# On Performance of Sphere Decoding and Markov Chain Monte Carlo Detection Methods

Haidong (David) Zhu, Behrouz Farhang-Boroujeny, and Rong-Rong Chen
ECE Department, Unversity of Utah, USA
emails: haidongz@eng.utah.edu, farhang@ece.utah.edu, rchen@ece.utah.edu

***Abstract*— In a recent work, it has been found that the suboptimum detectors that are based on Markov chain Monte Carlo (MCMC) simulation techniques perform significantly better than their sphere decoding (SD) counterparts. In this letter, we explore the sources of this difference and show that a modification to existing sphere decoders can result in some improvement in their performance, even though they still fall short when compared with the MCMC detector. We also present a novel SD detector that is an exact realization of max-log-MAP detector. We call this exact max-log SD detector. Comparison of the results of this detector with those of the max-log version of the MCMC detector reveals that the latter is near optimal.**

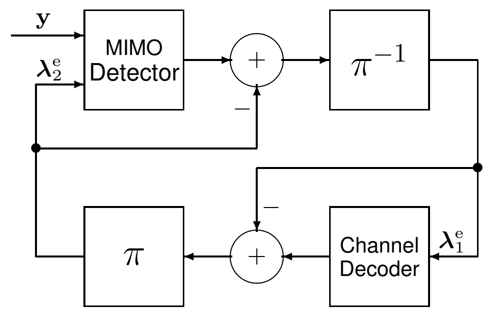


Fig. 1. Receiver structure. MIMO detector and channel decoder are soft-in soft-out blocks that exchange information, $\boldsymbol{\lambda}_1^e$ and $\boldsymbol{\lambda}_2^e$, in a turbo loop.

## I. Introduction

Two major research activities have dominated design of power and bandwidth efficient wireless communication systems in recent years: (i) communication through multiple antennas, known as multiple-input multiple-output (MIMO) communication, and (ii) iterative decoding/detection techniques. MIMO communication allows simultaneous transmission of multiple symbols from multiple transmit antennas. This results in a linear increase in the channel capacity proportional to the number of transmit antennas, when there are sufficient number of receive antennas. Iterative decoding/detection, on the other hand, is a feasible method that greatly improves the bit-error-rate (BER) performance of communication systems, bringing them very close to the Shannon channel capacity.

Combination of MIMO and iterative decoding/detection techniques has naturally been studied as a means of approaching the capacity of MIMO channels. Fig. 1 presents a block diagram of a MIMO receiver that works based on this principle. This follows the receiver structure that was first proposed in [1] for code-division multiple-access (CDMA) systems and later applied to MIMO channels, e.g., [2]. Here, the MIMO channel plays the role of an inner code. The outer code is a channel code that may be a convolutional code or a more advanced turbo or LDPC (low-density parity-check) code. We assume a MIMO channel with $M$ transmit antennas and $N$ receive antennas and a flat fading model

$$\mathbf{y} = \mathbf{H}\mathbf{d} + \mathbf{n} \tag{1}$$

where $\mathbf{d} = [d_1 \ d_2 \ \cdots \ d_M]^{\mathrm{T}}$ is a vector of transmitted symbols, $\mathbf{H}$ is the channel gain matrix (of size $N \times M$), $\mathbf{n}$ is a vector of channel additive noise, $\mathbf{y}$ is the received signal vector, and the superscript T denotes transpose.

Hochwald and ten Brink [3] have followed the receiver structure of Fig. 1 and used sphere decoding (SD) as a soft-in soft-out MIMO detector and reported some results that approach the capacity of MIMO channels. Vikalo et al [4] have also studied the use of SD and demonstrated similar results. The difference between the two approaches is that [3] uses a candidate list (a list of likely choices of $\mathbf{d}$ that matches the channel output $\mathbf{y}$) which is chosen independently of the *a priori* information from the channel decoder, while [4] includes these information in finding the list.

The SD may be thought of as a method of searching the $M$-dimensional space spanned by the vector of transmitted symbols, $\mathbf{d}$, and choosing a list of candidates that match $\mathbf{y}$. In SD, this is done in a deterministic manner by finding samples of $\mathbf{d}$ that result in a small distance between $\mathbf{y}$ and $\mathbf{Hd}$. Gibbs sampling (GS), a statistical method based on Markov chain Monte Carlo (MCMC) simulation techniques [5], [6], is an alternative method that may be used for choosing a list of $\mathbf{d}$ with the same goal. In a recent work [8] (also see [7] for our initial results), we have proposed a MIMO receiver that follows the structure of Fig. 1, but instead of using SD, applies GS to obtain samples of $\mathbf{d}$. The results presented in [8] are superior to those in [3] and [4]. In this paper, we explore the source of this performance gain and show that there are two contributers: (i) the search method for obtaining the samples of $\mathbf{d}$, and (ii) the way the samples are handled to produce the LLR values. The first contributer is inherent to SD and thus cannot be resolved, unless new search methods are found. However, a solution to the second contributer can be readily proposed by borrowing a relevant idea from [8]. This results in a new implementation that we call modified SD. Moreover, we present a novel SD detector that is optimum in the sense of max-log-MAP [9]. We thus call it exact max-log SD detector. Comparison of the results of the exact max-log SD detector with those of the max-log version of the MCMC detector reveals that the latter incurs very little loss, thus is identified as a near optimum detector.

The rest of this letter is organized as follows. A brief review of SD is presented in the next section. Besides giving the basic idea of SD, we also emphasize on the tree search method that is used to obtain samples of $\mathbf{d}$. This is important in drawing some of the conclusions of this paper. In Section III, we present a brief review of the MCMC detector. A modified SD algorithm is presented in Section IV. The exact max-log SD detector is presented in Section V. This detector, although impractical due to very high complexity, can be used to evaluate the effectiveness of suboptimal detectors. Computer simulations are presented in Section VI. The complexity of SD and MCMC detectors are discussed in Section VII. The conclusions are drawn in Section VIII.

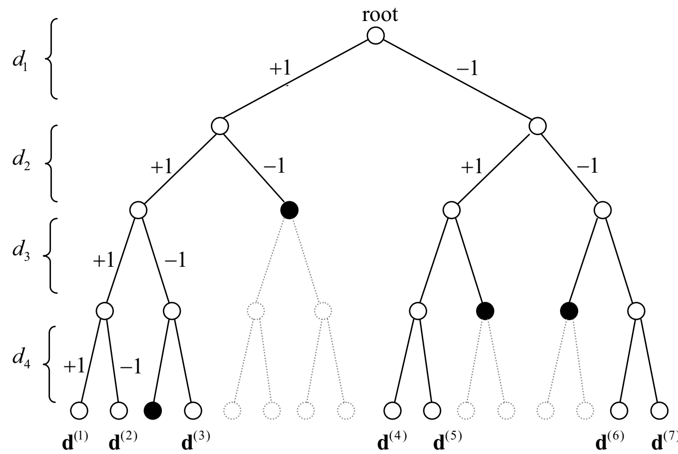


Fig. 2. An example of the tree search in SD.

## II. Review of the Results from Sphere Decoding

Similar to the previous works, [3] and [4], we assume that each vector of the transmitted symbols, $\mathbf{d}$, is obtained, through a mapping, from a vector $\mathbf{b}$ of information bits $b_1$, $b_2$, $\cdots$, $b_{M \cdot M_c}$, where each group of $M_c$ bits is mapped to one element of $\mathbf{d}$. We also recall from [4] that a candidate list $\mathcal{L}$ of $\mathbf{d}$ may be generated by picking samples of $\mathbf{d}$ that satisfy the inequality

$$(\mathbf{d}-\hat{\mathbf{d}})^\dagger \mathbf{H}^\dagger \mathbf{H}(\mathbf{d}-\hat{\mathbf{d}}) - 2\sigma^2 \ln P^{\mathrm{e}}(\mathbf{d}) < r^2 \tag{2}$$

where the superscript $\dagger$ denotes Hermitian, $P^{\mathrm{e}}(\mathbf{d})$ is the (extrinsic) probability of $\mathbf{d}$ according the feedback information from the channel decoder, $2\sigma^2$ is the variance of each element of $\mathbf{n}$, $\hat{\mathbf{d}} = (\mathbf{H}^\dagger\mathbf{H})^{-1}\mathbf{H}^\dagger\mathbf{y}$ is the center of the sphere and $r$ is its radius. In [3], the second term on the left-hand side of (2) is not included, and thus the list is generated independent of the *a priori* information.

Throughout this paper, we often refer to the inequality (2) as (interior of) a sphere, even though, strictly speaking, this is not an accurate statement. The presence of the prior information term $2\sigma^2 \ln P^{\mathrm{e}}(\mathbf{d})$ leads of a $\mathbf{d}$-dependent radius $r' = \sqrt{r^2 + 2\sigma^2 \ln P^{\mathrm{e}}(\mathbf{d})}$.

Given a candidate list $\mathcal{L}$ of $\mathbf{d}$, the following equation is used to obtain an estimate of the extrinsic LLR value of $b_k$, [3], [4],

$$\begin{aligned}\lambda_1^{\mathrm{e}}(b_k) = &\max_{\mathbf{d}\in\mathcal{L}_k^+} \left\{ -\frac{1}{2\sigma^2}||\mathbf{y}-\mathbf{H}\mathbf{d}||^2 + \frac{1}{2}\mathbf{b}_{-k}^{\mathrm{T}}\boldsymbol{\lambda}_{2,-k}^{\mathrm{e}} \right\} \\ &- \max_{\mathbf{d}\in\mathcal{L}_k^-} \left\{ -\frac{1}{2\sigma^2}||\mathbf{y}-\mathbf{H}\mathbf{d}||^2 + \frac{1}{2}\mathbf{b}_{-k}^{\mathrm{T}}\boldsymbol{\lambda}_{2,-k}^{\mathrm{e}} \right\}\end{aligned} \tag{3}$$

where $\mathcal{L}_k^+$ and $\mathcal{L}_k^-$ denote the subsets of $\mathcal{L}$ for which $b_k$ is $+1$ and $-1$, respectively, $\mathbf{b}_{-k}$ is obtained from $\mathbf{b}$ by removing $b_k$, and $\boldsymbol{\lambda}_{2,-k}^{\mathrm{e}}$ is the vector of the extrinsic LLR values of $\mathbf{b}_{-k}$ from the channel decoder.

SD adopts a tree search approach to obtain samples of $\mathbf{d}$. Fig. 2 presents an example of the tree search adopted in [3], [4]. The search starts from a root node and begins with examining possible choices of $d_1$ that may satisfy (2). For each choice of $d_1$, the possible choices of $d_2$ are examined. The procedure continues for the rest of the elements of $\mathbf{d}$, similarly. Each choice of an element of $\mathbf{d}$ is indicated by a branch in the tree. Also, for convenience of demonstration, it is assumed that the elements of $\mathbf{d}$ are binary. The search collects samples of $\mathbf{d}$ by running through the branches of the tree from top to bottom and left to right. This, for the example of Fig. 2, results in the selection of the respective samples

$$\mathbf{d}^{(1)} = \begin{bmatrix} +1\\+1\\+1\\+1 \end{bmatrix}, \quad \mathbf{d}^{(2)} = \begin{bmatrix} +1\\+1\\+1\\-1 \end{bmatrix}, \quad \mathbf{d}^{(3)} = \begin{bmatrix} +1\\+1\\-1\\-1 \end{bmatrix}, \quad \cdots$$

It is worth noting that many searches in the tree encounter nodes where no branches beyond them can lead to a point within the sphere, i.e., satisfy (2). In Fig. 2, these are indicated by bold nodes. We refer to these as *terminal nodes*, since the search cannot proceed beyond them. More will be said about the SD tree search and the potential problems that it may create in a soft detector in Section VII.

## III. Review of the Results from MCMC Detection Method

In the MCMC detection method, a list $\mathcal{L}$ of samples of $\mathbf{d}$ is generated by running one or more Markov chains that converge towards the conditional distribution

$$P(\mathbf{d}|\mathbf{y}, \boldsymbol{\lambda}_2^{\mathrm{e}}) = K \exp\left(-\frac{||\mathbf{y}-\mathbf{H}\mathbf{d}||^2}{2\sigma^2}\right) P^{\mathrm{e}}(\mathbf{d}) \tag{4}$$

where $K$ is a constant scalar added to make $P(\mathbf{d}|\mathbf{y}, \boldsymbol{\lambda}_2^{\mathrm{e}})$ a proper density function. This procedure, clearly, leads to generation of samples of $\mathbf{d}$ that result in small values for $||\mathbf{y}-\mathbf{H}\mathbf{d}||^2 - 2\sigma^2 \ln P^{\mathrm{e}}(\mathbf{d})$. Hence, the candidate list generated by this procedure is similar to those generated through SD; see (2), above.

GS is the common procedure of generating samples of $\mathbf{d}$ in MCMC detectors. It visits successive elements of $\mathbf{d}$, sequentially. After one cycle of visiting all elements of $\mathbf{d}$, a sample is generated and the process proceeds with generation of the next sample. The following procedure summarizes these steps:

- `Initialize` $\mathbf{d}^{(-N_\mathrm{b})}$ `(randomly),`
- `for` $n = -N_\mathrm{b}+1$ `to` $N_\mathrm{s}$

  `draw` $d_1^{(n)}$ `from` $P(d_1|d_2^{(n-1)},\ldots,d_M^{(n-1)},\mathbf{y},\boldsymbol{\lambda}_2^\mathrm{e})$

  `draw` $d_2^{(n)}$ `from` $P(d_2|d_1^{(n)},d_3^{(n-1)},\ldots,d_M^{(n-1)},\mathbf{y},\boldsymbol{\lambda}_2^\mathrm{e})$

  $\vdots$

  `draw` $d_M^{(n)}$ `from` $P(d_M|d_1^{(n)},\ldots,d_{M-1}^{(n)},\mathbf{y},\boldsymbol{\lambda}_2^\mathrm{e})$

In this procedure $\mathbf{d}^{(-N_\mathrm{b})}$ is initialized randomly, taking into account the *a priori* information $\boldsymbol{\lambda}_2^\mathrm{e}$. The first $N_\mathrm{b}$ iterations of the 'for' loop, called burn-in period, is to let Markov chain converge to near its stationary distribution. Hence, the samples of $\mathbf{d}$ used for LLR computations are $\mathbf{d}^{(n)} = [d_1^{(n)}\ d_2^{(n)}\ \cdots\ d_M^{(n)}]^\mathrm{T}$, for $n = 1, 2, \cdots, N_\mathrm{s}$. The samples generated by a single Markov chain are usually highly correlated [5]. In [8], it is noted that the performance of the MCMC detection method significantly improves if a parallel set of Markov chain are used to collect samples for a list $\mathcal{L}$. We adopt this strategy while generating the simulation results in Section VI.

From the above, we note that SD and GS are fundamentally similar in the sense that they both attempt to find a set of samples $\mathbf{d}$ that result in small values of $||\mathbf{y}-\mathbf{Hd}||^2 - 2\sigma^2 \ln P^\mathrm{e}(\mathbf{d})$. However, [8] has adopted a different approach of handling the list $\mathcal{L}$. More particularly, the lists $\mathcal{L}_k^+$ and $\mathcal{L}_k^-$, in (3), are, respectively, replaced by the expanded lists $\mathcal{U}_k^+$ and $\mathcal{U}_k^-$ which are obtained from $\mathcal{L}$ as follows. We note that in mapping the vector $\mathbf{b}$ of information bits to transmit symbols, $b_k$ is mapped to one of the elements of $\mathbf{d}$. The expansion begins with identifying the element of $\mathbf{d}$ that the desired bit, $b_k$, is mapped to. The set $\mathcal{L}$ is then expanded by giving this element of sample vectors $\mathbf{d}^{(n)}$ all possible values that it can take, from the symbol alphabet. This results in a set which we call $\mathcal{U}_k$. $\mathcal{U}_k^+$ and $\mathcal{U}_k^-$ are then generated as the subsets of $\mathcal{U}_k$ in which $b_k$ is $+1$ and $-1$, respectively.

## IV. Modified Sphere Decoding

Clearly, one can also propose an implementation of SD using the expanded lists $\mathcal{U}_k^+$ and $\mathcal{U}_k^-$ instead of $\mathcal{L}_k^+$ and $\mathcal{L}_k^-$. The simulation results presented in Section VI show that this modification to SD improves its performance. We call this implementation of SD, *modified SD*.

The improved behavior of the modified SD may be explained as follows. We recall from [3] that when a particular bit, say $b_k$, is more likely to be $+1$ (or $-1$), the subset $\mathcal{L}_k^-$ (or $\mathcal{L}_k^+$) may be empty. In such cases, [3] suggests that $\lambda_1^\mathrm{e}(b_k)$ be replaced by some extreme values, which, of course, may be inaccurate. Expansion of the lists, i.e., using $\mathcal{U}$ in place $\mathcal{L}$, avoids such undesirable cases and assures a better estimate of the LLR value. We confirm the validity of this claim through BER curves that are obtained through computer simulations.

We note that replacement of the list $\mathcal{L}$ by the expanded list $\mathcal{U}$ effectively increases the length of the list. In order to confirm that the improved performance of the modified SD is not because of the use of a larger list, in Section VI we choose the parameters of the modified SD such that the expanded list $\mathcal{U}$, on average, have a similar length to the list length in [3] and comparisons are made accordingly.

## V. Exact max-log SD Detector

The exact max-log-MAP estimate of $b_k$ is given by

$$\lambda_1^\mathrm{e}(b_k) = \max_{\mathbf{d}\in\mathcal{D}_k^+}\left\{-\frac{1}{2\sigma^2}||\mathbf{y}-\mathbf{Hd}||^2 + \frac{1}{2}\mathbf{b}_{-k}^\mathrm{T}\boldsymbol{\lambda}_{2,-k}^\mathrm{e}\right\} - \max_{\mathbf{d}\in\mathcal{D}_k^-}\left\{-\frac{1}{2\sigma^2}||\mathbf{y}-\mathbf{Hd}||^2 + \frac{1}{2}\mathbf{b}_{-k}^\mathrm{T}\boldsymbol{\lambda}_{2,-k}^\mathrm{e}\right\} \quad (5)$$

where $\mathcal{D}$ is a complete list of all possible choices of $\mathbf{d}$, and $\mathcal{D}_k^+$ and $\mathcal{D}_k^-$ are the subsets of $\mathcal{D}$ in which $b_k$ is $+1$ and $-1$, respectively.

Clearly, when the sizes of $\mathcal{D}_k^+$ and $\mathcal{D}_k^-$ are too large, a practical estimator can only be realized when these are replaced by some proper subsets. This can easily be done by applying SD to each of the terms on the right-hand side of (5), separately. With this, we concentrate on small subsets of $\mathcal{D}_k^+$ or $\mathcal{D}_k^-$ within some specified spheres. The true maximizer of the two terms on the right-hand side of (5) are within these subsets. Hence, using this approach one can realize an exact max-log SD detector without fully searching all the elements of $\mathcal{D}$. This requires $2M \cdot M_c$ independent sphere decoders that may be too complex to realize. Nonetheless, a study of such a detector is useful and can be considered as a benchmark for evaluation of other suboptimum detectors.

## VI. Computer Simulation

We present the BER results for two MIMO systems. One with 4 transmit and 4 receive antennas, and the other with 8 transmit and 8 receive antennas. Data symbols are chosen from a 16-point QAM constellation. The channel code is a rate 1/2 parallel concatenated (turbo) code with feedforward polynomial $1 + D^2$ and feedback polynomial $1 + D + D^2$. Each block of information bits has the length of 9216. The channel matrix $\mathbf{H}$ is independent over time and has complex i.i.d. entries. These parameters are similar to those of [3] and are selected so to facilitate the comparisons. We also recall that the SD of [3] uses lists of length 512 and 1024 in the cases of 4 and 8 antennas, respectively. For MCMC, $L$ parallel Gibbs samplers each of length $N_\mathrm{s}$ and with no burning periods (i.e., $N_\mathrm{b} = 0$) are used to collect samples of $\mathbf{d}^{(n)}$. Because of the stochastic nature of MCMC, repeated samples of $\mathbf{d}^{(n)}$ (that should be discarded) occur and this reduces the list length below the maximum length $LN_\mathrm{s}$. For modified SD, we start with a small radius $r$ and increase it gradually until we find sufficient number of samples of $\mathbf{d}$ that satisfy (2); say $N_\mathrm{min}$ samples. Since there is no clear relationship between $r$ and the number of choices of $\mathbf{d}$ that satisfy (2), one occasionally encounters situation where the number of choices of $\mathbf{d}$ that satisfy (2) is excessively large. In such cases, we stop choosing samples of $\mathbf{d}$ when certain number of samples, say $N_\mathrm{max}$, are collected.

For simulation results that are presented below, we choose the parameters $L$ and $N_\mathrm{s}$ of the MCMC detector, and $N_\mathrm{min}$ and $N_\mathrm{max}$ of the modified SD detector such that the average list lengths in these cases (after expansion of $\mathcal{L}$ to $\mathcal{U}$) are comparable with the list lengths used in the SD of [3]. We note that in both the MCMC and the implementation of SD that we have adopted (taking into account the a prior information from the channel decoder) the length of the list $\mathcal{L}$ decreases rapidly over iterations of the turbo loop. This has a significant impact on the average list size. For example, in our simulations we have observed that in the last few iterations the size of list often reduces to only one sample. This is because the choice of the samples become more limited due to increasingly accurate prior information from the channel decoder; also, see the comments around equation (6), below.

The results of the case 4 transmit and 4 receive antennas are presented in Fig. 3. The BER results of SD of [3] are presented along with those of the MCMC, modified SD and the exact max-log SD detectors. For the MCMC detector, the parameters $L = 10$ and $N_s = 10$ are used. For the modified SD detector, $N_{\max}$ is set to be 100, and $N_{\min}$ is set equal to 25, 20, 15,10, and 5 for the first, second, third, forth, and fifth iterations of the turbo loop, respectively. For the sixth iteration onwards, $N_{\min}$ is set equal to one. These choices of $N_{\min}$ are made on the following basis. In the first iterations, when no prior information from channel decoder is available, the term $2\sigma^2 \ln P^{\rm e}(\mathbf{d})$ on the left hand side of (2) is a constant, i.e., it does not vary with $\mathbf{d}$. Hence, (2) may be rearranged as

$$(\mathbf{d} - \hat{\mathbf{d}})^\dagger \mathbf{H}^\dagger \mathbf{H} (\mathbf{d} - \hat{\mathbf{d}}) < r'^2 \qquad (6)$$

where $r' = \sqrt{r^2 + 2\sigma^2 \ln P^{\rm e}(\mathbf{d})}$. The sphere defined by (6), when $r'$ exceeds a certain value, usually contains a relatively large number of samples of $\mathbf{d}$. In other words, there are many choices of $\mathbf{d}$ at about the same distance from $\hat{\mathbf{d}}$ and to obtain a good estimate of LLR values a good portion of these samples should be included in the list $\mathcal{L}$. On the other hand, in the later iterations when more prior information about $\mathbf{d}$ is available, $2\sigma^2 \ln P^{\rm e}(\mathbf{d})$ varies with $\mathbf{d}$. It becomes close to zero for more likely values of $\mathbf{d}$, when $P^{\rm e}(\mathbf{d}) \approx 1$, and becomes a large negative number for less likely values of $\mathbf{d}$, when $P^{\rm e}(\mathbf{d}) \ll 1$. In the latter case, $r'^2$ may become a very small or a negative number, and hence (6) may have no solution. From this argument, it is clear that one should expect more solutions to (2) in the first few iterations of the turbo loop and less solutions in the later iterations.

From the results of Fig. 3, we observe that the performance of the MCMC detector is very close to that of the exact max-log SD detector. The modified SD performs better than the original SD of [3], however, not as good as the MCMC and the exact max-log SD detectors. This loss can be explained as follows. As discussed in Section II, SD adopts a tree search approach to obtain samples of $\mathbf{d}$. From the example given in Fig. 2, the following observation is made. The samples of $\mathbf{d}$ are chosen sequentially starting from the left side of the tree and moving to the right. When for a given $r$ the number of samples that satisfy (2) is larger than $N_{\max}$, some of these samples will be excluded from the list $\mathcal{L}$. This truncation of the samples results in a list $\mathcal{L}$ that has a non-even distribution around the center of the sphere, $\hat{\mathbf{d}}$. Say, if in the example of Fig. 2, $N_{\max}$ is set equal to 3, all the samples taken will be from the left side of the tree, which means, for all samples, $d_1 = +1$. In GS, on the other hand, such a non-even distribution is naturally avoided since elements of $\mathbf{d}$ are visited on a regular basis. The accuracy of this argument can be confirmed by removing the upper limit $N_{\max}$ in our simulations. The results (not shown here) are very similar to those of the MCMC detector.

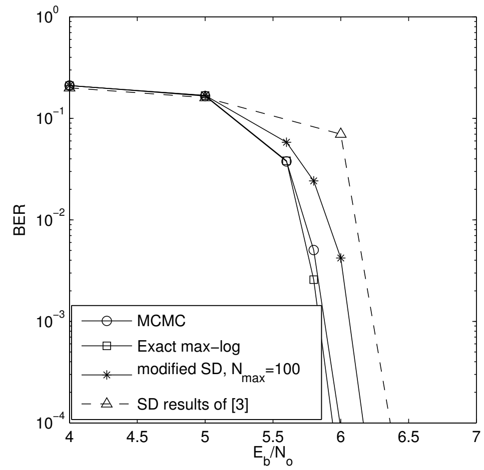


Fig. 3. BER results of the SD, modified SD, MCMC and exact max-log SD detectors for a MIMO system with 4 transmit and 4 receive antennas.

The above experiment was repeated for a MIMO system with 8 transmit and 8 receive antennas. The results are presented in Fig. 4. Here, we have presented the results of the modified SD and MCMC detectors and also those of SD of [3]. For the MCMC detector, the parameters $L = 20$ and $N_s = 20$ are used. For the modified SD detector, two choices of $N_{\max} = 400$ and 800 are considered, and $N_{min}$ is set equal to 50, 40, 30,20, and 10 for the first, second, third, forth, and fifth iterations of the turbo loop, respectively. For the sixth iteration onwards, $N_{\min}$ is set equal to one. We are unable to present the results of the exact max-log SD detector because of its prohibitive complexity. Here, the MCMC detector outperforms the SD of [3] by about 1 dB. The modified SD also outperforms the SD of [3]. However, similar to the case of Fig. 3, it does not perform as good as the MCMC detector. Also, note that the performance of the modified SD improves as $N_{\max}$ increases. This, of course, is at the cost of added complexity.

## VII. Computational Complexity

Hassibi and Vikalo [11] have studied the computational complexity of SD and have shown that the widely cited polynomial complexity of SD is only valid when SNR is high. When SNR is low, the complexity of SD is predicted as exponential. Similar results have been reported in [12], where the authors have discussed a number of branch and bound (BBD) algorithms, including SD.

The exponential complexity of SD is a direct consequence of the fact that the SD tree search algorithm often encounters many terminal nodes before finding a sample of $\mathbf{d}$, and it grows exponentially with the size of $\mathbf{d}$. The number of terminal nodes increases further in the particular case of iterative detectors where many samples of $\mathbf{d}$ have to be collected per channel use. Computational complexity of MCMC detector, on the other hand, is independent of SNR and merely is determined by the parameters $L$ and $N_{\rm s}$, the number of

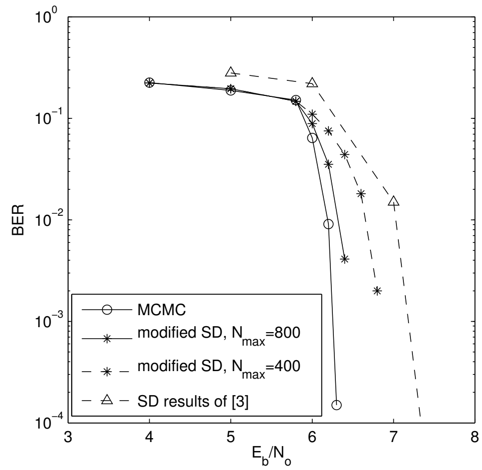


Fig. 4. BER results of the SD, modified SD, MCMC and exact max-log SD detectors for a MIMO system with 8 transmit and 8 receive antennas.

transmit antennas, and the constellation size of transmitted data symbols and has an order that is determined by the multiplication these numbers.

Table I presents the number of multiplication and addition operations per data block, counted while running our simulation programs, at the indicated SNR values. Measuring the complexity by these numbers, SD and MCMC detectors exhibit similar complexity in the case of a $4 \times 4$ MIMO channel. However, because of the exponential growth of the complexity of SD with the size of $\mathbf{d}$, in the case of an $8 \times 8$ channel, the complexity difference between MCMC and SD grows to over an order of magnitude.

TABLE I
COMPLEXITY STUDY OF SD AND MCMC DETECTORS.

| | SNR (dB) | Channel size | $L$, $N_s$ or $N_{\max}$ , $N_{\min}$ | No. of Operations |
|---|---|---|---|---|
| MCMC | 5.6 | $4 \times 4$ | 10, 10 | $1.32 \times 10^8$ |
| SD | 5.6 | $4 \times 4$ | 100, 10 | $1.16 \times 10^8$ |
| MCMC | 6.5 | $8 \times 8$ | 20, 20 | $9.72 \times 10^8$ |
| SD | 6.5 | $8 \times 8$ | 400, 10 | $1.40 \times 10^{10}$ |
| SD | 6.5 | $8 \times 8$ | 800, 10 | $1.69 \times 10^{10}$ |

While the numbers presented in Table I provide some measure of complexity of our current implementation, it is important to note that both SD and MCMC have a great deal of structural regularity which may be used to further improve the complexity of both algorithms. We thus wish to acknowledge that an accurate comparison of SD and MCMC requires a more thorough study, where the implementations of both algorithms are optimized.

Another point that should be noted is that the focus of this letter is on low SNR regime. The situation at higher values of SNR differs significantly. The SD, definitely, is a valuable technique for MAP detection in high SNR regime, where through a judicious choice of the radius $r$ and more elaborate tree search algorithms, such as [10], one can realize SD with moderate complexity. MCMC detector, on the other hand, may encounter some problems as discussed in [8]. A more thorough comparison of SD and MCMC detection algorithms for high SNR regime is underway and will be reported in future.

## VIII. CONCLUSIONS

In this paper, we explored the similarities and the differences of sphere decoding (SD) and the statistical methods that work based on Markov chain Monte Carlo (MCMC) simulation in soft MIMO detectors. It was noted that both methods start with construction of a candidate list of more likely samples of the transmitted symbols. While SD takes a deterministic approach (a tree search), MCMC detector obtains the samples through a statistical procedure. The candidate list is then used for computation of the LLR values of the information bits. We found that the way the candidate list is handled by the reported SD algorithms [3], [4] is different from what has been suggested in a recently proposed MCMC detector [7], [8]. By applying the method of [7], [8] to SD, we proposed a new detector and called it modified SD. Moreover, we proposed an exact max-log SD detector which may be used as benchmark for evaluation of the MCMC and SD detectors. Computer simulations revealed that while an MCMC detector with moderate complexity can approach the performance of the exact max-log SD detector, SD, even after the modification proposed in this letter, falls short in performance. We also briefly discussed the complexity of SD and MCMC detection and found that in low SNR regime the latter may offer a significantly lower computational complexity.

## REFERENCES


[1] X. Wang and H.V. Poor, "Iterative (Turbo) Soft Interference Cancellation and Decoding for Coded CDMA", *IEEE Trans. Commun.*, vol. 47, no. 7, July 1999.

[2] H.E. Gamal and A.R. Hammons, Jr., "A new approach to layered space-time coding and signal processing," *IEEE Trans Info. Theory,* vol. 47, no. 6, pp. 2321 - 2334, Sept. 2001.

[3] 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, March 2003.

[4] H. Vikalo, and B. Hassibi, "Modified Fincke-Pohst algorithm for low-complexity iterative decoding over multiple antenna channels, in *Proceedings of IEEE International Symposium on Information Theory,* ISIT 2002, pp. 390.

[5] C.P. Robert and G. Casella, *Monte Carlo Statistical Methods.* Springer-Verlag, New York, 1999.

[6] O. Häggström, *Finite Markov Chains and Algorithmic Applications.* Cambridge University Press, 2002.

[7] 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 Proceedings of Globecom'04, Hyatt Regency, Dallas, Nov. 29 to Dec. 02, 2004.

[8] B. Farhang-Boroujeny, H. Zhu, and Z. Shi, 'Markov chain Monte Carlo algorithms for CDMA and MIMO communication systems," Submitted for Publication.

[9] P. Robertson, E. Villebrun, P. Hoeher, "A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain." *IEEE International Conference on Communications,* ICC 95, Seattle, 18-22 June 1995, vol. 2, pp. 1009 - 1013.

[10] D. Pham, K. R. Pattipati, P. K. Willett, and J. Luo, "An improved complex sphere decoder for V-BLAST systems," *IEEE Sig. Proc. Letters,* vol. 11, no. 3, Sept. 2004, pp. 748-571.

[11] B. Hassibi and H. Vikalo, "On the sphere decoding algorithm: Part I, the expected complexity," submitted to IEEE Transactions on Signal Processing, available at http://www.caltech.edu.

[12] J. Luo, K.R. Pattipati, P. Willett, and G.G. Levchuk, "Fast optimal and suboptimal any-time algorithms for CDMA multiuser detection based on brach and bound," *IEEE Trans. Commun.*, vol. 52, no. 4, pp. 632-642, April 2004.