# A Maximum-likelihood Decoder with a New Reduction Strategy for MIMO Channel Systems

Xiao-Wen Chang
School of Computer Science
McGill University
Montreal, Quebec, Canada H3A 2A7

Xiaohua Yang
School of Computer Science
McGill University
Montreal, Quebec, Canada H3A 2A7

***Abstract*—An efficient maximum-likelihood decoder with a new reduction strategy is proposed for linear MIMO channel systems. Unlike the current reduction strategies which only reorder the columns of the channel matrix, the new reduction algorithm employs the so called integer Gauss transformations to reduce the off-diagonal entries of the upper triangular factor of the QR decomposition of the channel matrix. Simulation results show that this new decoding algorithm can be much more efficient than existing algorithms.**

## I. Introduction

For Gaussian multi-input multi-output (MIMO) antenna systems, decoding algorithms (see, e.g., [2], [3], [4], [6], [7], and [14]) have been developed to solve a box-constrained integer least squares (ILS) problem, obtaining the maximum-likelihood (ML) detection. A typical ML decoding algorithm consists of two stages: reduction process (or preprocessing) and search process. To make the search process easier and more efficient, the reduction process is needed to transform the channel matrix to an upper triangular matrix. For solving an unconstrained ILS problem, the often used reduction process is the well-known LLL reduction [9]. The LLL reduction can be described as a QR decomposition of the transformed channel matrix by an unimodular matrix from the right. But for a box-constrained ILS problem, the LLL reduction will make the box-constraint too complicated to be easily handled by a search process (see, e.g., [7]). So the LLL reduction is usually not used for solving a box-constrained ILS problem. In the literature a typical reduction process for solving a box-constrained ILS problem computes the QR decomposition of the column-reordered channel matrix. Column-reordering may have a significant effect on the speed of the search process. In [7], three column-reordering strategies including the well-known V-BLAST strategy were introduced for the reduction purpose. A column-reordering strategy which was proposed in [16] for finding a suboptimal solution of a box-constrained ILS problem can be used for the reduction purpose, see [4]. All these reduction strategies use only the information of the channel matrix. Recently, a new column-reordering strategy was proposed in [4], which uses all information of the box-constrained ILS problem, i.e., the channel matrix, the constraint and the received signal vector. This new reduction can be much more effective than the existing ones in reducing the computational cost of the search process, especially when the noise in the received signal vector is large. For the search process of a box-constrained ILS problem, two search algorithms based on the Phost enumeration strategy (see [11] and [14] ) and Schnorr-Euchner enumeration strategy (see [12]) were proposed in [7]. As in [1] for solving an unconstrained ILS problem, it was found in [7] that the Schnorr-Euchner strategy is usually more efficient than the Phost strategy for solving a box-constrained ILS problem. In [2], the Schnorr-Euchner based search process given in [1] for an unconstrained ILS problem was directly extended to a box-constrained ILS problem. Later in [4] an improved Schnorr-Euchner based search algorithm was proposed for a box-constrained ILS problem.

Since ML detection may become time-prohibitive for small signal-to-noise ratio (SNR) or the problem with large dimension, recently some algorithms have been proposed to obtain near-ML detection, see, e.g., [8], [10] and [15]. But such algorithms will cause performance loss and will not be discussed further in this paper.

To solve the box-constrained ILS problem for ML detection, in this paper we propose a new reduction algorithm. Unlike the usual reduction algorithms which only reorder the columns of the channel matrix, the new reduction algorithm also uses the so-called integer Gauss transformations (IGTs) which are unimodular matrices in the reduction process. Notice that IGTs are used in the LLL reduction to reduce the off-diagonal elements of the upper triangular matrix, (see, e.g., [1] and [5]). The key to the success of our new reduction algorithm is the upper triangular form of the product of the IGTs. Without this form, it would be difficult to handle the box constraint in the search process. We also modify the Schnorr-Euchner based search algorithm given in [4] for the search process.

In this paper, $\boldsymbol{e}_i$ denotes the $i$-th column of the identity matrix $\boldsymbol{I}$, $\boldsymbol{e} = [1, 1, \cdots, 1]^T$, and $\lfloor x \rceil$ denotes the nearest integer to $x$ and if there is a tire it denotes the one which has smaller magnitude. For matrix $\boldsymbol{A}$, $\boldsymbol{A}(i_1 : i_2, j_1 : j_2)$ denotes the submatrix formed by rows $i_1$ to $i_2$ and columns $j_1$ to $j_2$.

## II. Problem Formulation

In Gaussian MIMO linear flat-fading channel systems, the relation between the received signal vector and the transmit signal vector can be written as a complex linear system

$$\boldsymbol{y}_c = \boldsymbol{H}_c \boldsymbol{x}_c + \boldsymbol{v}_c \tag{1}$$

978-1-4244-2324-8/08/$25.00 © 2008 IEEE.

where $\boldsymbol{H}_c \in \mathbb{C}^{N_r \times N_t}$ represents the flat-fading channel matrix with $N_t$ transmitter antennas and $N_r$ receiver antennas and we assume $N_r \geq N_t$, and the elements of $\boldsymbol{H}_c$ are complex i.i.d Gaussian variables with (normalized) distribution $CN(0,1)$, $\boldsymbol{v}_c \in \mathbb{C}^{N_r}$ is the white Gaussian noise vector with distribution $CN(\boldsymbol{0}, 2\sigma^2\boldsymbol{I})$, and $\boldsymbol{x}_c \in \mathbb{C}^{N_t}$ is the unknown signal vector and its elements are odd numbers in the finite set

$$\mathcal{I}_c(q) = \{q_1 + q_2 j : q_1, q_2 = \pm 1, \pm 3, \cdots, \pm(2^q - 1)\}$$

where $j^2 = -1$. Note that $q = 1, 2, 3$ corresponds to QPSK (i.e., 4QAM), 16QAM, 64QAM constellations, respectively.

To avoid complex operations, we first transform (1) to the following real linear system

$$\tilde{\boldsymbol{y}} = \tilde{\boldsymbol{H}}\tilde{\boldsymbol{x}} + \tilde{\boldsymbol{v}}, \tag{2}$$

$$\tilde{\boldsymbol{y}} = \begin{bmatrix} \boldsymbol{y}_c^R \\ \boldsymbol{y}_c^I \end{bmatrix}, \tilde{\boldsymbol{H}} = \begin{bmatrix} \boldsymbol{H}_c^R & -\boldsymbol{H}_c^I \\ \boldsymbol{H}_c^I & \boldsymbol{H}_c^R \end{bmatrix}, \tilde{\boldsymbol{x}} = \begin{bmatrix} \boldsymbol{x}_c^R \\ \boldsymbol{x}_c^I \end{bmatrix}, \tilde{\boldsymbol{v}} = \begin{bmatrix} \boldsymbol{v}_c^R \\ \boldsymbol{v}_c^I \end{bmatrix},$$

where $\boldsymbol{A}_c^R$ and $\boldsymbol{A}_c^I$ denote the real part and image part of a complex matrix or vector $\boldsymbol{A}_c$, respectively. Obviously, $\tilde{\boldsymbol{H}} \in \mathbb{R}^{m \times n}$ with $m \triangleq 2N_r$, $n \triangleq 2N_t$ and $\tilde{h}_{ij} \sim N(0, 1/2)$, $\tilde{\boldsymbol{v}} \sim N(0, \sigma^2\boldsymbol{I}_m)$, and $\tilde{\boldsymbol{x}} \in \mathcal{I}(q)^n$ with

$$\mathcal{I}(q) \triangleq \{\pm 1, \pm 3, \cdots, \pm(2^q - 3), \pm(2^q - 1)\}. \tag{3}$$

To obtain the ML detection of the complex transmit vector $\boldsymbol{x}_c$ in (1) or the real vector $\tilde{\boldsymbol{x}}$ in (2), one solves the following *box-constrained integer least squares (BILS) problem*:

$$\min_{\tilde{\boldsymbol{x}} \in \mathcal{I}(q)^n} \|\tilde{\boldsymbol{y}} - \tilde{\boldsymbol{H}}\tilde{\boldsymbol{x}}\|_2^2. \tag{4}$$

For convenience, we define the following transformation

$$\boldsymbol{x} \triangleq \frac{(2^q - 1)\boldsymbol{e} + \tilde{\boldsymbol{x}}}{2},$$

so that $x_i \in \{0, 1, 2, \cdots, 2^q - 1\}$, $i = 1, 2, \cdots, n$. Define $\boldsymbol{y} \triangleq \tilde{\boldsymbol{y}} + (2^q - 1)\tilde{\boldsymbol{H}}\boldsymbol{e}$, $\boldsymbol{H} \triangleq 2\tilde{\boldsymbol{H}}$, then (4) becomes

$$\min_{\boldsymbol{x} \in \mathbb{Z}^n,\ \boldsymbol{l} \leq \boldsymbol{x} \leq \boldsymbol{u}} \|\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x}\|_2^2, \tag{5}$$

where $\boldsymbol{l} = \boldsymbol{0}$ and $\boldsymbol{u} = (2^q - 1)\boldsymbol{e}$. Notice that any permutation of the entries of $\boldsymbol{l}$ (or $\boldsymbol{u}$) does not change $\boldsymbol{l}$ (or $\boldsymbol{u}$).

## III. A New Maximum-Likelihood Decoding Method

A reduction process transforms the channel matrix $\boldsymbol{H}$ to an upper triangular matrix, which has good properties to make the search process more efficient. The typical reduction used in solving an unconstrained ILS problem is the LLL reduction, which can be described as the following decomposition:

$$\boldsymbol{H}\boldsymbol{Z} = [\boldsymbol{Q}_1, \boldsymbol{Q}_2] \begin{bmatrix} \boldsymbol{R} \\ \boldsymbol{0} \end{bmatrix} = \boldsymbol{Q}_1\boldsymbol{R},$$

where $\boldsymbol{Z} \in \mathbb{Z}^{n \times n}$ is unimodular, i.e., $\boldsymbol{Z}$ is an integer matrix with $|\det(\boldsymbol{Z})| = 1$, $[\underset{n}{\boldsymbol{Q}_1}, \underset{m-n}{\boldsymbol{Q}_2}] \in \mathbb{R}^{m \times m}$ is orthogonal, and $\boldsymbol{R} \in \mathbb{R}^{n \times n}$ is a nonsingular upper triangular matrix satisfying

$$|r_{ij}| \leq |r_{ii}|/2, \quad |r_{ii}| \leq \delta\sqrt{r_{i,i+1}^2 + r_{i+1,i+1}^2},\ 1 \leq \delta < 2 \tag{6}$$

for $j = i + 1 : n,\ i = 1 : n - 1$. The unimodular matrix $\boldsymbol{Z}$ is the product of a sequence of permutation matrices and the IGTs which have the following form (see [5] and [13])

$$\boldsymbol{G}_{ij} = \boldsymbol{I} - \zeta\boldsymbol{e}_i\boldsymbol{e}_j^T, \quad i \neq j, \quad \zeta \in \mathbb{Z}. \tag{7}$$

It is easy to verify that $\boldsymbol{G}_{ij}^{-1} = \boldsymbol{I} + \zeta\boldsymbol{e}_i\boldsymbol{e}_j^T \in \mathbb{Z}^{n \times n}$. These IGTs are used to reduce the off-diagonal entries of $\boldsymbol{R}$. If we take $\zeta = \lfloor r_{ij}/r_{ii} \rceil$, then for $\bar{\boldsymbol{R}} = \boldsymbol{R}\boldsymbol{G}_{ij}$, we have $|\bar{r}_{ij}| \leq |\bar{r}_{ii}|/2$. The properties in (6) can make a typical search process more efficient.

But using the LLL reduction to solve the BILS problem (5) would have difficulties to handle the box constraint $\boldsymbol{l} \leq \boldsymbol{x} \leq \boldsymbol{u}$. To solve (5), a typical reduction algorithm in the literature computes the QR decomposition with column permutation:

$$\boldsymbol{H}\boldsymbol{P} = [\boldsymbol{Q}_1, \boldsymbol{Q}_2] \begin{bmatrix} \boldsymbol{R} \\ \boldsymbol{0} \end{bmatrix} = \boldsymbol{Q}_1\boldsymbol{R}, \tag{8}$$

where $\boldsymbol{P} \in \mathbb{Z}^{n \times n}$ is a permutation matrix. Different reduction strategies use different permutation matrices.

To make the search process faster, like the LLL reduction, we would like to use IGTs to reduce the off-diagonal entries of $\boldsymbol{R}$. But we have to make sure that doing this will not cause difficulties to handle the box constraint in the search process. It turns out that the following form of the decomposition of $\boldsymbol{H}$ will do:

$$\boldsymbol{H}\boldsymbol{P}\boldsymbol{G} = [\boldsymbol{Q}_1, \boldsymbol{Q}_2] \begin{bmatrix} \boldsymbol{R} \\ \boldsymbol{0} \end{bmatrix} = \boldsymbol{Q}_1\boldsymbol{R}. \tag{9}$$

where $\boldsymbol{P} \in \mathbb{Z}^{n \times n}$ is a permutation matrix, $\boldsymbol{G} \in \mathbb{Z}^{n \times n}$ is product of the IGTs and is unit upper triangular, and the upper triangular $\boldsymbol{R}$ satisfies the first inequality in (6). In fact, using (9) we have

$$\|\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x}\|_2^2 = \|\boldsymbol{Q}_1^T\boldsymbol{y} - \boldsymbol{R}\boldsymbol{G}^{-1}\boldsymbol{P}^T\boldsymbol{x}\|_2^2 + \|\boldsymbol{Q}_2^T\boldsymbol{y}\|_2^2.$$

Define $\bar{\boldsymbol{y}} \triangleq \boldsymbol{Q}_1^T\boldsymbol{y}$, $\boldsymbol{z} \triangleq \boldsymbol{G}^{-1}\boldsymbol{P}^T\boldsymbol{x}$, then the BILS problem (5) is equivalent to

$$\min_{\boldsymbol{z} \in \mathbb{Z}^n,\ \boldsymbol{l} \leq \boldsymbol{G}\boldsymbol{z} \leq \boldsymbol{u}} \|\bar{\boldsymbol{y}} - \boldsymbol{R}\boldsymbol{z}\|_2^2. \tag{10}$$

If $\hat{\boldsymbol{z}}$ is the solution of (10), then $\hat{\boldsymbol{x}} = \boldsymbol{P}\boldsymbol{G}\hat{\boldsymbol{z}}$ is the solution of (5). Since $\boldsymbol{G}$ is upper triangular, we will see that there will be no difficulties in the search process for handling the box constraint.

In the following we first present a search algorithm for the problem (10), then we will show how to obtain an effective matrix decomposition (9), which can enhance the search speed.

### A. Search

Suppose the optimal solution to (10) satisfies

$$\|\bar{\boldsymbol{y}} - \boldsymbol{R}\boldsymbol{z}\|_2^2 < \beta^2 \tag{11}$$

for some $\beta$. The inequality (11) stands for a hyper-ellipsoid in terms of $\boldsymbol{z}$ or a hyper-sphere in terms of $\boldsymbol{R}\boldsymbol{z}$. Later we will say how to set the initial $\beta$. The search process is to seek the integer point within this hyper-ellipsoid which satisfies $\boldsymbol{l} \leq \boldsymbol{G}\boldsymbol{z} \leq \boldsymbol{u}$ and makes the left hand side of (11) smallest.

We rewrite (11) as

$$\sum_{k=1}^{n}\Big(\bar{y}_k-\sum_{j=k}^{n}r_{kj}z_j\Big)^2<\beta^2. \tag{12}$$

To simplify notation, define

$$c_n \triangleq \bar{y}_n/r_{nn}, \quad c_k \triangleq (\bar{y}_k-\sum_{j=k+1}^{n}r_{kj}z_j)/r_{kk}, \tag{13}$$

for $k=n-1,\cdots,1$. Note that $c_k$ depends on $z_{k+1}$, $z_{k+2}$, $\cdots$, $z_n$ and the former is determined when the latter are fixed. Then it follows from (12) that

$$\sum_{k=1}^{n}r_{kk}^2(z_k-c_k)^2<\beta^2. \tag{14}$$

Obviously any $\boldsymbol{z}$ satisfying (14) and $\boldsymbol{l}\leq\boldsymbol{G}\boldsymbol{z}\leq\boldsymbol{u}$ must also satisfy the following individual bounds:

$$\text{level n}: r_{nn}^2(z_n-c_n)^2<\beta^2, \tag{15}$$

$$z_n\in\big[l_n,u_n\big] \tag{16}$$

$$\vdots$$

$$\text{level k}: r_{kk}^2(z_k-c_k)^2<\beta^2-\sum_{j=k+1}^{n}r_{jj}^2(z_j-c_j)^2, \tag{17}$$

$$z_k\in\Big[l_k-\sum_{j=k+1}^{n}g_{kj}z_j,\ u_k-\sum_{j=k+1}^{n}g_{kj}z_j\Big] \tag{18}$$

$$\vdots$$

$$\text{level 1}: r_{11}^2(z_1-c_1)^2<\beta^2-\sum_{j=2}^{n}r_{jj}^2(z_j-c_j)^2, \tag{19}$$

$$z_1\in\Big[l_1-\sum_{j=2}^{n}g_{1j}z_j,\ u_1-\sum_{j=2}^{n}g_{1j}z_j\Big] \tag{20}$$

Based on these bounds, the Schnorr-Euchner based search algorithm proposed in [4] can be modified to find the optimal solution of (10).

Now we describe search process, which starts from level $n$. First we take $z_n$ to be the integer in the interval (16) which is nearest to $c_n$. If it does not satisfy the inequality (15), then the optimal solution of (10) is outside the hyper-ellipsoid (11) and we stop. Otherwise, we go to level $n-1$. At this level, we first compute $c_{n-1}$ by (13) and the two endpoints of the interval (18) with $k=n-1$, then we choose $z_{n-1}$ to be the integer in the interval which is nearest to $c_{n-1}$. If this $z_{n-1}$ does not satisfy the inequality (17) with $k=n-1$, then we go back to level $n$ and take $z_n$ to be the second nearest integer to $c_n$ in the interval (16), otherwise we proceed to level $n-2$. Continue this process until we reach level 1 and $z_1$ is determined. Then a full integer point $\hat{\boldsymbol{z}}$ is found. We store it, update $\beta^2$ by setting $\beta=\|\bar{\boldsymbol{y}}-\boldsymbol{R}\hat{\boldsymbol{z}}\|_2$. Then we start to search a new integer point in the new hyper-ellipsoid by updating $\hat{\boldsymbol{z}}$. We go back to level 2 and choose $z_2$ to be the next nearest integer in the interval (18) (with $k=2$) to $c_2$. If (17) (with $k=2$) holds for this new $z_2$, we move to level 1 to update $z_1$, otherwise we move to level 3 to try to update $z_3$, and so on. Finally we end up at level $n$, where we fail to get an integer for $z_n$ in the interval (16) to satisfy (15). Then the latest found integer point is the optimal solution of the problem (10). If the initial $\beta$ is set to $\infty$, then we refer to the first found integer point in the search process as the Babai integer point. In the following, we give a formal description of the search algorithm.

**Algorithm** Search

**Input:** The nonsingular upper triangular matrix $\boldsymbol{R}\in\mathbb{R}^{n\times n}$, the unit upper triangular matrix $\boldsymbol{G}\in\mathbb{Z}^{n\times n}$. the vector $\bar{\boldsymbol{y}}\in\mathbb{R}^n$, the lower bound vector $\boldsymbol{l}\in\mathbb{Z}^n$, the upper bound vector $\boldsymbol{u}\in\mathbb{Z}^n$, and the initial hyper-sphere radius $\beta$.

**Output:** The solution $\hat{\boldsymbol{z}}\in\mathbb{Z}^n$ of the problem (10).

**function**: $\hat{\boldsymbol{z}}=\text{SEARCH}(\boldsymbol{R},\boldsymbol{G},\bar{\boldsymbol{y}},\boldsymbol{l},\boldsymbol{u},\beta)$

Step 1) Set $k=n$, and $T_k=0$

Step 2) Compute $\tilde{l}_k=l_k-\sum_{j=k+1}^{n}g_{kj}z_j$,
  and $\tilde{u}_k=u_k-\sum_{j=k+1}^{n}g_{kj}z_j$
  Compute $c_k=(\bar{y}_k-\sum_{j=k+1}^{n}r_{kj}z_j)/r_{kk}$,
  and $z_k=\lfloor c_k\rceil$
  Set $lbound_k=0$, $ubound_k=0$
  **if** $z_k\leq\tilde{l}_k$, **then**
    Set $z_k=\tilde{l}_k$, $lbound_k=1$, and $\Delta_k=1$
  **else if** $z_k\geq\tilde{u}_k$, **then**
    Set $z_k=\tilde{u}_k$, $ubound_k=1$, and $\Delta_k=-1$
  **else**
    Set $\Delta_k=\text{sign}(c_k-z_k)$
  **end**

Step 3) **if** $T_k+r_{kk}^2(z_k-c_k)^2\geq\beta^2$, **then**
    Go to Step 4
  **else if** $k>1$, **then**
    Compute $T_{k-1}=T_k+r_{kk}^2(z_k-c_k)^2$,
    set $k=k-1$, go to Step 2
  **else**
    Compute $\beta^2=T_1+r_{11}^2(z_1-c_1)^2$, set $\hat{\boldsymbol{z}}=\boldsymbol{z}$,
    and $k=k+1$, go to Step 5
  **end**

Step 4) **if** $k=n$, **then**
    Terminate
  **else**
    Set $k=k+1$
  **end**

Step 5) **if** $ubound_k=1$ **and** $lbound_k=1$, **then**
    Go to Step 4
  **end**
  Set $z_k=z_k+\Delta_k$
  **if** $z_k=\tilde{l}_k$, **then**
    Set $lbound_k=1$, $\Delta_k=-\Delta_k-\text{sign}(\Delta_k)$
  **else if** $z_k=\tilde{u}_k$, **then**
    Set $ubound_k=1$, $\Delta_k=-\Delta_k-\text{sign}(\Delta_k)$
  **else if** $lbound_k=1$, **then** set $\Delta_k=1$
  **else if** $ubound_k=1$, **then** set $\Delta_k=-1$
  **else**
    Compute $\Delta_k=-\Delta_k-\text{sign}(\Delta_k)$
  **end**
  Go to Step 3.

## *B. Reduction*

In the search process, at level $i$, we have the inequality

$$r_{ii}^2(z_i - c_i)^2 < \beta^2 - \sum_{k=i+1}^{n} r_{kk}^2(z_k - c_k)^2, \tag{21}$$

which shows that in order to reduce the search range of $z_i$ for $i = n, n-1, \cdots, 1$, each term $r_{kk}^2(z_k - c_k)^2$ on the right hand side of (21) should be as large as possible and $r_{ii}^2$ should be large as well. This observation motivates the reduction strategy given by Chang and Han in [4], which will be referred to as the CH reduction for convenience.

Suppose $\boldsymbol{H}$ has the QR decomposition $\boldsymbol{H} = [\boldsymbol{Q}_1, \boldsymbol{Q}_2]\left[\begin{smallmatrix}\boldsymbol{R}\\ \boldsymbol{0}\end{smallmatrix}\right]$. In our new reduction, we will use the idea of the CH reduction for column reordering, but we will also use IGTs to reduce the off-diagonal entries of $\boldsymbol{R}$. Like the CH reduction, the new reduction reorders the columns of $\boldsymbol{H}$ or equivalently, the columns of $\boldsymbol{R}$ from the last one to the first one. We work with the $\boldsymbol{R}$ matrix in the reduction process. Suppose columns $n, n-1, \ldots, k+1$ of $\boldsymbol{R}$ have been determined, and now we want to determine the $k$-th column, i.e., choose a column from the first $k$ columns of $\boldsymbol{R}$ as the $k$-th column. For each $j$ satisfying $1 \le j \le k$, we first interchange column $j$ and column $k$ of $\boldsymbol{R}$, and then bring $\boldsymbol{R}$ back to an upper triangular matrix by applying Givens rotations from the left. Then we compute the corresponding term $|r_{kk}(z_k - c_k)|$ (see the next paragraph). The column which corresponds to the largest one among the $k$ values of $|r_{kk}(z_k - c_k)|$ is chosen to be the $k$-th column.

Now we show how to compute $|r_{kk}(z_k - c_k)|$. From (13) we see that $c_k$ depends on $r_{k,k+1}, \cdots, r_{kn}$ and $z_{k+1}, \ldots, z_n$. We apply $\boldsymbol{G}_k = \boldsymbol{G}_{k,k+1}\boldsymbol{G}_{k,k+2}\cdots\boldsymbol{G}_{k,n}$ to $\boldsymbol{R}$ from the right so that $|r_{kj}| \le |r_{kk}|/2$ for $j = k+1 : n$. Note that $\boldsymbol{G}_k$ is the identity matrix $\boldsymbol{I}_n$ except that $\boldsymbol{G}_k(k, k+1:n)$ is an integer row vector and can also be called an IGT. These new $r_{kj}$ for $j = k+1 : n$ will be used in computing $c_k$. We take $z_j$ to be the integer in the interval at level $j$ (cf. (18) for the interval at $k$) which is nearest to $c_j$ for $j = k+1 : n$ when we compute $c_k$. These $z_j$ are also used to determine the interval (18). After we compute $c_k$, like the CH reduction, we take $z_k$ to be the integer in the interval (18) which is the second nearest to $c_k$ in computing $|r_{kk}(z_k - c_k)|$, see [4] for the reason for this choice.

When the reduction is done, we have

$$\boldsymbol{H}\boldsymbol{P}_n\boldsymbol{P}_{n-1}\boldsymbol{G}_{n-1}\boldsymbol{P}_{n-2}\boldsymbol{G}_{n-2}\cdots\boldsymbol{P}_2\boldsymbol{G}_2\boldsymbol{G}_1 = [\boldsymbol{Q}_1, \boldsymbol{Q}_2]\begin{bmatrix}\boldsymbol{R}\\ \boldsymbol{0}\end{bmatrix}, \tag{22}$$

where $\boldsymbol{P}_k$ for $k = n, \cdots, 2$ is the permutation matrix which was used to interchange some column $j$ ($j \le k$) and column $k$. Notice that there are no $\boldsymbol{G}_n$ and $\boldsymbol{P}_1$ in (22). From the structure of $\boldsymbol{P}_j$ and $\boldsymbol{G}_k$ we can easily verify that $\boldsymbol{G}_k\boldsymbol{P}_j = \boldsymbol{P}_j\boldsymbol{G}_k$ for $j < k$. Thus with $\boldsymbol{P} \triangleq \boldsymbol{P}_n\boldsymbol{P}_{n-1}\cdots\boldsymbol{P}_2$, $\boldsymbol{G} \triangleq \boldsymbol{G}_{n-1}\boldsymbol{G}_{n-2}\cdots\boldsymbol{G}_1$, which is unit upper triangular, (22) can be written as

$$\boldsymbol{H}\boldsymbol{P}\boldsymbol{G} = [\boldsymbol{Q}_1, \boldsymbol{Q}_2]\begin{bmatrix}\boldsymbol{R}\\ \boldsymbol{0}\end{bmatrix}.$$

This has the same form as (9). The whole reduction process can be summarized as follows.

**Algorithm** Reduction

**Input:** The channel matrix $\boldsymbol{H} \in \mathbb{R}^{m\times n}$ with full column rank, the input vector $\boldsymbol{y} \in \mathbb{R}^m$, the lower bound vector $\boldsymbol{l} = \boldsymbol{0} \in \mathbb{Z}^n$, and the upper bound vector $\boldsymbol{u} = (2^q - 1)\boldsymbol{e} \in \mathbb{Z}^n$ .
**Output:** The reduced upper triangular matrix $\boldsymbol{R} \in \mathbb{R}^{n\times n}$, the permutation matrix $\boldsymbol{P} \in \mathbb{Z}^{n\times n}$, the upper triangular unimodular matrix $\boldsymbol{G} \in \mathbb{Z}^{n\times n}$, the vector $\bar{\boldsymbol{y}} \in \mathbb{R}^n$, and the Babai integer point $\hat{\boldsymbol{z}} \in \mathbb{Z}^n$.
**function**: $[\boldsymbol{R}, \boldsymbol{P}, \boldsymbol{G}, \bar{\boldsymbol{y}}, \hat{\boldsymbol{z}}] = \text{REDUCTION}(\boldsymbol{H}, \boldsymbol{y}, \boldsymbol{l}, \boldsymbol{u})$
Compute $[\boldsymbol{Q}_1, \boldsymbol{Q}_2]^T\boldsymbol{H} = \begin{bmatrix}\boldsymbol{R}\\ \boldsymbol{0}\end{bmatrix}$ by the Householder
  transformations and simultaneously compute
  $[\boldsymbol{Q}_1, \boldsymbol{Q}_2]^T\boldsymbol{y}$ and set $\bar{\boldsymbol{y}} = \boldsymbol{Q}_1^T\boldsymbol{y}$
Set $\boldsymbol{P} = \boldsymbol{I}_n$, $\boldsymbol{G} = \boldsymbol{I}_n$
**for** $k = n : -1 : 1$
    Set $\boldsymbol{R}_{tmp} = \boldsymbol{R}$, $\boldsymbol{G}_{tmp} = \boldsymbol{G}$, $\bar{\boldsymbol{y}}_{tmp} = \bar{\boldsymbol{y}}$, $\alpha = -1$
    **for** $j = 1 : k$
        Set $\boldsymbol{R}' = \boldsymbol{R}_{tmp}$, $\boldsymbol{G}' = \boldsymbol{G}_{tmp}$, $\bar{\boldsymbol{y}}' = \bar{\boldsymbol{y}}_{tmp}$
        Interchange columns of $j$ and $k$ of $\boldsymbol{R}'(1:k, 1:k)$
         and transform $\boldsymbol{R}'(1:k, 1:k)$ to an upper
         triangular matrix by Givens rotations
        Apply the same Givens rotations to $\bar{\boldsymbol{y}}'(1:k)$
         and $\boldsymbol{R}'(1:k, k+1:n)$
        Apply IGT $\boldsymbol{G}_k$ to $\boldsymbol{R}'$, i.e., $\boldsymbol{R}' = \boldsymbol{R}'\boldsymbol{G}_k$ such that
         $|r'_{kj}| \le |r'_{kk}|/2$ for $j = k+1 : n$
        Compute $\boldsymbol{G}' = \boldsymbol{G}'\boldsymbol{G}_k$,
         $\tilde{l}_k = l_k - \sum_{l=k+1}^{n} g'_{kl}\hat{z}_l$,
         $\tilde{u}_k = u_k - \sum_{l=k+1}^{n} g'_{kl}\hat{z}_l$
         $c_k = (\bar{y}'_k - \sum_{j=k+1}^{n} r'_{kj}\hat{z}_j)/r'_{kk}$
        Set $\hat{z}'_k$ and $\hat{z}''_k$ to be the first and second
         nearest integers on $[\tilde{l}_k, \tilde{u}_k]$ to $c_k$, respectively
        Compute $\alpha' = |r'_{kk}(\hat{z}''_k - c_k)|$
        **if** $\alpha' > \alpha$, **then**
            Set $\alpha = \alpha'$, $p = j$, $\hat{z}_k = \hat{z}'_k$,
             $\boldsymbol{R} = \boldsymbol{R}'$, $\boldsymbol{G} = \boldsymbol{G}'$, $\bar{\boldsymbol{y}} = \bar{\boldsymbol{y}}'$
        **end**
    **end**
    Interchange columns $p$ and $k$ of $\boldsymbol{P}$
**end**

The above algorithm gives the Babai integer point $\hat{\boldsymbol{z}}$, which can be used to set the initial $\beta$ by defining $\beta = \|\bar{\boldsymbol{y}} - \boldsymbol{R}\hat{\boldsymbol{z}}\|_2$ for Algorithm Search. The initial $\beta$ can also be set to $\infty$.

## IV. SIMULATION RESULTS

In this section we compare the computational costs of the proposed ML decoder, which is to be referred to as Algorithm CY for convenience, and the ML decoder with the CH reduction proposed in [4], which is to be called Algorithm CH. We also compare the performance of the Babai integer points obtained by the two algorithms. We consider

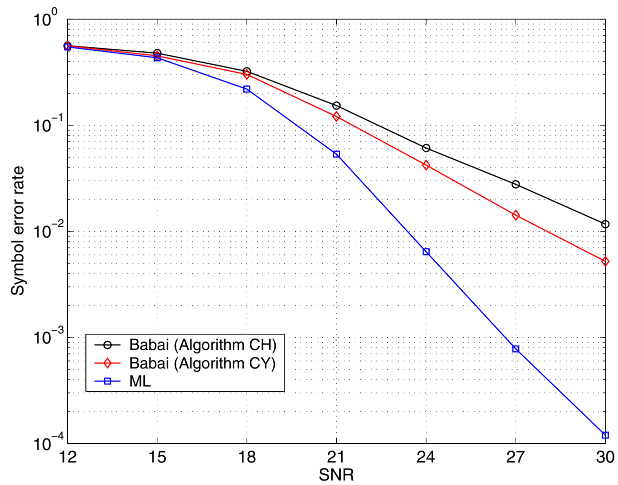


Fig. 1. Symbol error rate vs SNR (16QAM)

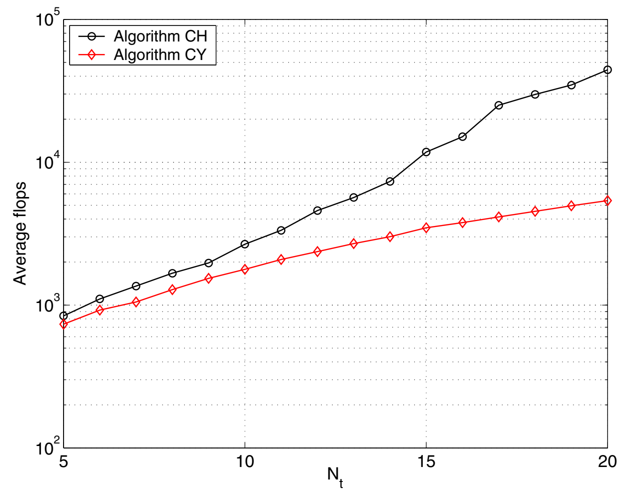


Fig. 2. Average flops vs dimension $N_t$ (16QAM)

the flat fading MIMO systems. All the simulations were run in MATLAB 7.0.

The channel matrix $\tilde{\boldsymbol{H}}$ and the receive vector $\tilde{\boldsymbol{y}}$ were generated according to the real system (2), where $\tilde{\boldsymbol{H}} \in \mathbb{R}^{m\times n}$ with $m = 2N_r$ and $n = 2N_t$. The performance and the computational cost are measured by symbol error rate (SER) and the number of flops, respectively. For each algorithm, the total flops of the reduction process and the search process are counted. For each case, we performed 500 runs and counted the average flops.

Fig. 1 in which $N_t = N_r = 12$ shows the symbol error rates (SER) of the Babai points obtained respectively by Algorithms CH and CY, and the SER of the ML estimate (obtained by Algorithm CY) versus different SNR for 16QAM. Here for $M$-QAM, SNR is defined by $\text{SNR} = 10\log_{10}\frac{(M-1)/3}{2\sigma^2}$. From this figure, we observe that Algorithm CY gives a better Babai point in terms of SER than Algorithm CH. This helps to understand why Algorithm CY costs less than Algorithm CH for finding the ML estimate, see Fig. 2.

Fig. 2 in which $N_t = N_r$ and $\text{SNR} = 20$ shows the average flops required for finding the ML estimate for 16QAM by Algorithm CH and Algorithm CY. When the dimension $N_t$ increases, Algorithm CY becomes more and more efficient than Algorithm CH. When $N_t = 20$ in Fig. 2, Algorithm CY is about eight times as fast as Algorithm CH.

## V. Summary

We have presented a new efficient maximum-likelihood decoder for the linear MIMO systems. The new reduction algorithm not only reorders the columns of the channel matrix by using the idea of [4] but also reduces the off-diagonal entries of the $\boldsymbol{R}$ factor of the QR decomposition of the channel matrix by integer Gauss transformations. These can significantly improve the search speed. We also presented a modified Schnorr-Euchner based search algorithm. Numerical simulations indicated that this decoder can be much more efficient than the decoder given in [4].

## References


[1] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger,"Closest point search in lattices," *IEEE Trans. Inform. Theory*, vol. 48, pp. 2201-2214, Aug. 2002.

[2] J. Boutros, N. Gresset, L. Brunel, and M. Fossorier, "Soft-input soft-output lattice sphere decoder for linear channels," in *IEEE 2003 Global Communications Conference*, San Francisco, U.S.A., Dec. 2003.

[3] L. Brunel, "Multiuser detection techniques using maximum likelihood sphere decoding in multicarrier CDMA systems," *IEEE Trans. Wireless Commu.*, vol. 3, no. 3, pp. 949-957, 2004.

[4] X.-W. Chang and Q. Han, "Solving box-constrained integer least squares problems," *IEEE Trans. Wireless Commun.*, vol. 7, no. 1, pp. 277-287, 2008.

[5] X.-W. Chang, X. Yang, and T. Zhou, "MLAMBDA: a modified LAMBDA method for integer least-squares estimation," *Journal of Geodesy*, vol. 79, pp. 552-565, 2005.

[6] M. Damen, A. Chkeif, and J.C. Belfiore,"Lattice code decoder for space-time codes," *IEEE Commun. Lett.*, vol. 4, no. 5, pp. 161-163, 2000.

[7] M. Damen, H. El Gamal, and G. Caire, "On maximum-likelihood detection and the search for the closest lattice point," *IEEE Trans. Inf. Theory*, vol. 49, no. 10, pp. 2389–2402, Oct. 2003.

[8] J. Foschini, G. Golden, R. Valenzuela, and P. Wolniansky, "Simplified processing for high spectral efficiency wireless communication employing multi-element arrays," *IEEE J. Select. Areas Commu.*, vol. 17, pp. 1841-1852, Nov. 1999.

[9] A.K. Lenstra, H.W. Lenstra, and L. Lovasz,"Factoring polynomials with rational coefficients," *Mathematicsche Annalen*, vol. 261, pp.515-534, 1982.

[10] W.K. Ma, T.N. Davidson, K.M. Wong, Z.-Q. Luo, and P.C. Ching, "Quasi-maximum-likelihood multiuser detection using semi-definite relaxation," *IEEE Trans. Signal Processing*, vol. 50, pp. 912-922, Apr. 2002.

[11] M. Pohst,"On the computation of lattice vectors of minimal length, successive minima and reduced bases with applications," *ACM SIGSAM bulletin*, vol. 15, pp. 37-44, Feb. 1981.

[12] C.P. Schnorr and M. Euchner,"Lattice basis reduction: Improved practical algorithms and solving subset sum problems," *Math. Programming*, vol. 66, pp. 181-191, 1994.

[13] P.J.G. Teunissen, "The least-squares ambiguity decorrelation adjustment: a method for fast GPS ambiguity estimation," *Journal of Geodesy*, vol. 70, pp. 65-82, 1995.

[14] E. Viterbo and J. Boutros, "A universal lattice code decoder for fading channel," *IEEE Trans. Infor. Theory*, vol. 45, pp. 1639-1642, Jul. 1999.

[15] C. Windpassinger and R. Fischer, "Low-complexity near maximum likelihood detection and precoding for MIMO systms using lattice reduction," in *Proc. IEEE Information Theory Workshop*, pp. 345-348, Paris, France, Apr. 2003.

[16] D. Wubben, R. Bohnke, J. Rinas, V. Kuhn, and K.D. Kammeyer, "Efficient algorithm for decoding layered space-time codes," *IEEE Electronics Letters*, vol. 37, pp. 1348–1350, October 2001.