Improved Sparsity Behaviour and Error Localization in Detectors for Large MIMO Systems

Abhishek*, Abhay Kumar Sah† and A. K. Chaturvedi‡
Department of Electrical Engineering
Indian Institute of Technology Kanpur, Kanpur, India 208016
Email: {abhie*, abhaysah†, akc‡}@iitk.ac.in

***Abstract*—Sparsity based techniques have been found to be promising for detection in large/massive multiple-input multiple-output (MIMO) systems. They initialize with an initial solution vector, localize its erroneous symbols and then correct them. However, the process works only if the errors in the initial solution vector are sparse enough and are localized accurately. In this paper, we improve the localizing capability by proposing a modification in the residual update computed by the generalized orthogonal matching pursuit algorithm. Subsequently, we show that the sparsity behaviour can be improved by concatenating error vectors to create a larger vector, with no increase the complexity. Combining both these strategies, we propose to concatenate MMSE solution vectors (say K) as a single vector and then apply the improved error localization algorithm. It is shown that the proposed strategy results in a significant improvement in error performance without compromising the complexity.**

I. Introduction

Recent years have seen a surge in research interest in large multiple-input multiple-output (MIMO) (often called $Massive$ MIMO) systems [1], [2]. Equipped with a large number of antennas at the transmitter and/or receiver side, the large MIMO system upscales the capacity [3], [4] from that of conventional MIMO and is promising in terms of offering high data rates. But the advantages served by this system are also met with challenges in several design aspects [2], one of them being the design of a reliable and computationally efficient detector.

Several detection algorithms exist in the literature for conventional MIMO systems. Non-linear detectors such as the sphere decoder (SD) [5] and its variants achieve close to maximum-likelihood (ML) performance, but their complexity scales exponentially with system size. In contrast, linear detectors such as the zero forcing (ZF) detector and the minimum mean square error (MMSE) detector are appealing for their low complexity but their reliability worsens as the number of antennas or/and constellation size increase. This gave way to the development of heuristic algorithms such as the likelihood ascent search (LAS) [6]–[8] and reactive tabu search (RTS) [9], which exhibit improved performance with increasing number of antennas for low-order modulation schemes but their performance deteriorates as the modulation order increases.

Compressive sensing has recently emerged as a promising candidate for use in large MIMO detection [10]–[12]. Since the transmitted symbol vectors in MIMO systems do not exhibit sparsity, it is not appropriate to directly apply sparsity based techniques to MIMO detection problems. The idea is to remodel the MIMO detection problem as a sparse error signal detection problem which can then be solved through sparse error recovery techniques [13]–[15]. One of the earliest approaches towards the application of sparse error recovery technique in the large MIMO paradigm has been explored in [11], [12]. Therein, an error vector is estimated by localizing the errors in an initial solution vector and then correcting them by employing a linear detection technique. The performance of such algorithms depends on the accuracy of localization, which in turn depends on the sparsity in the error vector.

In this paper, we attempt to improve both, the localization accuracy as well as the sparsity behaviour of the initial solution vector. Because of the (usually) low symbol error rate (SER) of the MMSE detector, [11], [12] use the MMSE detector to obtain an initial estimate of the transmitted signal. But the error vector obtained by the MMSE detector is sparse only in an average sense. The instantaneous sparsity level will sometimes be high, thus adversely affecting the error correcting capability of the recovery algorithm.

Here, we propose an approach to improve the sparsity behaviour ratio of the initial solution vector. For this, instead of recovering the error corresponding to a single transmitted signal vector, we wait for, say, K vectors to arrive. The MMSE detector obtains an estimate for each vector, which are subsequently concatenated, thus generating the initial estimate for the sparsity boosted scheme. We argue that this will improve the sparsity behaviour. It may appear that this will lead to a rise in complexity. We avoid this by exploiting the structure of the larger equivalent system.

Now coming to localization accuracy, [12] uses the generalized orthogonal matching pursuit (gOMP) algorithm [13] in which a residual vector is updated iteratively. In the update process, it compares the effect of the intermediate estimated error vector with the received vector. However, instead of the received vector, if we use the intermediate residual vector for update, the localization accuracy can be improved. Combining both the above strategies, we propose a concatenation based improved error localization (CBIEL) detector. Simulation results show that the proposed detector provides significant gains over the existing sparsity boosted iterative linear detector (SBIL) [12] without any compromise in complexity.

The rest of the paper is organized as follows. In Section

II, we discuss the system model and the preliminaries. The design of the proposed detector is described in Section III. In Section IV, we present a brief discussion on the advantage of concatenation. In Section V, we analyze the complexity of the proposed detection scheme and its dependence on the block size. In Section VI, simulation results are presented. We conclude the paper in Section VII. The following notations are used in this paper. Boldface upper and lower case letters denote matrices and vectors respectively. $\mathbf{A}^T$, $\mathbf{A}^H$, and $\mathbf{A}^\dagger$ stand for the transpose, conjugate transpose, and the Moore-Penrose pseudoinverse of matrix $\mathbf{A}$. The set operators $A\backslash B$, $|A|$, A^c denote, respectively, the elements in set A but not in set B, the cardinality of set A, and the complement of set A.

II. Preliminaries

Consider a MIMO system with n_r receive antennas and n_t transmit antennas $(n_r \geq n_t)$. Mathematically,

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

where $\mathbf{y}$ is the $n_r \times 1$ receive vector, $\mathbf{x}$ is the $n_t \times 1$ transmit vector, $\mathbf{H}$ is the $n_r \times n_t$ channel matrix, and $\mathbf{n}$ is the $n_r \times 1$ noise vector. The elements of the channel matrix are distributed as $\mathcal{CN}(0,1)$ and those of the noise vector as $\mathcal{CN}(0,\sigma^2)$. The elements of $\mathbf{x}$ are drawn from a finite alphabet set $\mathcal{A}$ such that $\mathrm{E}\|\mathbf{x}\|^2 = 1$.

The aim of the detection problem is to find the transmit vector which minimizes the Euclidean cost, mathematically expressed as

$$\hat{\mathbf{x}} = \underset{\mathbf{x}\in\mathcal{A}^{N_t}}{\operatorname{argmin}} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 \tag{2}$$

and known as the ML solution. However, the complexity of ML detection increases exponentially with the increase in the number of transmit antennas. This makes it infeasible even for conventional MIMO systems. To address this issue, recently algorithms which exploit the inherent sparsity [10]–[12] have been developed. These algorithms use a linear detector, such as MMSE, to estimate an initial solution and then apply a matching pursuit algorithm, like [13]–[15], to locate the incorrectly detected symbols. Using the information about locations of incorrectly detected symbols, the effect of correctly detected symbols is subtracted from the original received vector. This gives rise to a new MIMO system with an input dimension much lesser than the output dimension, after which application of a linear detector viz. ZF or an MMSE detector suffices for achieving better performance.

The error performance of these sparsity boosted algorithms depends on the sparsity of the errors in the initial solution. Let us denote the initial solution as $\hat{\mathbf{x}}$ and the error vector as $\mathbf{x}_e = \mathbf{x} - \hat{\mathbf{x}}$, i.e., the difference of initial solution vector to the transmitted vector. The errors in $\mathbf{x}_e$ can be successfully recovered only if the following criterion [16] is satisfied

$$n_r > c\,|S| \log\left(\frac{n_t}{|S|}\right) \tag{3}$$

where c is a constant and S denotes the support set of $\mathbf{x}_e$. Thus an error vector, which might not be sufficiently sparse, will degrade the performance. To address this issue, in the next section we propose a scheme so that the probability of (3) being satisfied is increased, thereby leading to an improvement in error recovery performance.

III. Proposed Detector

We begin with the premise that if we process K consecutive instances of receive vectors, say, $\{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_K\}$ as a single concatenated receive vector, we expect a better sparsity behaviour. Motivated by this, we formulate a concatenated system model as

$$\mathbf{y}_c = \mathbf{H}_c\mathbf{x}_c + \mathbf{n}_c, \tag{4}$$

where $\mathbf{y}_c = [\mathbf{y}_1^T\,\mathbf{y}_2^T\,\cdots\,\mathbf{y}_K^T]^T$ is the $Kn_r \times 1$ equivalent receive vector, $\mathbf{x}_c = [\mathbf{x}_1^T\,\mathbf{x}_2^T\,\cdots\,\mathbf{x}_K^T]^T$ is the $Kn_t \times 1$ equivalent transmit vector, $\mathbf{n}_c = [\mathbf{n}_1^T\,\mathbf{n}_2^T\,\cdots\,\mathbf{n}_K^T]^T$ is the $Kn_r \times 1$ equivalent noise vector, and $\mathbf{H}_c = blk(\mathbf{H}_1, \mathbf{H}_2, \cdots, \mathbf{H}_K)$[1] is the $Kn_r \times Kn_t$ equivalent channel matrix.

We propose to apply a sparsity based approach, which we refer to as concatenation based improved error localization (CBIEL) algorithm, over the resulting equivalent system model (4). By concatenating as in (4), the block diagonal type structure of the equivalent system model allows us to carry out the recovery algorithm separately on each block. As a result, the overall complexity remains low. It is worthwhile to mention here that the complexity will not be low if we use a combining scheme in which the sparsity boosted algorithm would have to operate on a large channel matrix. The complete CBIEL algorithm is explained below.

A. Initialization

The algorithm is initialized with the MMSE solutions of each of the K receive vectors i.e. $\mathbf{y}_c$. We express it as a concatenation of MMSE solutions i.e., $\hat{\mathbf{x}}_c = [\hat{\mathbf{x}}_1^T\,\hat{\mathbf{x}}_2^T\,\cdots\,\hat{\mathbf{x}}_K^T]^T$ where $\hat{\mathbf{x}}_i$ represents the MMSE solution for $\mathbf{y}_i$ given by

$$\hat{\mathbf{x}}_i = (\mathbf{H}_i^H\mathbf{H}_i + \sigma^2\mathbf{I})^{-1}\mathbf{H}_i^H\mathbf{y}_i. \tag{5}$$

It is worth mentioning that it is not necessary to initialize with an MMSE solution and that any low complexity solution will suffice for this purpose. These initial estimates are used for obtaining K residual vectors $\mathbf{r}_i^0 = \mathbf{y}_i - \mathbf{H}_i\hat{\mathbf{x}}_i\,\forall\, i = 1$ to K.

B. Improved Error Localization

In this section, the goal is to localize the errors in the concatenated MMSE solution $\hat{\mathbf{x}}_c$. For this, we estimate the error vector $\mathbf{x}_{e,i}$ for each individual initial solution vector $\hat{\mathbf{x}}_i$. This can be done by an application of the gOMP algorithm [13],

[1]Consider matrices $\{\mathbf{A}_1, \mathbf{A}_2, \cdots, \mathbf{A}_n\}$ such that $\mathbf{A}_i \in \mathbb{C}^{r_i\times c_i}\,\forall\, i \in \{1,2,\cdots,n\}$. Then the matrix $\mathbf{A} = blk(\mathbf{A}_1, \mathbf{A}_2, \cdots, \mathbf{A}_n)$ denotes the matrix

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 & \mathbf{0}_{r_1\times c_2} & \mathbf{0}_{r_1\times c_3} & \cdots & \mathbf{0}_{r_1\times c_n} \\ \mathbf{0}_{r_2\times c_1} & \mathbf{A}_2 & \mathbf{0}_{r_2\times c_3} & \cdots & \mathbf{0}_{r_2\times c_n} \\ \mathbf{0}_{r_3\times c_1} & \mathbf{0}_{r_3\times c_2} & \mathbf{A}_3 & \cdots & \mathbf{0}_{r_3\times c_n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{r_n\times c_1} & \mathbf{0}_{r_n\times c_2} & \mathbf{0}_{r_n\times c_3} & \cdots & \mathbf{A}_n \end{bmatrix}.$$

[15], which uses the intermediate residual vectors to compute a heuristic $\mathbf{f}_i = \mathbf{H}_i^H \mathbf{r}_i \, \forall \, i$, which denotes the correlation between columns of $\mathbf{H}_i$ and the corresponding residual $\mathbf{r}_i$. We then form the vector $[|\mathbf{f}_1|^T \, |\mathbf{f}_2|^T \, \cdots \, |\mathbf{f}_K|^T]$ from which the indices corresponding to the L largest elements are selected. These indices identify the erroneous locations. It may be pointed out that in contrast to the earlier schemes, we propose to select these L locations over all $\mathbf{f}_i$s from 1 to K. We denote the selected indices as a set Λ. Using this set, we form the subsets Λ_is $\forall \, i$ (explained later in this Section), the elements of which are used for estimating the error vectors as

$$\mathbf{x}_{e,i} = \mathbf{H}_i^{\dagger} \left(\Lambda_i\right) \mathbf{r}_i^0. \tag{6}$$

These error vectors are used for updating the corresponding residual vectors and the set Λ. We propose a residual update process which is different from [12]. Thus, we use the current residual vector instead of the receive vector y, as

$$\mathbf{r}_i = \mathbf{r}_i^0 - \mathbf{H}_i \left(\Lambda_i\right) \mathbf{x}_{e,i}. \tag{7}$$

This update strategy uses the intermediate residual information for improving the localization accuracy of the algorithm. This whole process is repeated for a fixed number of iterations N_{itr}. At the end, we have better estimates of $\mathbf{x}_{e,i}$s, and their support sets S_i.

C. Error Recovery

Subtracting the effect of the correctly detected symbols i.e. the elements of $S_i{}^c = [1 : n_t] \setminus S_i$, we have

$$\mathbf{y}_{r,i} = \mathbf{y}_i - \mathbf{H}_i \left(S_i^c\right) \hat{\mathbf{x}}_i \left(S_i^c\right) = \mathbf{H}_i \left(S_i\right) \mathbf{x}_i \left(S_i\right) + \mathbf{n}_i \tag{8}$$

$\forall \, i$ such that $\Lambda_i \neq \emptyset$. Here $\mathbf{x}_i \left(S_i\right)$ denotes the vector comprising of incorrectly detected symbols from the initial MMSE detector and $\mathbf{H}_i \left(S\right)$ denotes a sub-matrix of $\mathbf{H}_i$ comprising of columns corresponding to those symbols. Since the support set for each i is, in general, expected to satisfy $|S_i| \ll n_r$, we obtain new instances of MIMO systems (where $S_i \neq \emptyset$) whose input dimension is much less compared to its output dimension. Thereafter, a solution is obtained using a ZF detector over this new MIMO system. This solution will be used to update the initial MMSE solution and the residual vector. The steps in Sections III-B and III-C are repeatedly done until the update on the residual vector reduces its norm i.e. the Euclidean cost. The complete procedure is provided in Algorithm 1. For $K = 1$ i.e. the case without any concatenation, we refer this CBIEL algorithm simply as IEL algorithm.

Mapping from Λ to Λ_i: In the algorithm, Λ_i is used to identify the erroneous symbols in an individual MMSE solution vector. For determining Λ_i the elements of Λ is mapped to its corresponding blocks using μ and ν. Here, μ and ν are arrays such that, $\forall \, i \in \{1, \cdots, |\Lambda^{t_2}|\}$, $\mu^{t_2}(i) = \lceil \Lambda^{t_2}(i)/n_t \rceil$ and $\nu^{t_2}(i) = (\Lambda^{t_2}(i) - 1) \, \mathbf{modulo} \, n_t + 1$, where $\lceil \cdot \rceil$ and **modulo** denote the ceiling function and the standard modulus operation respectively. The set Ω^{t_2} is formed by selecting all the distinct elements of the array μ^{t_2}. Also, all instances of the statement '$\forall \, i$' imply '$\forall \, i \in \{1, \cdots, K\}$', unless stated otherwise.

Algorithm 1: The proposed CBIEL algorithm

```
Input  : $\{\mathbf{y}_1, \cdots, \mathbf{y}_K\}, \{\hat{\mathbf{x}}_1, \cdots, \hat{\mathbf{x}}_K\}, \{\mathbf{H}_1, \cdots, \mathbf{H}_K\}, t_{max},$
         $\eta, L, N_{itr}$
Output: $\{\check{\mathbf{x}}_1, \cdots, \check{\mathbf{x}}_K\}$
Initialization: $\check{\mathbf{x}}_i = \mathbf{x}_i^0 = \hat{\mathbf{x}}_i \; \forall \, i;$
$i^{th}$ residual $\mathbf{r}_i^{0,0} = \mathbf{y}_i - \mathbf{H}_i \hat{\mathbf{x}}_i \; \forall \, i; \; \mathbf{r}_i^{-1,0} = \infty \; \forall \, i;$
$t_1 = 0, \; t_2 = 1;$
while $\sum_{i=1}^K \|\boldsymbol{r}_i^{t_1,0}\|_2^2 < \sum_{i=1}^K \|\boldsymbol{r}_i^{t_1-1,0}\|_2^2$ & $t_1 < t_{max}$ do
    $t_1 = t_1 + 1;$
    Locations of incorrect symbols $\Lambda^0 \leftarrow \emptyset;$
    for $t_2 \leftarrow 1$ to $N_{itr}$ do
        $\mathbf{f}_i = \mathbf{H}_i^H \mathbf{r}_i^{t_1-1,t_2-1} \; \forall \, i;$
        $Ind \leftarrow$ select indices corresponding to $L$ largest
        elements in $[|\mathbf{f}_1|^T \, |\mathbf{f}_2|^T \, \cdots \, |\mathbf{f}_K|^T];$
        $\Lambda^{t_2} \leftarrow \Lambda^{t_2-1} \cup Ind;$
        $\mu^{t_2}, \nu^{t_2} \leftarrow$ generate arrays from $\Lambda^{t_2};$
        $\Omega^{t_2} \leftarrow$ generate set from $\mu;$
        $\Lambda_i^{t_2} = \{\nu(j) \mid \mu(j) = i\} \; \forall \, i \in \Omega^{t_2};$
        $\mathbf{x}_{e,i} = \mathbf{H}_i^{\dagger}(\Lambda_i^{t_2}) \, \mathbf{r}_i^{t_1-1,0} \; \forall \, i \in \Omega^{t_2};$
        $\mathbf{r}_i^{t_1-1,t_2} = \mathbf{r}_i^{t_1-1,0} - \mathbf{H}(\Lambda_i^{t_2}) \, \mathbf{x}_{e,i} \; \forall \, i \in \Omega^{t_2};$
    end
    $S_i^{t_1} = \{\Lambda_i^{N_{itr}}(j) \mid |\mathbf{x}_{e,i}(j)| > \eta, j = 1, 2, ...., LN_{itr}\}$
    $\forall \, i \in \Omega^{N_{itr}};$
    $\mathbf{y}_{r,i} = \mathbf{y}_i - \mathbf{H}_i((S_i^{t_1})^c) \, \mathbf{x}_i^{t_1-1}((S_i^{t_1})^c) \; \forall \, i \in \Omega^{N_{itr}};$
    $\mathbf{x}_i^{t_1-1}((S_i^{t_1})^c) = \mathbf{H}_i^{\dagger}(S_i^{t_1}) \, \mathbf{y}_{r,i} \; \forall \, i \in \Omega^{N_{itr}};$
    $\mathbf{x}_i^{t_1} = \mathbf{x}_i^{t_1-1} \; \forall \, i;$
    $\mathbf{r}_i^{t_1,0} = \mathbf{y}_i - \mathbf{H}_i \mathbf{x}_i^{t_1} \; \forall \, i;$
end
return $\check{\mathbf{x}}_i \leftarrow \mathbf{x}_i^{t_1-1} \; \forall \, i.$
```

Discussion on L and N_{itr}: It may be noted that there are two important parameters in the CBIEL algorithm, namely the number of indices selected at each iteration and the maximum number of iterations which we denote by L and N_{itr} respectively. Our choice for the value of L and N_{itr} is based on the relation from [12], $LN_{itr} = \lfloor Kn_r / \log(Kn_t) \rfloor$, where $\lfloor \cdot \rfloor$ denotes the floor function. This relation ensures that the computational complexity of the inner loop in Algorithm 1 remains unchanged, as will be discussed in detail in Section V. Here, since we are processing on a larger vector, a small value of L as in SBIL [12] will not be able to locate all the erroneous indices. Thus from practical considerations we scale it by a factor of K and the value of N_{itr} will change accordingly without affecting the complexity of the inner loop.

IV. Effect of Concatenation on the Sparsity Behaviour

We know that a better initial solution will lead to a more sparse error vector $\mathbf{x}_e$, thus ensuring a better recovery. Let us examine the sparsity behaviour corresponding to the concatenated initial solution $\hat{\mathbf{x}}_c$ as compared to the individual initial solutions $\hat{\mathbf{x}}$. Assume that the sparsity ratio corresponding to an individual MMSE solution is distributed with mean μ

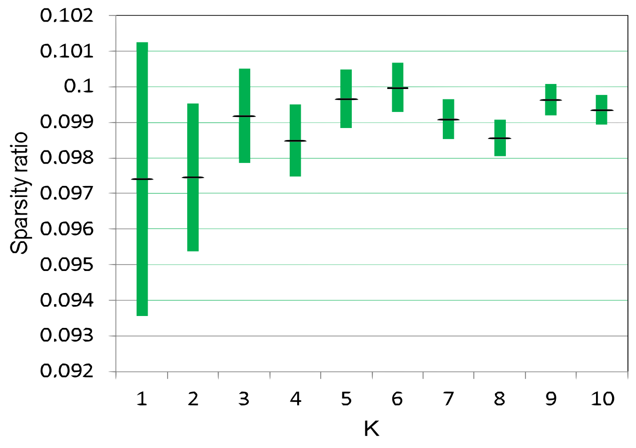

Fig. 1. Variation of sparsity ratio with the increasing value of K for a MMSE detector for a 32×32 system with 4-QAM at SNR $= 10$ dB. Here, the green bars denote the spread of sparsity ratio around the mean value (black mark at center).

and variance σ^2. Then the sparsity ratio for a block of K MMSE solutions will be distributed with the same mean μ and variance $\frac{\sigma^2}{K}$. We have verified this for a 32×32 system with 4-QAM modulation at $E_s/N_0 = 10$ dB and shown the results in Fig. 1. In the figure, the green bars denote the spread of sparsity ratio around its mean value i.e. the black line at the center of each green bar. One can observe that as the value of K increases, the variance in the sparsity ratio reduces. Therefore $\hat{\mathbf{x}}_c$ is more likely to satisfy (3) than a standalone $\hat{\mathbf{x}}$. As a result, an error recovery over $\hat{\mathbf{x}}_c$ is better ensured than that over $\hat{\mathbf{x}}$.

One natural question that arises is whether there is an alternative way to achieve the above. Consider, for instance, that we spread the errors over a block of K vectors (such as by interleaving) but process the individual vectors alone. Then the extension of (1) would be intricate. The gOMP would have to operate on a system of a much larger size even for an individual vector, and this will lead to a rise in complexity.

V. Complexity Analysis of CBIEL

In this section we analyze the complexity of the proposed algorithm by examining the complexity of the individual steps one by one.

1) Identification: In the t_2^{th} iteration, the gOMP performs $(2n_r - 1)n_t$ flops (n_r multiplications followed by $n_r - 1$ additions for each of the n_t rows) for the matrix-vector multiplication $\mathbf{H}_i^H \mathbf{r}_i$, and the modulus operation on the resultant vector requires $4n_t$ flops. These operations are performed for K instances. Sorting to find L largest indices would further require $O(Kn_t \log(L))$ complexity. Taking into account N_{itr} iterations gives an overall complexity $O(K^2 n_r^2 n_t / L log(Kn_t))$. If we account that L has been scaled by K, the resulting overall complexity will be $O(Kn_r^2 n_t / log(Kn_t))$.

2) Estimation of $\boldsymbol{x}_e$: This step requires computation of the LS solution $\mathbf{x}_{e,i}\ \forall i \in \Omega$. Using the QR decomposition of $\mathbf{H}_i$ ($\mathbf{H}_i = \mathbf{Q}_i \mathbf{R}_i$), we have

$$\mathbf{x}_{e,i} = (\mathbf{R}_i^H \mathbf{R}_i)^{-1} \mathbf{R}_i^H \mathbf{Q}_i^H \mathbf{r}_i \tag{9}$$

In the t_2^{th} iteration, the newly selected L indices would result either/both in update (addition of new columns) of $\mathbf{H}_i(\Lambda_i^{t_2-1})$s for $i \in \Omega^{t_2-1}$ or/and in creation of $\mathbf{H}_i(\Lambda_i^{t_2-1})$s corresponding to $i \in \Omega^{t_2} \backslash \Omega^{t_2-1}$. An efficient method of computing (9) for the updated matrices is by employing the modified Gram-Schmidt(MGS) algorithm which makes use of the QR decomposition of the $(t_2-1)^{th}$ iteration. This requires a complexity $O(n_r N_i l_i)$ where N_i and l_i denote, respectively, the number columns in $\mathbf{H}_i(\Lambda_i^{t_2-1})$ and the new columns being added to it in t_2^{th} iteration. The complexity of computing (9) for $\mathbf{H}_i(\Lambda_i^{t_2})$ corresponding to $i \in \Omega^{t_2} \backslash \Omega^{t_2-1}$ is of the order $O(n_r l_i^2)$.

To analyze the worst case complexity of this step, two cases depending upon the relative values of N_{itr} and K are considered. We assume that the LN_{itr} indices selected across the N_{itr} iterations all correspond to different columns in the matrix $[\mathbf{H}_1 \cdots \mathbf{H}_K]$ so that $LN_{itr} \leq Kn_t$. We first consider the case $N_{itr} \leq K$. If $L > n_t$, the i^{th} iteration selects columns with indices ranging from $(i-1)L+1$ to iL in the matrix $[\mathbf{H}_1 \cdots \mathbf{H}_K]$. This gives a complexity $O(n_r^2 n_t / log(Kn_t))$. When $L \leq n_t$, the i^{th} iteration selects L columns from $\mathbf{H}_i$, giving a total complexity of order $O(K^2 n_r^2 / log(Kn_t))$. The second case occurs when $N_{itr} > K$, which implies that $L < n_t$. We further make subcases based upon the relative values of N_{itr} and $K\lfloor n_t/L \rfloor$. One must keep in mind that there is a definite expression relating the values of L and N_{itr}, and that these subcases are only for the purpose of understanding. If $N_{itr} \leq K\lfloor n_t/L \rfloor$, the i^{th} iteration of the R^{th} run involves selecting columns with indices from $(R-1)L+1$ to RL in $\mathbf{H}_i$, where a run is comprised of K iterations with the i^{th} iteration selecting L columns from $\mathbf{H}_i$. There will be a total of $\lfloor N_i tr/K \rfloor$ such runs. The manner in which columns are selected remains the same for the rest of the $N_{itr} - K\lfloor N_i tr/K \rfloor$ iterations. For the other subcase, consider $N_{itr} > K\lfloor n_t/L \rfloor$. Here the $\lfloor N_i tr/K \rfloor$ runs proceed in a manner similar to the earlier subcase. The i^{th} iteration from amongst the last $N_{itr} - K\lfloor N_i tr/K \rfloor$ iterations selects columns with indices ranging from $(i-1)L+1$ to iL in the matrix formed by concatenating columns remaining in K matrices. Overall, the complexity of this step is of the order $O(Kn_r^3/(log(Kn_t))^2)$.

3) Residual update: The t_2^{th} iteration for this step involves the matrix-vector multiplication $\mathbf{H}_i(\Lambda_i^{t_2})\,\mathbf{x}_{e,i}$, requiring $(2l_i - 1)n_r$ flops, and followed by n_r subtractions, where l_i denotes the number of columns selected from $\mathbf{H}_i(\Lambda_i^{t_2})$. This step is carried out $\forall i \in \Omega$. Observing that $\sum_{i=1}^{|\Omega|} l_i = t_2 L$ and summing over N_{itr} iterations, we get an overall complexity $O(Kn_r^3/(log(Kn_t))^2)$.

4) Shrinkage: This step involves shrinking the set comprising the incorrectly detected symbols through a thresholding function and requires $O(1)$ complexity.

5) ZF demodulation: Demodulation via ZF detector requires computing $\mathbf{H}_i^{\dagger}(S_i)\,\mathbf{y}_{r,i}\ \forall i \in \Omega$. The complexity of this step will be of the order $O(n_r l_i^2)$, where l_i is the number of columns in $\mathbf{H}_i^\dagger(S_i)$. The worst-case complexity results when the LN_{itr} columns are distributed such that $l_i = n_t$ for $\lfloor LN_{itr}/n_t \rfloor$ number of $i \in \Omega$ and $LN_{itr} - n_t\lfloor LN_{itr}/n_t \rfloor$ columns are selected for the one remaining $i \in \Omega$. This leads to a total complexity of the order $O(Kn_r^2 n_t/log(Kn_t))$.

6) Residual initialization: Updating residuals require a total of $2Kn_r n_t$ flops and subsequent l_2-norm calculations involve $3Kn_r$ flops, resulting in a complexity $O(Kn_r n_t)$.

Therefore the overall complexity of our detector for a system comprising of K blocks is of order $O(Kn_r^2 n_t/log(Kn_t))$, leading to a complexity of the order of $O(n_r^2 n_t/log(Kn_t))$ per transmit vector. It is of roughly the same order as that of the SBIL [12], which is of the order of $O(n_r^3/(\log n_t)^2)$. The dominating term in both the expressions are of the same order i.e. n_r^3 $(n_r = n_t)$. In general, complexity depends on the system parameters. It may be noted that the above expressions of complexity do not consider the complexity of initialization which is of the order of $O(n_t^2 n_r)$ for MMSE based initialization. Including this term as well, we have the same order of overall complexity for both the algorithms.

VI. Simulation Results

In this section, we examine the performance of the CBIEL and IEL (i.e. the CBIEL without concatenation) detectors and compare them with SBIL [12]. Before comparing, we would like to investigate the effect of K on the error performance of CBIEL. We have already illustrated through Fig. 1 that the variance of the sparsity ratio decreases with increase in the value of K. The question arises whether this decrease will translate into any improvement in error performance. Thus, simulations were performed for 32×32 and 64×64 MIMO systems with 4-QAM modulation at $E_b/N_0 = 10$ dB considering the simulation parameters L, η and t_{max} as $2K$, 0.5 and 10 respectively. It may be noted that all these simulation parameters are same as in [12], except the value of L, which is upscaled by a factor of K. The reason for doing this has been explained in Section III. We have shown the results in Fig. 2. From the figure, it is clear that the error reduces with increasing value of K, but only up to a certain value of K, i.e., $K = 10$, after which there is no significant reduction. This value of K is in accordance with the result in Fig. 1. Hence for simulation purpose, we choose $K = 10$ for CBIEL.

Next, we examine the error performance of CBIEL and IEL detectors for two different systems i.e. a 32×32 system with 4-QAM and a 128×128 system with 16-QAM. For the simulation purpose we have taken the same simulation parameters as above i.e. i.e. $K = 10$, $L = 2K$, $\eta = 0.5$ and $t_{max} = 10$. The simulation results have been shown in Fig. 3. From the figure, it can be observed that both the proposed detectors outperform the SBIL detector. For example, for 32×32 system with 4-QAM at a bit error rate (BER) of 10^{-4}, the IEL and CBIEL $(K = 10)$ outperforms SBIL by 2 dB and 3 dB, respectively. This gain improves to 5 dB and 7

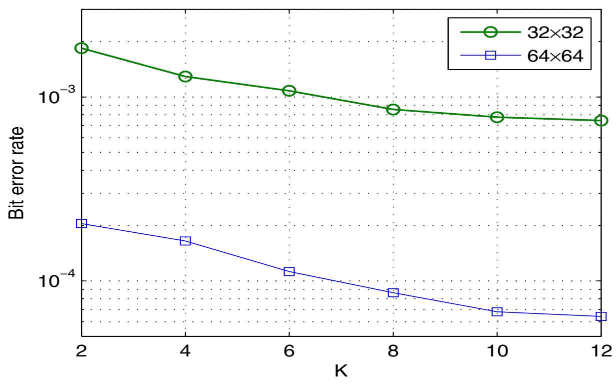

Fig. 2. BER performance of the proposed detector for 4-QAM modulation at $E_b/N_0 = 10$ dB with increasing block size i.e. K.

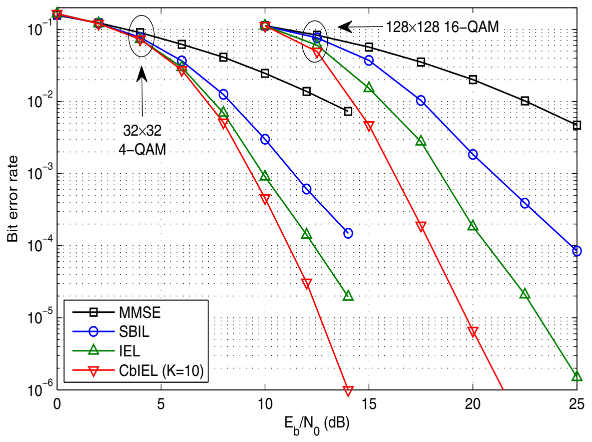

Fig. 3. BER comparison for a) a 32×32 system with 4-QAM modulation and b) a 128×128 system with 16-QAM modulation.

dB, respectively, for the 128×128 system with 16-QAM at the same BER of 10^{-4}, as illustrated in Fig. 3.

Next, we evaluate the error performance of IEL and CBIEL with the increasing number of antenna pairs for 16-QAM at a $E_b/N_0 = 18$ dB. The simulation results have been shown in Fig. 4 considering the same simulation parameters. From the figure, it can be observed that with the increase in the number of antennas, the error performance improves. One can also observe that to achieve the same BER, IEL and CBIEL $(K = 10)$ both require fewer antennas compared to SBIL. For example, IEL and CBIEL $(K = 10)$ can achieve a BER of 3×10^{-3} using only 64 and 44 antenna pairs while SBIL will require 128.

Furthermore, we compare the computations per bit for IEL, CBIEL, and SBIL detectors in terms of arithmetic operations i.e. multiplications and additions. The results for 4-QAM modulation are shown in the upper half of Table I while the numbers for 16-QAM modulation are provided in the lower half. Interestingly, it can be observed from the table that CBIEL requires the least number of arithmetic operations per bit among the three. Specifically for a 64×64 system with

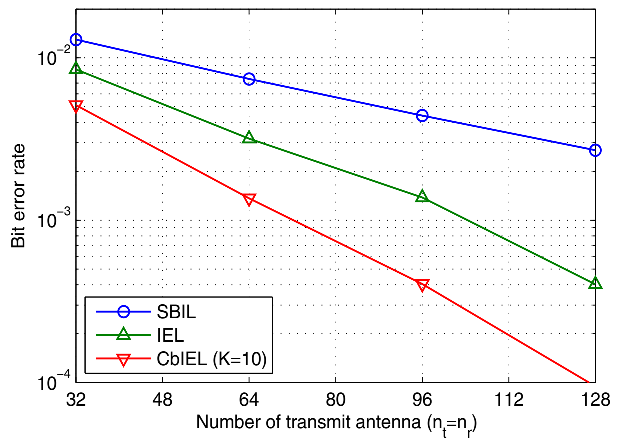

Fig. 4. BER comparison with increasing number of antennas for 16-QAM modulation at $E_b/N_0 = 18$ dB.

TABLE I
Comparison based on Arithmetic Operations ($\times 10^3$) per bit

Detection Algorithm	4-QAM Modulation at $E_b/N_0 = 10$ dB			
	32×32		64×64	
	Additions	Multiplications	Additions	Multiplications
SBIL	0.930	0.948	2.823	2.853
IEL	0.987	1.012	3.315	3.367
CBIEL ($K = 10$)	0.883	0.901	2.049	2.071
16-QAM Modulation at $E_b/N_0 = 15$ dB				
	64×64		128×128	
SBIL	2.936	2.966	13.494	13.557
IEL	2.279	2.316	7.269	7.337
CBIEL ($K = 10$)	3.055	3.098	9.972	10.062

4-QAM at $E_b/N_0 = 10$ dB, CBIEL, IEL, and SBIL require 2049, 3315, and 2823 number of additions, respectively while the number of multiplications are 2071, 3367 and 2853, respectively. Comparing the same for 16-QAM, we observe a slight increase in the number of arithmetic operations for CBIEL (slightly more than SBIL), while the order of computations remains the same. With the increase in antenna size i.e. for a 128×128 system with 16-QAM, CBIEL, and IEL require considerably lower number of arithmetic operations than the SBIL. Although the number of computations for CBIEL is on the lower side, one can observe that all the three algorithms require approximately the same order of computations, which is in accordance with the Section V i.e. the same order of complexity.

VII. Conclusion

We propose improved error localization and concatenation for sparsity based detection in large MIMO systems. The proposed algorithm provides two-fold gain. One is due to better localization, and the other is because of the improved sparsity behaviour of the initial solution. In addition to providing better sparsity, concatenation also equips us with the advantage of independent applications of the improved error localization method over separate blocks thereby preventing any increase in complexity. The proposed CBIEL detector has been shown to provide a significantly better error performance than the existing SBIL detector. The applicability of the scheme can be explored for other matching pursuit algorithms as well.

References

[1] M. Wu, B. Yin, G. Wang, C. Dick, J. Cavallaro, and C. Studer, "Large-scale MIMO detection for 3GPP LTE: Algorithms and FPGA implementations," *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 5, pp. 916–929, Oct. 2014.

[2] L. Lu, G. Li, A. Swindlehurst, A. Ashikhmin, and R. Zhang, "An overview of massive MIMO: Benefits and challenges," *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 5, pp. 742–758, Oct. 2014.

[3] F. Rusek *et al.*, "Scaling up MIMO: Opportunities and challenges with very large arrays," *IEEE Signal Process. Mag.*, vol. 30, no. 1, pp. 40–60, Jan. 2013.

[4] W. Yang, G. Durisi, and E. Riegler, "On the capacity of large-MIMO block-fading channels," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 2, pp. 117–132, Feb. 2013.

[5] B. Hassibi and H. Vikalo, "On the sphere-decoding algorithm I. Expected complexity," *IEEE Tran. Signal Process.*, vol. 53, no. 8, pp. 2806–2818, Aug. 2005.

[6] S. Mohammed, A. Chockalingam, and B. Sundar Rajan, "A low-complexity near-ML performance achieving algorithm for large MIMO detection," in *Proc. 2008 IEEE Int. Symp. Inf. Theory, (ISIT)*, pp. 2012–2016.

[7] P. Li and R. Murch, "Multiple output selection-LAS algorithm in large MIMO systems," *IEEE Commun. Lett.*, vol. 14, no. 5, pp. 399–401, May 2010.

[8] A. K. Sah and A. K. Chaturvedi, "Reduced neighborhood search algorithms for low complexity detection in MIMO systems," in *Proc. 2015 IEEE Global Commun. Conf. (GLOBECOM)*, pp. 1–6.

[9] N. Srinidhi, T. Datta, A. Chockalingam, and B. Rajan, "Layered tabu search algorithm for large-MIMO detection and a lower bound on ML performance," *IEEE Tran. Commun.*, vol. 59, no. 11, pp. 2955–2963, Nov. 2011.

[10] Y. Fadlallah, A. Aissa-El-Bey, K. Amis, D. Pastor, and R. Pyndiah, "New iterative detector of MIMO transmission using sparse decomposition," *IEEE Tran Veh. Technol.*, vol. 64, no. 8, pp. 3458–3464, Aug. 2015.

[11] J. W. Choi and B. Shim, "New approach for massive MIMO detection using sparse error recovery," in *Proc. 2014 IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2014, pp. 3754–3759.

[12] X. Peng, W. Wu, J. Sun, and Y. Liu, "Sparsity-boosted detection for large MIMO systems," *IEEE Commun. Lett.*, vol. 19, no. 2, pp. 191–194, Feb. 2015.

[13] J. Wang, S. Kwon, P. Li, and B. Shim, "Recovery of sparse signals via generalized orthogonal matching pursuit: A new analysis," *IEEE Tran. Signal Process.*, vol. 64, no. 4, pp. 1076–1089, Feb. 2016.

[14] S. Kwon, J. Wang, and B. Shim, "Multipath matching pursuit," *IEEE Tran. Inf. Theory*, vol. 60, no. 5, pp. 2986–3001, May 2014.

[15] J. Wang, S. Kwon, and B. Shim, "Generalized orthogonal matching pursuit," *IEEE Tran. Signal Process.*, vol. 60, no. 12, pp. 6202–6216, Dec. 2012.

[16] E. J. Candes and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 21–30, Mar. 2008.