This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE ICC 2010 proceedings

Performance Evaluation of the Markov Chain Monte Carlo MIMO Detector based on Mutual Information

Martin Senst and Gerd Ascheid
Institute for Integrated Signal Processing Systems
RWTH Aachen University
Templergraben 55, 52056 Aachen, Germany
{msenst, ascheid}@iss.rwth-aachen.de

Helge Lüders
Institute of Communication Systems and Data Processing
RWTH Aachen University
Templergraben 55, 52056 Aachen, Germany
lueders@ind.rwth-aachen.de

***Abstract*—Recently, MIMO detectors which are based on Markov Chain Monte Carlo (MCMC) simulation techniques have been proposed as alternatives to, e.g., the well-known sphere detector. In this paper, we present a systematic analysis of the performance of MCMC detectors. We study the impact of several parameters such as the list size and the number of independently running chains. As a performance criterion, our analysis is based on the mutual information over an equivalent modulation channel, rather than on coded bit error rates, because this metric is independent of the outer channel code and provides valuable insights over the whole SNR range of interest. Furthermore, we show that combining the MCMC detector with a hard-output sphere detector removes the error floor at high SNR, which is a well-known problem of the MCMC principle.**

I. Introduction

The multiple-input multiple-output (MIMO) technique is an interesting option for future wireless communication systems. By *spatial multiplexing*, i.e., transmitting several data streams in parallel, MIMO systems can achieve significantly higher data rates compared to single-antenna systems, without an increase in bandwidth or transmit power [1]. Because of its promising performance, spatial multiplexing is an important component of the next-generation wireless communication standards IEEE 802.11n and 3GPP LTE.

Due to the exponential complexity (in the number of transmitted bits per channel use) of the optimal MIMO detector, a lot of work has been devoted to the study of low-complexity alternatives. In particular, many different flavors of the so-called *sphere detector* have been proposed, which conduct a tree search for the most likely data symbols [2], [3].

In [4], a fundamentally different approach to this search has been proposed. The idea is to simulate stochastic processes, which output candidate symbols according to their posterior probability (i.e., conditioned on the channel observation). If these processes are run sufficiently long, a list of candidates is obtained which containes—with high probability—the symbol with largest posterior probability. However, as opposed to a deterministic tree search, there is no guarantee that this symbol is found. Since the stochastic processes are implemented by Markov chains, this technique is referred to as *Markov Chain Monte Carlo* (MCMC) [5].

This work has been supported by the UMIC Research Centre, RWTH Aachen University.

In the present work, we carry out a systematic analysis of the average performance that can be expected by MCMC MIMO detectors. We quantify the performance in terms of the mutual information over an equivalent modulation channel. Compared to bit error rates, this criterion has the advantage of being independent of the outer channel code, and it provides meaningful results for the whole SNR range of interest.

The paper is structured as follows: Section II introduces the system model and defines the problem of soft-output MIMO detection. Section III discusses the MCMC detection algorithm, followed by the definition of our performance criterion in Section IV. In Section V we present the results of our study, with a focus on both the medium and the high SNR regime. Finally, Section VI concludes the paper.

II. System Model

A. Transmitter and Channel Model

We consider an $N_t \times N_r$ MIMO system based on bit-interleaved coded modulation (BICM), as depicted in Fig. 1. A vector $\boldsymbol{b} \in \mathbb{F}_2^{N_b}$ of N_b information bits is encoded by a channel code of rate $r = N_b/N_c$ and interleaved by a random interleaver, yielding the vector $\boldsymbol{c} \in \mathbb{F}_2^{N_c}$. The code bits are demultiplexed into N_t spatial streams, where N_t is equal to the number of transmit antennas. In each stream, groups of $\log_2 M$ bits are mapped to complex data symbols from the alphabet $\mathcal{A} \subset \mathbb{C}$ with $|\mathcal{A}| = M$. We denote the total number of code bits per transmitted data vector as $K = N_t \log_2 M$, and the data rate in [bit / channel use] as $R = rK$.

The data vectors $\boldsymbol{x}_n$ are transmitted over a frequency-flat, spatially uncorrelated Rayleigh fading channel, described by the matrix $\boldsymbol{H} \in \mathbb{C}^{N_r \times N_t}$. After A/D conversion at the receiver, the vector of received data samples can be written as

$$\boldsymbol{y}_n = \boldsymbol{H}_n \boldsymbol{x}_n + \boldsymbol{w}_n \tag{1}$$

where $\boldsymbol{w}_n$ is zero-mean AWGN with $\mathbb{E}[\boldsymbol{w}_n \boldsymbol{w}_n^H] = N_0 \boldsymbol{I}_{N_r}$.

We denote $\mathbb{E}[\|\boldsymbol{x}\|_2^2] = E_s$ and assume the normalization $\mathbb{E}[|h_{ij}|^2] = 1$, where h_{ij} is the (i,j)th entry of $\boldsymbol{H}$. Thus, the average signal-to-noise ratio is given as $\mathsf{SNR} = E_s/N_0$.

Furthermore, we assume that the fading process is ergodic, and that perfect channel state information is available at the receiver.

978-1-4244-6404-3/10/$26.00 ©2010 IEEE

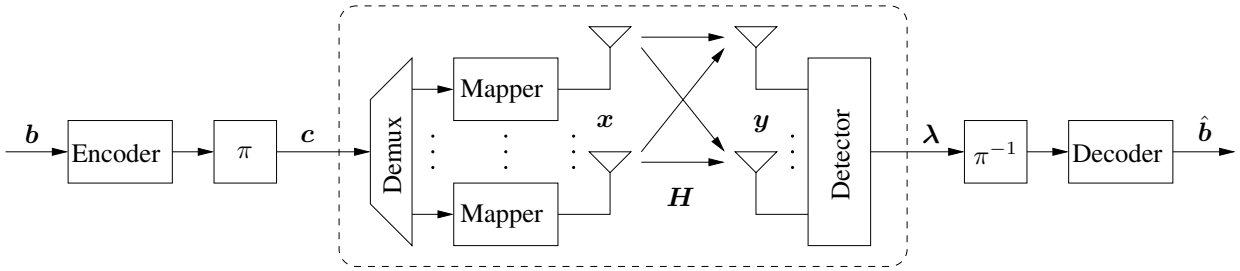

Fig. 1. System model. The dashed box highlights the binary-input, continuous-output modulation channel

B. MIMO Detection

The received vectors $\boldsymbol{y}_n$ are processed by the MIMO detector, which calculates (approximations of) the posterior L-values of the code bits, defined as[1]

$$\lambda_{n,k} \triangleq \log \frac{P(c_{n,k}=0|\boldsymbol{y}_n)}{P(c_{n,k}=1|\boldsymbol{y}_n)}\,. \tag{2}$$

Here $\boldsymbol{c}_n$ is the vector of the K code bits that are mapped to $\boldsymbol{x}_n$, and $c_{n,k}$ with $k=0,\ldots,K-1$, is its kth entry.

Since the MIMO detector processes the data vectors independently of each other, we drop the time index n for convenience.

With $\boldsymbol{x}(\boldsymbol{c})$ denoting the data vector $\boldsymbol{x}$ associated to the bit label $\boldsymbol{c}$, and because $P(\boldsymbol{c}) = 2^{-K}\ \forall \boldsymbol{c}$ due to a lack of prior information, (2) can be written as

$$\begin{aligned}
\lambda_k &= \log \frac{\sum_{\boldsymbol{c}:c_k=0} P(\boldsymbol{c}|\boldsymbol{y})}{\sum_{\boldsymbol{c}:c_k=1} P(\boldsymbol{c}|\boldsymbol{y})} && (3)\\
&= \log \frac{\sum_{\boldsymbol{c}:c_k=0} p(\boldsymbol{y}|\boldsymbol{c})}{\sum_{\boldsymbol{c}:c_k=1} p(\boldsymbol{y}|\boldsymbol{c})} && (4)\\
&= \log \frac{\sum_{\boldsymbol{c}:c_k=0} \exp(-N_0^{-1}\|\boldsymbol{y}-\boldsymbol{H}\boldsymbol{x}(\boldsymbol{c})\|_2^2)}{\sum_{\boldsymbol{c}:c_k=1} \exp(-N_0^{-1}\|\boldsymbol{y}-\boldsymbol{H}\boldsymbol{x}(\boldsymbol{c})\|_2^2)} && (5)\\
&\approx \max_{\boldsymbol{c}:c_k=0} \mu(\boldsymbol{c}) - \max_{\boldsymbol{c}:c_k=1} \mu(\boldsymbol{c}) && (6)
\end{aligned}$$

where we have defined the symbol metric[2]

$$\mu(\boldsymbol{c}) \triangleq -N_0^{-1}\|\boldsymbol{y}-\boldsymbol{H}\boldsymbol{x}(\boldsymbol{c})\|_2^2 = \log p(\boldsymbol{y}|\boldsymbol{c}) + \text{const}\,. \tag{7}$$

The max-log approximation in (6) leads to a significant complexity reduction and is assumed by most MIMO detection algorithms.

We have thus shown that soft MIMO detection is equivalent (under the max-log approximation) to finding the two maxima of $\mu(\boldsymbol{c})$, where the kth bit of $\boldsymbol{c}$ is constrained to 0 and 1, respectively. One of these two maxima will be the same for all K bits of one symbol vector; it is equal to $\mu(\boldsymbol{c}^{\mathrm{ML}})$, where the maximum likelihood (ML) symbol

$$\boldsymbol{c}^{\mathrm{ML}} \triangleq \arg\max_{\boldsymbol{c}} \mu(\boldsymbol{c}) \tag{8}$$

is the solution of the unconstrained optimization problem. Thus, the soft detector needs to find $K+1$ symbols: The ML symbol, and for each bit $k=0,\ldots,K-1$ the best counter-hypothesis

$$\bar{\boldsymbol{c}}^{(k)} \triangleq \arg\max_{\boldsymbol{c}:c_k\neq c_k^{\mathrm{ML}}} \mu(\boldsymbol{c})\,. \tag{9}$$

The desired L-values are then given as

$$\lambda_k = \begin{cases} \mu(\boldsymbol{c}^{\mathrm{ML}}) - \mu(\bar{\boldsymbol{c}}^{(k)}) & \text{if } c_k^{\mathrm{ML}}=0 \\ \mu(\bar{\boldsymbol{c}}^{(k)}) - \mu(\boldsymbol{c}^{\mathrm{ML}}) & \text{if } c_k^{\mathrm{ML}}=1 \end{cases}. \tag{10}$$

III. Markov Chain Monte Carlo MIMO Detection

Since the number of candidate symbols increases exponentially in K, a brute force search for the maximizers of (8) and (9) is only possible for very small K. An interesting question is therefore how to solve these optimization problems with feasible computational complexity.

One approach that has gained a lot of attention is the well-known sphere-detector. By means of a QR-decomposition of the channel matrix, the problem of finding candidate vectors with large metrics $\mu(\boldsymbol{c})$ is transformed into a tree search where the leaves correspond to the candidate symbols. In [3], the authors propose an algorithm which finds the ML symbol as well as all counter-hypotheses within a single tree search. However, this approach still suffers from an exponential worst-case complexity, which necessitates heuristic complexity reductions, such as stopping the algorithm at a maximum number of visited nodes.

A. The Markov Chain Monte Carlo Principle

A different approach to solving (8) and (9) approximately, which has been proposed in [4], is based on the principle of Monte Carlo integration. The basic idea is as follows: Imagine we were able to draw candidate symbols $\boldsymbol{c}$ randomly according to their posterior distribution $P(\boldsymbol{c}|\boldsymbol{y})$. We could then create a list $\mathcal{L} = (\boldsymbol{c}^{(1)},\ldots,\boldsymbol{c}^{(L)})$ by sampling L times from $P(\boldsymbol{c}|\boldsymbol{y})$. By construction, most entries of $\mathcal{L}$ will have a large metric $\mu(\boldsymbol{c}^{(l)})$, and if L is chosen sufficiently large, $\boldsymbol{c}^{\mathrm{ML}} \in \mathcal{L}$ with high probability. If we now constrain our optimization problems to $\boldsymbol{c}\in\mathcal{L}$ we obtain approximations of λ_k whose quality increases with the list size L. This parameter therefore provides a natural trade-off between accuracy and computational complexity.

Unfortunately, obtaining samples from an arbitrary high-dimensional distribution is in general a hard problem. A

[1]For simplicity, we do not make the dependence on $\boldsymbol{H}_n$ explicit.

[2]Note that this definition is in contrast to the literature about sphere detection, where the symbol metric is usually defined as the *negative* log-likelihood.

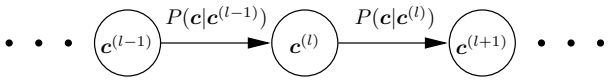

Fig. 2. Illustration of a Markov chain

promising approach is based on first order Markov processes, giving rise to the term *Markov Chain Monte Carlo*.

In the MCMC algorithm, a Markov chain is initialized with the state $\boldsymbol{c}^{(0)}$, which can be obtained by a "clever guess", or simply chosen at random. Then, in each time step l, a new state is drawn randomly from the transition kernel $\boldsymbol{c}^{(l)} \sim P(\boldsymbol{c}|\boldsymbol{c}^{(l-1)})$. If this kernel is chosen appropriately, then for $l \to \infty$, $P(\boldsymbol{c}^{(l)})$ converges to the desired distribution, independently of the initial state.

A possible realization of the transition kernel, which has been successfully applied to MIMO detection, is the Gibbs sampler, which we now briefly describe for our specific problem. A more general description of Monte Carlo statistical methods can be found in [5], [6].

Assume that we are at time step l. The Markov chain is currently in state $\boldsymbol{c}^{(l-1)}$ and we want to draw a new state $\boldsymbol{c}^{(l)} \sim P(\boldsymbol{c}|\boldsymbol{c}^{(l-1)}, \boldsymbol{y})$, where we have made the dependence on the channel observation explicit. To proceed, the Gibbs sampler iterates over the bits $k = 0, \ldots, K-1$ and draws each new bit according to the conditional distribution

$$c_k^{(l)} \sim P(c_k|c_0^{(l)}, \ldots, c_{k-1}^{(l)}, c_{k+1}^{(l-1)}, \ldots, c_{K-1}^{(l-1)}, \boldsymbol{y}) . \tag{11}$$

An efficient log-domain implementation of (11) has been proposed in [7]. With the abbreviation

$$\boldsymbol{c}_{\setminus k}^{(l)} \triangleq (c_0^{(l)}, \ldots, c_{k-1}^{(l)}, c_{k+1}^{(l-1)}, \ldots, c_{K-1}^{(l-1)}) \tag{12}$$

we can describe the probability in (11) via the L-value

$$\gamma_k^{(l)} \triangleq \log \frac{P(c_k = 0|\boldsymbol{c}_{\setminus k}^{(l)}, \boldsymbol{y})}{P(c_k = 1|\boldsymbol{c}_{\setminus k}^{(l)}, \boldsymbol{y})} \tag{13}$$

$$= \log \frac{p(\boldsymbol{y}|\boldsymbol{c}_{\setminus k}^{(l)}, c_k = 0)}{p(\boldsymbol{y}|\boldsymbol{c}_{\setminus k}^{(l)}, c_k = 1)} \tag{14}$$

$$= N_0^{-1}(\|\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x}_1\|_2^2 - \|\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x}_0\|_2^2) \tag{15}$$

where $\boldsymbol{x}_\beta$ denotes the symbol vector with the bit label $(\boldsymbol{c}_{\setminus k}^{(l)}, c_k = \beta)$. Since the labels of the vectors $\boldsymbol{x}_0$ and $\boldsymbol{x}_1$ differ in only one bit, the vectors themselves differ in only one entry, which permits an efficient numerical evaluation of (15).

B. Running Several Chains in Parallel

As noted above, the Gibbs sampler produces candidate vectors with the correct distribution as the list size L tends to infinity. However, due to the Markov characteristic, the generated samples are not uncorrelated, and the chain might need to be run for a long time in order to produce samples from all regions with significant probability mass. In order to reduce this time, and hence the computational complexity, it has been proposed to run several Markov chains, initialized with independently chosen states, in parallel [8]. Obviously, the states of different chains will be mutually independent, leading to a faster exploration of the probability space.[3]

In [9] the idea of running several Markov chains is developed one step further. The authors propose that, after an estimate $\hat{\boldsymbol{c}}^{\mathrm{ML}}$ of the ML symbol has been obtained, K parallel *constrained* Gibbs samplers should be run, where the kth bit of the kth sampler is fixed to $1 - \hat{c}_k^{\mathrm{ML}}$. These chains are thus devoted to the search for the kth counter-hypothesis, while the unconstrained chains are used for finding the ML symbol. The rationale is that, especially at high SNR, the transition probability in (11) can become very low. In this case, the kth bit of the chain keeps its value for a long time, and the list

$$\mathcal{L}_k \triangleq \mathcal{L} \cap \{\boldsymbol{c} : c_k \neq \hat{c}_k^{\mathrm{ML}}\} \tag{16}$$

that is used for solving (9) can become very short, or even empty. The adoption of Markov chains with constrained Gibbs samplers, however, guarantees a minimum list size $|\mathcal{L}_k| > 0$.

In the following, we refer to the unconstrained Markov chains as *ML-chains*, and to the constrained ones as *C-chains*.

IV. The Equivalent Modulation Channel

In most publications, the performance of MIMO detectors is measured in terms of bit error rates. This approach, however, has the drawback that it is dependent on the specific channel code and the applied decoding algorithm. In a recent paper [10], the authors propose to benchmark MIMO detectors based on the mutual information over an equivalent modulation channel. This equivalent channel, marked in Fig. 1 with a dashed box, consists of the symbol mapper, the physical channel, and the detector, i.e., its input are code bits c_k and its output are real-valued log-likelihood ratios λ_k. The average mutual information $I(C;\Lambda)$ over this channel depends only on the number of antennas, the modulation alphabet, the physical channel, and the detection algorithm; in particular, it is independent of the outer channel code.

With $P(c) = 0.5, c \in \{0, 1\}$, the mutual information of the modulation channel can be calculated as follows:

$$I(C;\Lambda) = \mathbb{H}(C) - \mathbb{H}(C|\Lambda) \tag{17}$$

$$= 1 - \mathbb{E}_{C,\Lambda}\left[\log_2 \frac{1}{P(c|\lambda)}\right] \tag{18}$$

$$= 1 - \mathbb{E}_{C,\Lambda}\left[\log_2 \underbrace{\frac{p(\lambda|C=0) + p(\lambda|C=1)}{p(\lambda|C=c)}}_{\alpha}\right] . \tag{19}$$

The argument of the logarithm is

$$\alpha = \begin{cases} 1 + \frac{p(\lambda|C=1)}{p(\lambda|C=0)} = 1 + e^{-\lambda} & \text{if } c = 0 \\ 1 + \frac{p(\lambda|C=0)}{p(\lambda|C=1)} = 1 + e^{\lambda} & \text{if } c = 1 \end{cases} \tag{20}$$

which leads to

$$I(C;\Lambda) = 1 - \mathbb{E}_{C,\Lambda}\left[\log_2\left(1 + e^{(2c-1)\lambda}\right)\right] . \tag{21}$$

[3] A further advantage of running independent chains is that it introduces parallelism into the algorithm, which can be exploited in implementations.

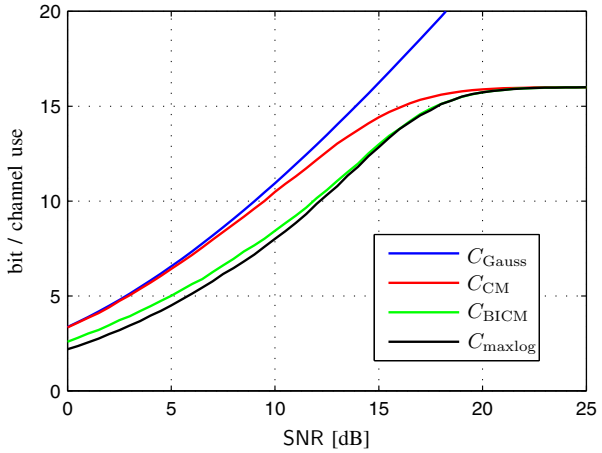

Fig. 3. Capacity of a 4×4 MIMO system under various constraints

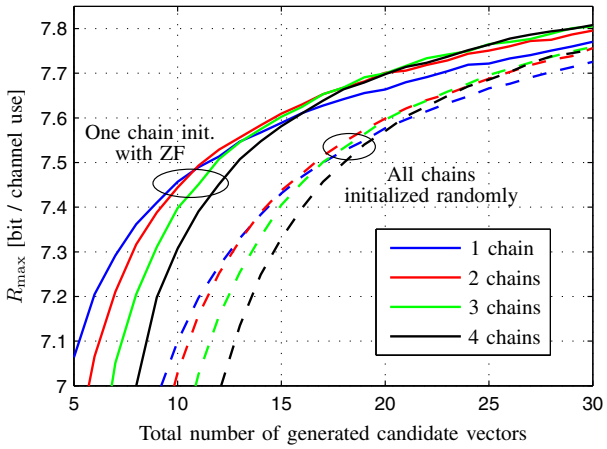

Fig. 4. Achievable data rate as a function of the number of candidate vectors that are generated by the ML-chains, for different number of independently running chains, at SNR = 10 dB, using 2 C-chains per bit of length 3 each

For most cases of practical interest, there is no analytical expression available for (21), but the expectation can easily be evaluated by Monte Carlo simulations.

An intuitive interpretation of this quantity is that it provides an upper bound on the data rate which can be achieved using a specific detection algorithm,

$$R \leq R_{\text{max}} \triangleq K \cdot I(C;\Lambda)\,. \tag{22}$$

V. Performance Evaluation

In the following, we examine the performance of the MCMC detector based on R_{max}. We study the trade-off between accuracy and computational complexity which can be tuned by varying the number of Markov chains and the number of generated candidate vectors, as well as different initialization strategies. We compare the achieved performance with the upper bound that is obtained by solving the optimization problems (8) and (9) exactly, denoted as C_{maxlog} in the following.

As a running example, we consider a 4×4 MIMO system using a 16-QAM alphabet with Gray mapping, i.e., there are $K = 16$ code bits per transmitted symbol vector. In order to get an idea of the achievable rates that we should expect, we present in Fig. 3 system capacities under various constraints. The plot C_{Gauss} shows the maximum rate under a Gaussian input distribution, which is equal to the information theoretic channel capacity. C_{CM} is the achievable rate under the constraint of a 16-QAM alphabet per antenna. As expected, this curve converges to a rate of 16 bit / channel use. The plot C_{BICM} further constrains the receiver to BICM, meaning that the detector outputs bitwise L-values according to (2). The last curve finally is the maximum achievable rate under the max-log approximation (6), which constitutes an upper bound to the performance of the MCMC detector.

A. Medium SNR Range

In this section, we study the performance of the MCMC detector at an SNR of 10 dB. At this point, C_{maxlog} equals almost exactly 8 bit / channel use, which corresponds to a code rate of $r = 1/2$.

Fig. 4 shows the achievable rate as a function of the list size L. For the different plots, $1, \ldots, 4$ parallel ML-chains have been used. In order to allow for a fair comparison, the x-axis shows the total number of generated samples (i.e., not the length of a single chain). Furthermore, for the plots drawn by solid lines, one chain was initialized by the hard ZF estimate (i.e., the zero-forcing estimate $\boldsymbol{H}^{-1}\boldsymbol{y}$, quantized elementwise to the closest constellation point), while the other chains were initialized uniformly at random. For the plots drawn by dashed lines, all chains were initialized randomly. It is evident that the initialization of one chain to the ZF estimate[4] results in a significant performance gain. If, however, the list size is large enough ($L \gtrapprox 10$), having other chains which are initialized randomly becomes beneficial.

In the following, we will use 4 parallel chains, each of length 5, as this choice realizes a good compromise between accuracy and complexity.

Fig. 5 presents an analogous experiment for the C-chains. Here, for each bit k we initialized one chain to

$$\arg \max_{\boldsymbol{c} \in \mathcal{L}_k^{\text{init}}} \mu(\boldsymbol{c})\,, \tag{23}$$

where the list of candidate vectors is defined as

$$\mathcal{L}_k^{\text{init}} \triangleq (\mathcal{L}^{\text{ML}} \cap \{\boldsymbol{c} : c_k \neq \hat{c}_k^{\text{ML}}\}) \cup \{\hat{\boldsymbol{c}}_{\sim k}^{\text{ML}}\}\,. \tag{24}$$

Here, $\mathcal{L}^{\text{ML}}$ is the list generated by the ML-chains, and $\hat{\boldsymbol{c}}_{\sim k}^{\text{ML}}$ is the ML estimate, but with the kth bit flipped. Including this symbol in the list $\mathcal{L}_k^{\text{init}}$ ensures that it is nonempty. As above, the other chains are initialized uniformly at random.

Again, a smooth trade-off between accuracy and complexity is visible. We choose to use 2 C-chains per bit, each of length 3, in the following. With this choice, the achievable rate is

[4] We also used the hard MMSE estimate in our experiments, but the resulting performance was virtually the same.

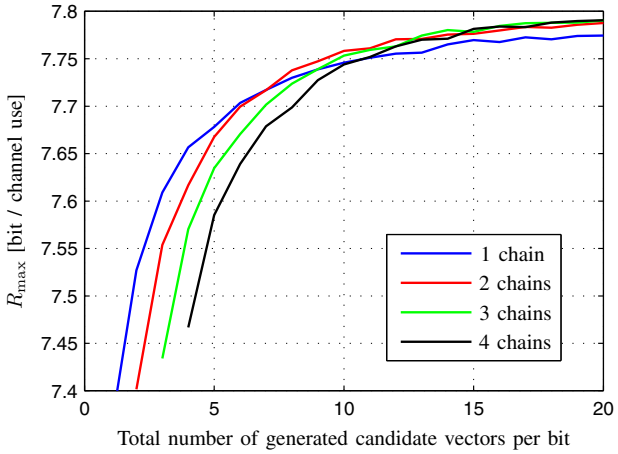

Fig. 5. Achievable data rate as a function of the number of candidate vectors per bit generated by the C-chains, for different number of independently running chains, at SNR = 10 dB, using 4 ML-chains of length 5 each

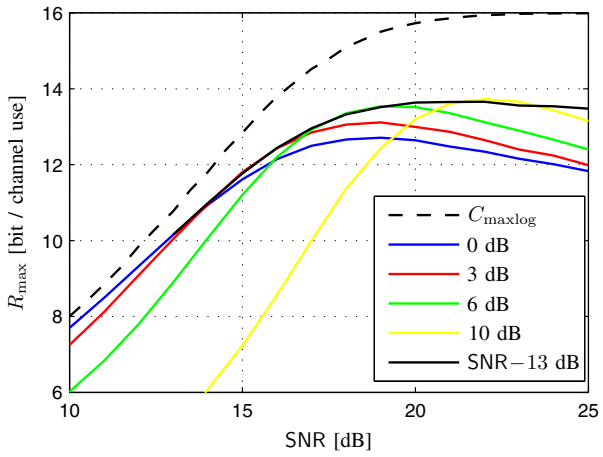

Fig. 6. Effect of increasing the noise power used by the Gibbs sampler. The legend shows the difference between the assumed and the true noise power

$R_{\max} \approx 7.7$ bit / channel use, which corresponds to a loss in SNR of about 0.4 dB. Of course, a system designer is free to make other choices, based on the available computational resources.

B. High SNR Range

It is well-known that the MCMC algorithm suffers from a significant performance degradation at high SNR. For example, consider the plot labeled with "0 dB" in Fig. 6, which shows $R_{\max}$ for the MCMC detector using 4 ML-chains of length 5 each, and 2 C-chains per bit of length 3 each (we will keep this parameterization in the following). It is evident that the rate fails to reach the maximum of 16 bit / channel use, and even degrades at high SNR.

This at first glance counter-intuitive behaviour can be explained by the inability of the Gibbs sampler to explore all regions of significant probability mass, due to very low transition probabilities, as explained in Section III-B. A possible counter-measure is an artificial increase of the transition probabilities, which can be achieved by assuming a larger noise power $\tilde{N}_0 > N_0$ in the Gibbs sampler (of course, the symbol metrics are still calculated based on the true N_0) [4]. In the statistical physics literature, this is referred to as increasing the *temperature* of the system; for example, in simulated annealing (an optimization heuristic) this technique is used to overcome local minima.

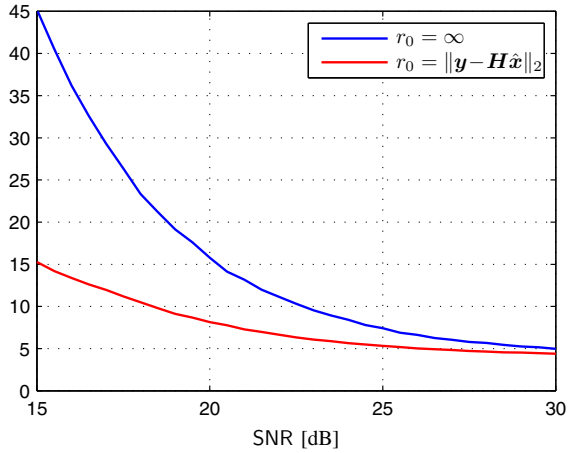

Fig. 7. Average number of visited nodes of the hard-output sphere detector (r_0 is the initial radius, and $\hat{\boldsymbol{x}}$ denotes the hard MMSE estimate)

The effect of increasing the temperature is shown in Fig. 6. The legend denotes ΔN_0 [dB] $\triangleq 10\log_{10}\tilde{N}_0 - 10\log_{10} N_0$, i.e., the amount of increase in dB. We see that increasing the temperature indeed helps to a certain extent in the high-SNR range, albeit at the cost of a significant performance degradation at lower SNR. Judging from the figure, a reasonable heuristic is to choose the artificial temperature increase as ΔN_0 [dB] $= \max\{0 \text{ dB}, \text{SNR} - 13 \text{ dB}\}$, which approximately leads to the pointwise maximum of the other curves. However, there remains a significant gap to capacity at high SNR.

It turns out that the main reason for this gap is the failure of finding the ML symbol (as opposed to the counter-hypotheses) in some cases. In low SNR, using the wrong symbol as ML hypothesis introduces only a negligible error, since the most likely symbols all have similar metrics. In high SNR, however, the second-best symbol metric already differs significantly from the ML metric.

A simple remedy of this problem is thus to find the ML symbol by other means, and select it as the initial state of one ML-chain.[5] As mentioned in [9], a viable method for this is to run a sphere detector, which, if implemented without approximations, is guaranteed to find the ML solution.

At first glance this proposal might seem odd, since the MCMC detector is supposed to be an alternative to the sphere detector. However, note (a) that only a hard-output sphere detector is needed here, which has a significantly lower

[5]When the ML symbol is already known, one could skip the ML-chains and only use C-chains, but since the results of the ML-chains also extend the lists $\mathcal{L}_k$, we do not recommend to do so.

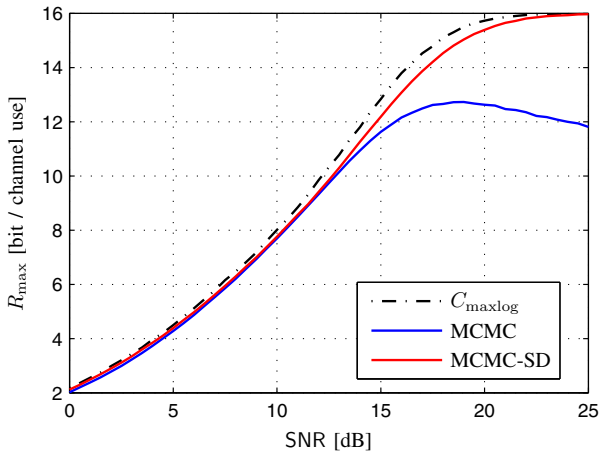

Fig. 8. Achievable rate of the MCMC detector, alone and in combination with a hard-output sphere detector

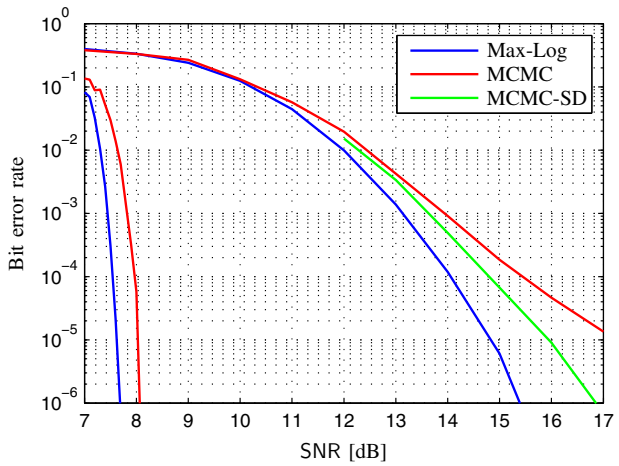

Fig. 9. Bit error rates for a rate-$1/3$ Turbo code (left) and a rate-$1/2$ convolutional code (right)

complexity than a full soft-output sphere detector in terms of, e.g., silicon area and runtime; and (b) that running the sphere detector becomes necessary only at high SNR, where the tree search terminates quickly because the subtrees can be pruned early. Fig. 7 verifies this claim by showing the average number of nodes that are visited by the sphere detector. The two plots differ in the choice for the initial sphere radius, which is once set to ∞, and once to the metric of the hard MMSE estimate, which can easily be obtained. While the latter initialization reduces the number of visited nodes significantly, both curves decay quickly with the SNR.

As a summary of this section, Fig. 8 shows the achievable rate of the discussed MCMC detector as a function of the SNR. In the low-to-medium SNR range, the difference between the MCMC and the max-log capacity is less than 0.5 dB, and at high SNR, the gap to capacity stays below 1 dB, thanks to the support of the hard-output sphere detector. By increasing the list size, the gap could be reduced further.

Finally, in order to demonstrate the validity of our performance evaluation based on mutual information, we present bit error rate plots in Fig. 9 which were obtained by two different outer codes. As an example of a strong code, operating at low SNR, we used a systematic Turbo code of rate $r = 1/3$ with two constituent $(1, 15/13)_8$ recursive systematic convolutional codes. The Turbo cliffs for the max-log and the MCMC detector are about 0.4 dB apart, which matches exactly with our prediction. In order to study the behaviour at high SNR, we used a rate-$1/2$ nonsystematic convolutional code with polynomials $(133, 171)_8$. The figure clearly shows the improvement that can be achieved by augmenting the MCMC detector with a hard-output sphere detector.

VI. Summary

We have presented a systematic study of the performance of Markov Chain Monte Carlo MIMO detectors based on the mutual information over an equivalent modulation channel. We have demonstrated the smooth trade-off between accuracy and computational complexity that is available to the system designer by adapting the number and/or the length of the employed Markov chains. Specific values for these parameters have been suggested which achieve a performance gap of less than 0.5 dB to max-log capacity at medium SNR with a reasonable complexity.

Furthermore, we have demonstrated that the error floor at high SNR, which is a well-known problem of the MCMC approach, can be removed by augmenting the MCMC detector with a hard-output sphere detector which is responsible for finding the ML symbol. Since this modification is only needed at high SNR, it can be implemented with a moderate increase in complexity.

References

[1] I. E. Telatar, "Capacity of multi-antenna Gaussian channels," *European Transactions on Communications*, vol. 10, pp. 585–595, Nov. 1999.

[2] B. M. Hochwald and S. ten Brink, "Achieving near-capacity on a multiple-antenna channel," *IEEE Trans. Commun.*, vol. 51, no. 3, pp. 389–399, Mar. 2003.

[3] C. Studer, A. Burg, and H. Bölcskei, "Soft-output sphere decoding: Algorithms and VLSI implementation," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 2, pp. 290–300, Feb. 2008.

[4] B. Farhang-Boroujeny, H. Zhu, and Z. Shi, "Markov chain Monte Carlo algorithms for CDMA and MIMO communication systems," *IEEE Trans. Signal Process.*, vol. 54, no. 5, pp. 1896–1909, May 2006.

[5] C. P. Robert and G. Casella, *Monte Carlo Statistical Methods*, 2nd ed. Berlin, Germany: Springer, 2004.

[6] A. Doucet and X. Wang, "Monte Carlo methods for signal processing," *IEEE Signal Process. Mag.*, vol. 22, no. 6, pp. 152–170, Nov. 2005.

[7] S. A. Laraway and B. Farhang-Boroujeny, "Implementation of a Markov chain Monte Carlo based multiuser/MIMO detector," *IEEE Trans. Circuits Syst. I*, vol. 56, no. 1, pp. 246–255, Jan. 2009.

[8] X. Mao, P. Amini, and B. Farhang-Boroujeny, "Markov chain Monte Carlo MIMO detection methods for high signal-to-noise ratio regimes," in *Proc. of IEEE Global Telecommunications Conference (GLOBECOM'07)*, Washington, D.C., Nov. 2007.

[9] S. Akoum, R. Peng, R.-R. Chen, and B. Farhang-Boroujeny, "Markov chain Monte Carlo detection methods for high SNR regimes," in *Proc. of IEEE International Conference on Communications (ICC'09)*, Dresden, Germany, Jun. 2009.

[10] P. Fertl, J. Jaldén, and G. Matz, "Performance assessment of MIMO-BICM demodulators based on system capacity," in *Proc. of IEEE 9th Workshop on Signal Processing Advances in Wireless Communications (SPAWC'08)*, Recife, Brazil, Jul. 2008.