Advanced Markov Chain Monte Carlo Methods for Iterative (Turbo) Multiuser Detection

Markus A. Dangl[1], Zhenning Shi[2], Mark C. Reed[2], Jürgen Lindner[1]

[1]University of Ulm, Dept. of Information Technology, Albert-Einstein-Allee 43, 89081 Ulm, Germany
Email: {markus.dangl,juergen.lindner}@uni-ulm.de
[2] National ICT Australia and Australian National University
Locked Bag 8001, Canberra ACT 2601, Australia
Email: {zhenning.shi,mark.reed}@nicta.com.au

Abstract

Recently, Markov Chain Monte Carlo (MCMC) sampling methods have evolved as new promising solutions to both multiuser and multiple-input multiple-output (MIMO) detection problems. Approaches based on Gibbs sampling as a special type of MCMC methods are well suited due to their good trade-off between performance and complexity. However, it is known that detection methods based on Gibbs sampling show a performance degradation in the high signal-to-noise ratio (SNR) regime. We propose an improved version of a soft-input soft-output algorithm, where this degradation effect is considerably mitigated. Employing the algorithm for turbo multiuser detection in overloaded code-division multiple-access (CDMA) systems yields excellent performance in comparison to other known detection schemes while offering moderate computational complexity.

1 Introduction

Multiuser detection has become a vital technique to achieve high spectral and power efficiency in CDMA systems. Due to the exponential complexity of optimum multiuser detection [1], algorithms with largely reduced cost have gained a huge attention. Among them, detection schemes that are able to process soft-input information and deliver soft-outputs are of special interest as they represent an efficient means to reduce interuser interference by being embedded in an iterative (turbo) loop [2]–[5]. Recently, algorithms based on the theory of Markov chains and employing Monte Carlo sampling techniques have become popular for various detection problems such as equalization of intersymbol interference (ISI) channels, equalization in MIMO systems, and multiuser detection [6]–[9]. In this paper, we review MCMC techniques for synchronous CDMA following the Gibbs sampling [10] approach in [11]. The Gibbs sampler is known to suffer from a possible performance degradation at high SNR when the system load becomes large. We propose a novel technique referred to as *forced state transitions* to significantly mitigate this problem. Numerical results indicate that our suggested method triggers convergence of the turbo scheme to low bit error rates (BERs) for those scenarios where the conventional MCMC algorithms fail. In addition, we provide a performance-wise comparison to known multiuser detection schemes from the literature.

The paper is organized as follows. Section 2 introduces the notation and the system model. Conventional multiuser detection techniques under consideration are briefly summarized in Section 3. We review MCMC techniques for multiuser detection in Section 4.1 and derive the proposed modified MCMC algorithm in Section 4.2. Numerical results are presented in Section 5. Section 6 wraps up with some conclusions.

Zhenning Shi and Mark C. Reed are with National ICT Australia and affiliated with the Australian National University. This work has been conducted while Markus A. Dangl was a visiting researcher at National ICT Australia.

2 System Model

We use the following notation: upper case bold letters indicate matrices and lower case bold letters column vectors. $\mathbf{I}_N$ denotes the $N \times N$ identity matrix. The probability density function (pdf) of an N-dimensional real-valued Gaussian random variable $\mathbf{z}$ with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Phi}$ is defined by $f(\mathbf{z}) = 1/(\sqrt{(2\pi)^N|\boldsymbol{\Phi}|}) \cdot \exp\big(-\frac{1}{2}(\mathbf{z}-\boldsymbol{\mu})^T\boldsymbol{\Phi}^{-1}(\mathbf{z}-\boldsymbol{\mu})\big)$. We briefly denote such a particular distribution as multivariate Gaussian.

Next, we introduce the CDMA system model under consideration as depicted in Fig. 1. A vector of equally probable source bits $\mathbf{q}_k$ of length N_q, belonging to user $k = 1, \ldots, K$, is protected by a convolutional code of rate $R_c = N_q/N_c$. We assume that all users apply the same channel code. The vector of encoded bits $\mathbf{c}_k$ of length N_c is fed into a random block permutor $\boldsymbol{\Pi}_k$ to form the interleaved vector $\mathbf{v}_k$. The permutor $\boldsymbol{\Pi}_k$ can be represented by a random $N_c \times N_c$ permutation matrix. Each bit in $\mathbf{v}_k$ corresponds to a specific discrete time index i and is mapped onto a BPSK symbol

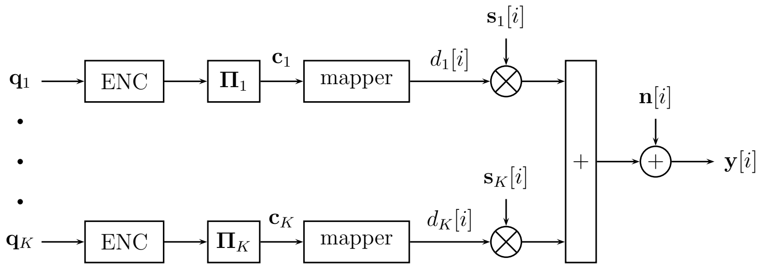

Fig. 1. Model for a synchronous CDMA system.

$d_k[i] \in \{-1; +1\}$. For conceptual ease, we consider a synchronous CDMA system over an AWGN channel. Thus, the received vector $\mathbf{y}[i]$, sampled at chip rate, is given by

$$\mathbf{y}[i] = \mathbf{S}[i]\mathbf{d}[i] + \mathbf{n}[i], \tag{1}$$

where $\mathbf{d}[i] = [d_1[i], d_2[i], \ldots, d_K[i]]^T$ denotes the vector of transmit symbols at time slot i and $\mathbf{S}[i]$ is an $N \times K$ matrix, whose columns are the users' binary spreading sequences $\mathbf{s}_k[i] \in \{-1/\sqrt{N}; +1/\sqrt{N}\}^N$. $\mathbf{n}[i]$ contains the samples of an AWGN process with spectral power density $\sigma_n^2 = N_0/2$. The users' spreading sequences are drawn randomly each time i the channel is used. We refer to $\beta = K/N$ as the load of the system.

At the receive side turbo multiuser detection (joint iterative multiuser detection and decoding) is employed as illustrated in Fig. 2. The multiuser detection algo-

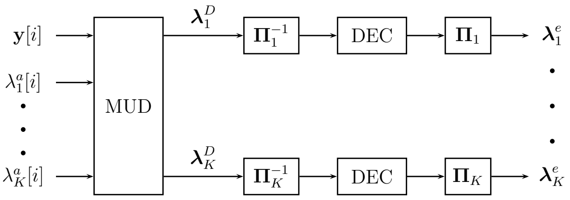

Fig. 2. Principle of turbo multiuser detection. Extrinsic output of decoders is fed back to the input of the MUD algorithm.

rithm ("MUD") expects the received vector $\mathbf{y}[i]$ and a-priori information of the code bits of all users in form of log-likelihood ratios (LLRs) $\lambda_k^a[i]$ ($k = 1, \ldots, K$). After processing all received vectors $\mathbf{y}[i]$ that correspond to a complete codeword, i.e., $i = 1, \ldots, N_c$, extrinsic output LLRs are generated for each user ($\boldsymbol{\lambda}_1^D, \ldots, \boldsymbol{\lambda}_K^D$). Here, we applied the short-hand notation $\boldsymbol{\lambda}_k^D = [\lambda_k^D[1], \ldots, \lambda_k^D[N_c]]^T$. The LLRs $\boldsymbol{\lambda}_k^D$ are fed into the corresponding decoder after deinterleaving ($\mathbf{\Pi}_k^{-1}$). The outputs of the decoders are extrinsic LLRs of the code bits that are combined into the vectors $\boldsymbol{\lambda}_k^e = [\lambda_k^e[1], \ldots, \lambda_k^e[N_c]]^T$ ($k = 1, \ldots, K$) after an interleaving step. The extrinsic LLRs $\lambda_k^e[i]$ act as a-priori information $\lambda_k^a[i]$ in a next iteration. In the last iteration, the decoders perform a hard decision on the information bits of the codewords. For the sake of clarity, we drop the time index i in the following.

3 Conventional Multiuser Detection Algorithms

For comparison, we consider three conventional turbo algorithms:

- Matched-filtering with parallel soft interference cancellation using extrinsic probabilities (MF-PIC). Details are given in [3].
- Matched-filtering with successive soft interference cancellation using a soft-decision function based on a-posteriori probabilities and variance estimates (MF-SIC). The scheme shows the structure of a recurrent neural network [12] and is described in detail for turbo equalization in [13].
- Filtering based on the minimum mean-square error criterion with parallel soft interference cancellation using extrinsic probabilities (MMSE-PIC) [5].

The computational complexity of the MUD algorithm (without decoding) per user, symbol, and iteration is about $\mathcal{O}(K)$ for both the MF-PIC and the MF-SIC, and $\mathcal{O}(K^2)$ for the MMSE-PIC.

4 Multiuser Detection based on MCMC Methods

In the following, we review Gibbs sampling combined with Monte Carlo integration as proposed in [11]. Our improved concept is presented in Section 4.2.

4.1 Review of MCMC Methods

The purpose of the MUD algorithm is to provide estimates of the a-posteriori probability (APP) of all users, given the received vector $\mathbf{y}$ and the vector of a-priori LLRs $\boldsymbol{\lambda}^a = [\lambda_1^a, \ldots, \lambda_K^a]^T$, i.e., $P(d_k = +1|\mathbf{y}, \boldsymbol{\lambda}^a)$, $k = 1, \ldots, K$. The APP can be expressed as a $(K-1)$-dimensional marginalization of joint probabilities yielding [11]

$$\begin{aligned} &P(d_k=+1|\mathbf{y},\boldsymbol{\lambda}^a)\\ &=\sum_{n=1}^{2^{K-1}} P(d_k=+1,\mathbf{d}_{-k}=\mathbf{d}_{-k}^{(n)}|\mathbf{y},\boldsymbol{\lambda}^a)\\ &=\sum_{n=1}^{2^{K-1}} P(d_k=+1|\mathbf{y},\mathbf{d}_{-k}=\mathbf{d}_{-k}^{(n)},\boldsymbol{\lambda}^a)\\ &\qquad\cdot P(\mathbf{d}_{-k}=\mathbf{d}_{-k}^{(n)}|\mathbf{y},\boldsymbol{\lambda}^a), \end{aligned} \tag{2}$$

where $\mathbf{d}_{-k}$ denotes the vector $\mathbf{d}$ without the kth symbol. The vectors $\mathbf{d}_{-k}^{(n)}$ are distinct and represent interfering candidate symbols (all other users except the user of interest k) taken from the set of all 2^{K-1} possible hypotheses. In [11] (2) was approximated by classical Monte Carlo integration yielding

$$\begin{aligned} &P(d_k=+1|\mathbf{y},\boldsymbol{\lambda}^a)\\ &\approx 1/N_s\sum_{n=1}^{N_s} P(d_k=+1|\mathbf{y},\mathbf{d}_{-k}=\mathbf{d}_{-k}^{(n)},\boldsymbol{\lambda}^a). \end{aligned} \tag{3}$$

The N_s (here not necessarily distinct) samples $\mathbf{d}_{-k}^{(n)}$ in (3) are drawn from the distribution $P(\mathbf{d}_{-k}|\mathbf{y},\boldsymbol{\lambda}^a)$.The probability $P(\mathbf{d}_{-k}=\mathbf{d}_{-k}^{(n)}|\mathbf{y},\boldsymbol{\lambda}^a)$ is unknown, but can be efficiently modeled by a Markov chain whose initial state is randomly chosen [11]. We outline the procedure of drawing the samples $\mathbf{d}_{-k}^{(n)}$ according to the probability $P(\mathbf{d}_{-k}=\mathbf{d}_{-k}^{(n)}|\mathbf{y},\boldsymbol{\lambda}^a)$ in the following. In the literature this approach is known as Gibbs sampling.
Algorithm 1 (Gibbs sampling):

draw initial sample vector $\mathbf{d}^{(-N_b)}$ randomly
for $n=-N_b+1:N_s$
  draw sample $d_1^{(n)}$ from
    $P(d_1|d_2^{(n-1)},\ldots,d_K^{(n-1)},\mathbf{y},\boldsymbol{\lambda}^a)$
  draw sample $d_2^{(n)}$ from
    $P(d_2|d_1^{(n)},d_3^{(n-1)},\ldots,d_K^{(n-1)},\mathbf{y},\boldsymbol{\lambda}^a)$
  ⋮
  draw sample $d_K^{(n)}$ from
    $P(d_K|d_1^{(n)},\ldots,d_{K-1}^{(n)},\mathbf{y},\boldsymbol{\lambda}^a)$
end

After one pass (one iteration) in the for-loop, K samples $d_k^{(n)}$ ($k=1,\ldots,K$) are drawn, i.e., a complete sample vector $\mathbf{d}^{(n)}$ is available. Each time a single sample $d_k^{(n)}$ is drawn, it influences the probabilities of all other samples to be drawn, since they are conditioned on it. After a sufficiently large number N_b of initial draws, the burn-in period, the Markov chain will converge to a stationary distribution that is the desired distribution $P(\mathbf{d}_{-k}|\mathbf{y},\boldsymbol{\lambda}^a)$ [14]. Thus, the samples $\mathbf{d}_{-k}^{(n)}$ (the interfering symbols) follow the distribution $P(\mathbf{d}_{-k}|\mathbf{y},\boldsymbol{\lambda}^a)$ without requiring an explicit calculation of the probability $P(\mathbf{d}_{-k}=\mathbf{d}_{-k}^{(n)}|\mathbf{y},\boldsymbol{\lambda}^a)$. After the burn-in period, N_s sample vectors $\mathbf{d}^{(n)}$ are drawn according to Algorithm 1. Gibbs sampling and the evaluation of (3) involves the calculation of the probability $\eta_k^{(n)} \triangleq P(d_k=+1|\mathbf{y},\mathbf{d}_{-k}=\mathbf{d}_{-k}^{(n)},\boldsymbol{\lambda}^a)$ given by

$$\begin{aligned} \eta_k^{(n)} &= 1/\big(1+\exp\big(-\lambda_k^{(n)}\big)\big)\\ \lambda_k^{(n)} &= (2/\sigma_n^2)\,\mathbf{s}_k^T(\mathbf{y}-\mathbf{S}_{-k}\mathbf{d}_{-k}^{(n)})+\lambda_k^a. \end{aligned} \tag{4}$$

The matrix $\mathbf{S}_{-k}$ is obtained from $\mathbf{S}$ by removing its kth column. We refer to the MCMC detection algorithm according to (3), Algorithm 1, and (4) as MCMC-I.

An improved version of MCMC-I based on importance sampling has been devised in [14]. There, the APPs are approximated by (5), see top of next page. The samples $\mathbf{d}_{-k}^{(n)}$ are obtained from Gibbs sampling according to Algorithm 1, but only L distinct samples $\mathbf{d}_{-k}^{(n)}$ are used for the evaluation of (5). The pdf $f(\mathbf{y}|\mathbf{d}_{-k}=\mathbf{d}_{-k}^{(n)},d_k=d)$ is multivariate Gaussian with mean $\mathbf{S}_{-k}\mathbf{d}_{-k}^{(n)}+\mathbf{s}_k d$ and covariance matrix $\sigma_n^2\mathbf{I}_N$. The probability $P(\mathbf{d}_{-k}=\mathbf{d}_{-k}^{(n)},d_k=d|\boldsymbol{\lambda}^a)$ represents the a-priori information that is easily evaluated as a K-fold product using the LLRs $\boldsymbol{\lambda}^a$. We refer to this second approach involving Algorithm 1 and (5) as MCMC-II. We remark that if the list in (5) was complete, i. e. $L=2^{K-1}$, (5) represented the optimal APP rule. Thus, MCMC-II might be interpreted as special type of a list detection algorithm. In contrast to MCMC-I no burn-in period is required. The extrinsic output LLRs of MCMC-I and MCMC-II, respectively, are finally given by (using either the approximation (3) or (5))

$$\lambda_k^D = \ln\frac{P(d_k=+1|\mathbf{y},\boldsymbol{\lambda}^a)}{P(d_k=-1|\mathbf{y},\boldsymbol{\lambda}^a)} - \lambda_k^a. \tag{6}$$

4.2 MCMC Techniques with Forced State Transitions

In order to maintain a moderate computational complexity for both MCMC-I and MCMC-II one should use a small number of samples N_s. However, at high SNR the transition probabilities in the Markov chain approach extreme values, thus, not all "important" candidate samples $\mathbf{d}_{-k}^{(n)}$ are likely to be drawn/found by Gibbs sampling. In [14] the usage of multiple Markov chains in parallel is suggested that allows to cover more states (candidate samples). In this paper, we propose another approach by introducing *forced state transitions* (FST) in the Markov chain. This offers the advantage of still running only one Markov chain. Let us first introduce a quantity for the event that a certain bit $d_k^{(n)}$ to be drawn by the Gibbs sampler will differ from the previous sample, i. e., that $d_k^{(n)} \neq d_k^{(n-1)}$. A suitable measure is the variance estimate v_k of d_k based on the probability $\eta_k^{(n-1)} = P(d_k=+1|\mathbf{y},\mathbf{d}_{-k}=\mathbf{d}_{-k}^{(n-1)},\boldsymbol{\lambda}^a)$. For clarity, we drop the sample index $(n-1)$ in the following and define $\check{\mathbf{d}}_{-k} \triangleq \mathbf{d}_{-k}^{(n-1)}$. Accordingly, v_k

$$P(d_k=+1|\mathbf{y},\boldsymbol{\lambda}^a) \approx \frac{\sum_{n=1}^{L} f(\mathbf{y}|\mathbf{d}_{-k}=\mathbf{d}_{-k}^{(n)}, d_k=+1)\, P(\mathbf{d}_{-k}=\mathbf{d}_{-k}^{(n)}, d_k=+1|\boldsymbol{\lambda}^a)}{\sum_{n=1}^{L}\sum_{d=\pm1} f(\mathbf{y}|\mathbf{d}_{-k}=\mathbf{d}_{-k}^{(n)}, d_k=d) P(\mathbf{d}_{-k}=\mathbf{d}_{-k}^{(n)}, d_k=d|\boldsymbol{\lambda}^a)}. \quad (5)$$

is computed by

$$v_k = \sum_{d=\pm1} (d-\mu_k)^2 P(d_k=d|\mathbf{y},\check{\mathbf{d}}_{-k},\boldsymbol{\lambda}^a), \quad (7)$$

where the mean μ_k is given by $\mu_k = 2\eta_k - 1$. Thus, v_k simplifies to

$$v_k = 4\eta_k(1-\eta_k). \quad (8)$$

Large values of v_k indicate a large probability that the kth bit will change, small values that it is unlikely to change. At high SNR some of the bits appear to be very stable (small v_k), since the Markov chain is likely to be trapped in a biased distribution. However, this distribution can be quite different from the target distribution $P(\mathbf{d}_{-k}|\mathbf{y},\boldsymbol{\lambda}^a)$ we are interested in, since the number of samples N_s might be not sufficiently large. In order to increase the chance to overcome local maxima of the sampling distribution and visit states in the Markov chain that otherwise would require a huge number of samples to be drawn, we introduce additional state transitions (FST). Therefore, we flip bits that did not change over the last m samples to force the Markov chain to cover more states and thus increase its efficiency in searching for candidate samples $\mathbf{d}_{-k}^{(n)}$. As a criterion which bit should be flipped we observe the variance estimate v_k for all users $k = 1, \ldots, K$. We consider two possibilities:

- Change the bit of user k that is most likely to be changed in the next iteration, i. e., that has largest v_k.
- Change the bit of user k that is least likely to be changed in the next iteration, i. e., that has smallest v_k.

Interestingly, both approaches can potentially increase the BER performance dramatically. The first criterion offers the advantage that less state changes in total will occur (less complex), whereas the second one yields a lower BER. Therefore, we will consider the second criterion only in the following. Numerical results indicate that in contrast to MCMC-II, MCMC-I does not benefit from additional FST. Thus, we combine only MCMC-II with FST and term the new scheme MCMC-II-FST.

5 Numerical Results

Next, we assess the performance of our proposed method by some numerical examples. Unless stated otherwise, the following parameters are valid for all simulation results. We consider a doubly loaded synchronous CDMA system with $K = 16$ users and spreading factor $N = 8$. As channel code we apply a terminated, non-systematic convolutional code with code rate $R_c = 1/2$, memory 2, and generator polynomials $[1+D^2, 1+D+D^2]$. The number of information bits per codeword is $N_q = 510$, yielding $N_c = 1024$ code bits per codeword. We use K random permutors for interleaving that are chosen once and then kept fixed throughout the whole simulation. The number of turbo iterations is 10. We employ APP channel decoding with the BCJR algorithm [15]. All E_b/N_0 values refer to the energy per information bit. We apply $N_s = 30$ samples (number of sample vectors within the Gibbs sampler) in total, i. e., if multiple Markov chains are employed the number of samples per chain is reduced accordingly. This allows a fair comparison. In the case of MCMC-I, in addition, a burn-in period of $N_b = 30$ samples is applied to allow the Markov chain to converge to a stationary distribution. The FST parameter m (memory) equals 10 when MCMC-II-FST is employed.

Fig. 3 shows a comparison of the BER performance of the proposed MCMC methods and standard turbo multiuser detection techniques. The conventional approaches MCMC-I and MCMC-II with 1 Markov chain fail to converge to a low BER at high SNR, since the number of samples N_s is too small. Introducing either multiple Markov chains (here, 3 chains) or FST (here, $m = 10$) a dramatic improvement and convergence to single-user performance between 5 and 6 dB occurs. Both approaches allow the Gibbs sampler to visit more distinct states and thus obtain a more reliable list of candidate vectors, which is crucial for soft-output calculation in iterative decoding schemes. As a reference, we include the results of the MMSE-PIC and the MF-SIC. The results of the MF-PIC are omitted in Fig. 3 due to a high bit error floor.

Fig. 4 shows the influence of the memory parameter m on the BER performance, using MCMC-II-FST. For values of m between 4 and 10 (not all depicted) similar results are obtained. If m is further increased the influence of FST become negligible and an error floor occurs ($m = 16$). Interestingly, at low SNR the performance for $m = 16$ is still close to the single-user case. We remark that computational complexity of the scheme becomes larger for smaller values of m due to the more frequent bit flipping by FST. Fig. 5 depicts the performance of the MCMC-II when multiple Markov chains are used, either with or without additional FST. As expected, the BER results improve when the number of Markov chains is small and additional bit flipping is applied allowing to visit more states in the Markov chains. However, if the number of samples per chain is

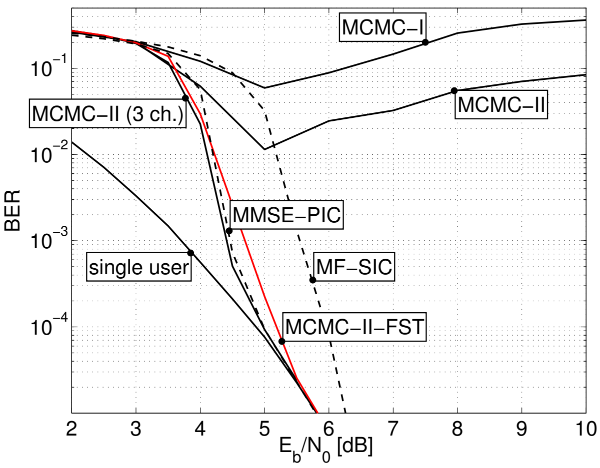

Fig. 3. Coded BER vs. E_b/N_0. Doubly loaded system, $K = 16$, $N = 8$. $N_s = 30$ samples (1×30 or 3×10). MCMC-II-FST: $m = 10$. 10 turbo iterations.

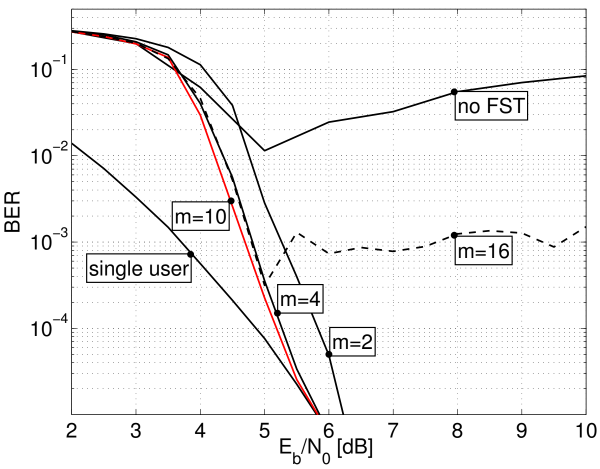

Fig. 4. Influence of parameter m on coded BER of MCMC-II-FST. Doubly loaded system, $K = 16$, $N = 8$. $N_s = 30$ samples, 1 Markov chain. 10 turbo iterations.

small (4 chains, i. e., 6 samples per chain), additional FST are not helpful, the performance becomes even worse. Fig. 6 depicts the coded BER performance of the MMSE-PIC, MF-SIC, MCMC-II with 3 Markov chains, and MCMC-II-FST (1 Markov chain) versus number of iterations at fixed $E_b/N_0 = 5\,\text{dB}$, for $K = 14$ users and spreading factor $N = 8$. MCMC-II with 3 Markov chains and MCMC-II-FST show a similar performance as the MMSE-PIC. The MCMC algorithms require about 3 iterations less than the MF-SIC.

Figs. 7 and 8 depict the average number of updates in the Gibbs sampler and average list size L, respectively, versus the iterations. An update in the Gibbs sampler is caused every time a certain sample value is distinct from the previous value ($d_k^{(n)} \neq d_k^{(n-1)}$). The number of updates and the list size can be interpreted as a measure for computational complexity. The remaining simulation parameters were chosen according to Fig. 3. Both MCMC-II with 3 Markov chains and MCMC-II-FST require more updates than MCMC-II with 1 Markov chain. As iteration proceeds, the convergence of the Gibbs sampler towards a stationary distribution is faster, since more reliable a-priori information is present, thus the number of updates decreases. In the beginning, MCMC-II-FST requires more updates than MCMC-II with 3 chains, however, at the last iterations the situation is vice versa. We observe that the average list size L of MCMC-II with 3 Markov chains and MCMC-II-FST is larger than that of MCMC-II with only 1 chain in early iterations. Interestingly, the list size of MCMC-II-FST is always larger than that of MCMC-II with 3 Markov chains, however, the performance of the latter one is slightly better. This indicates that not only the size of the list is crucial for performance but also which candidate vectors are included. With increasing iterations, i. e., more reliable channel decoders' feedback, the list size becomes much smaller, since less distinct candidates are found by the Gibbs sampler (a-priori information is dominating).

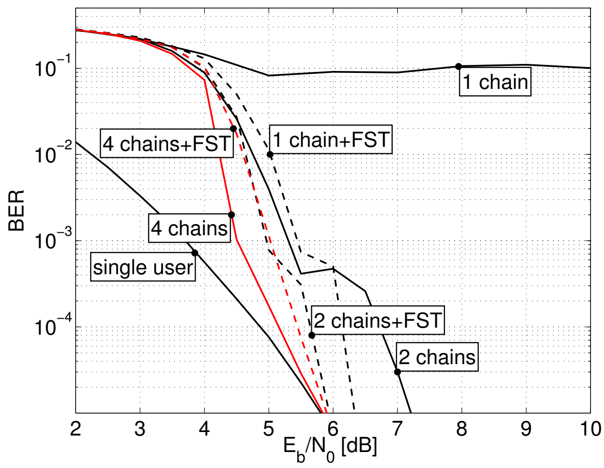

Fig. 5. Influence of multiple Markov chains on coded BER of MCMC-II. Doubly loaded system, $K = 16$, $N = 8$. $N_s = 20$ samples. MCMC-II-FST: $m = 4$. 10 turbo iterations.

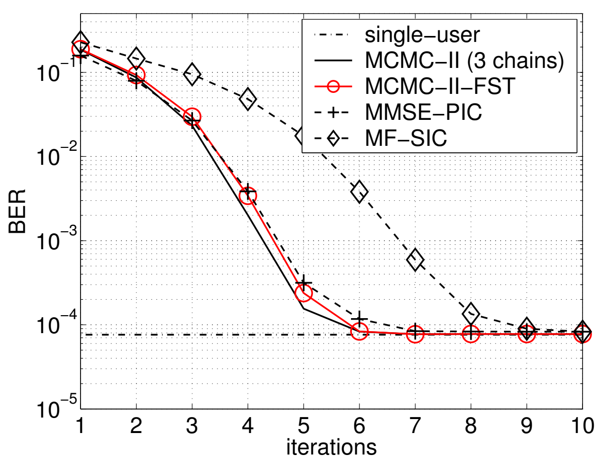

Fig. 6. Coded BER vs. number of iterations. System load $\beta = 1.75$, $K = 14$, $N = 8$. $E_b/N_0 = 5\,\text{dB}$. $N_s = 30$ samples.

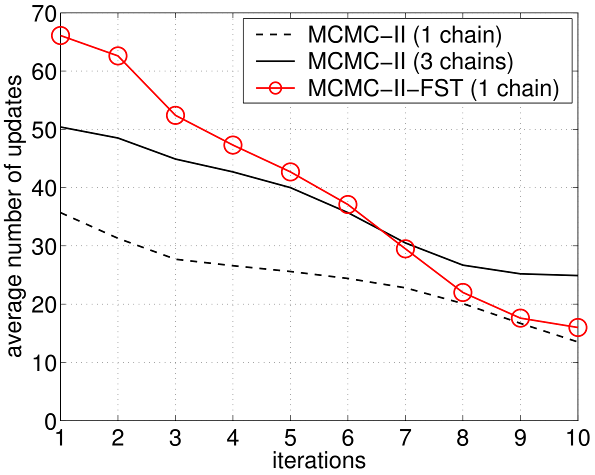

Fig. 7. Average number of updates vs. iterations. $E_b/N_0 = 5\,\text{dB}$.

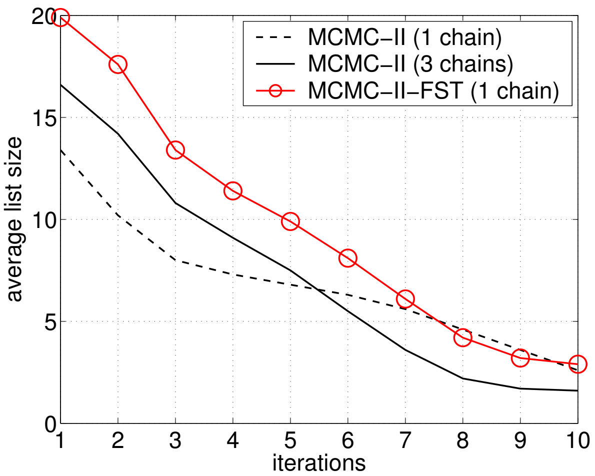

Fig. 8. Average list size vs. iterations. $E_b/N_0 = 5\,\text{dB}$.

A detailed study of computational complexity of the MCMC algorithms is beyond the scope of this paper. However, we remark that the computational complexity depends on both the list size L and the number of updates U, making an exact analysis difficult. Results not included in this paper indicate that by setting an upper limit for L and U the expected number of arithmetic operations can be made smaller than for the MMSE-PIC approach without sacrificing too much performance.

6 Conclusions

For high SNR and high system load conventional MCMC algorithms may suffer from a performance degradation. We presented forced state transitions as an efficient means to enhance performance of MCMC detection schemes, allowing a performance close to the MMSE-PIC. We remark that the proposed algorithm can be applied to other types of vector-valued transmission schemes [16] as well, e. g. , MIMO and multicarrier systems. The application to the asynchronous CDMA case is possible by formulating the algorithm as a sliding window version which has been suggested for the MMSE-PIC in [5].

Acknowledgement

National ICT Australia is funded through the Australian Government's *Backing Australia's Ability* initiative and in part through the Australian Research Council. Partial financial support by the German research foundation (Deutsche Forschungsgemeinschaft) under grant No. 447 AUS-111/1/05 is gratefully acknowledged.

References

[1] S. Verdú, *Multiuser Detection*. Cambridge: Cambridge University Press, 1998.

[2] M. Moher, "An iterative multiuser decoder for near-capacity communications," *IEEE Trans. Commun.*, vol. 47, pp. 870–880, July 1998.

[3] P. D. Alexander, A. J. Grant, and M. C. Reed, "Iterative detection in code-division multiple-access with error control coding," *European Trans. Telecommun.*, vol. 9, no. 5, pp. 419–426, Sept./Oct. 1998.

[4] 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.

[5] 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, July 1999.

[6] T. M. Schmidl, A. Gatherer, X. Wang, and R. Chen, "Interference cancellation using the Gibbs sampler," in *Proc. IEEE VTC-Fall*, vol. 1, Boston (MA), USA, Sept. 2000, pp. 429–433.

[7] V. Buchoux, O. Cappé, and E. Moulines, "Turbo multiuser detection for coded DS-CDMA systems: A Gibbs sampling approach," in *Proc. Asilomar Conf. on Signals, Systems, and Computers*, Pacific Grove (CA), USA, Oct./Nov. 2000, pp. 1426–1430.

[8] 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.

[9] B. Dong and X. Wang, "Sampling-based soft equalization for frequency-selective MIMO channels," *IEEE Trans. Commun.*, vol. 53, no. 2, pp. 278–288, Feb. 2005.

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

[11] Z. Shi, H. Zhu, and B. Farhang-Boroujeny, "Markov chain Monte Carlo techniques in iterative detectors: a novel approach based on Monte Carlo integration," in *Proc. IEEE Globecom*, vol. 1, Dallas (TX), USA, Nov./Dec. 2004, pp. 325–329.

[12] W. G. Teich and M. Seidl, "Code division multiple access communications: Multiuser detection based on a recurrent neural network structure," in *Proc. IEEE ISSSTA*, vol. 3, Mainz, Germany, Sept. 1996, pp. 979–984.

[13] M. A. Dangl, C. Sgraja, W. G. Teich, J. Lindner, and J. Egle, "Convergence behavior of iterative equalization and decoding schemes with memory," in *Proc. IEEE Globecom*, vol. 3, San Francisco (CA), USA, Dec. 2003, pp. 1710–1715.

[14] B. Farhang-Boroujeny, H. Zhu, and Z. Shi, "Markov chain Monte Carlo algorithms for CDMA and MIMO communication systems," *IEEE Trans. Signal Processing*, 2005, accepted for publication.

[15] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory*, vol. 20, pp. 284–287, Mar. 1974.

[16] J. Lindner, "MC-CDMA in the context of general multiuser/multisubchannel transmission methods," *European Trans. Telecommun.*, vol. 10, no. 5, pp. 351–367, July/Aug. 1999.