

# Convergence of Markov-Chain Monte-Carlo Approaches to Multiuser and MIMO Detection

Soren Henriksen, Brett Ninness, *Member, IEEE*, and Steven R. Weller *Member, IEEE*

**Abstract**—Markov-chain Monte-Carlo methods have been demonstrated to offer an attractive alternative to the design of approximate (near optimal) maximum a-posteriori (MAP) detectors for synchronous direct-sequence code-division multiple access (DS-CDMA) and multi-input, multi-output (MIMO) multiple antenna applications. Central to evaluating these methods is understanding their convergence properties. In other works, this has been established via simulation, and the underlying theoretical basis has been identified. The contribution of this paper is to extend the theoretical understanding by rigorously establishing both convergence and convergence rate results for a wide class of Metropolis-Hastings methods.

**Index Terms**—MIMO systems, multiaccess communication, pseudonoise coded communication, spread spectrum communication.

## I. INTRODUCTION

IT IS WELL known that while maximum a-posteriori (MAP) approaches to direct-sequence code-division multiple access (DS-CDMA) multi-user detection (MUD) and multiple-input, multiple-output (MIMO) multiple antenna detection are optimal, they exhibit computational complexities which increase exponentially with the number of users in a CDMA system [1] and the product of the number of transmit antennas with the logarithm of the modulation constellation size in a MIMO system [2].

Due to these prohibitive complexities of the MAP detector, enormous effort has been expended over the past two decades to find suboptimal detectors which strike a balance between performance and computational complexity [3], [4]. Recently, new suboptimal approaches have been developed that involve Monte-Carlo based averaging of simulated quantities and have been illustrated as capable of impressive performance.

These can be divided into two broad classes. Sequential importance resampling (particle filter) based techniques [5], [6] and Markov-chain Monte-Carlo (MCMC) methods [7]–[12]. The first are approaches for approximating the Chapman-Kolmogorov measurement and time update equations in non-linear and non-Gaussian cases where the Kalman filter is inapplicable. The second are random number generators that can be designed to realise samples from rather arbitrary target densities.

This paper considers the latter class of methods, where in the MAP detection context, the essential idea is to select the target density as a desired symbol posterior, or function

thereof, and then use (possibly smoothed) sample averages of realisations from the random number generator as a MAP estimate.

Clearly, it is essential to consider and understand the convergence and convergence rate properties of such a simulation based approach. Indeed, this has been considered in previous work. For example, the work [8]–[12] provides extensive empirical evidence by illustrating impressive performance with moderate number of iterations, and the recent contribution [7] develops further improvements via Rao-Blackwell based smoothing of realisations.

However, the theoretical analysis and understanding of the methods underpinning this promising MCMC-based approach is much less developed. While the above-mentioned works acknowledge that convergence of the methods they develop can be understood by results from Markov chain theory, this is argued only in fairly general terms. The most complete analysis to date is provided in [13], which gives important insights.

This purpose of the work here is to extend the theoretical analysis of MCMC approaches and provide a detailed, rigorous and self-contained treatment of the theory underlying them. This involves provision of precise conditions on the underlying MAP detection problem that are sufficient for convergence, and establishing convergence rate quantifications formulated in terms of physically relevant parameters.

Furthermore, to date, the bulk of attention has been given to so-called Gibb's sampling approaches, but this paper considers the wider class of Metropolis-Hastings approaches, of which the Gibb's sampler is a special case. Finally, this paper does not assume knowledge of necessary Markov-chain results, but instead provides a completely self contained treatment with the necessary material integrated into the development.

## II. SIGNAL MODEL

This paper considers the following complex baseband model

$$\mathbf{y} = \mathbf{H}s(\mathbf{x}) + \boldsymbol{\nu}, \quad (1)$$

where  $\mathbf{y} \in \mathbf{C}^{n_R}$  is a vector of received signals, and  $s(\mathbf{x}) \in \mathbf{C}^{n_T}$  is a vector of transmitted symbols that depends on the  $K$ -vector of transmitted bits  $\mathbf{x} \in \{+1, -1\}^K$ .

In the MIMO case  $n_R$  and  $n_T$  are, respectively, the number of receive and transmit antennas, while in the multi-user situation  $n_R = n_T = K$ , the number of users.

Meanwhile,  $\boldsymbol{\nu} \in \mathbf{C}^{n_R}$  is additive noise assumed to be zero mean and with covariance  $\mathbf{R} = \mathbf{E}\{\boldsymbol{\nu}\boldsymbol{\nu}^T\}$ , while  $\mathbf{H} \in \mathbf{C}^{n_R \times n_T}$  is a matrix whose specification depends on whether (1) models a multi-user CDMA or multi-antenna MIMO system as will now be described.

Manuscript received June 1, 2007; revised October 26, 2007.

The authors are with the School of Electrical Engineering and Computer Science, The University of Newcastle, Callaghan, NSW 2308, Australia (e-mail: soren.henriksen@newcastle.edu.au, brett.ninness@newcastle.edu.au, steven.weller@newcastle.edu.au).

Digital Object Identifier 10.1109/JSAC.2008.080408.

### A. Multi-User (MU) CDMA

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):

$$y(t) = \sum_{n=1}^K A_n x_n(t) s_n(t) + \sigma \boldsymbol{\nu}(t), \quad 0 \leq t \leq T. \quad (2)$$

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 signature waveform assigned to the  $n$ -th user which is normalised such that  $\int_0^T s_n^2(t) dt = 1$ . The additive noise  $\boldsymbol{\nu}(t)$  in (2) is white Gaussian, with unit power spectral density and single-sided spectral density  $N_0 = 2\sigma^2$ .

In this paper it is assumed that 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 [3]

$$y_n = \int_0^T y(t) s_n(t) dt = A_n \mathbf{x}_n + \sum_{j \neq n} A_j \mathbf{x}_j \rho_{jn} + \boldsymbol{\nu}_n, \quad (3)$$

where

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

The baseband model (3)–(4) for  $n = 1, \dots, K$  can be represented by (1) with

$$\begin{aligned} \mathbf{H} &= \boldsymbol{\Gamma} \mathbf{A}, \quad R = \sigma^2 \boldsymbol{\Gamma}, \quad [\boldsymbol{\Gamma}]_{i,j} = \rho_{i,j}, \quad \mathbf{A} = \text{diag}\{A_n\}, \quad (5) \\ \mathbf{y} &\triangleq [y_1, \dots, y_K]^T, \quad \mathbf{x} \triangleq [x_1, \dots, x_K]^T, \quad \boldsymbol{\nu} \triangleq [\boldsymbol{\nu}_1, \dots, \boldsymbol{\nu}_K]^T. \end{aligned} \quad (6)$$

In this case, the mapping from binary data to symbols is given simply as  $s(\mathbf{x}) = \mathbf{x}$ .

### B. Multi-Antenna MIMO

Assuming transmit symbol period  $T$  and a channel which is static over this duration  $T$ , exhibits frequency flat fading, and rich scattering, then the appropriate MIMO baseband model is (1) with

$$\mathbf{H} = \sqrt{\frac{E_s}{n_T}} \boldsymbol{\Gamma}, \quad (7)$$

where  $E_s$  is the average energy of the transmitted symbols in  $s(\mathbf{x})$ , and  $\boldsymbol{\Gamma}$  describes the fading channel [2]. The mapping  $s(\mathbf{x}) : \{-1, +1\}^K \rightarrow \mathbf{C}^{n_T}$ , where  $K = n_T \log_2 M$  between data bits and transmitted symbols is assumed here to realise a M-PSK or M-QAM scheme (although others are possible). Finally  $\boldsymbol{\nu}$  is taken as zero mean with covariance  $\mathbf{R} = \mathbf{I}$ .

## III. APPROXIMATELY OPTIMAL DETECTION

Given the observation  $\mathbf{y}$ , we seek to estimate the vector of transmitted bits  $\mathbf{x}$ . The approach considered here is that of MAP detection, so that the estimate  $\hat{\mathbf{x}}$  is taken as

$$\hat{\mathbf{x}} = \arg \max_{\mathbf{x} \in \{-1, +1\}^K} p(\mathbf{x} | \mathbf{y}), \quad p(\mathbf{x} | \mathbf{y}) = \frac{p(\mathbf{x})}{p(\mathbf{y})} p_{\boldsymbol{\nu}}(\mathbf{y} - \mathbf{H}s(\mathbf{x})) \quad (8)$$

where  $p_{\boldsymbol{\nu}}(\cdot)$  is the joint probability density function of the elements in the noise vector  $\boldsymbol{\nu}$  and  $p(\mathbf{y})$  is a quantity normalising the total probability to be one. When  $p_{\boldsymbol{\nu}}$  is Gaussian, and  $p(\mathbf{x})$  assigns equal prior probability to all transmitted bits, this leads to the well known explicit form

$$p(\mathbf{x} | \mathbf{y}) = \frac{\text{const}}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2} \|\mathbf{R}^{-1/2}(\mathbf{y} - \mathbf{H}s(\mathbf{x}))\|^2\right) \quad (9)$$

where  $\|\cdot\|$  denotes Euclidean norm. The MAP estimate  $\hat{\mathbf{x}}$  is therefore, in the Gaussian case, given by

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x} \in \{-1, +1\}^K} \|\mathbf{R}^{-1/2}(\mathbf{y} - \mathbf{H}s(\mathbf{x}))\|^2. \quad (10)$$

The minimisation of (10) is a combinatorial optimisation problem requiring, in general, an exhaustive search over the  $2^K$  possible combinations of transmitted bits. This motivates the search for computationally cheaper solutions.

This paper considers an MCMC approach to the problem that has been proposed by several authors [7]–[11]. This involves generating a realisation  $\{x_1, \dots, x_k, \dots, x_L\}$  with limiting distribution equal to the desired posterior; viz.

$$\lim_{k \rightarrow \infty} p(\mathbf{x}_k = \mathbf{x} | \mathbf{x}_0) = p(\mathbf{x} | \mathbf{y}), \quad \forall \mathbf{x}_0. \quad (11)$$

This simulated realisation  $\{\mathbf{x}_k\}$  is then used as if it were a random sample from  $p(\mathbf{x} | \mathbf{y})$ . 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  $f$ , it allows the numerical computation and consistent estimation of the conditional expectation  $\mathbf{E}\{f(\mathbf{x}) | \mathbf{y}\}$  as

$$\mathbf{E}\{f(\mathbf{x}) | \mathbf{y}\} = \sum_{\mathbf{x}} f(\mathbf{x}) p(\mathbf{x} | \mathbf{y}) \approx \frac{1}{L} \sum_{k=\ell+1}^{\ell+L} f(\mathbf{x}_k). \quad (12)$$

Here  $\ell$  is the length of an initial ‘‘burn-in’’ period of the algorithm. In particular, this approach allows the computation of rather arbitrary posterior densities according to

$$p(f(\mathbf{x}) \in F | \mathbf{y}) \approx \frac{1}{L} \sum_{k=\ell+1}^{\ell+L} \chi_{f^{-1}(F)}(\mathbf{x}_k), \quad (13)$$

where  $\chi$  is the indicator function and  $F$  is a  $f$ -measurable set, so that the right hand side of (13) is simply the sample histogram of  $f(\mathbf{x})$ .

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  $p(\mathbf{x} | \mathbf{y})$ . Perhaps surprisingly, a Markov chain can be constructed with limiting density equal to the required posterior  $p(\mathbf{x} | \mathbf{y})$ , and in a manner that is far more straightforward than might prima facie be thought possible by means of the ‘Metropolis–Hastings Algorithm’ [14], [15] defined in Algorithm 3.1.

While this provides a completely general definition of the essential algorithm to be employed here, some important specialisations and embellishments will be studied in order to maximise its performance for the MU/MIMO applications studied in this paper.

**Algorithm 3.1 (Metropolis–Hastings):**

- 1) Initialise  $\mathbf{x}_0$  at some value such that  $p(\mathbf{x}_0 | \mathbf{y}) > 0$ .
- 2) At iteration  $k$ , consider a candidate value  $\xi_k$  which is drawn from a proposal density  $\gamma(\cdot | \mathbf{x}_{k-1})$ . That is, find a realisation  $\xi_k$  of  $\mathbf{x}_k$  as

$$\xi_k \sim \gamma(\cdot | \mathbf{x}_{k-1}). \quad (16)$$

- 3) Compute the acceptance probability

$$\alpha(\xi_k | \mathbf{x}_{k-1}) = \min \left\{ 1, \frac{p(\xi_k | \mathbf{y})}{p(\mathbf{x}_{k-1} | \mathbf{y})} \cdot \frac{\gamma(\mathbf{x}_{k-1} | \xi_k)}{\gamma(\xi_k | \mathbf{x}_{k-1})} \right\}. \quad (17)$$

- 4) Accept the proposed candidate  $\mathbf{x}_k = \xi_k$  with probability  $\alpha(\xi_k | \mathbf{x}_{k-1})$ , otherwise leave  $\mathbf{x}_k$  unchanged and discard the candidate.
- 5) Increment  $k$  and if  $k < \ell + L$  return to step 2 otherwise terminate.

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

$$\frac{\gamma(\mathbf{x}_{k-1} | \xi_k)}{\gamma(\xi_k | \mathbf{x}_{k-1})} = 1, \quad (14)$$

and hence the acceptance probability (17) simplifies to

$$\alpha(\xi_k | \mathbf{x}_{k-1}) = \min \left\{ 1, \frac{p(\xi_k | \mathbf{y})}{p(\mathbf{x}_{k-1} | \mathbf{y})} \right\}. \quad (15)$$

In this case, Algorithm 3.1 is known simply as the ‘Metropolis Algorithm’.

To see how this symmetric proposal scenario might occur, consider the obvious ‘‘random walk’’ proposal density implied by

$$\xi_k = \mathbf{x}_k + \mathbf{e}_k \quad (18)$$

in the special, but common, case in which the probability density  $p_e(\cdot)$  governing  $\mathbf{e}_k$  is symmetric in that  $p_e(\mathbf{x}) = p_e(-\mathbf{x})$ . Then clearly  $\gamma(\xi | \mathbf{x}) = p_e(\xi - \mathbf{x}) = p_e(\mathbf{x} - \xi) = \gamma(\mathbf{x} | \xi)$ .

A further specialisation of Algorithm 3.1 occurs in the situation where it is possible to employ the proposal density  $\gamma(\xi | \mathbf{x}) = p(\xi^i | \mathbf{x}^{-i}, \mathbf{y})$  where  $\xi^i$  denotes the  $i$ -th sub-block of the vector  $\xi$ , which may be of arbitrary (but fixed) size, and  $\mathbf{x}^{-i}$  denotes the complement of this, namely everything *except* the  $i$ -th sub-block. Note that the term in the acceptance probability (15) affected by this choice becomes

$$\frac{p(\xi_k | \mathbf{y})}{p(\mathbf{x}_{k-1} | \mathbf{y})} \cdot \frac{\gamma(\mathbf{x}_{k-1} | \xi_k)}{\gamma(\xi_k | \mathbf{x}_{k-1})} = \frac{p(\xi_k^i | \xi_k^{-i}, \mathbf{y}) p(\xi_k^{-i} | \mathbf{y})}{p(\mathbf{x}_{k-1}^i | \mathbf{x}_{k-1}^{-i}, \mathbf{y}) p(\mathbf{x}_{k-1}^{-i} | \mathbf{y})} \times \frac{p(\mathbf{x}_{k-1}^i | \mathbf{x}_{k-1}^{-i}, \mathbf{y})}{p(\xi_k^i | \xi_k^{-i}, \mathbf{y})} = 1.$$

In this calculation, the definition of conditional probability  $p(A | B) = p(A, B)/p(B)$  and the fact that by design  $\xi_k^{-i} = \mathbf{x}_{k-1}^{-i}$  have both been used. This implies that the acceptance probability  $\alpha(\mathbf{x}_k | \mathbf{x}_{k-1})$  in (15) is one, and hence the proposals drawn from the density  $p(\xi^i | \mathbf{x}^{-i}, \mathbf{y})$  are always retained. In this special case, Algorithm 3.1 becomes an instance of the *Gibbs sampling* algorithm, whose application to multi-user detection is presented in [13] and to MIMO detection in [11].

Before continuing, it is important to note that while the special case of  $p_{\nu}(\cdot)$  being Gaussian was used in (9), (10) in the interests of providing a concrete example, in what follows  $p_{\nu}(\cdot)$  can be quite arbitrary, subject only to certain very mild assumptions that will be introduced as necessary in the following lemmas and theorems.

## IV. MU/MIMO DETECTION VIA MCMC

With the essentials of the general Metropolis–Hastings Algorithm 3.1 established, the paper now turns to the application of these techniques to the specific problem of MU/MIMO detection. This involves the study of two specialisations of Algorithm 3.1, namely the Gibbs sampler and the Metropolis algorithm.

## A. MU/MIMO Detection via the Gibbs Sampler

Recall that the Gibbs sampler is achieved as a special case of Algorithm 3.1 whereby the proposal density is chosen as

$$\gamma(\xi_k | \mathbf{x}_{k-1}) = p(\xi_k^i | \mathbf{x}_{k-1}^{-i}, \mathbf{y}) \quad (19)$$

and the acceptance probability is then always equal to one. This requires that the  $k$ -th realisation of a vector  $\mathbf{x}_k$  be divided into a variety of sub-blocks. In general, each sub-block may comprise a different number of bits, and sub-blocks may overlap. The choice and size of the sub-blocks involves a trade-off between the computation load imposed at each iteration, and the total number of iterations required to form the required sample histograms to sufficient accuracy.

This paper studies the case of the sub-blocks being simply one element of  $\mathbf{x}_k$ . In this case, and for the MU/MIMO detection applications of this paper, the Gibbs proposal (19) may be computed via the following Lemma.

*Lemma 4.1:* Assuming that the elements  $\{\mathbf{x}^i\}$  are statistically independent of one another, and are assigned prior probabilities  $p(\mathbf{x}^i = +1) = \rho_i$ , then

$$p(\xi_k^i = +1 | \mathbf{x}_{k-1}^{-i}, \mathbf{y}) = \frac{1}{1 + \beta_i^{-1}}, \quad (20)$$

where

$$\beta_i \triangleq \frac{p(\mathbf{y} | \mathbf{x}_{k-1}^{-i}, \xi_k^i = +1)}{p(\mathbf{y} | \mathbf{x}_{k-1}^{-i}, \xi_k^i = -1)} \cdot \frac{\rho_i}{1 - \rho_i} \quad (21)$$

with

$$p(\mathbf{y} | \mathbf{x}_{k-1}^{-i}, \xi_k^i) = p_{\nu}(\mathbf{y} - \mathbf{H}\mathbf{s}(\mathbf{z})), \quad \mathbf{z} \triangleq \xi_k^i \cup \mathbf{x}_{k-1}^{-i}. \quad (22)$$

*Proof:* By application of Bayes’ rule and the definition of conditional probability

$$p(\xi_k^i | \mathbf{x}_{k-1}^{-i}, \mathbf{y}) = \frac{p(\mathbf{y} | \xi_k^i, \mathbf{x}_{k-1}^{-i}) p(\xi_k^i | \mathbf{x}_{k-1}^{-i})}{p(\mathbf{y})}. \quad (23)$$

Therefore, since the symbols sent by different users are assumed independent of one another,

$$\frac{p(\xi_k^i = +1 | \mathbf{x}_{k-1}^{-i}, \mathbf{y})}{p(\xi_k^i = -1 | \mathbf{x}_{k-1}^{-i}, \mathbf{y})} = \frac{p(\mathbf{y} | \mathbf{x}_{k-1}^{-i}, \xi_k^i = +1) p(\xi_k^i = +1)}{p(\mathbf{y} | \mathbf{x}_{k-1}^{-i}, \xi_k^i = -1) p(\xi_k^i = -1)}. \quad (24)$$

Consequently, with the definitions (21) and

$$\alpha_i \triangleq p(\xi_k^i = +1 | \mathbf{x}_{k-1}^{-i}, \mathbf{y}), \quad (25)$$

**Algorithm 4.1 (Gibbs Sampler for MUD/MIMOD):**

- 1) Initialise  $\mathbf{x}_0$  at some value such that  $p(\mathbf{x}_0 | \mathbf{y}) > 0$ .
- 2) At iteration  $k$ , for a randomly chosen index  $i$ , compute the proposal density value

$$\gamma(\boldsymbol{\xi}_k | \mathbf{x}_{k-1}) = p(\boldsymbol{\xi}_k^i = +1 | \mathbf{x}_{k-1}^{-i}, \mathbf{y}) \quad (27)$$

according to (20)–(22).

- 3) Draw a realisation  $\zeta \sim \mathcal{U}[0, 1]$ , where  $\mathcal{U}[0, 1]$  represents a uniform distribution on the interval  $[0, 1]$ .
- 4) Set  $\mathbf{x}_k^i = +1, \mathbf{x}_k^{-i} = \mathbf{x}_{k-1}^{-i}$  if

$$\gamma(\boldsymbol{\xi}_k | \mathbf{x}_{k-1}) < \zeta, \quad (28)$$

otherwise set  $\mathbf{x}_k^i = -1, \mathbf{x}_k^{-i} = \mathbf{x}_{k-1}^{-i}$ .

- 5) Increment  $k$  and if  $k < L$  return to step 2 otherwise terminate.

**Algorithm 5.1 (Metropolis for MUD/MIMOD):**

- 1) Initialise  $\mathbf{x}_0$  at some value such that  $p(\mathbf{x}_0 | \mathbf{y}) > 0$ .
- 2) At iteration  $k$ , for a randomly chosen index  $i$ , set  $\boldsymbol{\xi}_k = -\mathbf{x}_{k-1}^i \cup \mathbf{x}_{k-1}^{-i}$ .
- 3) Compute the acceptance probability

$$\alpha(\boldsymbol{\xi}_k | \mathbf{x}_{k-1}) = \min \{1, \beta_i\} \quad (29)$$

where  $\beta_i$  is given by (21)–(22).

- 4) Accept the proposed  $\mathbf{x}_k = \boldsymbol{\xi}_k$  with probability  $\alpha(\boldsymbol{\xi}_k | \mathbf{x}_{k-1})$ .
- 5) Increment  $k$  and return to step 2.

*Theorem 5.1:* Suppose that the noise corruption  $\nu$  in (1) has an underlying density which satisfies  $0 < p_\nu(\cdot) < \infty$ . Suppose further that the a-priori symbol probability satisfies  $0 < p_i(\mathbf{x}^i = +1) < 1$ . Then Algorithms 4.1 and 5.1 both generate a realisation sequence  $\{\mathbf{x}_k\}$  for which

$$\lim_{M \rightarrow \infty} \frac{1}{M} \sum_{k=1}^M f(\mathbf{x}_k) = \sum_{\mathbf{x} \in \{-1, +1\}^K} f(\mathbf{x}) p(\mathbf{x} | \mathbf{y}) = \mathbf{E}\{f(\mathbf{x}) | \mathbf{y}\} \quad (30)$$

with probability one where  $f : \mathbf{R}^K \rightarrow \mathbf{R}$  is an arbitrary bounded function.

*Proof:* First, note that, as explained in Section III, Algorithms 4.1 and 5.1 are particular cases of Algorithm 3.1 that depend on the choice of the proposal density  $\gamma(\cdot | \cdot)$ . Therefore Algorithms 4.1 and 5.1 both implement a time homogeneous Markov chain for which the probability of observing  $\mathbf{x}_k \neq \mathbf{x}_{k-1}$  given  $\mathbf{x}_{k-1}$  is simply the probability  $\gamma(\mathbf{x}_k | \mathbf{x}_{k-1})$  of proposing  $\mathbf{x}_k$  times the probability  $\alpha(\mathbf{x}_k | \mathbf{x}_{k-1})$  of accepting this proposal

$$\kappa(\mathbf{x}_k | \mathbf{x}_{k-1}) = \alpha(\mathbf{x}_k | \mathbf{x}_{k-1}) \cdot \gamma(\mathbf{x}_k | \mathbf{x}_{k-1}) \quad (31)$$

while the probability of  $\mathbf{x}_k = \mathbf{x}_{k-1}$  staying the same is a sum accounting for the fact that this may occur either via rejection of a proposal, or via the proposal actually being  $\mathbf{x}_{k-1}$  and being accepted; viz.

$$\begin{aligned} \kappa(\mathbf{x}_k = \mathbf{x}_{k-1} | \mathbf{x}_{k-1}) &= \gamma(\mathbf{x}_{k-1} | \mathbf{x}_{k-1}) + \\ &\sum_{\mathbf{z}: \alpha(\mathbf{z}, \mathbf{x}_{k-1}) < 1} \gamma(\mathbf{z} | \mathbf{x}_{k-1}) [1 - \alpha(\mathbf{z} | \mathbf{x}_{k-1})]. \end{aligned} \quad (32)$$

Therefore, noting that by construction in Algorithms 4.1 and 5.1 the proposed  $\mathbf{x}_k$  is unequal to  $\mathbf{x}_{k-1}$ , according to (17), (31)

$$p(\mathbf{x}_{k-1} | \mathbf{y}) \kappa(\mathbf{x}_k | \mathbf{x}_{k-1})$$

$$\begin{aligned} &= \alpha(\mathbf{x}_k | \mathbf{x}_{k-1}) \gamma(\boldsymbol{\xi}_k | \mathbf{x}_{k-1}) p(\mathbf{x}_{k-1} | \mathbf{y}) \\ &= \min \{ \gamma(\boldsymbol{\xi}_k | \mathbf{x}_{k-1}) p(\mathbf{x}_{k-1} | \mathbf{y}), \gamma(\boldsymbol{\xi}_{k-1} | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{y}) \}. \end{aligned}$$

On the other hand, by the same reasoning

$$p(\mathbf{x}_k | \mathbf{y}) \kappa(\mathbf{x}_{k-1} | \mathbf{x}_k)$$

$$\begin{aligned} &= \alpha(\mathbf{x}_{k-1} | \mathbf{x}_k) \gamma(\boldsymbol{\xi}_{k-1} | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{y}) \\ &= \min \{ \gamma(\boldsymbol{\xi}_{k-1} | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{y}), \gamma(\boldsymbol{\xi}_k | \mathbf{x}_{k-1}) p(\mathbf{x}_{k-1} | \mathbf{y}) \}. \end{aligned}$$

equation (24) indicates that

$$\frac{\alpha_i}{1 - \alpha_i} = \beta_i \Rightarrow \alpha_i = \frac{1}{1 + \beta_i^{-1}} \quad (26)$$

which completes the proof.  $\blacksquare$

Combining the general Algorithm 3.1 with the proposal density (19) computed via (21) then delivers the Gibbs sampling approach to MU/MIMO detection [7], [13] specified in Algorithm 4.1.

**B. MU/MIMO Detection via the Metropolis Algorithm**

The Metropolis algorithm is another specialisation of Algorithm 3.1. However, it is a situation in which the proposal density  $\gamma$  may be chosen much more generally than is dictated in the Gibbs sampling case. This offers several potential benefits relative to the Gibbs sampler, including enhanced convergence rate, and the ability to handle more complex estimation/detection problems.

For the purposes of MU/MIMO 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 realisation  $\mathbf{x}_k$ . The proposal density  $\gamma$  is then taken as one in which a tentative new realisation  $\boldsymbol{\xi}_k$  is formed by the bits in the randomly chosen  $i$ -th sub-block having their signs reversed with probability one.

In this case  $\gamma(\boldsymbol{\xi}_k | \mathbf{x}_{k-1}) = \gamma(\mathbf{x}_{k-1} | \boldsymbol{\xi}_k) = 1$  which, according to (14) is necessary for the associated general Metropolis–Hastings Algorithm 3.1 to reduce to the simpler Metropolis case. It is then necessary to compute the acceptance probability  $\alpha(\boldsymbol{\xi}_k | \mathbf{x}_{k-1})$  according to (15) to deliver the algorithm shown in Algorithm 5.1.

This algorithm definition assumes that the proposed  $\boldsymbol{\xi}_k^i = +1$ . If not, use  $\beta_i^{-1}$  in (29). The remainder of this paper will be primarily concerned with studying and illustrating the properties and performance of Algorithms 4.1 and 5.1.

**V. CONVERGENCE ANALYSIS**

To begin this analysis, the following result confirms the strong convergence of sample estimators such as (12), (13) for both the Gibbs and Metropolis sampling cases.

That is  $p(\mathbf{x}_{k-1} | \mathbf{y})\kappa(\mathbf{x}_k | \mathbf{x}_{k-1}) = p(\mathbf{x}_k | \mathbf{y})\kappa(\mathbf{x}_{k-1} | \mathbf{x}_k)$  and hence the Markov chain implied by (31) is ‘reversible’ [16] with respect to the posterior density  $p(\mathbf{x} | \mathbf{y})$ . Therefore,

$$\begin{aligned} \sum_{\mathbf{z}} p(\mathbf{z} | \mathbf{y})\kappa(\mathbf{x} | \mathbf{z}) &= \sum_{\mathbf{z}} p(\mathbf{x} | \mathbf{y})\kappa(\mathbf{z} | \mathbf{x}) \\ &= p(\mathbf{x} | \mathbf{y}) \sum_{\mathbf{z}} \kappa(\mathbf{z} | \mathbf{x}) = p(\mathbf{x} | \mathbf{y}), \end{aligned} \quad (33)$$

so that  $p(\mathbf{x} | \mathbf{y})$  is an ‘invariant’ density [16] of that chain which is a necessary condition for the convergence (30) [16]. To investigate sufficiency conditions, note that the Markov chain implied by  $\kappa(\mathbf{x}_{k+1} | \mathbf{x}_k)$  is respectively termed *irreducible* and *aperiodic* if and only if for all states  $i$  and  $j$ , there exists an  $n > 0$  such that (again respectively)  $\kappa(\mathbf{x}_{k+n} = j | \mathbf{x}_k = i) > 0$  and for all  $\ell > n$ ,  $\kappa(\mathbf{x}_{k+\ell} = i | \mathbf{x}_k = i) > 0$  [16, Theorem 1.2.1].

In relation to this, consider first the Metropolis Algorithm 5.1 and the case of two states  $i$  and  $j$  which differ in only one bit position, and hence for which  $\gamma(\mathbf{x}_{k+1} = j | \mathbf{x}_k = i) = K^{-1}$  where  $K$  is the dimension of  $x$ . Then according to (31)

$$\begin{aligned} \kappa(\mathbf{x}_{k+1} = j | \mathbf{x}_k = i) &= \gamma(j | i)\alpha(j | i) \times \\ &\quad \frac{1}{K} \cdot \min \left\{ 1, \frac{p(j | \mathbf{y})}{p(i | \mathbf{y})} \right\}, \end{aligned} \quad (34)$$

where by (8)

$$\frac{p(j | \mathbf{y})}{p(i | \mathbf{y})} = \frac{p(\mathbf{x} = j) p_{\nu}(\mathbf{y} - \mathbf{H}\mathbf{s}(j))}{p(\mathbf{x} = i) p_{\nu}(\mathbf{y} - \mathbf{H}\mathbf{s}(i))}. \quad (35)$$

Furthermore, by the assumptions of the Theorem all the terms in (35) are non-zero and bounded, so by (34)  $\kappa(\mathbf{x}_{k+1} = j | \mathbf{x}_k = i) > 0$ .

Turning to the Gibbs sampling Algorithm 4.1, in this case  $\kappa(\mathbf{x}_{k+1} | \mathbf{x}_k) = \gamma(\mathbf{x}_{k+1} | \mathbf{x}_k)$  where the latter is given by (19), (20) and (21) with

$$\beta = \frac{p_{\nu}(\mathbf{y} - \mathbf{H}\mathbf{s}(\mathbf{z}))}{p_{\nu}(\mathbf{y} - \mathbf{H}\mathbf{s}(\mathbf{w}))} \cdot \frac{\rho}{1 - \rho} \quad (36)$$

where  $\mathbf{z} \triangleq \mathbf{x}_{k+1}^i = +1 \cup \mathbf{x}_k^{-i}$ ,  $\mathbf{w} \triangleq \mathbf{x}_{k+1}^i = -1 \cup \mathbf{x}_k^{-i}$ . Therefore, again by the assumptions of the theorem  $\kappa(\mathbf{x}_{k+1} = j | \mathbf{x}_k = i) > 0$ .

Consequently, for both algorithms, and states  $i$  and  $j$  differing in an arbitrary number  $d$  of bit positions, there is a  $d$ -step path by which  $j$  can be reached from  $i$  with probability

$$\begin{aligned} \kappa(\mathbf{x}_{k+d} = j | \mathbf{x}_k = i) &\geq \kappa(\mathbf{x}_{k+d} = j | \mathbf{x}_{k+d-1} = \xi_{d-1}) \times \\ &\prod_{\ell=2}^{d-1} \kappa(\mathbf{x}_{k+\ell} = \xi_{\ell} | \mathbf{x}_{k+\ell-1} = \xi_{\ell-1}) \kappa(\mathbf{x}_{k+1} = \xi_{\ell+1} | \mathbf{x}_k = i) > 0. \end{aligned} \quad (37)$$

Therefore, the chain implied by  $\kappa(\mathbf{x}_{k+1} | \mathbf{x}_k)$  is irreducible with invariant density  $p(\mathbf{x} | \mathbf{y})$ . Hence by [16, Theorem 1.7.7] it is positive recurrent so that (30) follows by [16, Theorem 1.10.2]. ■

In particular, Theorem 5.1 implies that by choosing  $f$  as the Kronecker delta function  $f(\mathbf{x}) = \delta(\mathbf{x} - \mathbf{z})$  for any

$$z \in \{-1, +1\}^K$$

$$\lim_{M \rightarrow \infty} \frac{1}{M} \sum_{k=1}^M \delta(\mathbf{x}_k - \mathbf{z}) = \sum_{\mathbf{x} \in \{-1, +1\}^K} \delta(\mathbf{x} - \mathbf{z}) p(\mathbf{x} | \mathbf{y}) = p(\mathbf{z} | \mathbf{y}) \quad (38)$$

with probability one. That is, the sample histogram formed from realisations of either Algorithm 4.1 or 5.1 converges (as  $M$  increases) to the true underlying posterior density  $p(\cdot | \mathbf{y})$  and therefore provides an estimator for it.

The most closely related work is [13] wherein, for the Gibbs sampling case, the convergence in (11) is addressed by noting that

$$\lim_{k \rightarrow \infty} \max_{\mathbf{x} \in \{-1, +1\}^K} |p(\mathbf{x}_k = \mathbf{x} | \mathbf{x}_0) - p(\mathbf{x} | \mathbf{y})| = 0. \quad (39)$$

Although this is not formally argued in [13], it is noted that it depends on aperiodicity and irreducibility of the underlying Markov chain.

The following result establishes the conditions under which the Gibbs Sampler in Algorithm 4.1 and the Metropolis Algorithm 5.1 both imply such an aperiodic and irreducible chain, and hence for which pointwise distributional convergence (which implies (39)) holds.

*Theorem 5.2:* Under the conditions of Theorem 5.1, together with the further assumption that there exists at least two symbol vectors  $\mathbf{x}_1$  and  $\mathbf{x}_2$  such that  $p(\mathbf{x}_1 | \mathbf{y}) \neq p(\mathbf{x}_2 | \mathbf{y})$ , Algorithms 4.1 and 5.1 both generate a realisation sequence  $\{\mathbf{x}_k\}$  for which

$$\lim_{k \rightarrow \infty} p(\mathbf{x}_k = \mathbf{x} | \mathbf{x}_0) = p(\mathbf{x} | \mathbf{y}) \quad (40)$$

for any  $\mathbf{x}_0 \in \{-1, +1\}^K$ .

*Proof:* The proof depends on establishing that the Markov chain implied by  $\kappa(\mathbf{x}_{k+1} | \mathbf{x}_k)$  is ‘aperiodic’ [16]. For this purpose, suppose that for all  $i, j \in \{-1, +1\}^K$  that differ in only one bit position,  $p(i | \mathbf{y}) = p(j | \mathbf{y})$ . Then by Lemma A.1  $p(i | \mathbf{y}) = p(j | \mathbf{y})$  for all  $i, j \in \{-1, +1\}^K$  which contradicts the assumptions of the Theorem, and hence there exist a pair of states  $i$  and  $j$  differing in only one bit position for which

$$p(i | \mathbf{y}) > p(j | \mathbf{y}). \quad (41)$$

Therefore, in the Metropolis algorithm case, by (8)

$$\alpha(\mathbf{x}_{k+1} = j | \mathbf{x}_k = i) = \min \left\{ 1, \frac{p(j | \mathbf{y})}{p(i | \mathbf{y})} \right\} < 1. \quad (42)$$

Therefore, again according to (31)

$$\begin{aligned} \kappa(\mathbf{x}_{k+1} = i | \mathbf{x}_k = i) &= \sum_{\ell} \gamma(\ell | i) [1 - \alpha(\ell | i)] \\ &\geq \gamma(\ell = j | i) [1 - \alpha(\ell = j | i)] > 0. \end{aligned}$$

In the Gibbs sampling case,  $\gamma(\mathbf{x}_{k+1} = i | \mathbf{x}_k = i) > 0$  from (19), and so it follows that

$$\kappa(\mathbf{x}_{k+1} = i | \mathbf{x}_k = i) > 0. \quad (43)$$

Therefore, by induction,  $\kappa(\mathbf{x}_{k+n} = i | \mathbf{x}_k = i) > 0 \forall n \geq 1$  and hence by definition [16] the Markov chain implied by  $\kappa(\mathbf{x}_{k+1} | \mathbf{x}_k)$  is aperiodic. Since it has already been established to also be irreducible and positive recurrent, then (40) follows by [16, Theorem 1.8.3]. ■

It may be initially puzzling that, in light of the sample histogram convergence (38) implied by Theorem 5.1, an extra condition is required in Theorem 5.2 in order to establish the distributional convergence (40). To clarify this issue, consider a Markov process with two states  $S_1, S_2$  and with transition probabilities of 1 for each of the moves  $S_1 \rightarrow S_2, S_2 \rightarrow S_1$ . Then clearly the sample histogram formed from samples of this chain will converge to one with equal 0.5 weights centred on  $S_1$  and  $S_2$ . However, it is equally clear that the probability density function of these samples does not converge to that density, or in fact to any density - on alternate samples there are alternate probabilities of either one or zero for being in  $S_1$  or  $S_2$ .

This highlights that it is actually strong convergence of the form (30) presented in Theorem 5.1 that is of most interest from a practical standpoint, despite the fact that in what theoretical analysis has been previously available, only distributional convergence has been considered. To argue this further, note that while distributional convergence such as (40) is reassuring, it does not imply the convergence (30) of the sample-average type estimators that have been proposed in the MCMC literature.

## VI. CONVERGENCE RATE ANALYSIS

Having now established the convergence of Algorithms 4.1 and 5.1, this section turns to the practically relevant question of *rate* of convergence. This is a very challenging problem, and for a typical Metropolis–Hastings algorithm only fairly general bounds are possible [17].

Consequently, in the interests of most clearly exposing the key factors affecting convergence rate, this paper takes the approach of studying a hypothetical, more tractable Markov chain, which is nevertheless closely related to Algorithms 4.1 and 5.1 in that it is still the base Metropolis–Hastings Algorithm 3.1. The only difference is that it has a proposal density  $\gamma$  constrained to be independent of previous realisations from the chain.

This independence will permit explicit analysis of convergence rate. Furthermore, in order for this rate to be as informative as possible for the practical chains implied by Algorithms 4.1 and 5.1, in following sections the probability of a given candidate  $\xi_k$  in these algorithms in steady state will be matched to that of the hypothetical (analysable) chain via special choice of the independent proposal.

### A. Metropolis–Hastings with Independent Proposal

To study this case, it is useful to represent the state transition probabilities as a  $2^K \times 2^K$  transition probability matrix  $\mathbf{P}$  given as

$$[\mathbf{P}]_{\ell,j} = \kappa(\mathbf{x}_{k+1} = j \mid \mathbf{x}_k = \ell) \quad (44)$$

with associated vector of conditional state probabilities  $\pi$  defined as

$$\pi = [p(\mathbf{x} = 1 \mid \mathbf{y}), p(\mathbf{x} = 2 \mid \mathbf{y}), \dots, p(\mathbf{x} = 2^K \mid \mathbf{y})], \quad (45)$$

where the state index is determined according to the sorting,

$$\frac{\pi_1}{\gamma_1} \geq \frac{\pi_2}{\gamma_2} \geq \dots \geq \frac{\pi_{2^K}}{\gamma_{2^K}}. \quad (46)$$

Then by (33)

$$\pi \mathbf{P} = \pi, \quad (47)$$

and hence  $\pi$  is a left eigenvector of  $\mathbf{P}$  with eigenvalue  $\lambda_1 = 1$ . The remaining eigenvalues, denoted  $\lambda_1, \dots, \lambda_{2^K}$  determine the rate of convergence to this stationary distribution  $\pi$ , and may be directly formulated as follows.

*Theorem 6.1:* The transition matrix  $\mathbf{P}$  implied by Algorithm 3.1 with proposal  $\gamma$  constrained to be independent of previous realisations has eigenvalues in order of decreasing size given by

$$\lambda_0 = 1, \quad \lambda_j = \sum_{i=j}^{2^K} \left( \gamma_i - \frac{\pi_i}{\pi_j} \gamma_j \right), \quad 1 \leq j < 2^K. \quad (48)$$

*Proof:* We apply a method due to Liu [18] by noting that the transition matrix  $\mathbf{P}$  defined in (44) is given by

$$\mathbf{P} = \begin{bmatrix} \gamma_1 + \lambda_1 & \gamma_1 \pi_2 / \pi_1 & \dots & \gamma_1 \pi_{2^K-1} / \pi_1 & \gamma_1 \pi_{2^K} / \pi_1 \\ \gamma_1 & \gamma_2 + \lambda_2 & \dots & \gamma_2 \pi_{2^K-1} / \pi_2 & \gamma_2 \pi_{2^K} / \pi_2 \\ \gamma_1 & \gamma_2 & \dots & \gamma_3 \pi_{2^K-1} / \pi_3 & \gamma_3 \pi_{2^K} / \pi_3 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \gamma_1 & \gamma_2 & \dots & \gamma_{2^K-1} + \lambda_{2^K-1} & \gamma_{2^K-1} \frac{\pi_{2^K}}{\pi_{2^K-1}} \\ \gamma_1 & \gamma_2 & \dots & \gamma_{2^K-1} & \gamma_{2^K} \end{bmatrix} \quad (49)$$

with  $\lambda_j$  given by (48). To establish this, notice that for the subdiagonal terms ( $[\mathbf{P}]_{\ell,j}, \ell > j$ ), according to Algorithm 3.1 the acceptance probability is

$$\alpha(j, \ell) = \min \left\{ 1, \frac{\pi_j}{\pi_\ell} \cdot \frac{\gamma_\ell}{\gamma_j} \right\} = \min \left\{ 1, \frac{\pi_j}{\gamma_j} \cdot \frac{\gamma_\ell}{\pi_\ell} \right\} = 1 \quad (50)$$

where the latter follows according to the ordering (46). Hence for  $\ell > j$  the transition probability is simply the proposal probability  $\gamma_j$ .

Similarly, for the superdiagonal terms where  $\ell < j$ , the ordering (46) implies a transition probability which is acceptance probability times proposal probability of

$$\alpha(j, \ell) = \left( \frac{\pi_j}{\pi_\ell} \cdot \frac{\gamma_\ell}{\gamma_j} \right) \gamma_j = \frac{\pi_j}{\pi_\ell} \gamma_\ell. \quad (51)$$

The terms below the diagonal correspond to transitions to a more likely state than the present state. According to Algorithm 5.1, such a transition will always be accepted if it is proposed. Thus, the sub-diagonal transition probabilities are equal to the associated proposal probability.

Since  $\mathbf{P}$  is a stochastic matrix, its row sums are unity and hence the diagonal terms are given by

$$\begin{aligned} [\mathbf{P}]_{j,j} &= 1 - \sum_{i < j}^{2^K} \gamma_i - \sum_{i > j}^{2^K} \frac{\pi_i}{\pi_j} \gamma_j \\ &= \sum_{i=j}^{2^K} \gamma_i - \left( \sum_{i=j}^{2^K} \frac{\pi_i}{\pi_j} \gamma_j - \gamma_j \right) = \gamma_j + \lambda_j \end{aligned} \quad (52)$$

where the fact that  $\sum_i \gamma_i = 1$  has been exploited.

With this formulation of  $\mathbf{P}$  now established, it is claimed that its eigenvalues are given by  $\gamma$ (48) with corresponding

right eigenvectors given by

$$\mathbf{v}_k = \left[ 0, 0, 0, \dots, 0, -\sum_{\ell=k+1}^{2^K-1} \frac{\pi_\ell}{\pi_k}, 1, 1, \dots, 1 \right]^T, \quad 0 < k < 2^K, \quad (53)$$

$$\mathbf{v}_0 = [1, 1, \dots, 1]^T. \quad (54)$$

where there are  $k - 1$  zeros in (53). This is now verified by demonstrating

$$\mathbf{P}\mathbf{v}_k = \lambda_k \mathbf{v}_k. \quad (55)$$

Let  $\mathbf{V}$  be a matrix whose columns are  $\mathbf{v}_k$ , and then set  $\mathbf{U} = \mathbf{P}\mathbf{V}$ . Then for  $j < k$

$$[\mathbf{U}]_{j,k} = \gamma_j \frac{\pi_k}{\pi_j} \left( -\sum_{\ell=k+1}^{2^K} \frac{\pi_\ell}{\pi_k} \right) + \sum_{\ell=k+1}^{2^K} \gamma_j \frac{\pi_\ell}{\pi_j} = 0. \quad (56)$$

For  $j = k$

$$[\mathbf{U}]_{j,k} = (\gamma_k + \lambda_k) \left( -\sum_{\ell=k+1}^{2^K} \frac{\pi_\ell}{\pi_k} \right) + \sum_{\ell=k+1}^{2^K} \gamma_k \frac{\pi_\ell}{\pi_k} \quad (57)$$

$$= -\lambda_k \sum_{\ell=k+1}^{2^K-1} \frac{\pi_\ell}{\pi_k}. \quad (58)$$

For  $j > k$

$$[\mathbf{U}]_{j,k} = \gamma_k \left( -\sum_{\ell=k+1}^{2^K} \frac{\pi_\ell}{\pi_k} \right) + \left[ \sum_{\ell=k+1}^{j-1} \gamma_\ell \right] + [\gamma_j + \lambda_j] + \left[ \sum_{\ell=j+1}^{2^K} \gamma_k \frac{\pi_\ell}{\pi_k} \right] \quad (59)$$

$$= - \left[ \left( \sum_{\ell=k+1}^{2^K} \gamma_k \frac{\pi_\ell}{\pi_k} \right) + \left( \sum_{\ell=1}^{k-1} \gamma_\ell \right) + \gamma_k + \lambda_k \right]$$

$$+ \left( \sum_{\ell=1}^{k-1} \gamma_\ell \right) + \gamma_k + \lambda_k + \left[ \sum_{\ell=k+1}^{j-1} \gamma_\ell \right]$$

$$+ [\gamma_j + \lambda_j] + \left[ \sum_{\ell=j+1}^{2^K} \gamma_k \frac{\pi_\ell}{\pi_k} \right] \quad (60)$$

$$= -1 + \lambda_k + \left( \sum_{\ell=1}^{j-1} \gamma_\ell \right)$$

$$+ [\gamma_j + \lambda_j] + \left[ \sum_{\ell=j+1}^{2^K} \gamma_k \frac{\pi_\ell}{\pi_k} \right] = \lambda_k \quad (61)$$

where the simplification results from observing that the first line of (60) and last line of (61) are the sums of the  $k$ 'th and  $j$ 'th rows of  $\mathbf{P}$  and hence equal to one. Finally, this fact of all row sums of  $\mathbf{P}$  being equal to 1 establishes that  $\mathbf{v}_0$  given by (53) is a right eigenvector with eigenvalue  $\lambda_0 = 1$ . ■

This exact formulation of the eigenvalues of  $\mathbf{P}$  in (44) makes possible a simple and explicit convergence analysis.

*Theorem 6.2:* For any initialisation  $\mathbf{x}_0$ , and under the assumptions of Theorem 5.2, the distribution  $p(\mathbf{x}_k | \mathbf{x}_0)$  of a

realisation  $\mathbf{x}_k$  from Algorithm 5.1 with  $\gamma$  modified to be independent of past realisations satisfies

$$|p(\mathbf{x}_k = \mathbf{x} | \mathbf{x}_0) - p(\mathbf{x} | \mathbf{y})| < R\lambda^k \quad (62)$$

where

$$\lambda = 1 - \min_{\mathbf{x}} \tau(\mathbf{x}), \quad \tau(\mathbf{x}) \triangleq \frac{\gamma(\mathbf{x})}{p(\mathbf{x} | \mathbf{y})}. \quad (63)$$

*Proof:* Denote an arbitrary initial distribution at  $k = 0$  as

$$\boldsymbol{\mu}_0 = [p(\mathbf{x}_0 = 1), p(\mathbf{x}_0 = 2), \dots, p(\mathbf{x}_0 = 2^K)] \quad (64)$$

and note that the distribution  $\boldsymbol{\mu}_k$  of the Markov chain defined by  $\kappa(\cdot | \cdot)$ ,  $\boldsymbol{\mu}_0$  at time  $k$  is given by

$$\boldsymbol{\mu}_k = \boldsymbol{\mu}_0 \mathbf{P}^k. \quad (65)$$

If  $\mathbf{P}$  has distinct eigenvalues  $\{\lambda_k\}$ , with associated left eigenvectors  $\{\mathbf{w}_k\}$ , define  $\{\alpha_k\}$  via the decomposition

$$\boldsymbol{\mu}_0 = \sum_{\ell=0}^{2^K-1} \alpha_\ell \mathbf{w}_\ell. \quad (66)$$

Then

$$\begin{aligned} \boldsymbol{\mu}_k &= \boldsymbol{\mu}_0 \mathbf{P}^k = \left( \sum_{\ell=0}^{2^K-1} \alpha_\ell \lambda_\ell \mathbf{w}_\ell \right) \mathbf{P}^{k-1} \\ &= \left( \sum_{\ell=0}^{2^K-1} \alpha_\ell \lambda_\ell^m \mathbf{w}_\ell \right) \mathbf{P}^{k-m} = \sum_{\ell=0}^{2^K-1} \alpha_\ell \lambda_\ell^k \mathbf{w}_\ell. \end{aligned} \quad (67)$$

Theorem 6.1 establishes that  $\lambda_0 = 1$  is a maximal eigenvalue with all others satisfying  $|\lambda_k| < 1$ ,  $0 < k < 2^K$ . Furthermore, via (47)  $\mathbf{w}_0 = \boldsymbol{\pi}$  and hence

$$\boldsymbol{\mu}_k = \boldsymbol{\pi} \alpha_0 + \boldsymbol{\varepsilon}_k, \quad \boldsymbol{\varepsilon}_k \triangleq \sum_{\ell=1}^{2^K-1} \alpha_\ell \lambda_\ell^k \mathbf{w}_\ell \quad (68)$$

where an arbitrary  $i$ 'th element of  $\boldsymbol{\varepsilon}_k$  satisfies

$$|\varepsilon_k^i| < R \cdot \max_{1 \leq \ell < 2^K} |\lambda_\ell|^k, \quad R \triangleq \sum_{\ell=1}^{2^K-1} |\alpha_\ell| |\mathbf{w}_\ell^i| \quad (69)$$

However, by Theorem 6.1

$$\max_{1 \leq \ell < 2^K} |\lambda_\ell| = \lambda_1 = 1 - \frac{\gamma_1}{\pi_1} < 1 \quad (70)$$

so that  $\lim_{k \rightarrow \infty} |\varepsilon_k^i| = 0$  and hence by Theorem 5.2  $\alpha_0 = 1$ . ■

Therefore, in the situation where the Metropolis Algorithm 5.1 is modified to have independent proposal, the convergence rate of the sample density  $p(\mathbf{x}_k | \mathbf{x}_0)$  to the posterior  $p(\mathbf{x} | \mathbf{y})$  is exponential, and governed by the maximum mismatch between the desired posterior  $p(\mathbf{x} | \mathbf{y})$  and the proposal  $\gamma$  as measured by the minimum value of the ratio  $\tau(\mathbf{x})$ .

In particular, at high SNR when the MAP estimate  $\hat{\mathbf{x}}$  implies  $p(\hat{\mathbf{x}} | \mathbf{y}) \approx 1$ , then for uniform proposal  $\gamma(\mathbf{x}) = 2^{-K}$

$$\min_{\mathbf{x}} \tau(\mathbf{x}) \leq \tau(\hat{\mathbf{x}}) \approx \frac{1}{2^K}. \quad (71)$$

For large  $K$  (eg. number of users in a CDMA system), this will imply  $\lambda \approx 1$  and hence slow convergence, which is consistent with the 'hold up' observed in [7], [13].

### B. Independent Proposal Choice

We now further investigate the choice of independent proposal density  $\gamma$ . In what follows, it will be selected so that the probability  $\gamma(\xi)$  of a given proposed Markov chain realisation  $\xi$  is identical to that arising from the original (non-independent proposal) Metropolis Algorithm 5.1 after it has converged.

The underlying argument is that via this construction, the convergence properties of the independent chain can be expected to be informative as possible for those of Algorithm 5.1.

In this case, the convergence rate can again be judged by the results of Theorem 6.2 in combination with the following Lemma quantifying  $\tau(\mathbf{x})$ .

*Lemma 6.1:* Suppose that the independent proposal  $\gamma(\xi)$  is chosen to be, as a function of  $\xi$ , identical to the dependent proposal  $\gamma(\xi | \mathbf{x})$  employed in Algorithm 5.1 after convergence has occurred. Then the quantity  $\tau(\mathbf{x})$  defined in equation (63) of Theorem 6.2 is given by

$$\tau(\mathbf{x}) = \frac{q(\mathbf{x} | \mathbf{y})}{p(\mathbf{x} | \mathbf{y})}, \quad q(\mathbf{x} | \mathbf{y}) \triangleq \sum_{\sigma} \frac{1}{K} p(\xi | \mathbf{y}), \quad (72)$$

$$\sigma \triangleq \{\xi : H(\mathbf{x}, \xi) = 1\} \quad (73)$$

where  $H(\mathbf{x}, \xi)$  is the number of elements in  $\mathbf{x}$  that are different to those in  $\xi$ .

*Proof:* Note that according to Theorem 5.2, the unmodified Algorithm 5.1 converges so that  $\mathbf{x}_k \sim p(\mathbf{x} | \mathbf{y})$ , in which case the probability of a candidate realisation  $\xi$  being proposed is

$$\sum_x \gamma(\xi | \mathbf{x}) p(\mathbf{x} | \mathbf{y}). \quad (74)$$

Furthermore, Algorithm 5.1 uses a (non-independent) proposal which involves randomly inverting one bit of  $\mathbf{x}$ . This may be expressed as

$$\gamma(\xi | \mathbf{x}) = \begin{cases} K^{-1} & ; \text{If } H(\xi, \mathbf{x}) = 1 \\ 0, & ; \text{Otherwise,} \end{cases} \quad (75)$$

Therefore, by combining (74) and (75)

$$\gamma(\xi) = \sum_{\mathbf{x}: H(\xi, \mathbf{x})=1} \frac{1}{K} p(\mathbf{x} | \mathbf{y}) \quad (76)$$

is an independent proposal that, after burn in, is identically distributed to the non-independent proposal used in Algorithm 5.1. Using this in (63) completes the proof. ■

To understand this result, note that  $\{\xi : H(\xi, \mathbf{x}) = 1\}$  is the set of elements differing from  $\mathbf{x}$  in only one bit position. Therefore, if the initial distribution for the transmitted bits is independent and equiprobable as  $p(x^i) = 1/K$ ,  $q(\mathbf{x} | \mathbf{y})$  is the posterior probability  $q(\mathbf{x} | \mathbf{y})$  that the transmitted symbol differs from  $\mathbf{x}$  by one bit,

Therefore,  $\tau(\mathbf{x})$  given by (72) is the likelihood ratio for testing the hypothesis that an estimate  $\mathbf{x}$  is in error by one bit.

Consequently, when the analysable independent proposal case is as closely matched as possible to the implementable Metropolis Algorithm 5.1, then according Theorem 5.2, the exponential convergence rate slows as the certainty in received bit decisions based on the posterior  $p(\mathbf{x} | \mathbf{y})$  grows.

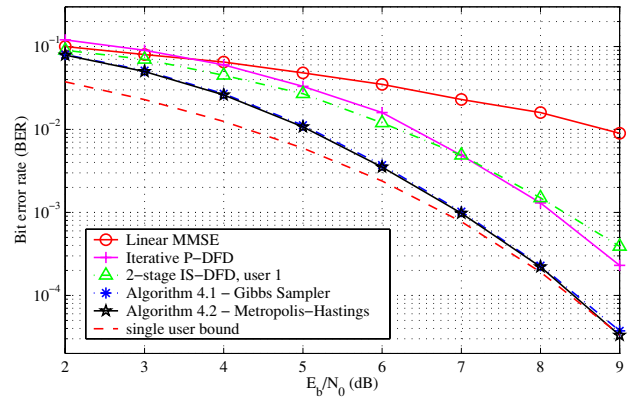


Fig. 1. Performance comparison of different receivers on an AWGN channel, with  $K = 96$  users and spreading factor  $N = 128$

In particular, assuming that  $\min_x \tau(\mathbf{x})$  occurs at the MAP estimate  $\hat{\mathbf{x}}$ , and denoting by  $L_E$  the likelihood ratio testing the hypothesis that  $\hat{\mathbf{x}}$  is in error by one bit

$$L_E = \tau(\hat{\mathbf{x}}) \quad (77)$$

then according to Theorem 5.2 the independent sampler tied to Algorithm 5.1 converges as

$$|p(\mathbf{x}_k = \mathbf{x} | \mathbf{x}_0) - p(\mathbf{x} | \mathbf{y})| < R[1 - L_E]^k \quad (78)$$

Therefore, since  $L_E$  becomes smaller as SNR increases, then the convergence rate will decrease, which has been empirically observed by other authors [7], [13].

## VII. SIMULATION EXAMPLE

Having now established various convergence properties of Algorithms 4.1 and 5.1, this section presents a brief simulation example to provide a concrete illustration of their performance. Readers interested in more extensive simulation-based performance profiling of Metropolis-Hastings type approaches are referred to [7], [8], [11], [13].

A multi-user detection problem of non-trivial size involving  $K = 96$  users with spreading factor  $N = 128$  and additive white Gaussian noise channel was chosen. The signature waveforms  $s_n(t) = \pm 1$  were chosen randomly and independently, and the choice  $A_k = 1, k = 1, \dots, K$  was taken for the transmit powers.

At each value of  $E_b/N_0$ , Algorithms 4.1 and 5.1 were both run for  $M = 5000$  iterations, with detection achieved by thresholding of the resulting posterior probability estimates. The bit error rate (BER) was then evaluated by repeating the simulation with different noise and spreading sequence realisations until a total of 5000 error events were observed.

The results are presented in Figure 1. Also shown there is the performance, under the same conditions, of the linear minimum mean-squared error (MMSE) detector [3], an iterative parallel decision-feedback detector (P-DFD) and a 2-stage iterative successive DFD (IS-DFD) [19], and the single-user bound. Clearly, and especially for higher SNR's, Algorithms 4.1 and 5.1 outperform the alternative MMSE, P-DFD and IS-DFD receivers. Furthermore, the Metropolis based Algorithm 5.1 enjoys a slight performance advantage relative to relative to the Gibbs sampling based Algorithm 4.1.

In these simulations, the variance of the proposal density  $\gamma$  was chosen to achieve an observed acceptance rate of approximately 30%, which was important for maintaining performance at high SNR's. This technique is well accepted and widely used in the MCMC literature [17].

### VIII. CONCLUSIONS

This paper has used a unified formulation to encompass a variety of Markov chain Monte Carlo approaches to a range of 'multi-dimensional' detection problems to facilitate the study of convergence properties including necessary conditions, and factors influencing rate of convergence.

While empirical study by the authors has exposed similar performance for the Gibbs sampling and Metropolis methods, because of the freedom in proposal density choice, the latter is more easily extensible beyond the 'detection-only' scenario considered here without the need for special 'conjugate' choices of assumed densities on unknown parameters (eg. channel state information).

The application of these same methods to problems of soft decoding and channel state estimation is believed worthy of further study.

### APPENDIX

#### AUXILIARY RESULT

*Lemma A.1:* For  $\mathbf{x}$  and  $\mathbf{y}$  vectors in  $\{-1, +1\}^K$ , let  $H(\mathbf{x}, \mathbf{y})$  be the number of elements in  $\mathbf{x}$  that are different to those in  $\mathbf{y}$ , and let  $f: \mathbf{R}^K \rightarrow \mathbf{R}$  be arbitrary. If  $f(\mathbf{x}_a) = f(\mathbf{x}_b)$  for all  $\mathbf{x}_a, \mathbf{x}_b$  such that  $H(\mathbf{x}_a, \mathbf{x}_b) = 1$ , then  $f(\mathbf{x})$  is a constant for all  $\mathbf{x} \in \{-1, +1\}^K$ .

*Proof:* Take  $\mathbf{x}_p$  and  $\mathbf{x}_q$  arbitrary and denote  $H(\mathbf{x}_p, \mathbf{x}_q) = d$ . Then there exists a sequence of  $d + 1$  states  $\{\mathbf{x}_0 = \mathbf{x}_p, \mathbf{x}_1, \mathbf{x}_2 \dots \mathbf{x}_d = \mathbf{x}_q\}$  with the property that  $H(\mathbf{x}_r, \mathbf{x}_{r+1}) = 1$  for  $0 \leq r < d$ . Consequently  $f(\mathbf{x}_r) = f(\mathbf{x}_{r+1})$  for  $0 \leq r < d$ . However,  $\mathbf{x}_p$  and  $\mathbf{x}_q$  were arbitrary, and hence  $f(\mathbf{x}_p) = f(\mathbf{x}_q)$  for all  $\mathbf{x}_p$  and  $\mathbf{x}_q$ . ■

### REFERENCES

- [1] S. Verdú, "Computational complexity of optimum multiuser detection", *Algorithmica*, vol. 4, no. 3, pp. 303–312, 1989.
- [2] Arogyaswami Paulraj, Rohit Nabar, and Dhananjay Gore, *Introduction to Space-Time Wireless Communications*, Cambridge University Press, 2003.
- [3] S. Verdú, *Multiuser Detection*, Cambridge University Press, Cambridge, United Kingdom, 1998.
- [4] Bertrand Hochwald and Stephen ten Brink, "Achieving near-capacity on a multiple-antenna channel", *IEEE Trans. Communications*, vol. 51, no. 3, pp. 389–399, 2003.
- [5] P.M. Djuric, J.H. Kotecha, J.Q. Zhang, Y.F. Huang, T. Ghirmai, M.F. Bugallo, and JMiguez, "Particle filtering", *IEEE Signal Processing Mag.*, vol. 20, no. 5, pp. 19–38, September 2003.
- [6] Yufei Huang and Petar M. Djuric, "Multiuser detection of synchronous code division multiple access signals by perfect sampling", *IEEE Trans. Signal Processing*, vol. 50, no. 7, pp. 1724–1734, July 2002.
- [7] Behrouz Farhang-Bououjny, Haidong Zhu, and Zhenning Shi, "Markov chain Monte Carlo algorithms for CDMA and MIMO communication systems", *IEEE Trans. Signal Processing*, vol. 54, no. 5, pp. 1896–1909, May 2006.
- [8] Bin Dong, Xiaodong Wang, and Arnaud Doucet, "A new class of soft MIMO demodulation algorithms", *IEEE Trans. Signal Processing*, vol. 51, no. 11, pp. 2752–2763, 2003.
- [9] 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.
- [10] Xiaodong Wang and Rong 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.
- [11] Haidong Zhu, Behrouz Farhang-Boroujny, and Rong Chen, "On performance of sphere decoding and Markov chain Monte Carlo detection methods", *IEEE Signal Processing Lett.*, vol. 12, no. 10, pp. 669–672, October 2005.
- [12] Arnaud Doucet and Xiaodong Wang, "Monte Carlo methods for signal processing", *IEEE Signal Processing Mag.*, vol. 22, no. 6, pp. 152–170, November 2005.
- [13] 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.
- [14] M. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller, "Equations of state calculations by fast computing machines", *J. Chemical Physics*, vol. 21, pp. 1087–1091, 1953.
- [15] W. K. Hastings, "Monte Carlo sampling methods using Markov chains and their applications", *Biometrika*, vol. 57, no. 1, pp. 97–109, April 1970.
- [16] J.R. Norris, *Markov Chains*, Cambridge University Press, 1997.
- [17] Christian Robert and George Casella, *Monte Carlo Statistical Methods*, Springer, 1999.
- [18] J. Liu, "Metropolized independent sampling and comparisons to rejection sampling and importance sampling", *Statistics and Computing*, vol. 6, pp. 113–119, 1995.
- [19] G. Woodward, R. Ratasuk, M. L. Honig, and P. B. Rapajic, "Minimum mean-squared error multiuser decision-feedback detectors for DS-CDMA", *IEEE Trans. Commun.*, vol. 50, no. 12, pp. 2104–2112, December 2002.



**Soren Henriksen** Soren completed a bachelor degree in computer engineering at the University of Newcastle, Australia in 1996. Since then he has worked in system identification and control, completing a masters degree at the University of Newcastle in 2001. He is currently studying for a PhD applying statistical methods to system identification.



**Brett Ninness** Brett Ninness was born in 1963 in Singleton, Australia and received his BE, ME and Ph.D degrees in Electrical Engineering from the University of Newcastle, Australia in 1986, 1991 and 1994 respectively. He has stayed with the School of Electrical Engineering and Computer Science at the University of Newcastle since 1993, where he is currently a Professor.

His research interests are in the areas of system identification and stochastic signal processing, in which he has authored approximately one hundred papers in journals and conference proceedings. He has served on the editorial boards of Automatica, IEEE Transactions on Automatic Control and is currently Editor in Chief for IET Control Theory and Applications.

Together with Håkan Hjalmarsson he jointly organised the 14th IFAC Symposium on System Identification in Newcastle, Australia in 2006. Further details of his professional activities are available at <http://sigpromu.org/brett>.



**Steven R. Weller** Steven R. Weller was born in Sydney, Australia, in 1965. He received the B.E. (Hons I) degree in Computer Engineering in 1988, the M.E. degree in Electrical Engineering in 1992, and the Ph.D. degree in Electrical Engineering in 1994, all from the University of Newcastle, Australia. From April 1994 to July 1997 he was a Lecturer in the Department of Electrical and Electronic Engineering, at the University of Melbourne, Australia. In July 1997 he joined the University of Newcastle, where he is currently an Associate Professor and

Head of School, Electrical Engineering and Computer Science. His current research interests include low-density parity-check codes, iterative decoding algorithms, and space-time coded communications.