Markov Chain Monte Carlo MIMO Detection Methods for High Signal-to-Noise Ratio Regimes
Xuehong Mao, Peiman Amini, and Behrouz Farhang-Boroujeny
ECE department, University of Utah
{mao, pamini, farhang}@ece.utah.edu

Abstract—Markov Chain Monte Carlo methods have recently been applied as front-end detectors in multiple-input multiple-output (MIMO) communication systems. Moreover, the near capacity behavior of such detectors in low signal-to-noise ratio (SNR) regimes have been demonstrated through computer simulations. However, it has also been found that the MCMC MIMO detectors degrade in high SNR regimes. This paper investigates into the source of this degradation and proposes a number of ad hoc methods to resolve this undesirable behavior of the MCMC MIMO detectors.

Keywords—Markov chain Monte Carlo, MIMO communications, Turbo detection

I. INTRODUCTION

Wireless communication systems that use multiple antennas at both transmitter and receiver have gained great momentum in recent years because of the improved transmission capacity that they offer compared to their single antenna counterparts [1]. To achieve near-capacity performance, a key block in the MIMO receiver is the MIMO detector whose role is to generate the soft information of the transmitted coded bits. These soft information that are often in the form of log-likelihood-ratio (LLR) values are passed to a channel decoder to extract the uncoded information bits. Moreover, to improve on the receiver performance, the soft information may be cycled in a loop consisting of the MIMO detector and the channel decoder for a number of iterations before final decision on the transmitted bits is made. This procedure may be thought of as a serially concatenated turbo detector, where the MIMO coding/detection is viewed as an inner code and the channel code as an outer code. The channel code can be a simple convolutional code, a turbo code, or low-density parity check (LDPC) code [2].

The optimal MIMO detector, known as maximum likelihood (ML) detector, has a complexity that grows exponentially with the number of bits per channel use. In fact, the complexity of the MCMC MIMO detector is not exponential with the number of bits per channel use [6].

The Markov chain Monte Carlo (MCMC) method, [13], is an alternative search technique that may also be used to generate a candidate list [8], [9], [10]. This method is different from the tree-search methods in two ways: (i) it is a stochastic search; (ii) the growth of the size of the list and thus the complexity of the MIMO detector is not exponential with the number of bits per channel use. In fact, the complexity of the MCMC MIMO detector only grows slightly faster than the linear. However, the past studies have shown that while the MCMC MIMO detector perform fantastically good in low SNR (near capacity) regime, it may suffer from a noise floor, or even its performance may degrade as SNR increases.

The goal of this paper is to investigate and identify the source of this undesirable behavior of the MCMC MIMO detector and propose a number of methods that resolve this shortcoming.

This paper is organized as follows. The system model is presented in Section II. A short review of the MCMC MIMO detector is presented in Section III. Several novel methods that enhance the behavior of the MCMC MIMO detector in high SNR regimes are also presented in Section IV. Simulation results are presented in Section V. The concluding remarks of the paper are drawn in Section VI.

II. SYSTEM MODEL

The block diagram of an \(N_r\)-by-\(N_t\) MIMO system is shown in Fig. 1. At the transmitter, the information word \(s\) is encoded by the channel encoder. The output of the channel encoder after passing through interleaver is divided into the blocks of \(M\) bits. These blocks form a vector sequence \(b(n)\), where \(n\) is the time index. Each \(b(n)\) is then mapped to the transmit symbol \(d(n) = [d_1(n), d_2(n), \ldots, d_{N_t}(n)]^T\), where the superscript ‘T’ denotes transpose. We also assume that each
element of \( \mathbf{d}(n) \) carries \( M_c = M/N_t \) coded bits and thus is chosen from a \( 2^{M_c} \)-ary QAM/PSK constellation. We also note that each value of \( n \) corresponds to one channel use and during each channel use, \( M \) coded bits are being transmitted. In the sequel, since most of our derivations correspond to one channel use, i.e., a fixed \( n \), we drop the time index \( n \), for brevity.

Assuming a flat fading channel, the received signal can be modeled as
\[
y = \mathbf{Hd} + \mathbf{n},
\]
where \( \mathbf{H} \) is the channel gain matrix and \( \mathbf{n} \) is the channel noise, a white Gaussian noise vector. We assume that \( \mathbf{n} \) has zero mean and the covariance matrix \( \mathbb{E}[\mathbf{n}\mathbf{n}^H] = \sigma^2 \mathbf{I} \). We also note that \( \mathbf{d} \) is the transmit vector that is obtained from a block of coded bits represented by \( \mathbf{b} \).

### III. Detection Methods

At the receiver, the MIMO detector provides the LLR values
\[
\lambda_1(b_k) = \ln \frac{p(b_k = +1 | \mathbf{y}, \lambda_2^d(b))}{p(b_k = -1 | \mathbf{y}, \lambda_2^d(b))},
\]
for all \( k \). Here, \( \lambda_2^d(b) \) is the extrinsic information from the channel decoder. The extrinsic information \( \lambda_2^d(b_k) = \lambda_1(b_k) - \lambda_2^d \) is then formed and passed to the channel decoder. By exchanging the extrinsic information between the MIMO detector and the channel decoder iteratively, the turbo principle is applied. This procedure reduces the bit-error rate (BER) over successive iterations and allow one to achieve a near capacity performance [2], [9].

Using the log-max approximation, we obtain
\[
\lambda_2^d(b_k) \approx \max_{\mathbf{d} \in D_k^+} \left\{ -\frac{\| \mathbf{y} - \mathbf{Hd} \|^2}{2\sigma^2} + \frac{1}{2} \mathbf{b}_k^T \lambda_2^d \mathbf{b}_k \right\} - \max_{\mathbf{d} \in D_k^-} \left\{ -\frac{\| \mathbf{y} - \mathbf{Hd} \|^2}{2\sigma^2} + \frac{1}{2} \mathbf{b}_k^T \lambda_2^d \mathbf{b}_k \right\},
\]
where \( D_k^+ \) is the set of all possible samples of \( \mathbf{d} \) with \( b_k = +1 \), and \( D_k^- \) is the set of \( \mathbf{d} \) with \( b_k = -1 \). \( \mathbf{b}_k \) is obtained from \( \mathbf{b} \) by removing \( b_k \), and \( \lambda_2^d \) is the vector of the extrinsic LLR values of \( \mathbf{b}_k \) from the channel decoder.

The key point and the main reason that has initiated the development of the tree-search methods (including the LSD) and the MCMC MIMO detector is that the complexity of realization of (3) grows exponentially with the number of bits in each channel use. In a MIMO system with \( M \) bits per channel use, each of the sets \( D_k^+ \) and \( D_k^- \) have the size of \( 2^{M-1} \). Both the tree-search methods and the MCMC MIMO detector are designed to find small subsets of \( D_k^+ \) and \( D_k^- \) that with a high probability contain the desired terms that maximize both terms on the right-hand side of (3).

The MCMC MIMO detector uses a stochastic search method called Gibbs sampler. The Gibbs sampler is a particular Markov chain process that searches the state space defined by \( \mathbf{b} \). It walks through this space in a stochastic manner with the goal of finding the samples of \( \mathbf{b} \) that result in small values of \( \frac{\| \mathbf{y} - \mathbf{Hd} \|^2}{2\sigma^2} - \frac{1}{2} \mathbf{b}_k^T \lambda_2^d \mathbf{b}_k \). In other words, the Gibbs sampler looks for important sample of \( \mathbf{b} \) that maximize the two terms on the right-hand side of (3). For details of the Gibbs sampler, when applied to MIMO detection, we request the reader to refer to [9]. Also, for a comparison of the MCMC MIMO detector and LSD the reader may refer to [10]. A hardware architecture for efficient implementation of the MCMC MIMO detector can be found in [11].

### IV. MCMC MIMO Detector for High SNR Regimes

Studies performed in [9] have revealed that while the MCMC MIMO detector performs very well in low SNR regimes, it does not perform so well as SNR increases. The source of this behavior was explored in [9]. It was noted that at higher values of SNR some of the transition probabilities in the underlying Markov chain may become very small and as a result the Markov chain may effectively be divided into a number of nearly disjoint chains. The term nearly disjoint here means the transition probabilities that allow movement between the disjoint chains are very low. As a result, a Gibbs sampler that is started from a random point will remain within the set of points surrounding the initial point and thus may not get a chance of visiting sufficient points to find the maxima of the terms on the right-hand side of (3). In [9] two solutions for solving this problem were proposed: (i) run a number of parallel Gibbs samplers with different starting points; (ii) while running the Gibbs sampler assume a noise variance higher than what actually is and use the correct noise variance while evaluating (3). These two methods turned out to be effective for low and medium size SNRs, as is evident from the excellent results presented in [9], [10], [12].

In many situations in practice, communication systems operate in SNR range that are relatively high; many decibels away from the capacity. In such cases, if
The MIMO detector can obtain reasonably correct values for the LLRs, one would expect to detect the transmitted information with a very low probability of error through the channel decoder and without any need to run any extra iteration between the MIMO detector and the channel decoder. However, simulations, some of which are presented below, reveals that the above two measures are insufficient to remedy the problem. In this paper, we propose additional methods to resolve the problem of MCMC MIMO detector in high SNR regimes. We also introduce a trivial, yet novel, method for minimizing the receiver complexity.

A. Non-turbo receiver

We note that, in the absence of the extrinsic information from the channel decoder, the desired solutions that maximize the two terms on the right-hand side of (3) are those that result in relatively small values for $|\|y - H\|_{\infty}$

Such solutions are known and can be obtained using a ZF or MMSE equalizer. Through computer simulations, we have found that by initializing one of the Gibbs samplers using either ZF or MMSE solution and initializing the rest of the Gibbs samplers randomly, we obtain results that are much better than those that would be obtained if all of the Gibbs samples were initialized randomly.

Fig. 2 presents a sample of our simulation results. For this result, as well as the subsequent results in this paper, we simulate a MIMO system with 4 transmit and 4 receive antennae. The channel code is the rate $R = 1/2$ convolutional code with the generator polynomials $1$ and $1 + D^2 + D^7$. The data are transmitted in packets of length 1600 uncoded (3200 coded) bits. The channel $H$ is random, but quasi static, meaning that it is fixed over each packet. However, it is chosen independently for each packet. The elements of $H$ are complex-valued Gaussian iid zero mean random variables with variance of unity. There are $M=16$ bits per channel use. The 16 bits are divided into 4 blocks of 4 bits each that are mapped to a 16-QAM symbols using Gray coding.

For the results presented in Fig. 2 there is no iteration between the channel decoder and the MIMO detector. The soft information generated by the MIMO detector is passed to the channel decoder and the output of the channel decoder is used to decide on the information bits. We use the normalized SNR $E_b/N_0$ which is related to the SNR $E_b/N_0$ as

$$E_b \left|_{\text{rmdB}} \right. = E_b \left|_{\text{rmdB}} \right. + 10 \log_{10} \frac{N_r}{N_t R M_x}. \quad (4)$$

There are four plots in Fig. 2. The first plot is obtained by running 5 parallel randomly initialized Gibbs samplers. Each Gibbs sampler has depth of 5, i.e., it runs over the elements of $b$ 5 times. This will result in $5 \times 5 = 25$ samples for each of the terms on the right-hand side of (3). The second plot is obtained by running a single Gibbs sampler, initialized with the ZF solution, and for a depth of 25; so the sample sets have the same size as in the first plot. The third and fourth plots are generated using 5 parallel Gibbs samplers each of depth 5, with 4 of the Gibbs samplers initialized randomly and 5th one initialized with the solution obtained from ZF and MMSE equalizers, respectively.

The following conclusions are drawn from the results shown in Fig. 2.

- The use of only randomly initialized Gibbs samplers results in a MIMO detector that degrades at high SNR values.
- The use of a single Gibbs sampler initialized with the ZF (or MMSE) solution results in a much improved performance.
- The combination of a number of randomly initialized Gibbs samplers and one Gibbs sampler that is initialized with the ZF (or MMSE) solution further improves the results.
- The level of improvement achieved through ZF and MMSE initializations are the same.

We have the following explanation to these observations. When Gibbs samplers are initialized randomly, there is always a chance that none of the Gibbs samplers do not approach the portions of the state-space defined by $b$ that correspond to the maximum terms on the right-hand side of (3). As a result, for a relatively large percentage of the channel uses, the MIMO detector may generate incorrect LLR values. The ZF (or MMSE) initialization has a very high likelihood of giving an initial $b$ within the vicinity of the points that minimize the two terms on the right-hand side of (3). The randomized Gibbs samplers result in some level of improvement by adding more samples to the list in the cases where ZF (or MMSE) fails in giving a good initial point. The fact that both ZF and MMSE initialization results in the same improvement can be explained if we realize in high SNR, where such initializations help, the solutions to both cases are about the same.

From the above results, we observe that although the combination of ZF (or MMSE) and randomized initial-
ize the Gibbs samples greatly helps in reducing the BER in high SNR regimes, the BER curves presented in Fig. 2 still show some error floor. A number of approaches can be taken to further improve the performance of the receiver. One approach is to increase the number of Gibbs samplers and/or increase their depth. Fig. 3 presents a sample result that shows how this measure helps. Here, by increasing the number of parallel Gibbs samples from 5 to 15 and the depth of each Gibbs sampler from 5 to 15 (a 9-fold increase in complexity), we can achieve two orders of magnitude improvement in BER. However, the error floor problem is not resolved.

The following additional measures may be used to improve on the above BER curves and hopefully remove the error floor. (i) Add an additional code with error correcting capability (such as a Reed-Salomon code) prior to the channel encoder. The presence such code can get rid of the residual errors, as long as the number of errors is sufficiently small. (ii) Run iterations between MIMO detector and the channel decoder. We pursue the latter approach in the rest of this paper.

B. Turbo receiver

To reduce the receiver complexity, we first note when SNR is high and sufficiently accurate estimates of the LLR values are generated by the MIMO detector, error free recovery of a good majority of the packets occurs in the first iteration of turbo loop. In other words, most of the packets are recovered after the first pass through the MIMO detector and the channel decoder. We thus suggest by adding a parity check (e.g., a CRC check [14]) to each packet one may examine the correctness of the detected packet. If the packet is detected correctly, no further iteration of the receiver will be executed. If not, soft information from the channel decoder are fed back to the MIMO detector to continue with second iteration. Similarly, if after the second iteration, still the parity check does not confirm the correctness of the detected packet, iterations continue until the packet is correctly detected or the detection process is terminated after a maximum number of iterations is reached.

Other measures that we empirically (i.e., through computer simulations) found improve the performance of the receiver are:
- After each iteration, one may use the soft information from the channel decoder to randomize the initial settings of the Gibbs samplers for the next iteration.
- Although the latter method greatly helps, for some packets it does not work, not matter how many iterations of the turbo detector is executed. Detailed exploration of the simulation results reveals that in such cases the number of bit errors increases with iteration number. In other words, the turbo system can be subject to error propagation. We empirically found a good strategy for solving this problem is to restart the detection process if the turbo loop fails to detect the correct packet after a number of iterations. We refer to each restart of the turbo loop as one stage and number the successive stages as 1, 2, 3, · · · .
- As the receiver proceeds with a new stage, the number of parallel Gibbs samplers and/or the depth of each Gibbs sampler is increased. This, obviously, is done to improve the accuracy of the LLR values generated by the MIMO detector.

The simulation results presented in the next section reveals that the above measures lead to a MIMO receiver in which BER converges to zero as SNR increases.

V. SIMULATION STUDIES

Unfortunately, any theoretical analysis of the MCMC MIMO detector, discussed in this paper, turns out to be a very difficult task, and as of today no such analysis is available. We thus proceed with drawing some conclusions based on numerical studies. The numerical results that are presented in this section are for the 4 × 4 MIMO system that was introduced in Section IV-A. In addition, to be able to check successful detection of data packets after each iteration of the turbo loop, a length 16 CRC parity checker is added to the coded data bits. For each data packet, the turbo detector is run for three stages; namely, Stage 1, Stage 2, and Stage 3. Stage 1 consists of at most 5 iterations and in each iteration the Gibbs sampler operates based on 25 samples for each bit; 5 Gibbs samplers, each of depth 5, are run. For Stage 2, the number parallel Gibbs samplers is increased to 10, and their depth is extended to 10. The number of iteration is also increased to 7. In Stage 3, the number parallel Gibbs samplers is increased to 20, and their depth is extended to 20. The number of iteration is increased to 9. The detection process stops when CRC check indicates a correctly detected data packet, or when the three stages of the detection are completed without successful detection of the packet. The simulations are run for 100,000 packets.

Table I presents the percentages of the successfully detected packets after each iteration of the turbo loop, for
**TABLE I**

Percentages of the successfully detected packets at successive iterations and stages of the turbo loop.

<table>
<thead>
<tr>
<th>Iteration No.</th>
<th>$E_b/N_0$, dB</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>99.941%</td>
</tr>
<tr>
<td>12</td>
<td>99.957%</td>
</tr>
<tr>
<td>14</td>
<td>99.997%</td>
</tr>
<tr>
<td>16</td>
<td>100.00%</td>
</tr>
<tr>
<td>18</td>
<td>100.00%</td>
</tr>
<tr>
<td>20</td>
<td>99.987%</td>
</tr>
<tr>
<td>22</td>
<td>99.995%</td>
</tr>
<tr>
<td>24</td>
<td>99.996%</td>
</tr>
<tr>
<td>26</td>
<td>99.999%</td>
</tr>
</tbody>
</table>

$E_b/N_0$ values of 10 to 26 dB. The cumulative percentages of the successfully detected packets are also shown. Referring to the results in this table, the following observations are made:

- Most of the packets are correctly detected within the first stage.
- As SNR increases, the number of iterations required to correctly detect each packet decreases.
- At higher values of $E_b/N_0$ a large percentage of the packets are correctly detected within the first iteration. For instance at $E_b/N_0 = 22$ dB, 93.686% of the packets are correctly detected within the first iteration. This number increases to 95.311% at $E_b/N_0 = 24$ dB and to 96.557% at $E_b/N_0 = 26$ dB.
- For values of $E_b/N_0 \geq 18$ dB all the packets are correctly detected before completion of the third stage.
- At high SNR, since most of the packets are recovered within the first iteration, the average complexity of the receiver is only slightly more than one iteration of the turbo detector.

**VI. CONCLUSION**

We proposed a number of measures to overcome the poor performance of the MCMC MIMO detector in high SNR regimes. The proposed measures/solutions were studied through computer simulations. They were found to be very effective and able to solve the problem. Error free detection of 100,000 packets, each of length 1600 uncoded information bits, was observed in the $E_b/N_0$ range of 18 to 26 dB.

**REFERENCES**


