Transactions Letters

Uniquely Decodable Codes with Fast Decoder for Overloaded Synchronous CDMA Systems

Omid Mashayekhi, *Student Member, IEEE*, and Farokh Marvasti, *Senior Member, IEEE*

***Abstract*—In this paper, we introduce a new class of signature matrices for overloaded[1] synchronous CDMA systems that have a very low complexity decoder. While overloaded systems are more efficient from the bandwidth point of view, the Maximum Likelihood (ML) implementation for decoding is impractical even for moderate dimensions. Simulation results show that the performance of the proposed decoder is very close to that of the ML decoder. Indeed, the proposed decoding scheme needs neither multiplication nor addition and requires only a few comparisons [1]. Furthermore, the computational complexity and the probability of error vs. Signal to Noise Ratios (SNR) are derived analytically.**

***Index Terms*—Coding, code division multiple access (CDMA), overloaded codes, uniquely decodable codes, fast decoding, computational complexity, bit-error rate (BER).**

I. Introduction

CODING schemes for overloaded synchronous Code Division Multiple Access (CDMA) originates from [2]. These codes have fewer chips than the number of users and use relatively fast recursive decoders. Nevertheless, the decoder fails to work in noisy environments, which is an intrinsic characteristic of any CDMA system. While some other overloaded codes have been proposed in [3]-[6], they are all designed for noiseless channels. In fact, in noisy channels, they need an ML decoder to determine the received vector, a process which is NP-hard.

Recently, in [7] and [8], a class of binary matrices for overloaded CDMA systems with simplified ML decoders have been proposed. Moreover, some overloaded matrices with finite signature alphabets are introduced in [9], for which a simplified ML decoder proposed in [7] is applied.

In this paper, we introduce a recursive matrix construction method for highly overloaded synchronous CDMA systems. While the overloading factor[2], β, increases for a sequence of these matrices, they remain uniquely decodable. Moreover, they are designed so that the user data can be extracted at the receiver end with a very simple decoder which uses only a few comparisons. Although this decoder has much lower complexity than the ML one, its performance is almost as good as the performance of the ML decoder [1]. These novel matrices are called Logical Signature Matrices (LSM).

Section II proposes the LSMs and their corresponding recursive matrix construction method. In section III, the fast logical decoder is introduced. Section IV considers analytical results for the probability of error and computational complexity. Simulation results are discussed in section V. Conclusions and future studies are in section VI.

II. Logical Signature Matrix (LSM)

In a synchronous CDMA system, each user multiplies an allocated chip by its data and sends it through the channel. These vectors from different users are added up in the channel and the noisy sum reaches the receiver end. In a system with n users and m chips, let $\mathbf{S}_{m\times n}$ be a signature matrix where the columns are the user vectors. Thus, this kind of CDMA can be modeled as

$$Y_{m\times 1} = \mathbf{S}_{m\times n} X_{n\times 1} + N_{m\times 1} \tag{1}$$

where X is a column vector containing user data, N is the channel noise vector, and Y is the the vector at the receiver end. In (1), it is assumed that there is a perfect power control. Now, let us define an LSM.

Definition 1: Let $\{\lambda_1, \lambda_2, \ldots, \lambda_j\}$ be a set of j algebraically independent numbers[3] and $\bar{\lambda}$ be a linear combination of λ_is. In the following, it is assumed that the set of M input symbols $\Psi = \{\xi_1, \ldots, \xi_M\}$ is a subset of $\{\lambda_1, \lambda_2, \ldots, \lambda_j, \bar{\lambda}\}$. In addition, let $\hat{Y}_{(m-1)\times 1} = \hat{\mathbf{S}}_{(m-1)\times n} X_{n\times 1}$, where $\hat{\mathbf{S}}$ is derived by eliminating one of the rows of $\mathbf{S}_{m\times n}$.

$\mathbf{S}_{m\times n}$ is said to be LSM over the input set Ψ, if the following constraints hold:

1) $\mathbf{S}$ is one-to-one over Ψ.
2) If the number of different symbols in data vector $\mathbf{X}$ is known, it is possible to decipher the user data from $\hat{Y}$ uniquely.

Paper approved by L. Yang, the Editor for Ultra Wideband of the IEEE Communications Society. Manuscript received August 3, 2011; revised March 30, 2012.

O. Mashayekhi was with the Advanced Communications Research Institute (ACRI) and the Department of Electrical Engineering, Sharif University of Technology, Tehran, Iran. Currently, he is a graduate student in the Electrical Engineering Department, Stanford University, Stanford, California 94305, USA (e-mail: omidm@stanford.edu).

F. Marvasti is with the Advanced Communications Research Institute (ACRI) and the Department of Electrical Engineering, Sharif University of Technology, Tehran, Iran. Currently, he is an Honorary Senior Research Associate at the Electrical Engineering Department, University College London, London, UK (e-mail: marvasti@sharif.edu).

Digital Object Identifier 10.1109/TCOMM.2012.081012.110178

[1] Overloaded CDMA implies that the number of users is greater than the number of chips.

[2] Overloading factor is defined to be the number of users divided by the number of chips.

[3] In an algebraically independent set the linear combinations of the numbers with integer coefficients cannot become zero.

A. Recursive Matrix Construction

Now, we introduce a recursive method for constructing uniquely decodable codes. Starting from an LSM, $\mathbf{S}^1_{(m_1 \times n_1)}$, the following recursive relation defines a sequence of matrices. The k^{th} generated matrix $\mathbf{S}^k$ is an $m_k \times n_k$ matrix formed as follows:

$$\mathbf{S}^k = \begin{bmatrix} +\alpha_k & \dots & +\alpha_k & +\alpha_k & +\alpha_k & \dots & +\alpha_k \\ +\beta_k & \dots & +\beta_k & 0 & -\beta_k & \dots & -\beta_k \\ & & & 0 & & & \\ & \hat{\mathbf{S}}^{k-1} & & 0 & & \mathbf{0} & \\ & & & \vdots & & & \\ & \mathbf{0} & & 0 & & \hat{\mathbf{S}}^{k-1} & \\ & & & 0 & & & \end{bmatrix} \tag{2}$$

where $m_k = 2^{k-1}m_1$, $n_k = 2^{k-1}(n_1 + 1) - 1$, and $\hat{\mathbf{S}}^{k-1}$ is derived by eliminating the first row of $\mathbf{S}^{k-1}$; α_k and β_k are two arbitrary numbers. It can be seen that the loading factor increases for the sequence of matrices and approaches $(n_1 + 1)/m_1$ in infinity. In the following theorem, we will show that the resultant matrices are also LSM.

Theorem 1: By starting from a small $m_1 \times n_1$ LSM and by using the recursive algorithm in (2), the resultant sequence of matrices are also LSM.

Proof: The proof is based on induction on k. Assume that $\mathbf{S}^{k-1}$ is LSM and $Y = \mathbf{S}^k X$. Let $\Psi = \{\xi_1, \dots, \xi_M\}$ be the set of M input symbols as defined in Definition 1 and, N_i be the number of ξ_i in the data vector $X_{n\times 1}$. It is easy to show that

$$\sum_{i=1}^{M} N_i = n \quad , \quad \sum_{i=1}^{M} N_i \xi_i = Y_1/\alpha_k \tag{3}$$

Because there are at least $M - 1$ algebraically independent ξ_is, it is possible to find N_is uniquely from (3). On the other hand, let N_{Fi} and N_{Li} be the number of ξ_i in the first and last n_{k-1} elements of X, respectively. Then

$$\sum_{i=1}^{M} (N_{Fi} - N_{Li}) = 0 \ , \ \sum_{i=1}^{M} (N_{Fi} - N_{Li})\, \xi_i = Y_2/\beta_k \tag{4}$$

By just the same argument, it can be deduced that $(N_{Fi} - N_{Li})$ can be determined uniquely from (4). Note thet If the middle element of X is ξ_z, then $N_{Fi} - N_{Li} \equiv N_i - 1$ (mod2) if and only if $i = z$. Thus, it is possible to decipher the middle element of X, say ξ_z. Furthermore, $N_{Fz} + N_{Lz} = N_z - 1$ and $N_{Fi} + N_{Li} = N_i$ for $i \neq z$.

In summary, from Y_1 and Y_2, the middle element of X can be decoded. Moreover, N_{Fi}s and N_{Li}s can be determined from their sum and difference. Since $\mathbf{S}^{k-1}$ is LSM, by considering the second constraint in Definition 1, it can be seen that $Y_3, \dots, Y_{m_k}$ uniquely specify the first and the last n_{k-1} elements of X. Hence, $\mathbf{S}^k$ is one-to-one over the set Ψ. Note that, Y_1 is not needed for the decoding algorithm but rather the number of various symbols of X is required. Thus, the second constraint in Definition 1 also holds and the proof is complete.■

TABLE I
THE FIRST THREE MATRICES IN AN LSM SEQUENCE.

$$\mathbf{S}^1 \quad \begin{bmatrix} +1 & +1 & +1 \\ +1 & 0 & -1 \end{bmatrix}$$

$$\mathbf{S}^2 \quad \begin{bmatrix} +1 & +1 & +1 & +1 & +1 & +1 & +1 \\ +1 & +1 & +1 & 0 & -1 & -1 & -1 \\ +1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & +1 & 0 & -1 \end{bmatrix}$$

$$\mathbf{S}^3 \quad \begin{bmatrix} +1 & +1 & +1 & +1 & +1 & +1 & +1 & +1 & +1 & +1 & +1 & +1 & +1 & +1 & +1 \\ +1 & +1 & +1 & +1 & +1 & +1 & +1 & 0 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ +1 & +1 & +1 & 0 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ +1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & +1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & +1 & +1 & +1 & 0 & -1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & +1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & +1 & 0 & -1 \end{bmatrix}$$

In the following an example will be provided based on Theorem 1.

Example 1: It can be easily shown that

$$\mathbf{S}^1_{2\times 3} = \begin{bmatrix} +1 & +1 & +1 \\ +1 & 0 & -1 \end{bmatrix} \tag{5}$$

is an LSM. By starting from this matrix, the recursive construction with $\alpha_i = \beta_i = 1$ for $i = 1, \dots, k$ from (2) results in a sequence of LSM; the first three sequences are given in Table I. The k^{th} sequence $\mathbf{S}^k$ is a $2^k \times (2^{k+1} - 1)$ matrix and the loading factor approaches 2 as k tends to infinity.■

Theorem 1 suggests a recursive mechanism to construct larger uniquely decodable codes; however, it is not the only way. One common approach in producing larger matrices is by leveraging the characteristics of Kronecker multiplication. For the sake of completeness, it is also considered here as a Note.

Note 1: Given an invertible matrix $\mathbf{P}_{r\times r}$, and an LSM $\mathbf{S}_{m\times n}$, then $\mathbf{P} \otimes \mathbf{S}$ is also an $rm \times rn$ uniquely decodable matrix. By multiplying the received vector by $\mathbf{P}^{-1} \otimes \mathbf{I}$ from the left, r logical decoder can be applied to extract the user data [7].

LSMs have a quality that they can be decoded with a very simple decoder. In the next section, their fast logical decoder will be explained.

III. THE PROPOSED FAST LOGICAL DECODER

We describe the proposed algorithm for fast decoding by considering a system that uses $\{\pm 1\}$ as the input symbols and the ternary class of LSMs based on Example 1 as the encoder (Table I shows the first three matrices).

We first explain the decoder of $\mathbf{S}^1$. In the case that $Y = [y_1, y_2]^T$ and $\hat{X}$ is the decoded data, the decoding scheme has the following steps:

Step 1: Pass y_1 to a quaternary Analog to Digital Converter (ADC) with constellation of $\{\pm 1, \pm 3\}$. The output of this decoder z shows the number of $+1$s and -1s in $\hat{X}$. If $z = +3$ or $z = -3$, then $\hat{X}$ consists purely of $+1$s or -1s, respectively, and the process is terminated. Otherwise, the process goes to the next step.

Step 2: Based on y_2 and z, the decoding continues as follows:

- If $z = +1$, then $\hat{X}$ contains exactly one -1. By passing y_2 to a ternary ADC with constellation of $\{0, \pm 2\}$, it can be determined which user has sent this -1.
- If $z = -1$, then $\hat{X}$ contains exactly one $+1$. By passing y_2 to a ternary ADC with constellation of $\{0, \pm 2\}$, it can be determined which user has sent this $+1$.

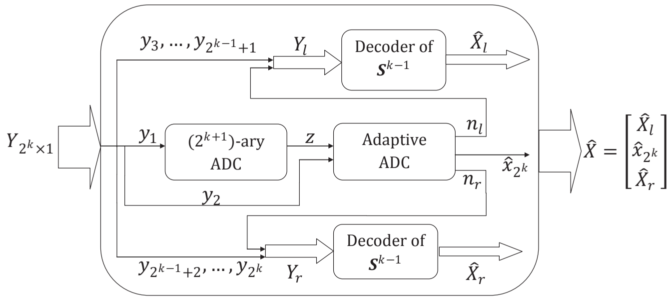

Fig. 1. The proposed logical decoder for $\mathbf{S}^k,\ k \geq 2$.

Now, for $\mathbf{S}^k,\ k \geq 2$, the decoding scheme is recursive. As is depicted in Fig. 1, there are three main steps:

Step 1: Pass y_1 to a (2^{k+1})-ary ADC with a constellation of $\{\pm 1, \pm 3, \pm(2^{k+1}-1)\}$. The output of this decoder z shows the number of $+1$s and -1s in $\hat{X}$. If z is $+(2^{k+1}-1)$ or $-(2^{k+1}-1)$, then the process is terminated and $\hat{X}$ is composed purely of $+1$s or -1s, respectively. Otherwise, the process moves to the next step.

Step 2: Pass y_2 to a $(2^{k+1}-|z|)$-ary ADC with a constellation of $\{0, \pm 2, \pm(2^{k+1}-1-|z|)\}$. By combining the output information of this decoder and the knowledge about the total number of -1s and $+1$s in $\hat{X}$, from the argument in proof of Theorem 1, the $2^{k^{th}}$ entry of $\hat{X}$ (the middle element) can be deciphered. Moreover, one may determine the number of $+1$s and -1s in the first and last $2^k - 1$ entries of $\hat{X}$, denoted by n_l and n_r, respectively.

Step 3: Apply the decoder of $\mathbf{S}^{k-1}$ with the inputs of

$$Y_l = \begin{bmatrix} 2^{k+1}-1-2n_l \\ y_3 \\ \vdots \\ y_{2^{k-1}+1} \end{bmatrix} \text{ and } Y_r = \begin{bmatrix} 2^{k+1}-1-2n_r \\ y_{2^{k-1}+1} \\ \vdots \\ y_{2^k} \end{bmatrix} \tag{6}$$

to identify the first and the last $2^k - 1$ entries of $\hat{X}$.

It is a straightforward extrapolation to extend this method for an M-ary input system. In addition, for any other sequence of matrices, we should just modify the decoder of $\mathbf{S}^1$ and the recursive algorithm remains the same.

In the following section, the analytical performance of this decoder is considered. Specifically, the probability of incorrect decoding and computational complexity are derived. Moreover, we discuss the fairness of these codes.

IV. Performance Analysis

In this section, the analytical performance of the logical decoder is discussed. In the following, we consider the binary input system with entries of $\{\pm 1\}$ and encoding matrices based on Example 1 (see Table I); although, all the results and statements are also applicable to the M-ary input case and any other set of matrices.

Assume that the system uses $\mathbf{S}^k$ and the standard deviation of the channel noise is σ. Let $P_c^k(\sigma)$ be the probability of decoding the received vector correctly using the logical decoder (i.e., $P(\hat{X} = X)$). For simplicity, we define

$$E_\sigma = 0.5 + Q(1/\sigma) \quad \text{and} \quad D_\sigma = 2Q(1/\sigma) \tag{7}$$

where $Q(x) = \int_0^x \frac{1}{\sqrt{2\pi}} e^{-t^2/2} dt$. Hence,$E_\sigma$ is the probability of correctly decoding one of the end points in the one dimensional constellation of the ADC, and D_σ is the probability of correctly decoding one of the middle points.

Let us start by calculating $P_c^1(\sigma)$. Since the inputs are equally likely, one reasonable approach would be to average over all possible input vectors. However, since the decoder is symmetric over ± 1s, one can only consider half of the cases. Specifically, consider the input vectors with no or a single -1 entry (the compliment case would be the vectors with no or a single $+1$). In addition, the probability of correctly decoding $[-1, +1, +1]^T$ or $[+1, +1, -1]^T$ would be $D_\sigma E_\sigma$, since both of them are middle points in the first ADC and end points in the second ADC. By exactly the same argument, the probability of correctly decoding $[+1, +1, +1]^T$ (it does not need the second ADC) and $[+1, -1, +1]^T$ would be E_σ and D_σ^2, respectively. Putting these things together, we will have

$$P_c^1(\sigma) = \frac{1}{4}[E_\sigma + D_\sigma(D_\sigma + 2E_\sigma)] \tag{8}$$

We can expand this approach to the general case. Specifically, for any $k \geq 2$, by classifying different input vectors and by considering the recursive form of the logical decoder, it can be deduced that

$$\begin{aligned} P_c^k(\sigma) \quad = \quad & \frac{1}{2^{2^{(k+1)}-2}} \Big[\overbrace{E_\sigma}^{1} + \overbrace{D_\sigma(D_\sigma + 2E_\sigma)}^{2} \\ & + \overbrace{4D_\sigma(E_\sigma + D_\sigma)\hat{P}_c^{k-1} + 4(D_\sigma \hat{P}_c^{k-1})^2}^{3} \Big] \end{aligned} \tag{9}$$

where

$$\hat{P}_c^{k-1}(\sigma) \quad = \quad \frac{1}{D_\sigma}\Big[2^{(2^k-2)} P_c^{k-1}(\sigma) - E_\sigma\Big] \tag{10}$$

is the modified version of the $P_c^{k-1}(\sigma)$ by excluding the terms resulted from the first ADC in the calculations. This modification comes from the fact that when the recursive decoder applies the previous step decoder, it already knows the number of different symbols in each sub vector, and thus the first ADC is not needed anymore. Also, it is worth mentioning that in(9)

- We are averaging over half of the cases ($\frac{2^{2^{(k+1)}-1}}{2} = 2^{2^{(k+1)}-2}$) due to symmetric behavior of the decoder. In this case, all inputs with at most $(2^k - 1)$ entries of -1.
- The first term takes into account the pure $+1$ input vector, which only needs the first ADC.
- The second term represents three cases; the vector with only one -1 as the middle element and the two vectors with all -1 in the first or last $(2^k - 1)$ entries.
- The last term aggregates all other situations, when the right sub decoder, the left sub decoder, or both of them are needed for the decoding of the input.

Now, let us take a look at the computational complexity of the proposed scheme. Interestingly enough, the logical decoder needs neither any multiplications nor any additions; it only requires a few comparisons with respect to the thresholds of the one dimensional ADC. Let C_k be the average number of comparisons used by the decoder of $\mathbf{S}^k$. Simply, $C_1 = \frac{7+4}{4} =$

2.75 (it becomes more clear shortly). In order to give a general relation for C_k, let us define a couple of factors.

$$F_k = \sum_{i=0}^{2^k-1} \left\{ \binom{2^{(k+1)}-1}{i} (i+1) \right\} \quad (11)$$

$$\begin{aligned} S_k = & \overbrace{\sum_{i=1}^{2^k-1} \left\{ \binom{2^k-1}{\lceil \frac{i-1}{2} \rceil}^2 (i+1)\right.}^{1} \\ & + \overbrace{2 \sum_{j=0}^{\lfloor \frac{i-1}{2} \rfloor} \binom{2^k-1}{j} \binom{2^k-1}{i-j} (2j+1)}^{2} \\ & + \overbrace{\left. 2 \sum_{j=0}^{\lfloor \frac{i-2}{2} \rfloor} \binom{2^k-1}{j} \binom{2^k-1}{i-j-1} (2j+2) \right\}}^{3} \end{aligned} \quad (12)$$

$$\begin{aligned} L_k = & \overbrace{4(2^{2^k-1} - 2)}^{1} \\ & + \overbrace{2 \times \sum_{i=2}^{2^k-1} \left\{ \binom{2^k-1}{\lceil \frac{i-1}{2} \rceil}^2 + 2 \sum_{j=1}^{\lfloor \frac{i-1}{2} \rfloor} \binom{2^k-1}{j} \binom{2^k-1}{i-j} \right.}^{2} \\ & + \overbrace{\left. 2 \sum_{j=1}^{\lfloor \frac{i-2}{2} \rfloor} \binom{2^k-1}{j} \binom{2^k-1}{i-j-1} \right\}}^{2} \end{aligned} \quad (13)$$

In fact, F_k is the number of comparisons that may be needed in the first ADC for all possible input vectors. The key point here is that if an input vector contains i, -1s, then it would be the i^{th} point in the constellation of the first ADC, and thus needs $(i+1)$ comparisons in the first step. Note that due to symmetry, only half of the cases are considered and thus, the dummy variable i iterates over the number of -1's in the input vector up to $(2^k - 1)$. By exactly the same argument, S_k is related to the number of comparisons needed in the second ADC by classifying the input vectors to three different groups; those which are symmetric, those with the middle entry equals to $+1$, and those with -1 as the middle entry. The main point here is that each of these groups lies in a different position in the constellation and hence needs different number of comparisons on the second step. The last term, L_k, shows the number of times the previous step decoder is called. The first term in this expression takes into account the input vectors which may need either the left sub decoder or the right sub decoder, but not both. On the other hand, the second term counts the inputs which may need both left and right sub decoders and thus, they double the number of calls (the coefficient 2 before it).

All in all, we could obtain the following relation in general for any $k \geq 2$

$$C_k = \frac{1}{2^{2^{(k+1)}-2}} \left[F_k + S_k + L_k \times \hat{C}_{k-1} \right] \quad (14)$$

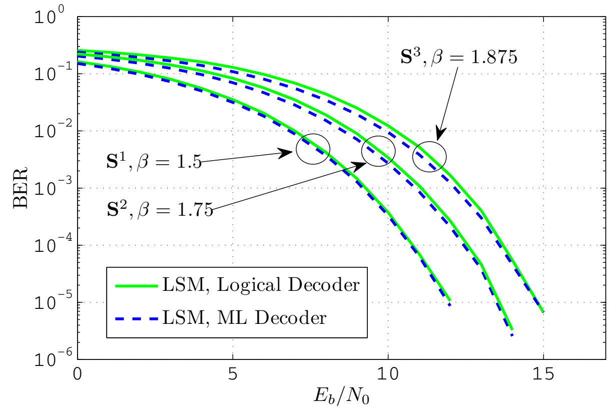

Fig. 2. Comparison of the BER vs. E_b/N_0 for the proposed logical and ML decoders for the three LSMs described in Table I.

where

$$\hat{C}_{k-1} = \frac{1}{2^{(2^k-2)} - 1} \left[2^{(2^k-2)} C_{k-1} - F_{k-1} \right] \quad (15)$$

is the modified version of the C_{k-1} by excluding the number of comparisons resulted from the first ADC in calculations. The reason of the " -1" in the denominator of (15) is that we should exclude the pure $+1$ vector from averaging since it only uses the first ADC. Now, it is easier to justify the previously mentioned value for C_1 since one can see $F_1 = 7$, $S_1 = 4$, and it does not call any sub decoder.

It is worth to classify users based on the number of ±1s in their allocated vectors (columns of the matrix). From this point of view, a system using $\mathbf{S}^k$ contains $k+1$ different groups of users, say $\mathbf{A}_i$, $i = 1, \ldots, k+1$. In the i^{th} group, there are 2^{i-1} users with i number of ±1s in their allocated vector. Because of symmetry in the matrices and their decoders, all users in the same group manifest the same BER in a noisy channel. Moreover, let ϵ_i be the BER of a user in the i^{th} group; then

$$\epsilon_1 > \epsilon_2 > \cdots > \epsilon_{k+1} \quad (16)$$

In other words, although the users with the greater number of ±1s in their allocated vector spend more power for data transmission, they take the advantage of a more reliable communication (i.e., lower BER).

We prove (16) for $k = 1$. ϵ_1 is the probability of deciding in favor of $[x, -1, x]^T$ when $[x, +1, x]^T$ is transmitted and vice versa. On the other hand, ϵ_2 is the probability of deciding in favor of $[-1, x, x]^T$ when $[+1, x, x]^T$ is transmitted and vice versa. While ϵ_1 and ϵ_2 are equally likely for the first ADC, ϵ_1 is more probable for the second ADC. In fact, the existence of 0 in the second column of $\mathbf{S}^1$ makes the distance between the received vectors in the form of $[x, -1, x]^T$ and $[x, +1, x]^T$ shorter than that of the other pair. The same arguments holds for any $k \geq 2$, in a recursive manner.

In the next section, the BER of the logical decoder is compared to the ML one. Moreover, the performance of LSM with the logical decoder is compared to GCO [9], for which a simplified ML decoder has been proposed.

V. Simulation Results

In this section, a BPSK system with different values of E_b/N_0 is simulated. Note that the LSMs are those proposed

TABLE II
COMPUTATIONAL COMPLEXITY OF THE LOGICAL AND ML DECODERS OF LSM WITH SIMPLIFIED ML DECODER OF GCO. ALL MATRICES ARE TERNARY.

Matrix	Decoder		(2×3)	(4×7)	(8×15)
LSM	Logical	Mul.+Add.	None	None	None
		Comparisons	2.75	7.86	21.30
LSM	ML	Mul.+Add.	24	896	491520
		Comparisons	7	123	32768
GCO	ML	Mul.+Add.	18	280	17280
		Comparison	1	7	63

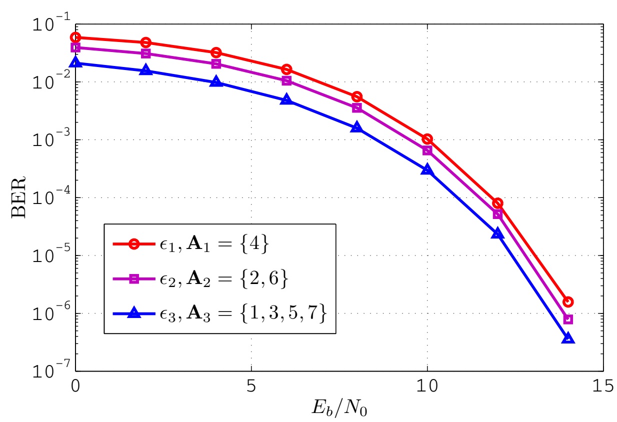

Fig. 3. The BER for different groups of users in a system using $\mathbf{S}^2$.

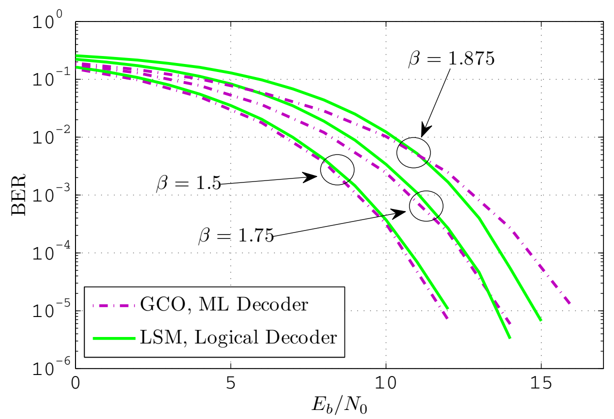

Fig. 4. Comparison of the BER vs. E_b/N_0 for the system using LSM with the proposed logical decoder and the one using GCO with ML decoder. Simulations have been performed for the three LSMs in Table I and GCOs with the same size.

in Table I and input symbols are $\{\pm 1\}$. Fig. 2 compares the BER of the proposed logical decoder versus the ML one for the first three LSMs. The first two rows of Table II show the computational complexity of these decoders. While the ML decoder is rather complex, the logical decoder requires only a few comparisons. Nevertheless, the performance of the ML decoder is slightly better than the logical one.

Fig. 3 shows the BER for three different groups of users in a system using $\mathbf{S}^2$. As one can observe, the more the energy per bit, the less the BER of the user becomes, as in (16).

We have also simulated systems using GCOs of the same size with $\{0, \pm 1\}$ entries [9]. It is noteworthy that the (2×3) GCO is the same as $\mathbf{S}^1$. Fig. 4 shows the results. Although GCO has a simplified ML decoder, it is more complex than the logical decoder (see Table II). In addition, the BER of the LSM with the logical decoder becomes better for moderate values of E_b/N_0.

Please note that, decoding an overloaded system is a general problem of solving a set of underdetermined linear equation, which has been considered extensively in the literature. However, these decoders has way worse performance than the ML decoder. For example, one of the best known non-ML decoder is the iterative decoder which is still way more complex than our decoder and has a poor performance from BER point of view[7].

VI. CONCLUSIONS AND FUTURE STUDIES

In this paper, we have introduced a class of M-ary matrices for overloaded synchronous CDMA systems. These matrices are constructed recursively and while the overloading factor increases for the sequence of these matrices, they remain uniquely decodable. Moreover, the decoding is very fast and requires few comparisons without any multiplications or additions. Simulation results confirm good performance of these types of decoders.

Finding optimum matrices from BER or channel capacity points of view and compatible with the proposed decoder would be valuable for future studies. Extending this method for asynchronous CDMA systems would be another worthwhile research project.

REFERENCES

[1] O. Mashayekhi and F. Marvasti, "Uniquely decodable codes and decoder for overloaded synchronous cdma systems," Apr. 7 2011, U.S. Patent 13/082,084.
[2] S. Chang and E. Weldon, "Coding for T-user multiple-access channels," *IEEE Trans. Inf. Theory*, vol. 25, no. 6, pp. 684–691, 1979.
[3] S. Chang and J. Wolf, "On the T-user M-frequency noiseless multiple-access channel with and without intensity information," *IEEE Trans. Inf. Theory*, vol. 27, no. 1, pp. 41–48, 1981.
[4] T. Fergusen, "Generalized T-user codes for multiple-access channels," *IEEE Trans. Inf. Theory*, vol. 28, no. 5, pp. 775–778, 1982.
[5] S. Chang, "Further results on coding for T-user multiple-access channels (corresp.)," *IEEE Trans. Inf. Theory*, vol. 30, no. 2, pp. 411–415, 1984.
[6] G. Khachatrian and S. Martirossian, "Code construction for the-user noiseless adder channel," *IEEE Trans. Inf. Theory*, vol. 44, no. 5, pp. 1953–1957, 1998.
[7] P. Pad, F. Marvasti, K. Alishahi, and S. Akbari, "A class of errorless codes for over-loaded synchronous wireless and optical CDMA systems," *IEEE Trans. Inf. Theory*, vol. 51, no. 6, pp. 2705–2715, 2009.
[8] R. Learned, A. Willsky, and D. Boroson, "Low complexity optimal joint detection for oversaturated multiple access communications," *IEEE Trans. Signal Process.*, vol. 45, no. 1, pp. 113–123, 1997.
[9] K. Alishahi, S. Dashmiz, P. Pad, and F. Marvasti, "Design of signature sequences for overloaded CDMA and bounds on the sum capacity with arbitrary symbol alphabets," *IEEE Trans. Inf. Theory*, vol. 58, no. 3, pp. 1441–1469, Mar. 2012.