# A Low-Complexity Near-ML Performance Achieving Algorithm for Large MIMO Detection

Saif K. Mohammed, A. Chockalingam, and B. Sundar Rajan
Department of ECE, Indian Institute of Science, Bangalore 560012, INDIA

***Abstract*—In this paper, we present a low-complexity, near maximum-likelihood (ML) performance achieving detector for *large* MIMO systems having *tens of transmit and receive antennas*. Such large MIMO systems are of interest because of the high spectral efficiencies possible in such systems. The proposed detection algorithm, termed as *multistage likelihood-ascent search (M-LAS)* algorithm, is rooted in Hopfield neural networks, and is shown to possess excellent performance as well as complexity attributes. In terms of performance, in a $64 \times 64$ V-BLAST system with 4-QAM, the proposed algorithm achieves an uncoded BER of $10^{-3}$ at an SNR of just about 1 dB away from AWGN-only SISO performance given by $Q(\sqrt{SNR})$. In terms of coded BER, with a rate-3/4 turbo code at a spectral efficiency of 96 bps/Hz the algorithm performs close to within about 4.5 dB from theoretical capacity, which is remarkable in terms of both high spectral efficiency as well as nearness to theoretical capacity. Our simulation results show that the above performance is achieved with a complexity of just $O(N_tN_r)$ per symbol, where $N_t$ and $N_r$ denote the number of transmit and receive antennas.**

## I. Introduction

MIMO techniques have become popular in realizing spatial diversity and high data rates through the use of multiple transmit antennas [1]. We consider large MIMO systems with *tens* of transmit and receive antennas, which are of interest due to the high spectral efficiencies possible in such systems. The key issues in realizing large MIMO systems include low-complexity detection, channel estimation, and communication terminal size to accommodate large number of antennas. We address the issue of low-complexity detection in large MIMO systems here. More recent approaches to low-complexity multiuser detection and MIMO detection involve application of techniques from belief propagation [2], neural networks [3],[4], Markov chain Monte-Carlo methods [5], probabilistic data association [6], etc. Detectors based on these techniques have been shown to achieve an average per-bit complexity linear in number of users, while achieving near-optimal performance in large multiuser CDMA system settings. These powerful techniques are increasingly being adopted in MIMO detection. In [4], we presented a Hopfield neural network based likelihood ascent search (LAS) algorithm for large MIMO detection; we showed that the LAS detector achieves near-AWGN SISO performance in a large MIMO setting with hundreds of antennas, while performing close to within 4.6 dB from theoretical capacity.

Our present work here on *multistage LAS (M-LAS) detector* differs from that in [4] in two key aspects, namely, $i$) while the LAS algorithm in [4] operates only on 1-symbol updates in each search step, in the present M-LAS algorithm we *devise a low-complexity multiple symbol update strategy* that results in improved performance compared to that of LAS in [4] with a small increase in complexity, and $ii$) in addition, we present a method to generate soft bit values from the M-LAS output vector to be fed as input to the turbo decoder; soft bit values generation method is not available in [4]. Our simulation results show that the M-LAS detector achieves near AWGN SISO performance even with tens of antennas, while the LAS needed hundreds of antennas to achieve near AWGN performance. This performance advantage of M-LAS over LAS in the regime of tens of antennas has interesting practical implications, as tens of antennas can be easily placed in moderately sized communication terminals (e.g., laptops) enabling large MIMO systems to be viable in practice. With an outer turbo code, the M-LAS is shown to perform close to within about 4.5 dB from theoretical capacity.

## II. System Model

Consider a V-BLAST system with $N_t$ transmit antennas and $N_r$ receive antennas, $N_t \leq N_r$, where $N_t$ symbols are transmitted from $N_t$ transmit antennas simultaneously. Let $\mathbf{x}_c \in \mathbb{C}^{N_t \times 1}$ be the symbol vector transmitted. Each element of $\mathbf{x}_c$ is an $M$-PAM or $M$-QAM symbol. $M$-PAM symbols take discrete values from $\{A_m, m = 1, 2, \cdots, M\}$, where $A_m = (2m - 1 - M)$, and $M$-QAM is nothing but two PAMs in quadrature. Let $\mathbf{H}_c \in \mathbb{C}^{N_r \times N_t}$ be the channel gain matrix, such that the $(p, q)$th entry $h_{p,q}$ is the complex channel gain from the $q$th transmit antenna to the $p$th receive antenna. Assuming rich scattering, we model the entries of $\mathbf{H}_c$ as i.i.d $\mathcal{CN}(0, 1)$. Let $\mathbf{y}_c \in \mathbb{C}^{N_r \times 1}$ and $\mathbf{n}_c \in \mathbb{C}^{N_r \times 1}$ denote the received signal vector and the noise vector, respectively, at the receiver, where the entries of $\mathbf{n}_c$ are modeled as i.i.d $\mathcal{CN}(0, \sigma^2)$. The received signal vector can then be written as

$$\mathbf{y}_c = \mathbf{H}_c\mathbf{x}_c + \mathbf{n}_c. \tag{1}$$

Let $\mathbf{y}_c$, $\mathbf{H}_c$, $\mathbf{x}_c$, and $\mathbf{n}_c$ be decomposed into real and imaginary parts as follows:

$$\begin{aligned} \mathbf{y}_c &= \mathbf{y}_I + j\mathbf{y}_Q, \quad \mathbf{x}_c = \mathbf{x}_I + j\mathbf{x}_Q, \\ \mathbf{n}_c &= \mathbf{n}_I + j\mathbf{n}_Q, \quad \mathbf{H}_c = \mathbf{H}_I + j\mathbf{H}_Q. \end{aligned} \tag{2}$$

Further, we define $\mathbf{H}_r \in \mathbb{R}^{2N_r \times 2N_t}$, $\mathbf{y}_r \in \mathbb{R}^{2N_r \times 1}$, $\mathbf{x}_r \in \mathbb{R}^{2N_t \times 1}$, and $\mathbf{n}_r \in \mathbb{R}^{2N_r \times 1}$ as

$$\mathbf{H}_r = \left( \begin{array}{cc} \mathbf{H}_I & -\mathbf{H}_Q \\ \mathbf{H}_Q & \mathbf{H}_I \end{array} \right),$$

$$\mathbf{y}_r = [\mathbf{y}_I^T \ \ \mathbf{y}_Q^T]^T, \ \ \mathbf{x}_r = [\mathbf{x}_I^T \ \ \mathbf{x}_Q^T]^T, \ \ \mathbf{n}_r = [\mathbf{n}_I^T \ \ \mathbf{n}_Q^T]^T. \tag{3}$$

Now, (1) can be written as

$$\mathbf{y}_r = \mathbf{H}_r\mathbf{x}_r + \mathbf{n}_r. \tag{4}$$

Henceforth, we shall work with the real-valued system in (4). For notational simplicity, we drop subscripts $r$ in (4) and write

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}, \tag{5}$$

where $\mathbf{H} = \mathbf{H}_r \in \mathbb{R}^{2N_r \times 2N_t}$, $\mathbf{y} = \mathbf{y}_r \in \mathbb{R}^{2N_r \times 1}$, $\mathbf{x} = \mathbf{x}_r \in \mathbb{R}^{2N_t \times 1}$ and $\mathbf{n} = \mathbf{n}_r \in \mathbb{R}^{2N_r \times 1}$. With the above real-valued

978-1-4244-2571-6/08/$25.00 ©2008 IEEE

system model, the real-part of the original complex data symbols will be mapped to $[x_1, \cdots, x_{N_t}]$ and the imaginary-part of these symbols will be mapped to $[x_{N_t+1}, \cdots, x_{2N_t}]$. For $M$-PAM, $[x_{N_t+1}, \cdots, x_{2N_t}]$ will be zeros since $M$-PAM symbols take only real values. In the case of $M$-QAM, $[x_1, \cdots, x_{N_t}]$ can viewed to be from an underlying $M$-PAM signal set and so is $[x_{N_t+1}, \cdots, x_{2N_t}]$. Let $\mathbb{A}_i$ denote the $M$-PAM signal set from which $x_i$ takes values, $i = 1, 2, \cdots, 2N_t$. For example, for 4-PAM, $\mathbb{A}_i = \{-3, -1, 1, 3\}$ for $i = 1, 2, \cdots, N_t$ and $\mathbb{A}_i = \{0\}$ for $i = N_{t+1}, \cdots, 2N_t$. Similarly, for 4-QAM, after transforming the system into an equivalent real-valued system, $\mathbb{A}_i = \{1, -1\}$ for $i = 1, 2, \cdots, 2N_t$. Now, define a $2N_t$-dimensional signal space $\mathbb{S}$ to be the Cartesian product of $\mathbb{A}_1$ to $\mathbb{A}_{2N_t}$. The ML solution vector, $\mathbf{d}_{ML}$, is given by

$$\mathbf{d}_{ML} = \underset{\mathbf{d} \in \mathbb{S}}{\arg\min} \; \|\mathbf{y} - \mathbf{H}\mathbf{d}\|^2 = \underset{\mathbf{d} \in \mathbb{S}}{\arg\min} \; \mathbf{d}^T\mathbf{H}^T\mathbf{H}\mathbf{d} - 2\mathbf{y}^T\mathbf{H}\mathbf{d}, \quad (6)$$

whose complexity is exponential in $N_t$. We present a low-complexity high-performance detection algorithm next.

## III. Proposed M-LAS Detector

The proposed M-LAS algorithm essentially consists of a sequence of likelihood-ascent search stages, where the likelihood increases monotonically with every search stage. Each search stage consists of several iterations, where we update one symbol per iteration such that the likelihood monotonically increases from one iteration to the next until a local minima is reached. Upon reaching this local minima, we try a 2-symbol and/or a 3-symbol update in order to further increase the likelihood. If this likelihood increase happens, we initiate the next search stage starting from this new point. The algorithm terminates at the stage from where further likelihood increase does not happen.

The M-LAS algorithm starts with an initial solution $\mathbf{d}^{(0)}$, given by $\mathbf{d}^{(0)} = \mathbf{B}\mathbf{y}$, where $\mathbf{B}$ is the initial solution filter, which can be a matched filter (MF) or zero-forcing (ZF) filter or MMSE filter. The index $m$ in $\mathbf{d}^{(m)}$ denotes the iteration number in a given search stage. The ML cost function after the $k$th iteration in a given search stage is given by

$$C^{(k)} = \mathbf{d}^{(k)^T}\mathbf{H}^T\mathbf{H}\mathbf{d}^{(k)} - 2\mathbf{y}^T\mathbf{H}\mathbf{d}^{(k)}. \quad (7)$$

Each search stage would involve a sequence of 1-symbol updates followed by a 2 and/or a 3 symbol update.

### *A. One-Symbol Update*

Let us assume that we update the $p$th symbol in the $(k+1)$th iteration; $p$ can take value from $1, \cdots, N_t$ for $M$-PAM and $1, \cdots, 2N_t$ for $M$-QAM. The update rule can be written as

$$\mathbf{d}^{(k+1)} = \mathbf{d}^{(k)} + \lambda_p^{(k)}\mathbf{e}_p, \quad (8)$$

where $\mathbf{e}_p$ denotes the unit vector with its $p$th entry only as one, and all other entries as zero. Also, for any iteration $k$, $\mathbf{d}^{(k)}$ should belong to the space $\mathbb{S}$, and therefore $\lambda_p^{(k)}$ can take only certain integer values. For example, in case of 4-PAM or 16-QAM (both have the same signal set $\mathbb{A}_p = \{-3, -1, 1, 3\}$), $\lambda_p^{(k)}$ can take values only from $\{-6, -4, -2, 0, 2, 4, 6\}$. Using (7) and (8), and defining a matrix $\mathbf{G}$ as

$$\mathbf{G} \;\; \stackrel{\triangle}{=} \;\; \mathbf{H}^T\mathbf{H}, \quad (9)$$

we can write the cost difference $\Delta C_p^{k+1} \stackrel{\triangle}{=} C^{(k+1)} - C^{(k)}$ as

$$\Delta C_p^{k+1} \;\; = \;\; \lambda_p^{(k)^2}(\mathbf{G})_{p,p} - 2\lambda_p^{(k)} z_p^{(k)},$$

where $\mathbf{h}_p$ is the $p$th column of $\mathbf{H}$, $\mathbf{z}^{(k)} = \mathbf{H}^T(\mathbf{y} - \mathbf{H}\mathbf{d}^{(k)})$, $z_p^{(k)}$ is the $p$th entry of the $\mathbf{z}^{(k)}$ vector, and $(\mathbf{G})_{p,p}$ is the $(p,p)$th entry of the $\mathbf{G}$ matrix. Also, let us define $a_p$ and $l_p^{(k)}$ as

$$a_p = (\mathbf{G})_{p,p}, \quad l_p^{(k)} = |\lambda_p^{(k)}|. \quad (10)$$

With the above variables defined, we can rewrite (10) as

$$\Delta C_p^{k+1} \;\; = \;\; l_p^{(k)^2} a_p - 2 l_p^{(k)} |z_p^{(k)}| \operatorname{sgn}(\lambda_p^{(k)}) \operatorname{sgn}(z_p^{(k)}), \quad (11)$$

where sgn(.) denotes the signum function. For the ML cost function to reduce from the $k$th to the $(k+1)$th iteration, the cost difference should be negative. Using this fact and that $a_p$ and $l_p^{(k)}$ are non-negative quantities, we can conclude from (11) that the sign of $\lambda_p^{(k)}$ must satisfy

$$\operatorname{sgn}(\lambda_p^{(k)}) \;\; = \;\; \operatorname{sgn}(z_p^{(k)}). \quad (12)$$

Using (12) in (11), the ML cost difference can be rewritten as

$$\mathcal{F}(l_p^{(k)}) \;\; \stackrel{\triangle}{=} \;\; \Delta C_p^{k+1} \;\; = \;\; l_p^{(k)^2} a_p - 2 l_p^{(k)} |z_p^{(k)}|. \quad (13)$$

For $\mathcal{F}(l_p^{(k)})$ to be non-positive, the necessary and sufficient condition from (13) is that

$$l_p^{(k)} \;\; < \;\; \frac{2|z_p^{(k)}|}{a_p}. \quad (14)$$

However, we can find the value of $l_p^{(k)}$ which satisfies (14) and at the same time gives the largest descent in the ML cost function from the $k$th to the $(k+1)$th iteration (when symbol $p$ is updated). Also, $l_p^{(k)}$ is constrained to take only certain integer values, and therefore the brute-force way to get optimum $l_p^{(k)}$ is to evaluate $\mathcal{F}(l_p^{(k)})$ at all possible values of $l_p^{(k)}$. This would become computationally expensive as the constellation size $M$ increases. However, for the case of 1-symbol update, we could obtain a closed-form expression for the optimum $l_p^{(k)}$ that minimizes $\mathcal{F}(l_p^{(k)})$, which is given by

$$l_{p,opt}^{(k)} \;\; = \;\; 2\left\lfloor \frac{|z_p^{(k)}|}{2a_p} \right\rceil, \quad (15)$$

where $\lfloor . \rceil$ denotes the rounding operation. If the $p$th symbol in $\mathbf{d}^{(k)}$, i.e., $d_p^{(k)}$, were indeed updated, then the new value of the symbol would be given by

$$\tilde{d}_p^{(k+1)} \;\; = \;\; d_p^{(k)} + l_p^{(k)} \operatorname{sgn}(z_p^{(k)}). \quad (16)$$

However, $\tilde{d}_p^{(k+1)}$ can take values only in the set $\mathbb{A}_p$, and therefore we need to check for the possibility of $\tilde{d}_p^{(k+1)}$ being greater than $(M-1)$ or less than $-(M-1)$. If $\tilde{d}_p^{(k+1)} > (M-1)$, then $l_p^{(k)}$ is adjusted so that the new value of $\tilde{d}_p^{(k+1)}$ with the adjusted value of $l_p^{(k)}$ (using (16)) is $(M-1)$. Similarly, if $\tilde{d}_p^{(k+1)} < -(M-1)$, then $l_p^{(k)}$ is adjusted so that the new value of $\tilde{d}_p^{(k+1)}$ is $-(M-1)$. That is, if $\tilde{d}_p^{(k+1)} > (M-1)$, the adjustment equation is

$$l_p^{(k)} \;\; = \;\; l_p^{(k)} - \operatorname{sgn}(z_p^{(k)})\big(\tilde{d}_p^{(k+1)} - (M-1)\big), \quad (17)$$

and if $\tilde{d}_p^{(k+1)} < -(M-1)$, the adjustment equation is

$$l_p^{(k)} \;\; = \;\; l_p^{(k)} - \operatorname{sgn}(z_p^{(k)})\big(\tilde{d}_p^{(k+1)} + (M-1)\big). \quad (18)$$

Let $\tilde{l}^{(k)}_{p,opt}$ be obtained from $l^{(k)}_{p,opt}$ by using the adjustment equations (17) and (18). It can be shown that if $\mathcal{F}(l^{(k)}_{p,opt})$ is non-positive, then $\mathcal{F}(\tilde{l}^{(k)}_{p,opt})$ is also non-positive. We compute $\mathcal{F}(\tilde{l}^{(k)}_{p,opt}), \forall\, p = 1, \cdots, 2N_t$, Now, given $\mathcal{F}(\tilde{l}^{(k)}_{p,opt}), \forall\, p$, let

$$s \quad = \quad \underset{p}{\arg\min} \;\; \mathcal{F}(\tilde{l}^{(k)}_{p,opt}). \tag{19}$$

If $\mathcal{F}(\tilde{l}^{(k)}_{s,opt}) < 0$, the update for the $(k+1)$th iteration is

$$\mathbf{d}^{(k+1)} \;=\; \mathbf{d}^{(k)} + l^{(k)}_{s,opt}\,\mathrm{sgn}(z^{(k)}_s)\,\mathbf{e}_s, \tag{20}$$

$$\mathbf{z}^{(k+1)} \;=\; \mathbf{z}^{(k)} - l^{(k)}_{s,opt}\,\mathrm{sgn}(z^{(k)}_s)\,\mathbf{g}_s. \tag{21}$$

where $\mathbf{g}_s$ is the $s$th column of $\mathbf{G}$. The update in (21) follows from the definition of $\mathbf{z}^{(k)}$ in (10). If $\mathcal{F}(\tilde{l}^{(k)}_{s,opt}) \geq 0$, then the 1-symbol update search terminates. The data vector at this point is referred to as ‘1-symbol update local minima’. After reaching the 1-symbol update local minima, we look for a further decrease in the cost function by updating multiple symbols simultaneously.

### *B. Why Multiple Symbol Updates?*

The motivation for trying out multiple symbol updates can be explained as follows. Let $\mathbb{L}_K \subseteq \mathbb{S}$ denote the set of data vectors such that for any $\mathbf{d} \in \mathbb{L}_K$, if a $K$-symbol update is performed on $\mathbf{d}$ resulting in a vector $\mathbf{d}'$, then $||\mathbf{y} - \mathbf{Hd}'|| \geq ||\mathbf{y} - \mathbf{Hd}||$. We note that $\mathbf{d}_{ML} \in \mathbb{L}_K, \forall\, K = 1, 2, \cdots, 2N_t$, because any number of symbol updates on $\mathbf{d}_{ML}$ will not decrease the cost function. We define another set $\mathbb{M}_K = \bigcap_{j=1}^{K} \mathbb{L}_j$. Note that $\mathbf{d}_{ML} \in \mathbb{M}_K, \forall\, K = 1, 2, \cdots, 2N_t$, and $\mathbb{M}_{2N_t} = \{\mathbf{d}_{ML}\}$, i.e., $\mathbb{M}_{2N_t}$ is a singleton set with $\mathbf{d}_{ML}$ as the only element. Also, $|\mathbb{M}_{K+1}| \leq |\mathbb{M}_K|$, $K = 1, 2, \cdots, 2N_t - 1$. For any $\mathbf{d} \in \mathbb{M}_K$, $K = 1, 2, \cdots, 2N_t$ and $\mathbf{d} \neq \mathbf{d}_{ML}$, it can be seen that $\mathbf{d}$ and $\mathbf{d}_{ML}$ will differ in $K+1$ or more locations. Since $\mathbf{d}_{ML} \in \mathbb{M}_K$, and $|\mathbb{M}_K|$ decreases monotonically with increasing $K$, there will be lesser non-ML data vectors to which the algorithm can converge to for increasing $K$. In addition, at moderate to high SNRs, $\mathbf{d}_{ML} = \mathbf{x}$ with high probability. Therefore, the separation between any $\mathbf{d} \in \mathbb{M}_K$ and $\mathbf{x}$ will monotonically increase with increasing $K$ with high probability. Therefore, the probability of the noise vector $\mathbf{n}$ inducing an error would decrease with increasing $K$. This indicates that $K$-symbol updates with large $K$ could get near to ML performance. However, the overall complexity with up to $K$-symbol simultaneous updates allowed would be of order $O(N_t^K)$. So, in order to limit the complexity to $O(N_t^2)$ per symbol, we restrict the updates to $K = 3$. Since only up to 3-symbol updates are considered in the proposed algorithm, it follows that the algorithm would always converge to a data vector in $\mathbb{M}_3$.

### *C. Two-Symbol Update*

Let us consider 2-symbol update in this subsection. Let us assume that we update the $p$th and $q$th symbols in the $(k+1)$th iteration; $p$ and $q$ can take values from $1, \cdots, N_t$ for $M$-PAM and $1, \cdots, 2N_t$ for $M$-QAM. The update rule for the 2-symbol update can be written as

$$\mathbf{d}^{(k+1)} = \mathbf{d}^{(k)} + \lambda^{(k)}_p \mathbf{e}_p + \lambda^{(k)}_q \mathbf{e}_q. \tag{22}$$

For any iteration $k$, $\mathbf{d}^{(k)}$ should belong to the space $\mathbb{S}$, and therefore $\lambda^{(k)}_p$ and $\lambda^{(k)}_q$ can take only certain integer values. In particular, $\lambda^{(k)}_p \in \mathbb{A}^{(k)}_p$, and $\lambda^{(k)}_q \in \mathbb{A}^{(k)}_q$. If $\mathbb{A}_p$ is the $M$-PAM signal set, then $\mathbb{A}^{(k)}_p \triangleq \{x | (x + d^{(k)}_p) \in \mathbb{A}_p, x \neq 0\}$, and so is the definition for $\mathbb{A}^{(k)}_q$. Here, $d^{(k)}_p$ refers to the $p$th symbol in the data vector $\mathbf{d}^{(k)}$. For example, both 4-PAM and 16-QAM will have the same set $\mathbb{A}_p = \{-3, -1, 1, 3\}$, and $d^{(k)}_p$ is -1, then $\mathbb{A}^{(k)}_p = \{-2, 2, 4\}$. Similar definitions can be obtained for non-square $M$-QAM signal sets as well.

If the symbols were updated as given by (22), then using (7), we can write the cost difference function $\Delta C^{k+1}_{p,q}(\lambda^{(k)}_p, \lambda^{(k)}_q) \triangleq C^{(k+1)} - C^{(k)}$ as

$$\begin{aligned}\Delta C^{k+1}_{p,q}(\lambda^{(k)}_p, \lambda^{(k)}_q) \;&=\; {\lambda^{(k)}_p}^2 (\mathbf{G})_{p,p} + {\lambda^{(k)}_q}^2 (\mathbf{G})_{q,q} \\ &+ 2\lambda^{(k)}_p \lambda^{(k)}_q (\mathbf{G})_{p,q} - 2\lambda^{(k)}_p z^{(k)}_p - 2\lambda^{(k)}_q z^{(k)}_q,\end{aligned} \tag{23}$$

where $\lambda^{(k)}_p \in \mathbb{A}^{(k)}_p$ and $\lambda^{(k)}_q \in \mathbb{A}^{(k)}_q$. We can write this compactly as $(\lambda^{(k)}_p, \lambda^{(k)}_q) \in \mathbb{A}^{(k)}_{p,q}$, where $\mathbb{A}^{(k)}_{p,q}$ denotes the Cartesian product of $\mathbb{A}^{(k)}_p$ and $\mathbb{A}^{(k)}_q$. For a given $p$ and $q$, in order to decrease the ML cost function, we would like to choose a pair $(\lambda^{(k)}_p, \lambda^{(k)}_q)$ such that $\Delta C^{k+1}_{p,q}$ given by (23) is negative. If multiple pairs exist for which $\Delta C^{k+1}_{p,q}$ is negative, we choose the pair which results in the most negative value of $\Delta C^{k+1}_{p,q}$.

Unlike 1-symbol update, for 2-symbol update $\Delta C^{k+1}_{p,q}(\lambda^{(k)}_p, \lambda^{(k)}_q)$ in (23) is a function of two discrete valued variables, and so we do not have a closed-form expression for $(\lambda^{(k)}_{p,opt}, \lambda^{(k)}_{q,opt})$. Consequently, a brute-force method is to evaluate $\Delta C^{k+1}_{p,q}(\lambda^{(k)}_p, \lambda^{(k)}_q)$ over all possible values of $(\lambda^{(k)}_p, \lambda^{(k)}_q)$, i.e.,

$$(\lambda^{(k)}_{p,opt}, \lambda^{(k)}_{q,opt}) = \underset{(\lambda^{(k)}_p, \lambda^{(k)}_q) \in \mathbb{A}^{(k)}_{p,q}}{\arg\min} \Delta C^{k+1}_{p,q}(\lambda^{(k)}_p, \lambda^{(k)}_q). \tag{24}$$

We denote the minimum value of the $\Delta C^{k+1}_{p,q}(\lambda^{(k)}_p, \lambda^{(k)}_q)$ obtained from the above minimization as

$$\Delta C^{k+1}_{p,q,opt} \;\triangleq\; \Delta C^{k+1}_{p,q}(\lambda^{(k)}_{p,opt}, \lambda^{(k)}_{q,opt}). \tag{25}$$

The computational complexity in (24) $O(M^2)$ for $M$-PAM and $O(M)$ for $M$-QAM. Approximate methods can be adopted to solve (24) using lesser complexity. One such method which can give closed-form expression for the solution is as follows. The cost difference function in (23) can be rewritten as

$$\Delta C^{k+1}_{p,q}(\lambda^{(k)}_p, \lambda^{(k)}_q) = {\mathbf{\Lambda}^{(k)}_{p,q}}^T \mathbf{F}_{p,q}\, \mathbf{\Lambda}^{(k)}_{p,q} - 2{\mathbf{\Lambda}^{(k)}_{p,q}}^T \mathbf{z}^{(k)}_{p,q}, \tag{26}$$

where $\mathbf{\Lambda}^{(k)}_{p,q} \triangleq [\lambda^{(k)}_p \lambda^{(k)}_q]^T$ and $\mathbf{z}^{(k)}_{p,q} \triangleq [z^{(k)}_p z^{(k)}_q]^T$. Also, $\mathbf{F}_{p,q} \in \mathbb{R}^{2\times 2}$ is the $2 \times 2$ sub-matrix of $\mathbf{G}$ containing only the elements in the $p$th and $q$th rows and columns. Therefore, $(\mathbf{F}_{p,q})_{1,1} \triangleq (\mathbf{G})_{p,p}$, $(\mathbf{F}_{p,q})_{1,2} \triangleq (\mathbf{G})_{p,q}$, $(\mathbf{F}_{p,q})_{2,1} \triangleq (\mathbf{G})_{q,p}$, and $(\mathbf{F}_{p,q})_{2,2} \triangleq (\mathbf{G})_{q,q}$. Since $\Delta C^{k+1}_{p,q}(\lambda^{(k)}_p, \lambda^{(k)}_q)$ is a strictly convex quadratic function (the Hessian $\mathbf{F}_{p,q}$ is always positive definite), a unique global minima exists, and is given by

$$\tilde{\mathbf{\Lambda}}^{(k)}_{p,q} = \mathbf{F}^{-1}_{p,q}\; \mathbf{z}^{(k)}_{p,q}. \tag{27}$$

However, the solution given by (27) need not lie in $\mathbb{A}^{(k)}_{p,q}$, and, therefore, we first round-off the solution to the nearest elements in $\mathbb{A}_{p,q}$, where $\mathbb{A}_{p,q}$ is the Cartesian product of $\mathbb{A}_p$ and $\mathbb{A}_q$. We do the rounding as follows

$$\widehat{\mathbf{\Lambda}}^{(k)}_{p,q} = 2\Big\lfloor 0.5\tilde{\mathbf{\Lambda}}^{(k)}_{p,q}\Big\rceil. \tag{28}$$

In (28), the operation is done element-wise since $\tilde{\mathbf{\Lambda}}^{(k)}_{p,q}$ is a vector. Further, let $\widehat{\mathbf{\Lambda}}^{(k)}_{p,q} \triangleq [\widehat{\lambda}^{(k)}_p \widehat{\lambda}^{(k)}_q]^T$. It is possible that the solution $\widehat{\mathbf{\Lambda}}_{p,q}^{(k)}$ in (28) need not lie in $\mathbb{A}_{p,q}^{(k)}$. This would result in $d_p^{(k+1)} \notin \mathbb{A}_p$. For example, if $\mathbb{A}_p$ is $M$-PAM, then $d_p^{(k+1)} \notin \mathbb{A}_p$ if $d_p^{(k)} + \widehat{\lambda}_p^{(k)} > (M-1)$. In such cases, we propose the following adjustment to $\widehat{\lambda}_p^{(k)}$:

$$\widehat{\lambda}_p^{(k)} = \begin{cases} (M-1) - d_p^{(k)}, & \text{when } \widehat{\lambda}_p^{(k)} + d_p^{(k)} > (M-1) \\ -(M-1) - d_p^{(k)}, & \text{when } \widehat{\lambda}_p^{(k)} + d_p^{(k)} < -(M-1). \end{cases} \quad (29)$$

Similar adjustment is done for $\widehat{\lambda}_q^{(k)}$ also. After these adjustments, we are guaranteed that $\widehat{\Lambda}_{p,q}^{(k)} \in \mathbb{A}_{p,q}^{(k)}$. We can therefore evaluate the cost difference function value as $\Delta C_{p,q}^{k+1}(\widehat{\lambda}_p^{(k)}, \widehat{\lambda}_q^{(k)})$. It is noted that the complexity of this approximate method does not depend on the size of the set $\mathbb{A}_{p,q}^{(k)}$, i.e., it has constant complexity. Through simulations, we have observed that this approximation results in a performance close to that of the brute-force method.

We define the optimum pairs, $(r, s)$ from the brute-force method and $(\hat{r}, \hat{s})$ from the approximate method, respectively, as

$$(r, s) \;=\; \underset{(p,q)}{\text{arg min}} \; \Delta C_{p,q,opt}^{k+1}, \quad (30)$$

and

$$(\hat{r}, \hat{s}) \;=\; \underset{(p,q)}{\text{arg min}} \; \Delta C_{p,q}^{k+1}(\widehat{\lambda}_p^{(k)}, \widehat{\lambda}_q^{(k)}). \quad (31)$$

The corresponding minimum values of the cost difference functions are given by

$$\Delta C_{opt}^{k+1} \;\stackrel{\triangle}{=}\; \Delta C_{r,s,opt}^{k+1}, \quad (32)$$

and

$$\Delta \widehat{C}_{opt}^{k+1} \;\stackrel{\triangle}{=}\; \Delta C_{\hat{r},\hat{s}}^{k+1}(\widehat{\lambda}_{\hat{r}}^{(k)}, \widehat{\lambda}_{\hat{s}}^{(k)}). \quad (33)$$

The update rule for the $\mathbf{z}^{(k)}$ vector is given by

$$\mathbf{z}^{(k+1)} \;=\; \mathbf{z}^{(k)} - (\lambda_{r,opt}^{(k)} \mathbf{g}_r + \lambda_{s,opt}^{(k)} \mathbf{g}_s) \quad (34)$$

$$\mathbf{d}^{(k+1)} \;=\; \mathbf{d}^{(k)} + \lambda_{r,opt}^{(k)} \mathbf{e}_r + \lambda_{s,opt}^{(k)} \mathbf{e}_s \quad (35)$$

for the brute-force method, and

$$\mathbf{z}^{(k+1)} \;=\; \mathbf{z}^{(k)} - (\widehat{\lambda}_{\hat{r}}^{(k)} \mathbf{g}_{\hat{r}} + \widehat{\lambda}_{\hat{s}}^{(k)} \mathbf{g}_{\hat{s}}) \quad (36)$$

$$\mathbf{d}^{(k+1)} \;=\; \mathbf{d}^{(k)} + (\widehat{\lambda}_{\hat{r}}^{(k)} \mathbf{e}_{\hat{r}} + \widehat{\lambda}_{\hat{s}}^{(k)} \mathbf{e}_{\hat{s}}) \quad (37)$$

for the approximate method. A similar procedure can be devised for the 3-symbol update also. The complexity of the M-LAS algorithm can be shown to be $O(N_t N_r)$ per symbol (we do not present the details of the 3-symbol update and the complexity analysis here due to page limit).

## IV. BER Performance of the M-LAS Detector

### A. Uncoded M-LAS Performance

*Performance as a function of increasing $N_t{=}N_r$:* In Fig. 1, we present the uncoded BER performance of the proposed M-LAS detector for different values of $N_t = N_r$ and 4-QAM obtained through simulations. MMSE filter is used as the initial filter. We label the M-LAS detector with MMSE initial filter as 'MMSE-MLAS' in all the figures. MMSE filter (without M-LAS) performance as well as AWGN-only SISO performance are also plotted for comparison. In generating the plots in Figs. 1 and 2, perfect channel knowledge is assumed at the receiver. From Fig. 1, it can be observed that the performance of the proposed MMSE-MLAS improves with increasing $N_t = N_r$, such that for $N_t = N_r = 64$ it achieves an uncoded BER of $10^{-3}$ at just 1 dB away from the AWGN-only SISO performance. With $N_t = N_r = 128$ and 256, the MMSE-MLAS performance moves even closer to the AWGN performance (to within 0.5 dB). This is an impressive result which illustrates the ability of the proposed MMSE-MLAS to achieve single-antenna AWGN performance even in a large multi-antenna setting, essentially removing 'almost' all the spatial interference from other antennas.

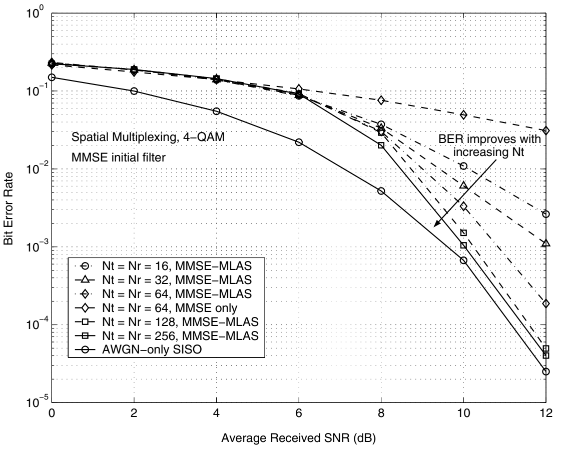

Fig. 1. Uncoded BER performance of the proposed M-LAS detector for different values of $N_t = N_r$. MMSE initial filter. 4-QAM. BER improves with increasing $N_t = N_r$.

*M-LAS versus LAS:* We further point out that the LAS detector we presented in [4] also achieves near-AWGN performance, but only when the number of antennas are very large (of the order of several hundreds). Whereas, a key advantage of the present M-LAS detector is that it is able to achieve near-AWGN performance even with tens of antennas (e.g., $N_t = N_r = 64$). This observation is illustrated in Fig. 2, where we compare the performance of the MMSE-MLAS with that of the MMSE-LAS in [4] (i.e., LAS with MMSE initial filter), for $N_t = N_r = 64, 32$ and 4-QAM. It can be seen that MMSE-MLAS outperforms MMSE-LAS. This performance improvement is due to the 2- and 3-symbol updates performed in M-LAS, in addition to the 1-symbol updates performed in LAS. As pointed out earlier, the 2- and 3-symbol updates in M-LAS increase the complexity a little, but the average per-symbol complexity (defined as total complexity divided by the total number of symbols, $N_t$) still remains as $O(N_t N_r)$. The performance advantage of M-LAS over LAS in the regime of tens of antennas has interesting practical implications, as tens of antennas can be easily placed in moderately sized communication terminals (e.g., laptops) enabling large MIMO systems to be viable in practice.

### B. Turbo Coded M-LAS Performance

We evaluated the coded BER performance of the M-LAS detector without and with channel estimation errors. In [4], hard decision data output from the LAS detector (i.e., ±1 valued data output vector) was fed to the turbo decoder. However, performance can be improved if soft output values can be generated and used instead. Consequently, here we propose a method for generating soft output from M-LAS.

*Soft Bit Values Generation:* We generate soft values at the M-LAS output for all the individual bits that constitute the modulation symbols ($M$-PAM/$M$-QAM) mounted on all the transmit antennas as follows. These soft output values are fed as inputs to the turbo decoder. Let $\mathbf{d} = [\widehat{x}_1, \widehat{x}_2, \cdots, \widehat{x}_{2N_t}]$, $\widehat{x}_i \in \mathbb{A}_i$ denote the detected output vector from the M-LAS algorithm. Let $\widehat{x}_i$ map to the bit vector $\mathbf{b}_i = [b_{i,1}, b_{i,2}, \cdots, b_{i,K_i}]^T$, where $K_i = \log_2 |\mathbb{A}_i|$, and $b_{i,j} \in \{+1, -1\}$, $i = 1, 2, \cdots, 2N_t$ and $j = 1, 2, \cdots, K_i$. Let $\tilde{b}_{i,j} \in \mathbb{R}$ denote the soft value for the $j$th bit of the $i$th symbol. Given $\mathbf{d}$, we need to find $\tilde{b}_{i,j}, \forall\,(i,j)$.

Now, define vectors $\mathbf{b}_i^{j+}$ and $\mathbf{b}_i^{j-}$ to be the $\mathbf{b}_i$ vector with its $j$th entry forced to +1 and -1, respectively. Let $\mathbf{b}_i^{j+}$ and $\mathbf{b}_i^{j-}$ demap to $x_i^{j+}$ and $x_i^{j-}$, respectively, where $x_i^{j+}, x_i^{j-} \in \mathbb{A}_i$. Also, define vectors $\mathbf{d}_i^{j+}$ and $\mathbf{d}_i^{j-}$ to be the $\mathbf{d}$ vector with its $i$th entry forced to $x_i^{j+}$ and $x_i^{j-}$, respectively. Using the above definitions, we obtain the soft output value for the $j$th bit of the $i$th symbol as

$$\tilde{b}_{i,j} \;=\; \frac{\|\mathbf{y} - \mathbf{H}\mathbf{d}_i^{j-}\|^2 - \|\mathbf{y} - \mathbf{H}\mathbf{d}_i^{j+}\|^2}{\|\mathbf{h}_i\|^2}. \tag{38}$$

The RHS of the above equation can be efficiently computed in terms of the known variables $\mathbf{z}$ and $\mathbf{G}$ as follows. Since $\mathbf{d}_i^{j+}$ and $\mathbf{d}_i^{j-}$ differ only in the $i$th entry, we can write

$$\mathbf{d}_i^{j-} \;=\; \mathbf{d}_i^{j+} + \lambda_{i,j}\mathbf{e}_i. \tag{39}$$

Since we know $\mathbf{d}_i^{j-}$ and $\mathbf{d}_i^{j+}$, we know $\lambda_{i,j}$ from (39). Substituting (39) in (38), we can write

$$\begin{aligned} \tilde{b}_{i,j}\,\|\mathbf{h}_i\|^2 &= \|\mathbf{y} - \mathbf{H}\mathbf{d}_i^{j+} - \lambda_{i,j}\mathbf{h}_i\|^2 - \|\mathbf{y} - \mathbf{H}\mathbf{d}_i^{j+}\|^2 \\ &= \lambda_{i,j}^2\|\mathbf{h}_i\|^2 - 2\lambda_{i,j}\mathbf{h}_i^T(\mathbf{y} - \mathbf{H}\mathbf{d}_i^{j+}) \qquad (40) \\ &= -\lambda_{i,j}^2\|\mathbf{h}_i\|^2 - 2\lambda_{i,j}\mathbf{h}_i^T(\mathbf{y} - \mathbf{H}\mathbf{d}_i^{j-}). \qquad (41) \end{aligned}$$

If $b_{i,j} = 1$, then $\mathbf{d}_i^{j+} = \mathbf{d}$ and substituting this in (40) and dividing by $\|\mathbf{h}_i\|^2$, we get

$$\tilde{b}_{i,j} \;=\; \lambda_{i,j}^2 - 2\lambda_{i,j}\frac{z_i}{(\mathbf{G})_{i,i}}. \tag{42}$$

On the other hand, if $b_{i,j} = -1$, then $\mathbf{d}_i^{j-} = \mathbf{d}$ and substituting this in (41) and dividing by $\|\mathbf{h}_i\|^2$, we get

$$\tilde{b}_{i,j} \;=\; -\lambda_{i,j}^2 - 2\lambda_{i,j}\frac{z_i}{(\mathbf{G})_{i,i}}. \tag{43}$$

It is noted that $\mathbf{z}$ and $\mathbf{G}$ are already available upon the termination of the M-LAS algorithm, and hence the complexity of computing $\tilde{b}_{i,j}$ in (42) and (43) is constant. Hence, the overall complexity in computing the soft values for all the bits is $O(N_t \log_2 M)$. We also see from (42) and (43) that the magnitude of $\tilde{b}_{i,j}$ depends upon $\lambda_{i,j}$. For large size signal sets, the possible values of $\lambda_{i,j}$ will also be large in magnitude. We therefore have to normalize $\tilde{b}_{i,j}$ for the turbo decoder to function properly. It has been observed through simulations that normalizing $\tilde{b}_{i,j}$ by $\left(\frac{\lambda_{i,j}}{2}\right)^2$ resulted in good performance.

*Coded BER Results:* Figure 3 shows the rate-3/4 turbo coded BER of the M-LAS detector for $N_t = N_r = 64, 128$, 4-QAM and MMSE initial vector. We have also shown the minimum SNR required to achieve theoretical capacity for a MIMO system with perfect CSI at the receiver, given by [1]

$$C \;=\; E\left[\log\det\left(\mathbf{I}_{N_r} + (\gamma/N_t)\,\mathbf{H}\mathbf{H}^H\right)\right], \tag{44}$$

where $\gamma$ is the average SNR per receive antenna. With soft decision inputs to the turbo decoder, the performance improves by about 1 dB compared to hard decision inputs. With perfect channel knowledge, the M-LAS performs close to within about 4.5 dB from theoretical capacity. With a Gaussian channel estimation error model, the performance loss incurred is only less than a dB for 2% estimation error variance. We note that we have also adopted the M-LAS algorithm to decode full-rate non-orthogonal STBCs from division algebras [7], achieving near-capacity performance [4].

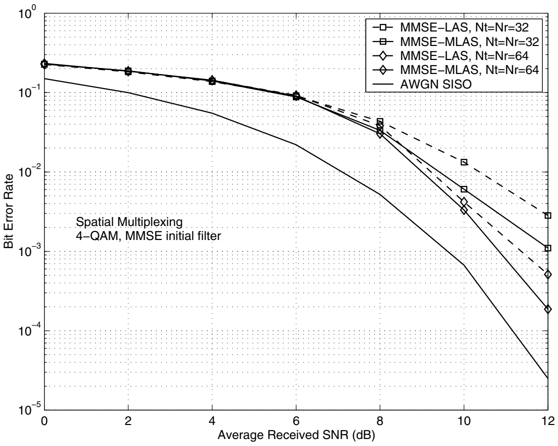

Fig. 2. Comparison of M-LAS and LAS performance in the tens of antennas regime. $N_t = N_r = 64, 32$. MMSE initial filter. 4-QAM. M-LAS outperforms LAS.

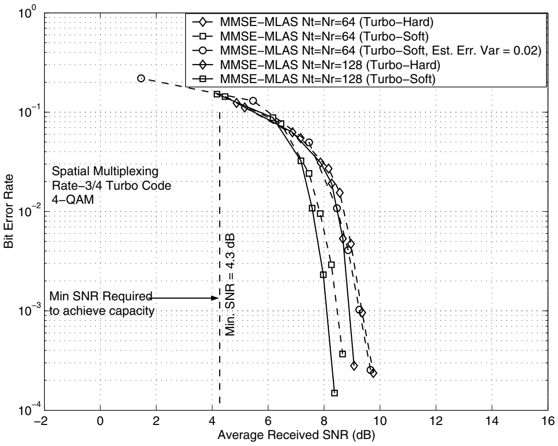

Fig. 3. Turbo coded BER performance of the proposed M-LAS detector for $N_t = N_r = 64$ and 128. MMSE initial filter. 4-QAM. Rate-3/4 turbo code. M-LAS detector performs to within about 4.5 dB from theoretical capacity.

## References

[1] H. Jafarkhani, *Space-Time Coding: Theory and Practice*, Cambridge University Press, 2005.

[2] X. Yang, Y. Xiong, F. Wang, "An adaptive MIMO system based on unified belief propagation detection," *Proc. IEEE ICC'2007*, June 2007.

[3] Y. Sun, "A family of linear complexity likelihood ascent search detectors for CDMA multiuser detection," *Proc. IEEE 6th Intl. Symp. on Spread Spectrum Tech. & App.*, September 2000.

[4] K. Vishnu Vardhan, S. K. Mohammed, A. Chockalingam, and B. Sundar Rajan, "A low-complexity detector for large MIMO systems and multicarrier CDMA systems," *IEEE JSAC Spl. Iss. on Multiuser Detection for Adv. Commun. Systems and Networks*, vol. 26, no.3, pp. 473-485, April 2008. Online arXiv:0804.0980v1 [cs.IT] 7 Apr 2008.

[5] B. Farhang-Boroujeny, H. Zhu, and Z. Shi, "Markov chain Monte Carlo algorithms for CDMA and MIMO communication systems," *IEEE Trans. on Sig. Proc.*, vol. 54, no. 5, pp. 1896-1908, May 2006.

[6] D. Pham, K. Pattipati, P. Willett, and J. Luo, "A generalized probabilistic data association detector for multiple antenna systems," *IEEE Commun. Lett.*, vol. 8, no. 4, pp. 205207, April 2004.

[7] B. A. Sethuraman, B. S. Rajan, and V. Shashidhar, "Full-diversity, high-rate space-time block codes from division algebras," *IEEE Trans. Inform. Theory*, vol. 49, no. 10, pp. 2596-2616, October 2003.