

# A Soft-Output Detector for the Golden Code

David L. Milliner, Mohammed O. Sinnokrot, John R. Barry  
 School of Electrical and Computer Engineering  
 Georgia Institute of Technology, Atlanta, Georgia 30332-0250  
 Email: {dlm,sinnokrot,barry}@ece.gatech.edu

**Abstract**—Soft-output detection of the golden code is an important but computationally difficult task. We propose a low- and fixed- complexity soft-output detector for the golden code that uses linear equalization to simplify the task of finding a list of candidate values for one pair of information symbols, and then – for each pair on the list – it uses decision-feedback equalization to find candidate values for the remaining pair of information symbols. We propose a simple ordering algorithm that exploits the golden code’s structure to ensure that the overall algorithm performs well. Numerical results indicate that the proposed algorithm is significantly less complex than previously reported algorithms, yet performs comparably.

## I. INTRODUCTION

A communication system having multiple antennas at both the transmitter and receiver is capable of achieving higher data rates and greater robustness to fading than a single-antenna system. These gains can be achieved in part through the use of space-time coding [1], [2]. The golden code is a code for two transmit and two or more receive antennas proposed independently in [3] and [4] that has many advantages: it is a full-rate code; it is a full-diversity code; it has a maximal coding gain; and it performs better than all previously reported full-rate codes with two transmit antennas in terms of the SNR required to achieve a target error probability.

To manage complexity, a practical multiantenna (MIMO) receiver will separate the tasks of MIMO detection and error-control decoding. In this setting, the best detector is a soft-output detector that computes the *a posteriori probability* (APP) for each of the coded bits. Soft-output detection is important for two reasons: First, the accuracy with which the detector approximates the true APPs can significantly affect performance, and second, the high complexity of soft-output detection can easily dominate the other receiver tasks such as error-control decoding.

In this work we present a soft-output detector for the golden code with low- and fixed- computational complexity. Our algorithm employs a linear equalizer followed by a simple enumeration algorithm that finds a joint list for the first and second symbol detected. A soft-output detector for the golden code was proposed in [5] that has a fixed computational complexity of  $\mathcal{O}(q^2)$ , where  $q$  is the alphabet size, which can be prohibitive for large alphabets. Like [5] our detector has computational complexity  $\mathcal{O}(q^2)$  when the list length  $\ell = q^2$ . However, for  $\ell \ll q^2$ , our algorithm achieves comparable performance to [5] at much lower complexity.

The remainder of this paper is organized as follows. We present the system model in §II and state the soft-output

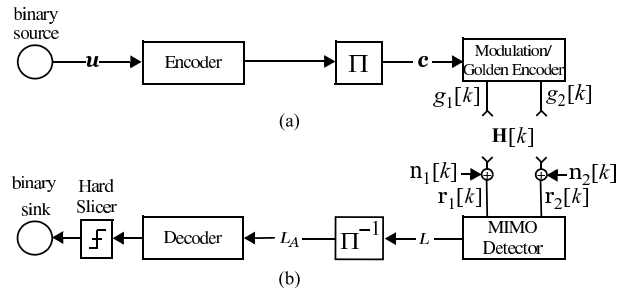


Figure 1. System model with (a) MIMO transmitter and (b) MIMO receiver.

MIMO detection problem in §III. We propose our soft-output detection algorithm for the golden code in §IV. Results are presented in §V and conclusions in §VI.

## II. GOLDEN CODE SYSTEM MODEL

We consider the transmitter shown in Fig. 1-a. The input is a vector  $\mathbf{u}$  of i.i.d. uniform information bits that is encoded and interleaved. The coded bit stream is partitioned into blocks  $\mathbf{c}$  of  $4\omega$  bits, where  $\omega = \log_2 q$  is the number of bits per symbol. Each block is mapped and encoded onto a vector of four complex information symbols  $\mathbf{a} = [a_1, a_2, a_3, a_4]^T$  whose components are taken from a QAM alphabet  $\mathcal{A}$  of size  $q = |\mathcal{A}| = 2^\omega$  and energy  $E/2$ . The golden code encodes and transmits these four information symbols over two symbol periods from two antennas, so that the rate of the space-time code is two symbols per signaling interval. The transmitted codeword can be expressed as a  $2 \times 2$  matrix:

$$\mathbf{G} = \begin{bmatrix} g_1[1] & g_2[1] \\ g_1[2] & g_2[2] \end{bmatrix}, \quad (1)$$

where  $g_i[k]$  denotes the symbol transmitted from antenna  $i \in \{1, 2\}$  at time  $k \in \{1, 2\}$ .

We assume the entries of the channel matrix are i.i.d. complex Gaussian random variables of variance 1, that  $N_r \geq 2$  and that the receiver knows the channel perfectly. The received signal  $r_l[k]$  at receive antenna  $l$  at time  $k$  is given by:

$$r_l[k] = \sum_{i=1}^2 g_i[k] h_{i,l}[k] + n_l[k] \quad (2)$$

where  $n_l[k]$  is the complex additive-white Gaussian noise at receive antenna  $l$  at time  $k$ ,  $\mathbb{E}|n_l[k]|^2 = N_0$ , and  $h_{i,l}[k]$  is the channel coefficient between the  $i$ -th transmit antenna and  $l$ -th receive antenna at time  $k$ . The signal-to-noise ratio (SNR) at any receive antenna is  $\text{SNR} = E/N_0$ .

In Fig. 1-b we show a receiver consisting of a MIMO detector (this paper's focus), a deinterleaver, and an error-control decoder. An iterative receiver based on the turbo principle can improve performance, but we limit our discussion to a system with no feedback from decoder to detector.

### III. EFFECTIVE CHANNEL MATRIX AND SOFT DETECTION

The Dayal-Varanasi golden code [4] encodes one pair of information symbols  $\mathbf{v} = [a_1, a_2]^T$  onto the main diagonal of  $\mathbf{G}$ , and it encodes a second pair of information symbols  $\mathbf{w} = [a_3, a_4]^T$  onto the off-diagonal, so that:

$$\mathbf{G} = \begin{bmatrix} \tilde{v}_1 & 0 \\ 0 & \tilde{v}_2 \end{bmatrix} + \phi \begin{bmatrix} 0 & \tilde{w}_1 \\ \tilde{w}_2 & 0 \end{bmatrix} \quad (3)$$

where:

$$\begin{aligned} \tilde{\mathbf{v}} &= \mathbf{M}\mathbf{v}, \quad \tilde{\mathbf{w}} = \mathbf{M}\mathbf{w}, \quad \mathbf{M} = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix}, \\ \theta &= \frac{1}{2} \tan^{-1}(2), \quad \phi = e^{j\pi/4}. \end{aligned} \quad (4)$$

Substituting (3) and (4) into (2), the vector of received samples  $\mathbf{r} = [r_1[1], r_1[2], r_2[1], r_2[2]]^T$  at a receiver with two antennas at the two time instances can be written as the output of an effective four-input four-output channel:

$$\mathbf{r} = \mathbf{H}\mathbf{a} + \mathbf{n}, \quad (5)$$

where  $\mathbf{n} = [n_1[1], \dots, n_2[2]]^T$  is the noise and  $\mathbf{H} = \bar{\mathbf{H}}\Psi$  is the *effective channel matrix*:

$$\mathbf{H} = \underbrace{\begin{bmatrix} h_{11}[1] & 0 & \phi h_{21}[1] & 0 \\ 0 & h_{21}[2] & 0 & \phi h_{11}[2] \\ h_{12}[1] & 0 & \phi h_{22}[1] & 0 \\ 0 & h_{22}[2] & 0 & \phi h_{12}[2] \end{bmatrix}}_{\bar{\mathbf{H}}} \underbrace{\begin{bmatrix} t & s & 0 & 0 \\ -s & t & 0 & 0 \\ 0 & 0 & t & s \\ 0 & 0 & -s & t \end{bmatrix}}_{\Psi} \quad (6)$$

where  $s = \sin(\theta)$  and  $t = \cos(\theta)$ .

The aim of a soft-output detector is to calculate or approximate the *a posteriori* probability (APP) for each of the bits  $c_j$  in a given codeword, where  $j \in \{1, \dots, 4\omega\}$  is the bit index. This probability is conveniently represented by the so-called a posteriori log-likelihood ratio (LLR):

$$L(c_j|\mathbf{r}) := \ln \frac{\Pr[c_j = +1|\mathbf{r}]}{\Pr[c_j = -1|\mathbf{r}]}. \quad (7)$$

The sign of  $L(c_j|\mathbf{r})$  is the maximum a posteriori (MAP) estimate for  $c_j$ , and the magnitude represents the reliability.

Let  $\mathcal{Z} = \mathcal{A}^4$  denote the set of all possible symbol vectors  $\mathbf{a} \in \mathcal{Z}$ , one for each binary vector  $\mathbf{c} \in \{\pm 1\}^{4\omega}$ , as determined by the mapping  $\mathbf{a}(\cdot)$  from coded bits to information symbols. After a series of manipulations that exploit (a) the application of Bayes' rule, (b) our assumption of no *a priori* information, (c) our assumption of AWGN and (d) the application of the max-log approximation, (7) reduces to:

$$L(c_j|\mathbf{r}) \approx \min_{\hat{\mathbf{a}} \in \mathcal{Z}_j^{-1}} \frac{\|\mathbf{r} - \mathbf{H}\hat{\mathbf{a}}\|^2}{N_0} - \min_{\hat{\mathbf{a}} \in \mathcal{Z}_j^{+1}} \frac{\|\mathbf{r} - \mathbf{H}\hat{\mathbf{a}}\|^2}{N_0}, \quad (8)$$

where  $\mathcal{Z}_j^{\pm 1}$  denotes a partitioning of  $\mathcal{Z}$  depending on whether the  $j^{\text{th}}$  bit label is 1 or  $-1$ , namely:

$$\mathcal{Z}_j^{+1} = \{\mathbf{a}(\mathbf{c}) : c_j = +1\}, \quad \mathcal{Z}_j^{-1} = \{\mathbf{a}(\mathbf{c}) : c_j = -1\}. \quad (9)$$

After a QR-decomposition of the channel matrix,  $\mathbf{H} = \mathbf{Q}\mathbf{R}$ , the cost function for a candidate  $\hat{\mathbf{a}}$  is:

$$J(\hat{\mathbf{a}}) = \|\mathbf{r} - \mathbf{H}\hat{\mathbf{a}}\|^2 = \|\mathbf{y} - \mathbf{R}\hat{\mathbf{a}}\|^2 = \sum_{i=1}^4 |y_i - \sum_{l=1}^i R_{il}\hat{a}_l|^2, \quad (10)$$

where  $\mathbf{R}$  is a lower triangular matrix,  $\mathbf{Q}$  is an orthogonal matrix and  $\mathbf{y} = \mathbf{Q}^*\mathbf{r}$ .

*List detection* is the process of finding a list of candidates  $\mathcal{L} \subseteq \mathcal{Z}$  of a desired length  $\ell = |\mathcal{L}|$ , with the intention of approximating (8) by:

$$L(c_j|\mathbf{r}) \approx \min_{\hat{\mathbf{a}} \in \mathcal{L} \cap \mathcal{Z}_j^{-1}} \frac{\|\mathbf{r} - \mathbf{H}\hat{\mathbf{a}}\|^2}{N_0} - \min_{\hat{\mathbf{a}} \in \mathcal{L} \cap \mathcal{Z}_j^{+1}} \frac{\|\mathbf{r} - \mathbf{H}\hat{\mathbf{a}}\|^2}{N_0}. \quad (11)$$

The difference between (8) and (11) is the insertion of the list  $\mathcal{L}$ . The list length  $\ell = |\mathcal{L}|$  plays a critical roll in the overall complexity and performance and is typically much less than  $|\mathcal{Z}| = q^4$ , so that (11) is significantly less complex than (8).

### IV. SOFT-OUTPUT DETECTION OF THE GOLDEN CODE

This section presents a low and fixed computational complexity soft-output detector for the golden code. The initial step is an *ordering* task that determines which two of the four information symbols to detect first. The second step is to find a candidate list of possible values for this first pair of information symbols; this task is facilitated by a linear zero-forcing filter. The final task is to find, for each candidate pair from the list, a complementary pair of values for the remaining information symbols using decision-feedback detection.

#### A. Ordering of the Effective Channel Matrix

We first determine the initial pair of information symbols detected, and then the order of detection for the remaining pair of symbols. Consider the system in (5), which can be rewritten:

$$\mathbf{r} = \bar{\mathbf{H}}\mathbf{x} + \mathbf{n}, \quad (12)$$

where  $\bar{\mathbf{H}} = \mathbf{H}\mathbf{P}$  is a permuted channel, where  $\mathbf{P}$  is a permutation matrix, and where  $\mathbf{x} = \mathbf{P}^{-1}\mathbf{a}$  is the permuted vector of information symbols. The ordering can be represented by the matrix  $\mathbf{P}$ , which approximately maximizes the SNR for the first pair of detected symbols ( $\hat{x}_1, \hat{x}_2$ ).

Fig. 2 provides pseudocode for the proposed ordering algorithm. In order to maximize the SNR for the first detected symbol, the ordering algorithm chooses the first column as the one with the largest norm. It is easily verified by direct computation that  $\|\mathbf{h}_1\| = \|\mathbf{h}_4\|$  and  $\|\mathbf{h}_2\| = \|\mathbf{h}_3\|$ . Therefore, we only need to compare the norm values of  $\mathbf{h}_1$  and  $\mathbf{h}_2$  (line 1). The first column of  $\bar{\mathbf{H}}$  (i.e.,  $\bar{\mathbf{h}}_1$ ) is  $\mathbf{h}_{k_1}$ , where  $k_1$  is given by line 2 or line 5. After deciding on the first column, there are three possibilities for the second column.

The three choices of the second column are given by  $\mathcal{C}_2$  (line 3 or line 6). For every choice of the second column, there are two ways to choose the third column and one way to choose the fourth column. To simplify the algorithm even further we allow for only one way to choose the third and fourth columns as shown in line 3 and line 6. In particular, we choose the third column as the one with the largest norm. Lines 1 through 7 can thus be interpreted as finding the index  $k_1$  and determining the possible choices for the second, third and fourth columns. For example, assume that  $\|\mathbf{h}_1\| > \|\mathbf{h}_2\|$ , then there are three possible permutations for the matrix  $\mathbf{P}$  as given by lines 2 and 3. These permutations are  $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_4, \mathbf{e}_3\}, \{\mathbf{e}_1, \mathbf{e}_3, \mathbf{e}_4, \mathbf{e}_2\}, \{\mathbf{e}_1, \mathbf{e}_4, \mathbf{e}_2, \mathbf{e}_3\}$ , where  $\mathbf{e}_j$  is the  $j$ -th column of the identity matrix  $\mathbf{I}_{4 \times 4}$ .

To maximize the SNR of the second symbol detected, we choose the second column index  $k_2$  such that  $\bar{\mathbf{h}}_1$  and  $\bar{\mathbf{h}}_2$  (or  $\mathbf{h}_{k_1}$  and  $\mathbf{h}_{k_2}$ ) are as orthogonal as possible, i.e. to minimize:

$$\delta = |\mathbf{h}_{k_1}^* \mathbf{h}_{k_2}| / (\|\mathbf{h}_{k_1}\| \|\mathbf{h}_{k_2}\|). \quad (13)$$

In line 8 we initialize the metric in (13). The algorithm then goes through each of three choices for the second, third and fourth columns (line 10) and computes the metric in (13). If the current metric value is less than the minimum value (line 12), then the current value becomes the minimum value (line 13) and the permutation matrix is updated.

The computational complexity for the ordering algorithm involves computing the norm for all four columns, the dot product for three pairs of columns, four comparisons and three real divisions. For a given list length, we show in section V that the proposed ordering algorithms outperforms other ordering algorithms, e.g. the sorted QR decomposition (SQRD) [6]. The main difference between the proposed algorithm and BLAST or SQRD ordering is that the proposed ordering algorithm not only maximizes the SNR for the pair of symbols  $(\hat{x}_1, \hat{x}_2)$  but also minimizes interference between the two detected symbols as described by (13).

### B. ZF Equalization, List Enumeration, and DF Detection

Consider the system in (12). After a QR decomposition  $\bar{\mathbf{H}} = \mathbf{Q}\mathbf{R}$  and multiplying the channel output by  $\mathbf{Q}^*$ , we obtain the effective channel output  $\mathbf{y} = \mathbf{Q}^* \mathbf{r}$ . Applying a linear filter  $\mathbf{R}^{-1}$  and performing a symbol slicing operation yields:

$$\hat{\mathbf{x}} = \mathcal{Q}(\mathbf{R}^{-1} \mathbf{y}) = \mathcal{Q}(\mathbf{x} + \tilde{\mathbf{n}}), \quad (14)$$

where  $\mathcal{Q}(\cdot)$  rounds its input to the nearest element of  $\mathcal{A}$  and  $\tilde{\mathbf{n}} = \mathbf{R}^{-1} \mathbf{Q}^* \mathbf{n}$ . The proposed enumeration algorithm exploits  $\hat{x}_1$  and  $\hat{x}_2$ , ignoring the outputs  $\hat{x}_3$  and  $\hat{x}_4$ . Specifically, we use  $\hat{x}_1$  to form a candidate list  $\mathcal{L}_1$  for the symbol  $x_1$  and  $\hat{x}_2$  to form a candidate list  $\mathcal{L}_2$  for  $x_2$ . Then, the list  $\mathcal{L}_{12}$  is formed by combining all possible symbols from the candidate lists  $\mathcal{L}_1$  and  $\mathcal{L}_2$ . Specifically,  $\mathcal{L}_{12} = \{(\mathcal{L}_1[m], \mathcal{L}_2[n])\} \forall m \in \{1, \dots, \sqrt{\ell}\}$  and  $\forall n \in \{1, \dots, \sqrt{\ell}\}$ , where  $\mathcal{L}_1[m]$  (resp.  $\mathcal{L}_2[n]$ ) is the  $m$ -th (resp.  $n$ -th) element of the list  $\mathcal{L}_1$  (resp.  $\mathcal{L}_2$ ).

Our enumeration algorithm for the list  $\mathcal{L}_1$  approximates the  $\sqrt{\ell}$  closest points to the decision  $\hat{x}_1$  by separating  $\hat{x}_1$  into its real and imaginary components. We do so by first finding

**Input: H    Output: P**

```

1 if  $\|\mathbf{h}_1\| > \|\mathbf{h}_2\|$  then
2    $k_1 = 1$ 
3    $\mathcal{C}_2 = [2, 3, 4], \mathcal{C}_3 = [4, 4, 2], \mathcal{C}_4 = [3, 2, 3]$ 
4 else
5    $k_1 = 2$ 
6    $\mathcal{C}_2 = [1, 3, 4], \mathcal{C}_3 = [3, 1, 3], \mathcal{C}_4 = [4, 4, 1]$ 
7 end
8  $\delta_{min} = \infty$ 
9 for  $i$  from 1 to 3 do
10   $k_2 = \mathcal{C}_2[i], k_3 = \mathcal{C}_3[i], k_4 = \mathcal{C}_4[i]$ 
11   $\delta = |\mathbf{h}_{k_1}^* \mathbf{h}_{k_2}| / (\|\mathbf{h}_{k_1}\| \|\mathbf{h}_{k_2}\|)$ 
12  if  $\delta < \delta_{min}$  then
13     $\delta_{min} = \delta$ 
14     $\mathbf{P} = [\mathbf{e}_{k_1}, \mathbf{e}_{k_2}, \mathbf{e}_{k_3}, \mathbf{e}_{k_4}]$ 
15  end
16 end

```

Figure 2. Ordering Algorithm

the  $\beta$  closest points on the real axis to  $\Re\{\hat{x}_1\}$  and imaginary axis to  $\Im\{\hat{x}_1\}$ , respectively, where  $\beta$  is assumed to be an odd integer and equal to  $\lfloor \ell^{1/4} \rfloor$ . These  $\beta$  points along the real axis and imaginary axis form a square grid of  $\beta^2$  candidates which form our initial approximation of  $\mathcal{L}_1$ . If  $\sqrt{\ell}$  is larger than  $\beta^2$  then additional points are enumerated beyond this grid. We propose to enumerate additional points by simply searching further along the real and imaginary axis for  $\hat{\mathbf{x}}_1$ .

Fig. 3 depicts the list  $\mathcal{L}_1$  for a decision  $\hat{x}_1 = 3 - 3j$  when  $\hat{x}_1$  is an (a) 16-QAM corner point and (b) 64-QAM interior point and  $\beta = 3$ . The decision  $\hat{x}_1$  is denoted using a gray square and the rest of the list is denoted using solid circles. Points not in the list are denoted using white circles. If  $\sqrt{\ell} = 11$  in Fig. 3-a then the two additional points beyond  $\beta^2 = 9$  are found by simply enumerating one additional candidate along the real and imaginary axes, respectively. These points are denoted using triangle markers. In Fig. 3-b we enumerate two additional points along both the real and imaginary axes so that  $\sqrt{\ell} = 13$ .

We find  $\mathcal{L}_2$  in exactly the same way as  $\mathcal{L}_1$ . Enumerating all combinations of the  $\sqrt{\ell}$  elements of  $\mathcal{L}_1$  and  $\sqrt{\ell}$  elements of  $\mathcal{L}_2$  yields a list of pairs  $\mathcal{L}_{12}$  for  $\{\hat{x}_1, \hat{x}_2\}$  of size  $\ell$ . Finally, note that the entire process for enumerating  $\mathcal{L}_{12}$  may be accomplished with a lookup table to form  $\mathcal{L}_1$  and also  $\mathcal{L}_2$  followed by a hardware combiner to form all  $\ell$  combinations.

For each of the  $\ell$  candidates in  $\mathcal{L}_{12}$  we observe that:

$$\begin{bmatrix} y_3 - R_{3,1}x_1 - R_{3,2}x_2 \\ y_4 - R_{4,1}x_1 - R_{4,2}x_2 \end{bmatrix} = \begin{bmatrix} R_{3,3} & 0 \\ R_{4,3} & R_{4,4} \end{bmatrix} \begin{bmatrix} x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} \tilde{n}_3 \\ \tilde{n}_4 \end{bmatrix}. \quad (15)$$

From this we can find  $\ell$  vectors  $\mathbf{x} = [\hat{x}_1 \hat{x}_2 \hat{x}_3 \hat{x}_4]^T$  by simple decision feedback for each of the  $\ell$  pairs of  $(\hat{x}_1, \hat{x}_2)$ :

$$\begin{aligned} \hat{x}_3 &= \mathcal{Q}((y_3 - R_{3,1}\hat{x}_1 - R_{3,2}\hat{x}_2)/R_{3,3}) \\ \hat{x}_4 &= \mathcal{Q}((y_4 - R_{4,1}\hat{x}_1 - R_{4,2}\hat{x}_2 - R_{4,3}\hat{x}_3)/R_{4,4}). \end{aligned} \quad (16)$$

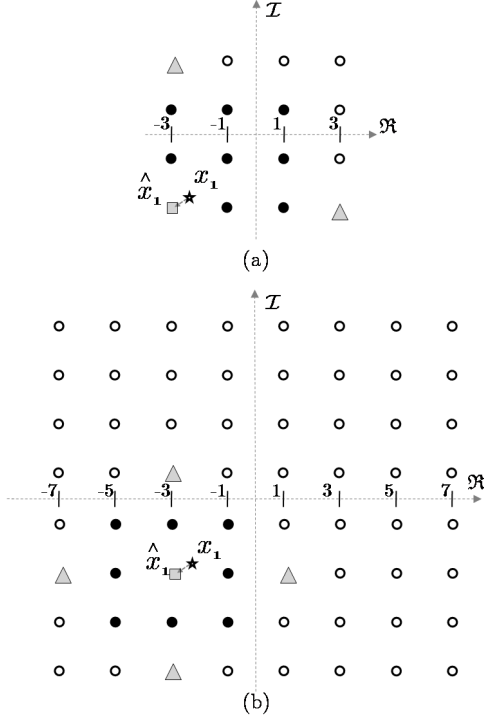


Figure 3. Enumerated list when  $\hat{x}_1 = 3 - 3j$  and  $\beta = 3$  for (a) 16-QAM with  $\sqrt{\ell} = 11$  and (b) 64-QAM with  $\sqrt{\ell} = 13$ .

A summary of the proposed algorithm is provided in Fig. 4. The inputs are the received vector  $\mathbf{r}$ , the effective channel matrix  $\mathbf{H}$  and the list length  $\ell$ . The output is soft-output information for each bit of the transmitted information vector.

### C. Quantifying Complexity

We measure computational complexity using real operations, which can be multiplies, additions, comparisons, quantizers, and divisions. We assume complex multiplies require 4 real multiplies and 2 real additions, and that complex additions require 2 real additions. The computational complexity of the proposed algorithm can be separated into the preprocessing step, which is an ordered QR decomposition (line 1 from Fig. 4), and a core processing step. The additional complexity of the ordering algorithm over a QR decomposition is 48 real operations, where 3 of the operations are real divides.

The core processing begins by inverting  $\mathbf{R}$ , multiplying  $\mathbf{R}^{-1}\mathbf{y}$  and quantizing the resultant vector  $\hat{\mathbf{x}}$  (line 3 from Fig. 4). The computation of  $\mathbf{R}^{-1}$  requires 104 real operations where 4 of the operations are real divides. As  $\mathbf{R}^{-1}$  is triangular, the multiplication  $\mathbf{R}^{-1}\mathbf{y}$  requires 44 real operations. Then the quantization to  $\hat{\mathbf{x}}$  requires 8 real quantizers and so line 3 requires 156 real operations for the entire information vector or  $156/2 = 78$  real operations per signaling interval.

The enumeration of the list  $\mathcal{L}_{12}$  (lines 4-11) can be accomplished with a lookup table and a dedicated hardware combiner used to form all combinations of the lists for each symbol, i.e.  $\mathcal{L}_{12} = \{(\mathcal{L}_1[m], \mathcal{L}_2[n])\} \forall m \in \{1, \dots, \sqrt{\ell}\}$  and  $\forall n \in \{1, \dots, \sqrt{\ell}\}$ . The remaining steps (lines 13-17)

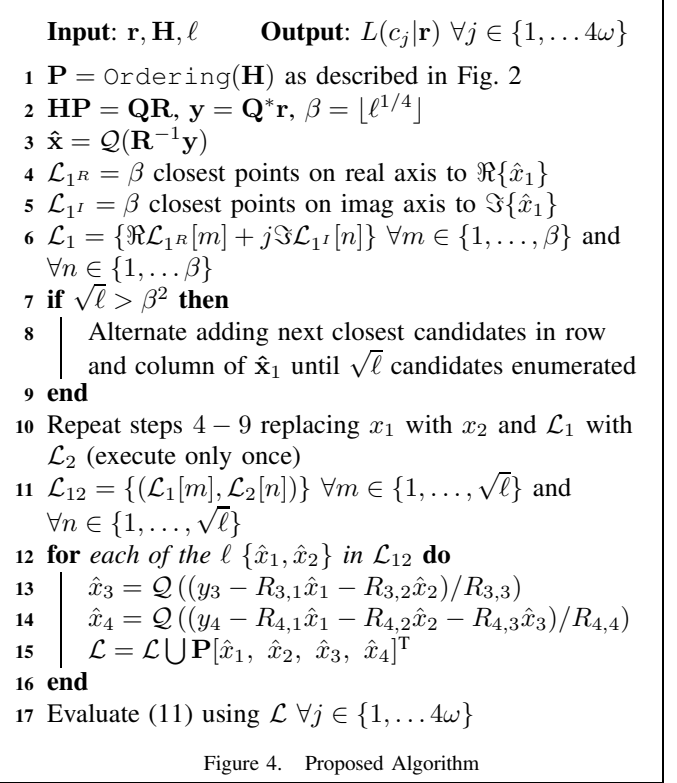


Figure 4. Proposed Algorithm

involve decision feedback, permutation and the computation of the soft-output information, where many of the intermediate variables used in the decision feedback computation may be reused for computation of the soft-output information. Line 13 requires 18 real operations (2 complex multiplies, 2 complex adds and 2 real divides) and line 14 requires 26 real operations (3 complex multiplies, 3 complex additions and 2 real divides) for a total of  $44\ell$  real operations per information vector.

In total lines 13 – 14 require  $44\ell$  real operations per information vector. Computing the cost for each of the  $\ell$  candidates in line 17, assuming reuse of computations between lines 13 – 14 and line 17, requires an additional  $44\ell$  real operations per information vector for a total of  $44\ell + 44\ell = 88\ell$  real operations per signaling interval. Consequently, the total number of real operations for the core processing plus the overhead required by the ordering algorithm over that of a QR decomposition, for one signaling interval, is  $78 + 44\ell$ .

## V. RESULTS

We now present results for the proposed algorithm. We assume the channel does not change during the duration of an entire frame and that the channel matrix entries are drawn anew with the transmission of each new frame. A convolutional code with code polynomial [133 171] and constraint length 7, punctured to code rate 3/4 is employed and the information block size (including tail bits) is 3456. Performance is measured in terms of the  $E_b/N_0$  in dB required to achieve a frame error rate (FER) of  $10^{-2}$ .

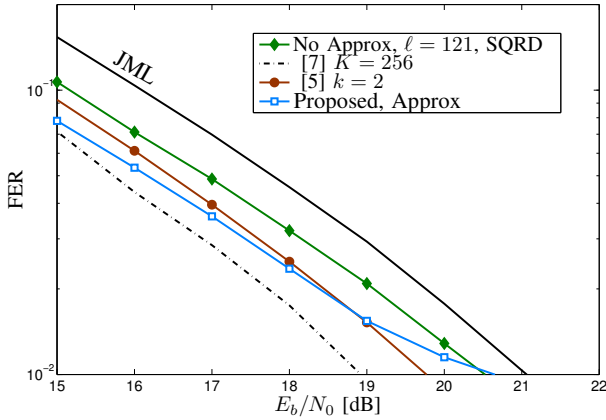


Figure 5. Coded performance for  $2 \times 2$  MIMO system (16-QAM).

Fig. 5 depicts performance results for a system employing 16-QAM transmission for  $\ell = 121$ . Also depicted is the joint maximum-likelihood (JML) detector and the K-best detector [7] employing MMSE-SQRD preprocessing for  $K = 256$ . We observe that the proposed algorithm with  $\ell = 121$  is 1.7 dB worse than the K-best detector, where the K-best detector requires over 3 times the complexity, in terms of real operations. Also depicted in Fig. 5 is the the G-LORD detector [5] with parameter  $k = 2$ , i.e. full enumeration at the first two layers which implies  $\ell = 256$ . The proposed algorithm is 0.9 dB worse than the G-LORD detector with  $\ell = 256$ , but the G-LORD detector is over 80% more complex. We also provide a curve listed as “No approx,  $\ell = 121$ , SQRD” demonstrating the significant impact of our proposed ordering algorithm. The curve was found by forming  $\mathcal{L}_{12}$  with the 11 *best* decisions, relative to the received signal  $\mathbf{r}$ , for  $x_1$  and  $x_2$ , respectively.

Fig. 6 depicts performance results for our proposed algorithm in a system employing 64-QAM transmission and  $\ell = 169$ . Also depicted are the JML detector and the K-best detector [7] employing MMSE-SQRD preprocessing for  $K = 512$ . We observe that the proposed algorithm with  $\ell = 169$  is 1.2 dB worse than the K-best detector, where the K-best detector requires over 120 times more real operations! Also depicted is the the G-LORD detector [5] with parameter  $k = 2$ , which implies  $\ell = 4096$ . The proposed algorithm is 0.8 dB worse than the G-LORD detector, but the G-LORD detector is over 18 times more complex, in terms of real operations.

Table V depicts the  $E_b/N_0$  required to achieve a FER of  $10^{-2}$  and the computational complexity for [5], [7] and the proposed algorithm, where complexity is measured in terms of the number of real operations performed per signaling interval. The complexity of the SQRD is ignored for all algorithms, but for the proposed algorithm we include the additional overhead of our ordering algorithm in the computational complexity. Table V supports the claim that the proposed algorithm has a desirable performance-complexity trade-off. The significant savings in complexity achieved by the proposed algorithm for 64-QAM transmission implies that the proposed algorithm is

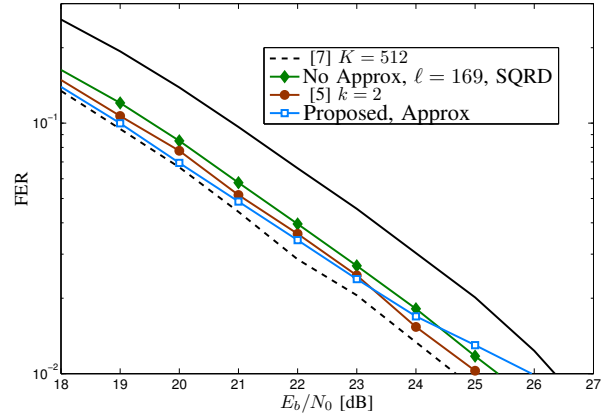


Figure 6. Coded performance for  $2 \times 2$  MIMO system (64-QAM).

16-QAM			
Algorithm	Prop.	[5]	[7]
$\ell$	121	256	256
$E_b/N_0$ [dB] for FER $10^{-2}$	20.6	19.8	18.9
Real Ops.	5402	9032	16738
64-QAM			
Algorithm	Prop.	[5]	[7]
$\ell$	169	4096	512
$E_b/N_0$ [dB] for FER $10^{-2}$	25.9	25.1	24.7
Real Ops.	7514	137440	931587

well-suited for higher-order modulation schemes.

## VI. CONCLUSIONS

In this work we presented an efficient algorithm for soft-output detection of the golden code. The proposed detector employs a linear equalization step followed by a simple list enumeration algorithm for the first two detected symbols. Results indicate a more desirable performance-complexity tradeoff than previous approaches and that the proposed algorithm is particularly attractive for 64-QAM transmission.

## REFERENCES

- [1] S. M. Alamouti, “A simple transmit diversity technique for wireless communication,” *IEEE Journal on Selected Areas in Communications*, vol. 6, pp. 1451–1458, Oct. 1998.
- [2] V. Tarokh, N. Seshadri, and A. R. Calderbank, “Space-Time Codes for High Data Rate Wireless Communication: Performance Criterion and Code Construction,” *IEEE Transactions on Information Theory*, vol. 44, pp. 744–765, Mar. 1998.
- [3] J.-C. Belfiore, G. Rekaya, and E. Viterbo, “The Golden Code: A 2x2 full rate Space-Time Code with Non Vanishing Determinants,” *IEEE Transactions on Information Theory*, vol. 51, 2005.
- [4] P. Dayal and M. K. Varanasi, “An Optimal Two Transmit Antenna Space-Time Code And Its Stacked Extensions,” *IEEE Transactions on Information Theory*, vol. 51, pp. 4348–4355, Dec. 2005.
- [5] C. Shen, M. Fitz, and M. Sidi, “Generalized Soft-Output Layered Orthogonal Lattice Detector for Golden Code,” in *IEEE Wireless Communications and Networking Conference, (WCNC’07)*, Mar. 2007, pp. 525–529.
- [6] D. Wübben, R. Böhnke, V. Kühn, and K. D. Kammeyer, “MMSE extension of V-BLAST based on sorted QR decomposition,” in *Proceedings of the IEEE Semiannual Vehicular Technology Conference (VTC2003-Fall)*, Orlando, USA, Oct. 2003.
- [7] K. W. Wong, C. Y. Tsui, R. S. K. Cheng, and W. H. Mow, “A VLSI architecture of a K-best lattice decoding algorithm for MIMO channels,” in *IEEE International Symposium on Circuits and Systems (ISCAS’02)*, vol. 3, 2002, pp. 273–276.