Low-Complexity Decoder for Overloaded Uniquely Decodable Synchronous CDMA

MICHEL KULHANDJIAN1, Senior Member, IEEE, HOVANNES KULHANDJIAN2, Senior Member, IEEE, CLAUDE D’AMOURS1, Member, IEEE, HALIM YANKOMEROGLU3, Fellow, IEEE, DIMITRIS A. PADOS4, Senior Member, IEEE and GURGEN KHACHATRIAN5, Member, IEEE

1School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, Ontario, K1N 6N5, Canada (e-mail: mkk6@buffalo.edu,cdamours@uottawa.ca)
2Department of Electrical and Computer Engineering, California State University, Fresno, Fresno, CA 93740, U.S.A. (e-mail: hkulhandjian@csufresno.edu)
3Department of Systems & Computer Engineering, Carleton University, Ottawa, Canada (e-mail: halim@sce.carleton.ca)
4Computer & Electrical Engineering & Computer Science & I-SENSE Center, Florida Atlantic University, U.S.A. (e-mail: dpados@fau.edu)
5College of Science & Engineering, American University of Armenia, Armenia, (e-mail: gurgenkh@aua.am)

Corresponding author: Michel Kulhandjian (e-mail: mkk6@buffalo.edu).

ABSTRACT We consider the problem of designing a low-complexity decoder for antipodal uniquely decodable (UD) errorless code sets for overloaded synchronous code-division multiple access (CDMA) systems, where the number of signals $K_{\text{max}}$ is the largest known for the given code length $L$. In our complexity analysis, we illustrate that compared to maximum-likelihood (ML) decoder, which has an exponential computational complexity for even moderate code lengths, the proposed decoder has a quasi-quadratic computational complexity. Simulation results in terms of bit-error-rate (BER) demonstrate that the performance of the proposed decoder has only a $1-2$ dB degradation in signal-to-noise ratio (SNR) at a BER of $10^{-3}$ when compared to ML. Moreover, we derive the proof of the minimum Manhattan distance of such UD codes and we provide the proofs for the propositions; these proofs constitute the foundation of the formal proof for the maximum number users $K_{\text{max}}$ for $L = 8$.

INDEX TERMS Uniquely decodable (UD) codes, overloaded CDMA, overloaded binary and ternary spreading spreading codes.

NOMENCLATURE

- 5G The fifth generation
- 6G The sixth generation
- AWGN Additive white Gaussian noise
- BER Bit-error-rate
- BPSK Binary phase-shift keying
- CDM Code-division multiplexing
- CDMA Code-division multiple access
- eMBB Enhanced mobile broadband
- FDA Fast detection algorithm
- FDMA Frequency division multiple access
- LDS Low-density spreading
- MAI Multiple-access interference
- MF Matched filter
- MIMO Multiple-input multiple-output
- ML Maximum-likelihood
- MMSE Minimum mean-square estimation
- mMTC Massive machine-type-communications
- mmWave Millimeter-wave
- MO Multiple-orthogonal
- NDA Noiseless detection algorithm
- NOMA Non-orthogonal multiple access
- OMA Orthogonal multiple access
- PDA Probabilistic data association
- PIC Parallel interference cancellation
- PN Pseudo-noise
- SCMA Sparse code multiple access
- SIC Successive code multiple access
- SNR Signal-to-noise ratio
- TDMA Time-division multiple access
- UD Uniquely decodable
- uRLLC Ultra-reliable low-latency communications
- ZF Zero-forcing
I. INTRODUCTION

In the last decade, wireless communication services have experienced explosive growth while communication technologies have progressed generation by generation. In the previous generations spanning from 1G to 4G, the multiple access schemes were mostly characterized by orthogonal multiple access (OMA) techniques, where users are assigned orthogonal resources in either frequency, (frequency-division multiple access (FDMA)), time, (time-division multiple access (TDMA)) or code, (code-division multiple access (CDMA)). CDMA [1] was the basic technology for 3G and for some 2G (IS-95) networks. High spectral- and power-efficiency, massive connectivity and low latency are among the requirements for next generation communications and these requirements are expected to increase in the future, as researchers turn their efforts towards sixth generation (6G) wireless communications. Enhanced mobile broadband (eMBB), ultra-reliable low-latency communications (uRLLC) and massive machine-type communication (mMTC) support a suite of compelling applications driving these requirements. Massive multiple-input multiple-output (MIMO), non-orthogonal multiple access (NOMA) and millimeter-wave (mmWave) communications constitute promising techniques of addressing these stringent requirements [2].

Supporting a large number of users communicating over a common channel may not be readily achievable by OMA techniques due to the multiple-access interference (MAI) in rank-deficient systems, where the number of users is greater than that of the resource blocks. To meet the demand of increased bandwidth efficiency in synchronous CDMA, a CDMA concept was introduced in [3], which can support many more users for a given code length compared to traditional CDMA. A number of signature design schemes have been studied for dense spreading in conventional CDMA, where low cross-correlation sequence sets are designed to minimize the overall MAI, which allows more users to simultaneously access the common channel. This in turn results in increased spectral efficiency. Finding suitable spreading codes and decoding schemes for such overloaded systems is a challenging optimization problem. To address these challenges, numerous non-uniquely decodable (non-UD) [8]–[10], and UD [11]–[35] construction based code sets have been proposed. Examples of such non-UD code sets are pseudo-noise spreading (PN) [4], [5], orthogonal/orthogonal CDMA (O/O), [6], [7], PN/orthogonal CDMA (PN/O) [8], multiple-orthogonal CDMA (MO) [8], improved O/O CDMA [9]. Those codes employ two or more sets of orthogonal signal waveforms, which allows the system to accommodate more users than the signature length $L$. As a consequence of this, a significant level of MAI exists at the output of each user’s matched filter due to the non-zero cross-correlation of different signatures.

Low cross-correlation sequence sets might not be the best criterion for very high rank-deficient systems. One important criterion in such rank-deficient systems is for the code set to be UD. By definition the UD codes are those in which the data of different users can be unambiguously decoded in a noiseless channel using linear recursive decoders [31]. Low-complexity linear decoders were introduced for these UD code sets using either binary $\{0,1\}$, or antipodal $\{\pm1\}$, or alternatively ternary $\{0,\pm1\}$ chips in [27]–[30], [32]–[35]. On the other hand, Lu et al. [36] proposed $M$-ary code sets for the multiple-access adder channel. Various applications have been implementing UD code sets framework such as in multi-way physical-layer network coding conceived in [33], [37].

All of these multiple access concepts were introduced in order to serve a number of excess users beyond the available resources. These multiple access schemes are characterized by NOMA techniques [38] in 5G and beyond wireless communications. Recently, several NOMA solutions have been actively investigated [2], which can be basically divided into two main categories, namely power-domain and code-domain NOMA. A few of the strong contenders of code-domain NOMA are low-density spreading aided CDMA (LDS-CDMA) [39], [40] and sparse code multiple access (SCMA) [41], [42]. LDS-CDMA [39], [40] must generally guarantee to be UD code set [43], which means non-zero Euclidean distance. The design of LDS type matrices offers flexible resource allocation, performs better in terms of handling the MAI that exists in rank-deficient systems and has low-complexity receivers compared to conventional CDMA.

Unlike the dense UD code set, the LDS-CDMA structure can be represented by a factor graph, the classic message passing algorithm (MPA) can be employed for its detection. Due to the reason that UD code sets were originally designed for adder channels, generally a simple noiseless detection is developed. However in practice, the wireless transmission channel exhibits, among other things, selective fading, multipath and the near-far problem, which leads to unequal received power among users. Consequently, if synchronization, channel equalization are compensated, low-complexity detectors can be applied to wireless channels.

Inspired by these attractive features of UD code sets, this paper investigates UD codes for synchronous uplink NOMA. In the case of dispersive fading channels, we can potentially equalize the channel effect by using channel precoding. Therefore, we consider developing a low-complexity detector for UD code sets that are proposed in [31] for overloaded synchronous CDMA systems. It is widely recognized that the complexity of an optimal detector is exponentially proportional to the number of users, which prohibits its practical implementation. Various suboptimal low-complexity detection techniques have been already proposed. These suboptimal approaches can be classified into two categories: linear and non-linear multiuser detectors. Linear multiuser detectors include among others, matched filter (MF), minimum mean-square error (MMSE), and zero-forcing (ZF), etc. In a non-linear subtractive interference cancellation detector the interference is first estimated and...
then it is subtracted from the received signal before detection. The cancellation process can be carried out either successively (SIC) [44], or in parallel (PIC) [45]–[47]. In non-linear iterative detectors [48]–[52], probabilistic data association (PDA) [53] aims to suppress the MAI in each iteration in order to improve the overall error performance. Suboptimal polynomial time detectors that are based on the geometric approach are studied in [54], [55].

In general, linear as well as non-linear detectors cannot separate users in overloaded systems even in the case of asymptotically vanishing noise. Therefore, the spreading codes must have property such that decoding can achieve asymptotically zero probability of error multiuser detection when the signal-to-noise (SNR) ratio becomes arbitrary large. The UD class of codes that guarantee “errorless” communication in an ideal (noiseless) synchronous CDMA/code-division multiplexing (CDM) also shows a good performance in the presence of noise.

Finding the overloaded UD class of codes for noiseless channel is directly related to coin-weighing problem, one of the problems that is discussed by Erdős and Rényi in [11]. It can be considered as a special case of a general problem where authors in [12], [13], [16], [25], [26] refer to them as detecting matrices. Lindström in [15] defines the same problem as the detecting set of vectors. Given an integer \( q \geq 2 \) and a finite set alphabet \( \mathcal{M} \) of rational integers, let \( v_k \) for \( 1 \leq k \leq K \) be \( L \)-dimensional (column) vectors with all components from \( \mathcal{M} \) such that the \( q^K \) sums

\[
\sum_{k=1}^{K} v_k \epsilon_k \quad (\epsilon_k = 0, 1, 2, \ldots, q - 1)
\]

are all distinctly unique, then \( \{v_1, \ldots, v_K\} \) are detecting set of vectors. Let \( F_q(L) \) be the maximal number of \( L \)-dimensional vectors and \( f_q(K) \) be the minimal vector length to form a detecting matrix for a given length \( L \) and a number of vectors \( K \). The problem of determining \( f_q(K) \) as a special case when \( q = 2 \), \( \mathcal{M} = \{0, 1\} \) that can be equivalently expressed as a coin-weighing problem: what is the minimal number of weighings on an accurate scale to determine all false coins in a set of \( K \) coins. The choice of coins for a weighing must not depend on results of previous weighings.

This problem was first introduced by Söderberg and Shapiro [12] for \( K = 5 \). The minimal number of weighings, \( L \), has only been found for a few different values of \( K \) in [17]. However, Lindström gives an explicit construction of \( L \times \gamma(L + 1) \) binary (alphabet \{0, 1\}) and \( L \times \gamma(L + 1) \) antipodal (alphabet \{±1\}) detecting matrices [14], where \( \gamma(L) \) is the number of ones in the binary expansion of all positive integers less than \( L \). He also proved that the lower bound in the case of \( \mathcal{M} = \{0, 1\} \) or \{±1\} is

\[
\lim_{K \to \infty} \frac{f_2(K) \log K}{K} = 2.
\]

\(^1\)As an example, \( \gamma(8) = 12 \).

Cantor and Mills [16] constructed a class of \( 2^i \times (i+1)2^{(i-1)} \) ternary (alphabet \{0, ±1\}) detecting matrices for \( i \in \mathbb{Z}^+ \), which implies that in the case of \( \mathcal{M} = \{0, ±1\} \) the lower bound is

\[
\lim_{K \to \infty} \frac{f_3(K) \log K}{K} \leq 2.
\]

In the literature, most of the explicit construction of UD code sets are recursive [18]–[25]. To the best of our knowledge, it is worth mentioning that the maximum number of vectors of the explicit constructions of binary, antipodal and ternary code sets are \( K_{\text{max}}^b = \gamma(L + 1) \), \( K_{\text{max}}^a = \gamma(L) + 1 \) and \( K_{\text{max}}^t = (i + 2)2^{(i-1)} \), as shown in Table 1, Table 2 and Table 3 respectively. Several authors have proposed decoders with linear complexity in noiseless scenarios for the explicit construction, where the detecting matrix has the known-to-us maximum number of vectors, \( K_{\text{max}} \).

For noisy channels, recently in [28], a class of antipodal code sequences, for overloaded CDM systems with simplified two-stage maximum-likelihood (ML) detection, has been proposed. In addition to that, other overloaded matrices over the ternary alphabet are introduced in [29] with a low-complexity decoding algorithm. Similarly, in [30] the authors propose overloaded code sets over the ternary alphabet that has a twin tree structured cross-correlation hierarchy that can be decoded with a simple multi-stage detector. Yet another construction of ternary codes that increases the number of columns, \( K \), of UD codes for a such length compared to those proposed in [29] and [30] with a low-complexity polynomial time decoder is proposed in [32]. The primary reason for such low-complexity decoders is that the code sets are constructed with a certain criteria, which entails lowering the maximum number of users \( K < K_{\text{max}} \), as shown in Table 3.

Apart from, binary, antipodal and ternary UD spreading codes higher alphabet \( k \)-ary spreading codes were studied by Lu et al. [56] for the multiple-access adder channels.

In this work, for the first time we consider the problem of designing a low-complexity decoder that has a complexity of \( O(LK \log_2(K)) \) for UD code sets in [31] having the maximum number of users \( K_{\text{max}}^a \). The code sets presented in [31] are also recursive, which make use of a linear map between vector spaces to Galois field extensions. These UD code sets are one possible construction of all possible distinct UD code sets, shown in Table 2. Simulation results in terms of bit-error-rate (BER) demonstrate that the proposed decoder has a degradation of only 1–2 dB in SNR compared to the ML detector at a BER of 10\(^{-3}\).

Our contributions are summarized as follows:

1. (1) We present the proofs for the important \textit{Propositions 1-4} that are presented for the first time in [31]. Those new prepositions are actually the broken down versions of the unique decodability (UD) property.

2. (2) Moreover, for the first time, based on \textit{Propositions 1-4}, we formally prove that the maximum number of users for the given \( L = 8 \) is \( K_{\text{max}}^a = 13 \).
TABLE 1. Binary Codes

| Year  | Authors and Publications | Rows | Columns | Decoder |
|-------|--------------------------|------|---------|---------|
| 1963  | Söderberg and Shapiro    | L    | <γ(L + 1) | No     |
| 1964  | Lindström               | L    | γ(L + 1)  | No     |
| 1966  | Cantor and Mills        | 2^i − 1 | i2^(i−1) | No     |
| 1989  | Martirosian and Khachatrian | L    | γ(L + 1)  | Yes    |
| 2019  | Kulhandjian et al.    | L    | γ(L + 1)  | Yes    |

*† Code set constructions that achieve the maximum number of users \(K_{\text{max}}\) are presented in bold.

TABLE 2. Antipodal Codes

| Year  | Authors and Publications | Rows | Columns | Decoder |
|-------|--------------------------|------|---------|---------|
| 1964  | Lindström               | L    | γ(L + 1)  | No     |
| 1987  | Khachatrian and Martirosian | 2^i | i2^(i−1) + 1 | Yes |
| 1995  | Khachatrian and Martirosian | 2^i | i2^(i−1) + 1 | Yes |
| 2012  | Kulhandjian and Pados  | 2^i | i2^(i−1) + 1 | Yes |

TABLE 3. Ternary Codes

| Year  | Authors and Publications | Rows | Columns | Decoder |
|-------|--------------------------|------|---------|---------|
| 1966  | Cantor and Mills        | 2^i | (1 + 2)2^(1−i) | No   |
| 1979  | Chang and Weldon        | 2^i | (1 + 2)2^(1−i) | Yes   |
| 1982  | Ferguson                | 2^i | (1 + 2)2^(1−i) | Yes   |
| 1984  | Chang                   | 2^i | (1 + 2)2^(1−i) | No    |
| 1998  | Khachatrian and Martirosian | 2^i | (1 + 2)2^(1−i) | Yes   |
| 2012  | Mashayekhi and Marvasti | 2^i | 2^(i+1) − 1 | Yes   |
| 2016  | Singh et al.            | 2^i | 2^(i+1) − 2 | Yes   |
| 2018  | Kulhandjian et al.      | 2^i | 2^(i−1) + 1 | Yes   |
| 2021  | Kulhandjian et al.      | 2^i | 2^(i−1) + 1 | Yes   |

(3) The minimum Manhattan distance is proved to be 4 for the recursive UD code sets in [31].

(4) We develop a low-complexity decoder that has a complexity of \(O(LK \log_2(K))\) for the recursive UD code sets in [31] and the maximum number of users \(K_{\text{max}}\).

(5) We compute the complexity of the deterministic noiseless detection algorithm (NDA) presented in [31] and perform some complexity analysis for the proposed fast (low-complexity) detection algorithm (FDA).

The rest of the paper is organized as follows. The minimum Manhattan distance of code sets in [31] is presented in Section IV, followed by the proofs of the Propositions 1-4 and the maximum number of \(K_{\text{max}}\) for the case of \(L = 8\) in Sections II and III, respectively. Detailed discussion of the FDA is presented in Section V. The complexity analysis for both NDA and FDA algorithms is presented in Section VI. After illustrating simulation results in Section VII, a few conclusions are drawn in Section VIII.

The following notations are used in this paper. All boldface lower case letters indicate column vectors and upper case letters indicate matrices, \((\cdot)^T\) denotes transpose operation, \(\text{mod}\) denotes the modulo operation, \(\text{sgn}\) denotes the sign function, and \(|\cdot|\) denotes cardinality of the set.

II. PROOF OF PROPOSITIONS

In order to facilitate the development of the proof it is beneficial to present the UD code set in [31] which are constructed as \(C = [H_LV_L] \in \{\pm 1\}^{L \times K}\), where \(V_L \in \{\pm 1\}^{L \times (K−L)}\). We recall that the Sylvester-Hadamard matrix of order 2 is \(H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\) and of order \(2^{p+1}\) for \(p = 1, 2, \ldots\) is \(H_{2^{p+1}} = \begin{bmatrix} H_{2^p} & H_{2^p} \\ H_{2^p} & -H_{2^p} \end{bmatrix}\).

Then, for any \(p = 1, 2, \ldots\), \(H_{2^p} = 2^pI_{2^p \times 2^p}\), where \(I_{N \times N}\) is the \(N \times N\) identity matrix. We introduce the notation \(H_4 = [h_0, h_1, h_2, h_3], [a_0, a_1, a_2, a_3] \triangleq [-1, 1, 1, 1]^T, [-1, -1, 1, 1]^T, [1, 1, -1, 1]^T, [1, 1, 1, -1]^T\), and the negation function \(\chi^{-} \triangleq -\chi\). We can see that set of vectors \(G = \{h_0, h_1, h_2, h_3, a_0, a_1, a_2, a_3\}\) together with \(\odot\)

4 VOLUME 5, 2016
operator form a finite group \((G, \circ)\). There exists an isomorphism \(\varphi\), shown in Table 4 from \(G\) to finite additive Abelian group \((\mathbb{F}_{2^2}, +)\) of extended Galois field \(\mathbb{F}_{2^4}\), in other words \(G\) is isomorphic to \((\mathbb{F}_{2^2}, +)\). \((G, \circ) \cong (\mathbb{F}_{2^2}, +)\). From linear algebra we know that there is an isomorphism from finite additive groups \((\mathbb{F}_{p^n}, +)\) to vector fields \((\mathbb{F}_{p^n}, +)\) and to \(\mathbb{Z}_{p^n}\), that is \((\mathbb{F}_{p^n}, +) \cong (\mathbb{Z}_{p^n}, +) \cong \mathbb{Z}_{p^n}\). Table 4 shows the mapping of the vectors \(h_0, \ldots, h_3, a_0, \ldots, a_3\) and its negated forms to elements in \(\mathbb{F}_{2^2}\) with primitive polynomial \(\alpha^2 + \alpha + 1 = 0\), where \(\alpha\) is the primitive element in extended Galois field \(\text{GF}(2^4)\). Notice carefully that operation of the finite group \(G\) is \(\circ\), whereas the finite additive group \(\mathbb{F}_{2^2}\) is +.

**TABLE 4. Isomorphism \(\varphi : G \rightarrow \mathbb{F}_{2^2}\)**

| Antipodal | Polynomial | Power |
|-----------|------------|-------|
| \(h_0\)   | 0          | 0     |
| \(h_1\)   | 1          | 1     |
| \(h_2\)   | \(\alpha\) | \(\alpha^2\) |
| \(h_3\)   | \(\alpha^3\) | \(\alpha^3\) |
| \(h_i\)   | \(\alpha\) | \(\alpha^3\) |
| \(h_j\)   | \(\alpha\) | \(\alpha^3\) |
| \(h_k\)   | \(\alpha\) | \(\alpha^3\) |
| \(a_0\)   | 0          | 0     |
| \(a_1\)   | 1          | 1     |
| \(a_2\)   | \(\alpha\) | \(\alpha^2\) |
| \(a_3\)   | \(\alpha\) | \(\alpha^2\) |
| \(a_4\)   | \(\alpha\) | \(\alpha^2\) |
| \(a_5\)   | \(\alpha\) | \(\alpha^2\) |
| \(a_6\)   | \(\alpha\) | \(\alpha^2\) |
| \(a_7\)   | \(\alpha\) | \(\alpha^2\) |
| \(a_8\)   | \(\alpha\) | \(\alpha^2\) |
| \(a_9\)   | \(\alpha\) | \(\alpha^2\) |
| \(a_{10}\) | \(\alpha\) | \(\alpha^2\) |
| \(a_{11}\) | \(\alpha\) | \(\alpha^2\) |
| \(a_{12}\) | \(\alpha\) | \(\alpha^2\) |
| \(a_{13}\) | \(\alpha\) | \(\alpha^2\) |
| \(a_{14}\) | \(\alpha\) | \(\alpha^2\) |

With the above formulation, the columns of \(V_L, F_{\varphi} : \{\pm 1\}^{L \times 1} \rightarrow \mathbb{F}_{2^2}^{1 \times L}, f_i = F_{\varphi}(v_i)\) maps the \(i\)-th column \(v_i\) to \(f_i\), for \(i = 0, \ldots, L = K - 1\), where \(F = [f_0, f_1, \ldots, f_L]^T, f_{i,j} \in \{0, 1, \alpha, \ldots, \alpha^{14}\}\), \(0 \geq j \geq i^T, L' = L/4 = 2^{p-2}\) for \(p \geq 2\). As an example for \(V_8\) the mapped vectors are as two-dimensional vectors \(f_i = [f_{i,0}, f_{i,1}]^T\).

In order to prove Proposition 1-4 that was first presented in [31], we will expose some very interesting claims. One property of the Hadamard matrix is the following, if we replace 1’s in Hadamard matrix with 0’s, and replace the –1’s with 1’s we create \(L - 1\) Hadamard binary channel codes, since the first column results in zero vector [58]. Let the Hadamard columns \(H_L = [h_0, \ldots, h_{L-1}]\) be mapped into linear binary codes \(B_L = [b_0, \ldots, b_{L-1}]\). There is an isomorphism between binary addition and multiplication of \(\{\pm 1\}\) elements. Consequently, since binary addition of Hadamard linear codes is a Hadamard code itself then it is equivalent to element-wise multiplication of any Hadamard codes \(H_L\) is also in \(H_L\). Therefore, some properties can be derived. We denote the mapping of \(h_i\) for \(i = 0, \ldots, L - 1\) vectors into binary \(b_i \in \{0, 1\}^{L \times 1}\) vectors by \(b_i = f(h_i)\) and inverse \(h_i = f^{-1}(b_i)\), where \(H_L = [h_0, \ldots, h_{L-1}]\) and \(B_L = [b_0, \ldots, b_{L-1}]\). Define the \(f(x) = (1 - x)/2\) and the inverse \(f^{-1}(y) = (-1)^{\log_2 y - 1}\) functions. Then, the \(j\)th column corresponds to linear combination of \(b_j = j_1b_1 \oplus j_2b_2 \oplus j_4b_4 \oplus \ldots \oplus j_{2^p}b_{2^p}\), where \(j\) has the binary representation \(j = j_1 + 2j_2 + 4j_4 + \ldots + 2^pj_{2^p}\) [58]. Here \(\oplus\) is modular 2 summation and \(j_i \in \{0, 1\}\) for \(i = \{1, 2, 4, \ldots, 2^p\}\). Therefore, the resulting code is linear code \(b_k = e_0b_0 \oplus e_1b_1 \oplus \ldots \oplus e_{L-1}b_{L-1}\) for all \(e_i \in \{0, 1\}\), where \(b_k \in B_L\).

**TABLE 5. Binary addition and \{±1\} multiplication**

| Binary | Bipolar |
|--------|--------|
| 0 + 0 = 0 | 1 \times 1 = 1 |
| 0 + 1 = 1 | 1 \times 1 = -1 |
| 1 + 0 = 1 | -1 \times 1 = -1 |
| 1 + 1 = 0 | -1 \times -1 = 1 |

Let us look at the problem and assume that \(\text{Null}(C) \subseteq \mathbb{Z}\), where \(\mathbb{Z} = \{0, \pm 1\}^{K \times 1}\) excluding trivial case \(\{0\}^{K \times 1}\) and let \(z \in \mathbb{Z}\). Then the nullspace of \(C\) can be formulated as such

\[
Cz = 0 \quad (4a)
\]

\[
[H_L V] z = 0_L \quad (4b)
\]

\[
H_L z_1 + V_L z_2 = 0 \quad (4c)
\]

\[
H_L z_1 = -V_L z_2 \quad (4d)
\]

where \(z = \begin{bmatrix} z^T_1 \ z^T_2 \end{bmatrix}^T, z_1 \in \{0, \pm 1\}^{L \times 1}, z_2 \in \{0, \pm 1\}^{(K-L) \times 1}\) and \(0 = \{0\}^{L \times 1}\). If we assume \(z_1 = 0\) and for some \(z_2\) [4] is true, then [4] can be expressed in terms of \(V_L\) only as

\[
V_L z_2 = 0. \quad (5)
\]

**Claim 1. Element-wise multiplication of \(H_L\)'s columns \(h_j = e_0 h_0 \oplus e_1 h_1 \oplus \ldots \oplus e_{L-1} h_{L-1}\) for all possible \(0 \leq i \leq L - 1, e_i \in \{0, 1\}\) is true, where \(h_j \in H_L\).**

**Claim 2. If \(V_L\) does not intersect 0 over the non-trivial \(\{0, \pm 1\}\) combinations and [4] is true, then \(\beta_0 v_0 \circ \beta_1 v_1 \circ \ldots \circ \beta_{K-L-1} v_{K-L-1} = \alpha h_j\) must be true for some \(\beta_i \in \{0, 1\}\), \(0 \leq i \leq K - L - 1, \alpha \in \{\pm 1\}\).**

**Claim 3. If [5] is true, then \(\beta_0 v_0 \circ \beta_1 v_1 \circ \ldots \circ \beta_{K-L-1} v_{K-L-1} = \pm 1\) must be true for some \(\beta_i \in \{0, 1\}\), \(0 \leq i \leq K - L - 1\) and [5] is true then it is not necessary that [5] is also true.**

**Claim 4. If \(V_L\) does not satisfy [4] and [5] then multiplying any column of \(V_L\) by \(-1\) will still not satisfy [4] and [5].**

Authors in [31] showed by exhaustive search that in case when \(L = 4\) then \(K_{\max} = 5\) and all the possible candidates are \([-1, 1, 1, 1]^T, [1, -1, 1, 1]^T, [1, 1, -1, 1]^T, [1, 1, -1, 1]^T\) and their negatives according to the Claim 4 Furthermore, we continue to present additional claims. Let \(v_i \in \{\pm 1\}^4\) for \(0 \leq i \leq N - 1\) then,
Claim 5. If \( \beta_0 v_0 \otimes \beta_1 v_1 \otimes \ldots \otimes \beta_{N-1} v_{N-1} = \alpha h_j \) and \( \alpha \in \{ \pm 1 \} \), \( h_j \in H_4 \) then \( V N \in N \), \([v_0, \ldots, v_{N-1}] A_v 1 = H_4 A_H 1 \) is true for some \( \beta_i \in \{ 0, 1 \} \), \( 0 \leq i \leq N - 1 \), \( N' \in N \) is not necessarily equal to \( N \), \( A_v \in \{ 0, \pm 1 \}^{N \times N} \), \( A_H \in \{ 0, \pm 1 \}^{4 \times N'} \), which have maximum one \( \pm 1 \) entry in each column and 0’s elsewhere.

Claim 6. If \( 0 = N \mod 2 \) and \( \beta_0 v_0 \otimes \beta_1 v_1 \otimes \ldots \otimes \beta_{N-1} v_{N-1} = \pm 1 \) then \([v_0, \ldots, v_{N-1}] A_v 1 = 0 \) is true for some \( \beta_i \in \{ 0, 1 \} \), \( 0 \leq i \leq N - 1 \), \( A_v \in \{ 0, \pm 1 \}^{N \times N} \), which have maximum one \( \pm 1 \) entry in each column and 0’s elsewhere, except when \( N = 4 \) and \( v_0 = [-1, 1, 1, 1]^T \), \( v_1 = [1, -1, 1, 1]^T \), \( v_2 = [1, 1, -1, 1]^T \), \( v_3 = [1, 1, 1, -1]^T \).

The generalization of Claims 1-6 formalizes the Propositions 1-4 and hence we presented the proofs.

III. PROOF

For the case when \( L = 8 \), we prove that the maximum number of columns we can append to \( H_8 \) is actually \( K_{max}^a = L = 5 \). Note that all proposed UD or another words one-to-one matrix constructions in literature are \( C \subset C \), where \( C \) is all possible antipodal UD code sets for a given \( L \). In order to prove for the maximum number of possible vectors \( K_{max}^a \), we should look at all possible \( V \) and count how many structure of \( V \) hits any of the forbidden lattice points \( H_m z_1 \) and how many does not. If for a given \( k \) column, we count the number of \( V \) that hits any forbidden lattice points is equal to the total number of possible \( V \) vector set, then we know that maximum number of columns of \( V \) that do not hit any forbidden lattice points should be smaller than \( k \).

First, we transform antipodal vectors into polynomials with integer coefficients \( Z[x] \), \( F : \{ \pm 1 \}^{m \times 1} \to Z[x] \). Those polynomials represent the row location and number of \( -1s \) or \( +1s \) in any antipodal \( v \in \{ \pm 1 \}^{m \times 1} \) with dimension \( m \). Let the polynomial be

\[
G(x) = a_0 x^0 + a_1 x^1 + a_2 x^2 + \ldots + a_{m-1} x^{m-1},
\]

where \( a_i \in Z \). Additions of vector \( v \) in vector space is equivalent to the addition of \( Z[x] \) in polynomial space. Each antipodal vectors, \( v_j \in \{ \pm 1 \}^{m \times 1} \), where \( 0 \leq j \leq 2^{m-1} - 1 \), is mapped to the corresponding polynomials \( G^i_a(x) \) and \( G^i_p(x) \) to represent the \( +1 \) and \( -1 \) of the \( v_j \). As an example, for \( m = 4 \), the antipodal vector, \( v_{10} = [1, -1, -1, -1]^T \), is mapped to \( G^{10}_n(x) = x^1 + x^3 \) and \( G^{10}_p(x) = x^0 + x^2 \) polynomials. Observe that for any antipodal vector \( v_j \), the addition of polynomials, \( G^i_a(x) + G^i_p(x) = x^0 + x^1 + \ldots + x^{m-1} \). In order to visualize polynomial additions in 2-dimensional Euclidean space we can further transform \( Z[x] \) vector space into \( A \subseteq Z^2 \) integer lattice, \( H : Z[x] \to Z^2 \). Let us define functions \( \sigma^a_1(m) = G^i_a(m) \), \( \sigma^p_1(m) = G^i_p(m) \), which are the evaluations of polynomials \( G^i_a(x) \) and \( G^i_p(x) \) at \( m \), where \( m \) is the dimension of vector \( v_j \). By setting the x-axis and y-axis to be \( G^i_a(x) \) and \( G^i_p(x) \), we can build \( A \subseteq Z^2 \) lattice points, since evaluations \( \sigma^a_1(m) \) and \( \sigma^p_1(m) \) for each antipodal \( v_j \) vectors are integers. Taking the above example of antipodal vectors having dimension of \( m = 4 \) the equivalent integer lattice points is shown in Table 6.

Let us define sublattices \( A_H \subseteq A \) and \( A_{V'} \subseteq A \) constructed by vectors of \([H_4 - H_4] \) and \( V'' \), where \( V'' \in \{ \pm 1 \}^{4 \times 8} \), \( v'_i \notin [H_4 - H_4] \) for \( 1 \leq i \leq 8 \). As a reminder, for the antipodal UD code set the goal is to construct \( V' \) with the maximum number of columns \( K_{max}^a - L \) such that \( A_H \cap A_{V'} = \emptyset \).

Fig. 1 demonstrates a hit, in other words \( A_H \cap A_{V'} \neq \emptyset \) for \( k = 2 \) and \( V' = [v_4, v_{14}] = [1, -1, -1, 1]^T [1, -1, -1, 1]^T \), which is equivalent to \([16, 69]^T \) and \([84, 1]^T \) in \( A_{V'} \). If we construct the third and the fourth columns to be \( v_6 = [1, -1, -1, 1]^T \) and \( v_{12} = [1, 1, -1, 1]^T \), equivalently \([20, 65]^T \) and \([80, 5]^T \) in \( A_H \) of \( H_4 \) adding them we get \([2, 0, -2, 0]^T \), which is shown as a lattice point \([100, 70]^T \) in Fig. 1. Clearly, addition of \([16, 69]^T \) and \([84, 1]^T \) lattice points also results \([100, 70]^T \), hence, this is why intersection of two sublattices is not an empty set. Therefore, it can be shown that all possible \((8)^2\) combination of two columns of \( V' \) will at least hit a lattice points of \( A_H \), which means that \( A_H \cap A_{V'} \neq \emptyset \) for \( k = 2 \). Hence, \( |V'| = 1 \) for the case of \( m = 4 \). Using this approach, we can construct sublattices \( A_H \) and \( A_{V'} \) for the case of \( m = 8 \) and find the upper bound of \( k \) such that \( A_H \cap A_{V'} = \emptyset \). This creates a framework to search the solution using geometric combinatorics (e.g. Minkowski sum, Minkowski geometry of numbers [59], partition and decomposition of \( A \) into equivalence classes, formed by sublattices and its’ cosets to prove the \( K_{max}^a \) we can have for a given \( m \) and how to generate those vectors in \( V' \).
Polynomials of $G^t_k(x)$ and $G^p_k(x)$ can also be represented by their exponents in the above example $G^{10}_n(x) \rightarrow E^{10}_n = \{1, 3\}$ and $G^{p10}_n(x) \rightarrow E^{p10}_n = \{0, 2\}$. Therefore, our problem of avoiding non-trivial combinations of $V$ to hit any forbidden lattice points becomes

$$\bigcup_{h \in H_m} E'_i \neq \bigcup_{v \in V} E'_j \quad i, j \in \{p, n\},$$

(7)

where $\bigcup$ denotes the multiset union, element-wise sums of each multisets and $1 \geq t \geq 2^m - 1$ integer value corresponds to antipodal vector. The objective is to avoid $V \bigcup_{v \in V} E'_i$ sums of multisets $E'_p \text{ or } E'_n$ of $V$ to hit any forbidden $\bigcup_{h \in H_m} E'_i$ sum of multisets $E'_p \text{ or } E'_n$ of $H_m$. Obviously, there are $3^m$ forbidden multisets of $H_m$ and we do not want $3^k - 1$ multisets of $V$ except trivial case hit any of the forbidden multisets. We can use multiset partition theories and study bipolar vectors by their $E'_p \text{ and } E'_n$ representations and prove the maximal number of $k$.

Note that in our matrix construction design of the $v_i$’s that they are distinct and not equal to any of columns of $H_m$ or $-H_m$. If, however, any of $v_i \in \pm H_m$, then the multiset of $v_i$ hits the forbidden multiset of $H_m$. Such $v_i$ vectors can never be included in vector set of UD codes. Additionally, the $v_i$’s can be replaced by $-v$ without violating the uniquely decodability property [4]. Since all possible combinations of vectors including $-v$ of $V$ do not hit any of the forbidden multisets. In other words, if $[H_m, V] \in C$ so is any number of columns of $V$ or $H_m$ multiplied by $-1$.

One way to approach this problem is to classify all $v$ bipolar vectors into groups then use inclusion-exclusion principle. There are $2^m$ variable's that consist of $|E_n| = i$, where $i = \{0, 1, ..., m\}$ with all the combinations makes $\sum_{i=0}^{m} (m-i) = 2^m$.

Let us divide the total number of $m$-dimensional bipolar vectors into two classes; $v \in B^+_m$ if $0 \leq |E_n| < m/2$, $|E_n| = m/2 |E_p = \{1, 2, ..\}$ excluding $m h_i$’s columns from $H_m$ and $v \in B^-_m$ if $m/2 < |E_n| \leq m$, $|E_n| = m/2 |E_n = \{1, 2, ..\}$ excluding $m - h_i$’s columns from $-H_m$. It’s clear that if any $v \in B^+_m$ then $-v \in B^-_m$. This narrows for our design to consider only distinct vector sets and the total number of such distinct $v_i$’s to be considered in our $V$ design is $\sum_{i=0}^{m/2} (m-i) - m = 2^{m-1} - m$. We need to construct from distinct vectors $v \in B^+_m$ such that they do not hit any forbidden multisets.

Therefore, the total possible number of $V$ sets with $k$ columns is $\binom{2^{m-1} - m}{k}$. Out of this total number of $V$ sets with $k$ columns only some satisfy UD or one-to-one condition when appended to $H_m$. If all the possible $V$’s do not satisfy one-to-one condition that means $k$ exceeds the maximum number of columns that can be added to $H_m$. Hence, we want to count how many combinations of $v \in B^+_m$ with $k$ columns hit the forbidden multisets. If the number of combination is equal the total number of $V$ sets then we know that $k$ is not the maximum. Counting that number will help us to prove the maximum number of columns $k$.

We classify $B^+_m$ into different groups, so that any combinations of vectors in similar group hits the forbidden multisets and therefore, such $v$ that belong to the same group must be avoided in our design.

In our example, for $m = 8$ and $k = 2$, we classify $B^+_8$ into groups and count how many $V$’s do not satisfy the UD condition or one-to-one condition out of $(2^{8-1}) = \binom{120}{2} = 7140$ possible vector sets. By looking at all possible combinations of $V = [v_{j1}, v_{j2}], [E^1_n, E^2_n] \in \{1, ..., 8\}$ we only consider $(|E^1_n|, |E^2_n|) = \{(1, 3), (2, 2), (3, 3), (4, 4)\}$.

Since to hit $[h_1 \pm h_i]$, where $2 \leq i \leq 8$, the vectors $[v_{j1}, v_{j2}]$ must have $(|E^1_n|, |E^2_n|) = (1, 3)$ and $(|E^1_n|, |E^2_n|) = (2, 2)$. We can easily show that if $[v_{j1}, v_{j2}]$ hit forbidden multisets with $(|E^1_n|, |E^2_n|) = (1, 3)$ and $(|E^1_n|, |E^2_n|) = (2, 2)$ then $-[v_{j1} - v_{j2}]$ with $(|E^1_n|, |E^2_n|) = (7, 5)$ and $(|E^1_n|, |E^2_n|) = (6, 6)$ hit $[-h_1 \pm h_i]$ and $i \neq l$. Also, if $[v_{j1}, v_{j2}]$ hit forbidden multisets with $(|E^1_n|, |E^2_n|) = (2, 2)$ then $-[v_{j1} - v_{j2}]$ and $[v_{j1} - v_{j2}]$ with $(|E^1_n|, |E^2_n|) = (2, 6)$ hit $[\pm h_1 \pm h_i]$ and $i \neq l$, $p \neq i$. Hitting $[\pm h_1 \pm h_i]$ where $2 \leq i \neq j \leq 8$ multisets then vectors must have $(|E^1_n|, |E^2_n|) = (3, 5)$ and $(|E^1_n|, |E^2_n|) = (4, 4)$. Since we only work with distinct vectors $B^+_8$ then $(|E^1_n|, |E^2_n|) = (3, 3)$ that has $[v_{j1} - v_{j2}]$ is equivalent to $[v_{j1}, v_{j2}]$ with $(|E^1_n|, |E^2_n|) = (3, 3)$.

Now, let us classify $(|E^1_n|, |E^2_n|) = (1, 3)$,

$(|E^1_n|, |E^2_n|) = (2, 2), (|E^1_n|, |E^2_n|) = (3, 3)$,

$(|E^1_n|, |E^2_n|) = (4, 4)$ into $A_i$’s, $D_i$’s, $F_i$’s and $G_i$’s groups.

We know that the total number of vs with $|E_n| = 1$ is $\binom{8}{1} = 8$, $|E_n| = 2$ is $\binom{8}{2} = 28$, $|E_n| = 3$ is $\binom{8}{3} = 56$ and $|E_n| = 4$ is $\binom{8}{4} = 70$.

Here is how we divide 56 of $|E_n| = 3$ and 8 of $|E_n| = 1$ into $A_i$’s, $1 \leq i \leq 8$ groups.

$$A_1 = \begin{bmatrix}
-1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\
1 & 1 & -1 & 1 & 1 & 1 & 1 & -1 \\
1 & 1 & 1 & -1 & 1 & 1 & -1 & 1 \\
1 & 1 & 1 & 1 & -1 & -1 & 1 & -1 \\
1 & 1 & 1 & 1 & 1 & -1 & -1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & -1 & -1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & -1 
\end{bmatrix}.$$
equivalently, we can then write them in multiset form as

\[
A_1 = \begin{bmatrix}
-1 & -1 & -1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & -1 & -1 & -1 \\
-1 & -1 & -1 & -1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1
\end{bmatrix},
E_p = \{0, 1, 2, 4, 5, 6, 7\}, \quad E_n = \{0\}
\]

\[
A_2 = \begin{bmatrix}
-1 & -1 & -1 & -1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 \\
-1 & -1 & -1 & -1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1
\end{bmatrix},
E_p = \{0, 2, 3, 4, 5, 6, 7\}, \quad E_n = \{1\}
\]

\[
A_3 = \begin{bmatrix}
-1 & -1 & -1 & -1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 \\
-1 & -1 & -1 & -1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1
\end{bmatrix},
E_p = \{0, 1, 2, 4, 5, 6, 7\}, \quad E_n = \{2\}
\]

\[
A_4 = \begin{bmatrix}
-1 & -1 & -1 & -1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 \\
-1 & -1 & -1 & -1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1
\end{bmatrix},
E_p = \{0, 1, 2, 3, 5, 6, 7\}, \quad E_n = \{3\}
\]

\[
A_5 = \begin{bmatrix}
-1 & -1 & -1 & -1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 \\
-1 & -1 & -1 & -1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1
\end{bmatrix},
E_p = \{0, 1, 2, 3, 4, 6, 7\}, \quad E_n = \{4\}
\]

\[
A_6 = \begin{bmatrix}
-1 & -1 & -1 & -1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 \\
-1 & -1 & -1 & -1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1
\end{bmatrix},
E_p = \{0, 1, 2, 3, 4, 5, 7\}, \quad E_n = \{5\}
\]

\[
A_7 = \begin{bmatrix}
-1 & -1 & -1 & -1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 \\
-1 & -1 & -1 & -1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1
\end{bmatrix},
E_p = \{0, 1, 2, 3, 4, 6, 7\}, \quad E_n = \{6\}
\]

\[
A_8 = \begin{bmatrix}
-1 & -1 & -1 & -1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 \\
-1 & -1 & -1 & -1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1
\end{bmatrix},
E_p = \{0, 1, 2, 3, 4, 5, 7\}, \quad E_n = \{7\}
\]

We can prove that
\[
E_1^i + E_1^l = E_2^j + E_2^l, \quad i, j \in \{p, n\}, \quad (8)
\]
holds only if \(t_1 = \{t_1|a_{t_1} \in A_h\}, t_2 = \{t_2|a_{t_2} \in A_h\}, t_1 \neq t_2\), \(h \in H_8|t \in \{2, ..., 8\}\) and \(h \neq h_2\) and does not hold if \(t_1 = \{t_1|a_{t_1} \in A_h\}, t_2 = \{t_2|a_{t_2} \in A_h\}\), where \(h_1 \neq h_2\). Therefore, we can conclude that we can append two vectors \(v_1 \in A_1\) and \(v_2 \in A_1\) to \(H_8\), which do not hit any forbidden multisets by choosing any two vectors from different groups \(1 \leq i \neq j \leq 8\).

Here is how we divide 28 of \(|E_n| = 2\) into \(D_i\)’s, \(1 \leq i \leq 7\) groups.
and \( D_2 = \begin{bmatrix} -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 \end{bmatrix} \),

\( D_3 = \begin{bmatrix} -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 \end{bmatrix} \),

\( D_4 = \begin{bmatrix} -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 \end{bmatrix} \),

\( D_5 = \begin{bmatrix} -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 \end{bmatrix} \),

\( D_6 = \begin{bmatrix} -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 \end{bmatrix} \),

\( D_7 = \begin{bmatrix} -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 \end{bmatrix} \)
equivalently, we can then write in multiset form as

\( \mathcal{D}_1 = \{2, 3, 4, 5, 6, 7\}, \mathcal{D}_2 = \{0, 1, 2, 3, 5, 6, 7\}, \mathcal{D}_3 = \{0, 1, 2, 3, 4, 5, 6\}, \mathcal{D}_4 = \{0, 1, 2, 3, 5, 6\}, \mathcal{D}_5 = \{0, 1, 2, 3, 4, 5\}, \mathcal{D}_6 = \{0, 1, 2, 3, 4\}, \mathcal{D}_7 = \{0, 1, 2, 3, 4\} \)

and \( t_2 = \{t_2|d_{t_2} \in \mathcal{D}_{h_2}\} \), where \( h_1 \neq h_2 \). Therefore, we can conclude that we can append two vectors \( v_1 \in \mathcal{D}_j \) and \( v_2 \in \mathcal{D}_i \) to \( H_m \), which do not hit any forbidden multisets by choosing any two vectors from different groups \( 1 \leq i \neq j \leq 7 \).

Dividing 56 of \(|E_n| = 3\) into \( F_i \), \( 1 \leq i \leq 8 \) groups is equivalent exactly \( \mathcal{A}_i \)'s. Similarly, we can prove that

\[ E_i^t + E_i^{t''} = E_i^{t_1} + E_i^{t_2} \quad i, j \in \{p, n\}, \quad (11) \]

holds only if \( t_1 = \{t_1|f_{t_1} \in \mathcal{F}_{h_1}\}, t_2 = \{t_2|f_{t_2} \in \mathcal{F}_{h_2}\}, t_1 \neq t_2, \) where \( h_1, h_2, h_{t_1}, h_{t_2}, h_{t_1''}, h_{t_2''}, t'' \in \{2, ..., 8\} \) and \( h \in \{1, ..., 7\} \) and does not hold if \( t_1 = \{t_1|f_{t_1} \in \mathcal{F}_{h_1}\}, t_2 = \{t_2|f_{t_2} \in \mathcal{F}_{h_2}\}, \) where \( h_1 \neq h_2 \). Therefore, we can conclude that we can append two vectors \( v_1 \in \mathcal{F}_j \) and \( v_2 \in \mathcal{F}_i \) to \( H_m \), which do not hit any forbidden multisets by choosing any two vectors from different groups \( 1 \leq i \neq j \leq 8 \).

Here is how we divide 70/2 = 28 of \(|E_n| = 4\) into \( \mathcal{G}_i \)'s, \( 1 \leq i \leq 7 \) groups.

\( \mathcal{G}_1 = \begin{bmatrix} -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 \end{bmatrix}, \mathcal{G}_2 = \begin{bmatrix} -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 \end{bmatrix} \)
equivalently, we can then write in multiset form as

\( \mathcal{G}_1 = \{0, 3, 4, 5, 6, 7\}, \mathcal{G}_2 = \{0, 1, 2, 3, 5, 6, 7\}, \mathcal{G}_3 = \{0, 1, 2, 3, 4, 5, 6\}, \mathcal{G}_4 = \{0, 1, 2, 3, 5, 6\}, \mathcal{G}_5 = \{0, 1, 2, 3, 4, 5\}, \mathcal{G}_6 = \{0, 1, 2, 3, 4\}, \mathcal{G}_7 = \{0, 1, 2, 3, 4\} \)

It can be proved that

\[ E_i^{t_1} + E_i^{t_2} = E_i^{t_1} + E_i^{t_2} \quad i, j \in \{p, n\}, \quad (9) \]

and

\[ E_i^{t_1} + E_i^{t_2} = E_i^{t_1} + E_i^{t_2} \quad i, j \in \{p, n\}, \quad (10) \]

holds only if \( t_1 = \{t_1|d_{t_1} \in \mathcal{D}_{h_1}\}, t_2 = \{t_2|d_{t_2} \in \mathcal{D}_{h_2}\}, t_1 \neq t_2, \) where \( h_1, h_2, h_{t_1}, h_{t_2}, h_{t_1''}, h_{t_2''}, t'' \in \{2, ..., 8\} \) and \( h \in \{1, ..., 7\} \) and does not hold if \( t_1 = \{t_1|d_{t_1} \in \mathcal{D}_{h_1}\}, \)
It can be proved that
\[ E'_i + E''_j = E'_j + E''_i, \quad i, j \in \{ p, n \}, \]
holds only if \( t_1 = \{ t_1 \mid g_{t_1} \subseteq G_h \} \), \( t_2 = \{ t_2 \mid g_{t_2} \subseteq G_h \} \), \( t_1 \neq t_2 \), where \( h \in \{ 1, \ldots, 7 \} \) and does not hold if \( t_1 = \{ t_1 \mid g_{t_1} \subseteq G_h \} \), and \( t_2 = \{ t_2 \mid g_{t_2} \subseteq G_h \} \), where \( h_1 \neq h_2 \). Therefore, we can conclude that we can append two vectors \( v_1 \in G_1 \) and \( v_2 \in G_2 \) to \( H_{t_1} \), which do not hit any forbidden multisets by choosing any two vectors from different groups \( 1 \leq i \neq j \). Note that \( A_i \)'s group can be constructed from \( A_i \)'s and \( D_i \)'s such as \( A_1 = \{ A_2 + D_1, A_3 + D_2, A_4 + D_3, A_5 + D_4, A_6 + D_5, A_7 + D_6, A_8 + D_7 \} \), \( A_2 = \{ A_1 + D_1, A_3 + D_2, A_4 + D_3, A_5 + D_4, A_6 + D_5, A_7 + D_6, A_8 + D_7 \} \), \( A_3 = \{ A_1 + D_1, A_2 + D_2, A_3 + D_3, A_4 + D_4, A_5 + D_5, A_6 + D_6, A_7 + D_7 \} \), \( A_4 = \{ A_1 + D_1, A_2 + D_2, A_3 + D_3, A_4 + D_4, A_5 + D_5, A_6 + D_6, A_7 + D_7 \} \), \( A_5 = \{ A_1 + D_1, A_2 + D_2, A_3 + D_3, A_4 + D_4, A_5 + D_5, A_6 + D_6, A_7 + D_7 \} \), \( A_6 = \{ A_1 + D_1, A_2 + D_2, A_3 + D_3, A_4 + D_4, A_5 + D_5, A_6 + D_6, A_7 + D_7 \} \), \( A_7 = \{ A_1 + D_1, A_2 + D_2, A_3 + D_3, A_4 + D_4, A_5 + D_5, A_6 + D_6, A_7 + D_7 \} \) and \( G_1 = \{ D_1 + D_6, D_2 + D_5, D_3 + D_4 \} \), \( G_2 = \{ D_1 + D_6, D_2 + D_5, D_3 + D_4 \} \), \( G_3 = \{ D_1 + D_6, D_2 + D_5, D_3 + D_4 \} \), \( G_4 = \{ D_1 + D_6, D_2 + D_5, D_3 + D_4 \} \), \( G_5 = \{ D_1 + D_6, D_2 + D_5, D_3 + D_4 \} \), \( G_6 = \{ D_1 + D_6, D_2 + D_5, D_3 + D_4 \} \), \( G_7 = \{ D_1 + D_6, D_2 + D_5, D_3 + D_4 \} \) and \( G_8 = \{ D_1 + D_6, D_2 + D_5, D_3 + D_4 \} \). Also the \( G_i \)'s group can be constructed from different combinations of \( A_i \)'s or \( D_i \)'s groups. For example, \( G_1 = \{ A_1 + A_5, A_2 + A_6, A_3 + A_7, A_4 + A_8, A_5 + A_9, A_6 + A_{10}, A_7 + A_{11} \} \), \( G_2 = \{ A_1 + A_7, A_2 + A_8, A_3 + A_9, A_4 + A_5, A_6 + A_7, A_7 + A_{10}, A_8 + A_{11} \} \), \( G_3 = \{ A_1 + A_6, A_2 + A_7, A_3 + A_8, A_4 + A_5, A_5 + A_6, A_6 + A_7, A_7 + A_{10}, A_8 + A_{11} \} \), \( G_4 = \{ A_1 + A_5, A_2 + A_6, A_3 + A_7, A_4 + A_8, A_5 + A_6, A_6 + A_7, A_7 + A_{10}, A_8 + A_{11} \} \) and \( G_5 = \{ A_1 + A_6, A_2 + A_7, A_3 + A_8, A_4 + A_5, A_5 + A_6, A_6 + A_7, A_7 + A_{10}, A_8 + A_{11} \} \). Also, the \( G_i \)'s group can be constructed from different combinations of \( A_i \)'s or \( D_i \)'s groups. The proof is complete. Next, we also show the different groups of such combinations.

In both construction examples \( V_1 \) and \( V_2 \), we cannot find any 2 vector combinations that belong to the same \( A_i \)'s, \( D_j \)'s, \( F_{i,j} \)'s and \( G_i \)'s. Therefore, all possible combinations of 2 vectors do not hit any of forbidden multisets.

However, once we add any other vector \( v_6 \in B_{7}^{+} \) to the above sets some of the combinations of resulting vector sets hit one of the forbidden multisets. This means that if \( k = 6 \) we compute the number of combinations that hit the forbidden multisets is exactly equal to \( \binom{7}{2} - 6 = \binom{12}{6} = 3,652,745,460 \). Hence, the maximum number of vectors \( v_i \) that can be appended to \( H_{8} \) is \( f_2(8) = 5 \). The proof is complete.

Let us count how many vector sets \( V = \{ v_1, v_2 \} \) hit forbidden multisets. We can easily count after classifying \( B_{8}^{+} \) into groups as discussed above. Hence, the total number of vector sets that hit forbidden matrices are
\[
\binom{8}{1} \times 7 + \binom{4}{2} \times 7 + \binom{7}{2} \times 8 + \binom{4}{2} \times 7 = 56 + 42 + 168 + 42 = 308.
\]

There are no other different combinations of two vectors in \( B_{8}^{+} \) that can hit the forbidden multisets. We find that our computed number of combination that hits the forbidden multiset 308 < 7140 is less than the total number of two vectors combination sets, then we can claim that the maximum number of vectors that can be added to \( H_{8} \) is \((K_{\text{max}} - L) \geq 2\). This method of classifying \( B_{8}^{+} \) into groups not only helps us to prove the maximum number of vectors but also on how to construct such vector sets that posses unique decodability property [4].

Similar computation can be carried out for the cases \( k = 3, 4, 5 \) and still claim that the number of computation is less than the total number of \( k = 3, 4, 5 \) vectors combination sets. As an example, we present two of such combinations below for the case of \( k = 5 \),
\[
\begin{align*}
V_1 &= \begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1
\end{bmatrix}, \\
V_2 &= \begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1
\end{bmatrix}.
\end{align*}
\]

In both construction examples \( V_1 \) and \( V_2 \), we cannot find any 2 vector combinations that belong to the same \( A_i \)'s, \( D_j \)'s, \( F_{i,j} \)'s and \( G_i \)'s. Therefore, all possible combinations of 2 vectors do not hit any of forbidden multisets.

Among all possible constructions, here we show that we can only have these group combinations \( (A, A, A, D, G), (A, A, A, D, D), (A, A, D, D, G), (A, A, A, D, G, G), (A, A, D, D, D), (A, A, D, D, G), (A, D, G, G, G) \). We take each combinations and using the rules developed in [31], we show in Appendix B that we cannot add any more columns from any groups.

### IV. MINIMUM DISTANCE OF CODE SETS

The Manhattan distance \[60\] equivalently \((\ell_1)\)-norm of two \(L\)-dimensional vectors \( y_i \) and \( y_j \) for \( i \neq j \) is defined as
\[
d_L(y_i, y_j) = \sum_{i=1}^{L} |y_{i,t} - y_{j,t}|, \quad t = 1, \ldots, L.
\]
where $|\cdot|$ denotes complex amplitude. Then the general minimum Manhattan distance of received vectors for a given antipodal code set can be formulated by

$$d_{\min}(C) = \arg\min_{\bar{y}_i, \bar{y}_j \in \{\pm 1\}^{K \times 1}} d_L(\bar{y}_i, \bar{y}_j).$$  \hspace{1cm} (14)$$

**Theorem 1.** Let $\mathcal{C} \in \{\pm 1\}^{L \times K}$ represent the set of all antipodal matrices constructed by distinct columns, then the number of the code set, $\delta(\mathcal{C})$, is equal to 4, where

$$\delta(\mathcal{C}) = \arg\min_{C' \in \mathcal{C}} d_{\min}(C').$$  \hspace{1cm} (15)$$

**Proof.** Let assume that $\delta(\mathcal{C}) = d_{\min}(C) = d_L(y_n, y_m)$, where $y_n = Cx_n$, $y_m = Cx_m$, $y_n, y_m \in \mathcal{N}^{L \times 1}$, $\mathcal{N} \in \{\pm K, \pm(K - 2), \ldots\}$, and $x_n, x_m \in \{\pm 1\}^{K \times 1}$. The minimum value obtained when the difference vector $y = y_n - y_m = C(x_n - x_m) = Cx$ has only one non-zero element $y_{c} \neq 0$, $y_{m,c} \neq y_{m,c}$, and $L - 1$ zeros $y_{1,i} = 0$, $y_{m,i} = y_{m,i}$ for $i \neq c$ where $c$ is the $i$-th row of matrix $C$. One obvious remark is that the number of ±2 elements in $x$ must be even for the inner product, $C'x'$, to be zero. One possible option is to have all zeros but two non-zero elements in $x$. In this case we have the minimum value $d_{\min}(C) = y_n - y_m = Cx$ results in $\delta(\mathcal{C}) = 4$. \hspace{2cm} \blacksquare$

Now that we proved that $\delta(\mathcal{C}) = 4$, we will try to find $d_{\min}(C)$ of our proposed UD code sets $C \subseteq \mathcal{C}$, where $\mathcal{C} \subseteq \{\pm 1\}^{L \times K}$ is the set of all the antipodal UD code sets. Let us follow the option of having all zeros but two non-zero elements in $x$. In the case of $L = 4$ the columns $c_1$ and $c_2$ differ in the first element only. If $x = [2, 0, 0, 0, -2]^T$ then the difference vector is $y = [4, 0, 0, 0]^T$. Note that even if we substitute the $c_1$ by $-c_1$ we can still obtain same $y$ with $x = [2, 0, 0, 0, 2]^T$. Based on our construction in [31], we look at the case of $L = 8$. Observe that all the elements of the columns $9$-th and $12$-th, $[\alpha^{13}, 0]^T$ and $[\alpha^{13}, \alpha^{13}]^T$ of the $C$ are equal except the $5$-th element in which they differ. If we select $x_{m,9} \neq x_{m,9}$, $x_{m,12} \neq x_{m,12}$, and $x_{m,i} = x_{m,i}$ for all $i \notin \{9, 12\}$ then $y_{n,5} = 2$ and $y_{m,5} = -2$ or $y_{n,5} = -2$ and $y_{m,5} = 2$, which will result in $d_L(y_n, y_m) = 4$. With this specific observation together with the Theorem 1 we conclude that $d_{\min}(C) = 4$. From this observation, we learn that if any two columns differ at one element in a UD code set we assure that $d_{\min}(C) = 4$. Similarly, for $L = 16$ columns $17$-th and $27$-th, $[\alpha^{13}, 0, \alpha^{13}]^T$ and $[\alpha^{13}, 0, 0, 0]^T$ differ in one element only. Due to our recursive construction in [31] for $L = 2^p$, where $p = 5, 6, \ldots$ columns $y^{2(p-2)} + 3$ and $(p-1)^2(p-1) + 3$ differ in one element only. Therefore, all the UD code set generated in [31] has $d_{\min}(C) = 4$.

**V. FAST DECODING ALGORITHM IN AWGN**

The recursive linear NDA decoder, discussed in [31], is not suitable for the noisy transmission channel. Let the received vector in the presence of noise be mathematically formulated as

$$y = Acx + n$$  \hspace{1cm} (16)$$

where $A$ denotes the amplitude, $c_k \in \{\pm 1\}^{L \times 1}$ are signatures for $1 \leq k \leq K$, $x \in \{\pm 1\}^{K \times 1}$ is user data and $n$ is the Additive white Gaussian noise (AWGN) channel noise vector with a variance of $\sigma^2$.

The objective of the receiver is as follows; recover the user data $\hat{x}$ given the received vector $y$ in (16) and $C$, so that the mean square error $E\{||x - \hat{x}||^2\}$ is minimized. The ML solution is given by

$$\hat{x} = \arg\min_{x \in \{-1, 1\}^{K \times 1}} ||y - Cx||^2.$$  \hspace{1cm} (17)$$

It is widely recognized that obtaining the ML solution is generally NP-hard [61].

Our detection problem, where the overloaded signature matrix has UD structure, can be solved efficiently if there is a function that maps $y \rightarrow \tilde{y} \in A \subseteq \mathcal{N}^{L \times 1}$, where $A$ is a $Z$-module with rank $L$. Therefore, it is equivalent to finding the closest vector point in a lattice $A$, such that

$$\tilde{y} = \arg\min_{\tilde{y} \in \mathcal{N}^{L \times 1}} d_L(y, \tilde{y}).$$  \hspace{1cm} (19)$$

Gaining the knowledge of $\tilde{y}$, one of the points in $A$ generated by $C$, we can obtain $\hat{x}$ unambiguously (uniquely) by applying the NDA [31], since $C$ satisfies the unique decodability criteria. However, there is no known polynomial algorithm to solve for $\tilde{y}$ from a given $y$.

Without loss of generality, we design our low-complexity decoding algorithm for code sets that are generated by the seed matrix $V_8$ which is given by:

$$V_8 = \begin{bmatrix}
\alpha^{14} & 1 & \alpha & \alpha^3 & \alpha^3
\end{bmatrix}.
$$  \hspace{1cm} (20)$$

However, we do not necessarily imply that our proposed decoder cannot be applied to other recursive UD code sets such as in [14, 21, 23]. A slight modification may be required depending on a given $C$ matrix. We present our proposed low-complexity FDA for $C_L$, $L \in 2^i$, where $i \in \{2,3,\ldots\}$, which is portrayed in Table 7. To quickly summarize, the FDA estimates iteratively the number and positions of $-1$s in $\tilde{x}$ from the received values in $y$. Those estimates are updated when the FDA compares each value of the vector $y$ against quantized levels. Those quantized levels...
are computed based on the information from the previous rows and the current values of the vector y.

| Merge, \text{meP} function |
|-----------------------------|
| **Input:** \(dP', m, n, K, r_c, m_{LR}, \text{meP} \) |
| 1: \(d\text{Part} \leftarrow \emptyset\), \(B \leftarrow \text{meP}(r_c), p' \leftarrow 1\) |
| 2: \(\text{for } p \leftarrow 1 \text{ to len}(dP')\) |
| 3: \(A_L \leftarrow dP'(p, 3), f \leftarrow \text{true}\) |
| 4: \(\text{if } A_L \neq 0\) |
| 5: \(d\text{Part}(p', 1: 2) \leftarrow [0, A_L], L_0 \leftarrow dP'(p, 5: 4 + A_L)\) |
| 6: \(L_1 \leftarrow L_0 \cap \{1, K\}, R_1 \leftarrow L_0 \cap \{1\}\) |
| 7: \(d\text{Part}(p', 3: 4 + A_L) \leftarrow \{[L_1], R_1, L_1\}\) |
| 8: \(r \leftarrow r_c, f \leftarrow \text{false}\) |
| 9: \(L_0 = \text{meP}(r - 1, 1)\) |
| 10: \(d\text{Part}(p', 1) \leftarrow m_{LR}(r - 1, 3), f \leftarrow 1\) |
| **else** |
| **while** \(f = \text{false} \text{ AND } r > 1\) |
| 11: \(T_r \leftarrow \{\{m'(i) = -1, i \in L_0\}\} \) |
| 12: \(d\text{Part}(p', 1) \leftarrow \sum_{i \in L_0} m(i), f \leftarrow \text{true}\) |
| **else** |
| \(L_0' \leftarrow [1 : K] \setminus L_0\) |
| 13: \(I_{L_0'} \leftarrow \{\{m'(i) = -1, i \in L_0'\}\} \) |
| **if** \(I_{L_0'} = \emptyset\) |
| 14: \(d\text{Part}(p', 1) \leftarrow n - \sum_{i \in L_0'} m(i), f \leftarrow \text{true}\) |
| **else** |
| \(r \leftarrow r - 1\) |
| 15: \(m \leftarrow \text{umissing}(m_{LR}(r, 3), \text{meP}(r, 1), m)\) |
| **if** \(f = \text{false}\) |
| **break** |
| 16: \(p' \leftarrow p' + 1\) |
| **else** |
| \(A_R \leftarrow dP'(p, 4)\) |
| **if** \(A_R \neq 0 \text{ AND } f = \text{true}\) |
| 17: \(d\text{Part}(p', 1: 2) \leftarrow [0, A_R]\) |
| 18: \(L_0 \leftarrow dP'(p, 5: 4 + A_R), L_2 \leftarrow L_0 \cap \{1, K\}, R_2 \leftarrow L_0 \cap \{1\}\) |
| 19: \(d\text{Part}(p', 3: 4 + A_R) \leftarrow \{[L_2], [R_2], [L_2, R_2]\}\) |
| **else** |
| \(d\text{Part}(p', 1) \leftarrow m_{LR}(r - 1, 4), f \leftarrow \text{true}\) |
| **else** |
| **while** \(f = \text{false} \text{ AND } r > 1\) |
| 20: \(I_{T_r} \leftarrow \{\{m'(i) = -1, i \in L_0\}\} \) |
| **if** \(I_{T_r} = \emptyset\) |
| 21: \(d\text{Part}(p', 1) \leftarrow \sum_{i \in L_0} m(i), f \leftarrow \text{true}\) |
| **else** |
| \(L_0' \leftarrow [1 : K] \setminus L_0\) |
| 22: \(I_{L_0'} \leftarrow \{\{m'(i) = -1, i \in L_0'\}\} \) |
| **if** \(I_{L_0'} = \emptyset\) |
| 23: \(d\text{Part}(p', 1) \leftarrow n - \sum_{i \in L_0'} m(i), f \leftarrow \text{true}\) |
| **else** |
| \(r \leftarrow r - 1\) |
| 24: \(m \leftarrow \text{umissing}(m_{LR}(r, 4), \text{meP}(r, 2), m)\) |
| **if** \(f = \text{false}\) |
| **break** |
| 25: \(p' \leftarrow p' + 1\) |

---

Therefore, the FDA attempts to find the closest lattice point that is generated by matrix C from the estimate \(\hat{x}\) to the received vector \(y\) \(\in \mathbb{R}^{L \times 1}\). This is achieved by performing quantization on each row of vector y to obtain \(z\) \(\in \mathbb{N}^{L \times 1}\) such that \(z\) is a valid lattice point. In order to demonstrate how the FDA works, it is beneficial to describe each function in detail. The quantizer \(Q : \mathbb{R} \rightarrow \mathbb{N}^\ast, z_1 = Q(y, -K, K, 2)\) maps a received real value \(y \in \mathbb{R}\) to one of the constellation in \(\{\pm K, \pm (K - 2), \ldots\}\) as follows,

\[
Q(y, t^-, t^+, s) = \begin{cases} 
 t^- & \text{if } y \leq t^- + s/2 \\
 t^- + s(t' - 3/2) & \text{if } t^- + s(t' - 1/2) \\
 t^+ & \text{if } y > t^+ - s/2 
\end{cases}
\]

(21)

where \(t^-\), \(t^+\) and \(s\) are the input parameters for the minimum, maximum, step-size values and \(t' \in \mathbb{N}\) is the internal value that decides the quantization level, respectively. Furthermore, let integer \(n\) and vector \(m\) denote the number of \(-1s\) and locations, where the \(-1s\) occur in \(\hat{x}\), respectively. Mathematically, \(n\) and \(m\) can be expressed as

\[
n = (K - \hat{x}^T 1)/2, \\
m = (1 - \hat{x})/2.
\]

(22)

(23)

**TABLE 7.** Fast Decoder Algorithm (FDA)

**FDA**

**Input:** \(y, K\)

1: \(z_1 \leftarrow Q(y_1, -K, K, 2)\)
2: \(z_1 \leftarrow K, \hat{x} \leftarrow \text{sgn}(z_1)K\)
3: \(\text{else}\)
4: \(m \leftarrow -1K, r_c \leftarrow -1, n \leftarrow (K - z_1)/2\)
5: \(m_{LR}(r_c, 3) \leftarrow n\)
6: \(dP(r_c) \leftarrow [n, K, m_{LR}(r_c, 1), m_{LR}(r_c, 2), mP(r_c)]\)
7: \(c_{AL} \leftarrow 0, z \leftarrow 0, s_j \leftarrow 1, c_T \leftarrow 1\)
8: \(s_l = 1\ AND\ c_T < N_c\)
9: \(s_l \leftarrow 0\)
10: \(\text{while } r_c < L, r_c \leftarrow r_c + 1\)
11: \([dP(r_c), m] \leftarrow \text{meP}(dP(r_c), n, K, r_c, m_{LR}, mP)\)
12: \(A^- \leftarrow \min T(dP(r_c)), A^+ \leftarrow \max T(dP(r_c))\)
13: \(z(r_c) \leftarrow Q(y, A^-, A^+)\)
14: \(m_{LR}(r_c, 3) \leftarrow n - m_{LR}(r_c, 3)\)
15: \(m \leftarrow \text{umissing}(m, m_{LR}, r_c, mP)\)
16: \(m \leftarrow f_l(m, m_{LR}, n)\)
17: \(t_d \leftarrow z - C(-2m + 1)\)
18: \(c_{AL}(r_c + 1) \leftarrow c_{AL}(r_c + 1) + 1\)
19: \(c_T \leftarrow c_T + 1\)
20: \(\text{Output: } \hat{x}\)

For the trivial case when \(z_1 = K\) or \(z_1 = -K\) the algorithm outputs the decision vector \(\hat{x}\) without proceeding to the next steps. Otherwise, it will initialize the vector \(m \leftarrow 1K,\) index \(r_c \leftarrow -1\) and \(n \leftarrow (K - z_1)/2\). Then the value of \(n\) is recorded in the table \(m_{LR}\) at the row \(r_c = 1\) and the column \(n_L\). The table \(m_{LR}\) keeps track of \(l_L, l_R, n_L\) and \(n_R\) values for each row and \(l_L\) and \(l_R\) are defined as the number of \(+1\)s and \(-1\)s of the row of code set \(C\), \(n_L\) and \(n_R\) are the number

\[i_{d_l}\] is the lowest index that \(t_d(i_d) \neq 0\).
of $-1$s of the estimated $\hat{x}$ that corresponds to locations of $+1$ and $-1$ of each row of the code set.

For example, $m_{LR}$ table for the code set in Fig. 6 can be constructed as

| $r_c$ | index $L$ | index $R$ |
|-------|-----------|-----------|
| 1     | {0}       | {1,2,3,4,5} |
| 2     | {2,4,5}   | {1,3}     |
| 3     | {3,4}     | {1,2,5}   |
| 4     | {2,3}     | {1,4,5}   |

where the last two columns presented as dots are filled up in the further steps of the algorithm. Whereas $mP(r_c)$ is the actual column indices of $+1$s and $-1$s in the row $r_c$. In the case of code set in Fig. 6, the matrix $mP$ is defined as

The algorithm proceeds by partitioning each row $r_c$ and saving the estimated number of $-1$s of vector $\hat{x}$, $n'$, the partition size, $K'$, the number of $+1$s and $-1$s, $lL$, and $lR$, of the specific partition in $dP(r_c)$. In Step 7, the adaptive parameter $c_{AL}$, stopping flag $s_I$, and the number of repetition count $c_T$ (e.g., algorithm can repeat steps from 10 to 16 up to a maximum of $N_c$ times, since it directly depends on the variance of the noise) are initialized. At each row $r_c$, $dP(r_c)$ gets updated by calling $meP()$ function. The function $meP(dP(r_c - 1), m, n, K, r_c, m_{LR}, mP)$ scans each partition of the row $r_c$ with updated values and if it finds one or more partitions completely identified the exact locations of $-1$s, hence will skip partitioning further. $A^-$ and $A^+$ are the minimum and maximum values calculated for a given partitions at each row.

### MinT function

**Input:** $dP$

1. $A^- ← 0$
2. for $i ← 1$ to $\text{len}(dP)$
   3. $A^- ← A^- + \text{minF}(dP(i, 1), dP(i, 3), dP(i, 2))$

**Output:** $A^-$

### MaxT function

**Input:** $dP$

1. $A^+ ← 0$
2. for $i ← 1$ to $\text{len}(dP)$
   3. $A^+ ← A^+ + \text{maxF}(dP(i, 1), dP(i, 3), dP(i, 2))$

**Output:** $A^+$

where $\text{minF}(n, L, K) = 2|n - L| - K$ and $\text{maxF}(n, R, K) = -2|n - R| + K$. In line 13 of FDA, we define $y' = y(r_c) + 2\text{sgn}(y(r_c) - z(r_c))c_{AL}(r_c)$, where $c_{AL}$ is an integer vector that is incremented in Step 19 by one if estimated $z$ is not one of the lattice vertices generated by $C$. In line 18, we scan the rows from 1 to $L$ to find the first $r_c$, where $z$ differ from estimated lattice vertex. The function $uM(m, m_{LR}, r_c, mP)$ updates $m$ with the given updated parameters as follows

### uM function

**Input:** $m, m_{LR}, r, mP$

1. if $m_{LR}(r, 3) = 0$
2. $m(mP(r, 1)) ← 0$
3. else $m_{LR}(r, 3) = m_{LR}(r, 1)$
4. $m(mP(r, 1)) ← 1$
5. else
6. $m ← u\text{Missing}(m_{LR}(r, 3), mP(r, 1), m)$
7. if $m_{LR}(r, 4) = 0$
8. $m(mP(r, 2)) ← 0$
9. else $m_{LR}(r, 4) = m_{LR}(r, 2)$
10. $m(mP(r, 2)) ← 1$
11. else
12. $m ← u\text{Missing}(m_{LR}(r, 4), mP(r, 2), m)$

**Output:** $m$

where the function $u\text{Missing}$ in lines 6 and 12 of uM are defined as

### uMissing function

**Input:** $n', A, m$

1. $I_{L} ← \{i | m(i) = -1, i ∈ A\}$
2. if $|I_{L}| = 1$
3. $m(I_{L}) ← n' - \sum_{i ≠ I_{L}} m(i)$

**Output:** $m$

In line 17 of the FDA, the function $f_c(m, m_{LR}, n)$ finds all the locations of $-1$s in $m$ based on the UD structure of $C$ codes. As an example, in case of code set presented in Fig. 6 the function $f_c$ is expressed as

### $f_c$ function, $L = 4$

**Input:** $m, m_{LR}, n$

1. $n' ← \sum_{i ∈ A_5} m_{LR}(i, 4)$
2. $m(5) ← \text{mod}(n', 2)$
3. $m' ← (n' - m(5))/2$
4. $m(1) ← n - m' - m(5)$
5. $m(2) ← m_{LR}(3, 3) - m(1) - m(5)$
6. $m(3) ← m_{LR}(2, 3) + m(1)$
7. $m(4) ← m_{LR}(2, 4) - m(2) - m(5)$

**Output:** $m$

where $A_5 = \{2, 3, 4\}$. In case the information in $m$ does not correspond to $z$, which is verified in line 18 of FDA it then sets $r_c$ to the row, where the discrepancy happened and re-runs from Step 10 again until it finds $m$ that correspondence to $z$.

For the case of Rayleigh fading channel instead of AWGN in [16] the proposed FDA is still applicable to perform detection. However, the channel coefficients for each user index $k$ should be known at the receiver side. For the frequency-selective fading channels, we can employ transmitter precoding scheme to overcome multipath channel effect as proposed by Fantuz and D’Amours, which is detailed in [62]. Briefly, this transmit precoding scheme exploits the knowledge of the channel impulse response for transforming the multipath channel into a single-path non-dispersive channel, which is equivalent over non-dispersive Rayleigh fading channel model.
VI. COMPLEXITY ANALYSIS

In this section, we discuss the complexity analysis of the proposed NDA and FDA algorithms. The NDA decoder for the noiseless transmission channels, discussed in [31], deciphers the data of all users at the receiver side in a recursive manner. At each step, it performs additions, comparisons and multiplications to decipher the bits of the users.

The NDA is deterministic with an exact number of execution steps. After $L/8$ number of execution steps it calls the NDA algorithm recursively, using two smaller vectors composed of the upper and lower $L/2$ elements of the received vector. For that reason to compute the complexity of the algorithm we first break the algorithm into two blocks $B_1$ and $B_2$ from steps 1 to 8 and 9 to 12.

We denote $N_1$ and $N_2$ to represent the total number of additions, comparisons and multiplications ($N_{\text{add}}^i$, $N_{\text{comp}}^i$, $N_{\text{mult}}^i$) of block $B_i$ at each recursive calls. As an example $N_1 = (N_{\text{add}}^1, N_{\text{comp}}^1, N_{\text{mult}}^1)$, where $N_{\text{add}}^1 = 10$, $N_{\text{comp}}^1 = 3$ and $N_{\text{mult}}^1 = 0$ since there is 3, 3 and 4 additions in steps 2, 4 and 6, and 1 comparison in steps 3, 5 and 8, respectively. Similarly, when $L = 16$ the $N_2 = (N_{\text{add}}^2, N_{\text{comp}}^2, N_{\text{mult}}^2)$, where $N_{\text{add}}^2 = (3 + L)L/2 + 4 = 156$, $N_{\text{comp}}^2 = 1$ and $N_{\text{mult}}^2 = L^2/2 = 128$, respectively. For the trivial case when $L = 4$ we have $N_4 = (N_{\text{add}}^4, N_{\text{comp}}^4, N_{\text{mult}}^4)$, where $N_{\text{add}}^4 = 4$, $N_{\text{comp}}^4 = 4$ and $N_{\text{mult}}^4 = 4$. 
As one would normally expect, the complexity of the proposed FDA decoder in Section 6 is not more complex than the NDA in terms of big $O$ notation. It is important to state that the proposed FDA requires neither matrix inversion nor decomposition, instead, only additions, comparisons and multiplications. The algorithm goes through each row of the received vector to decode one or more users. The best case scenario of our FDA would be to satisfy the condition in Step 2 with the complexity of maximum of $K$ comparisons, $N_{\text{comp}} = K$. We should note that unlike the deterministic NDA, FDA has an element of probability in the execution steps. Therefore, to compute the overall complexity of the FDA we need to consider the worst case scenario. In Step 11 the complexity of meP can be shown to have $6L$ additions and $(1 + \log_2(K))6L + 2$ comparisons. The complexity of steps 12 and 13 is 6 additions and $K$ comparisons. Similarly, the complexity of steps 14 and 15 is 4 additions. The complexity of uM method is $2K$ additions and $2K + 6$ comparisons. Those execution steps from 11 to 16 are repeated $L$ times. Finally, in Step 17 the complexity is $(2 + K)L + O(K)$ additions and $LK$ multiplications. Considering all the components, the total complexity can be shown to be $N_{\text{add}} = 9LK + 12L + O(LK)$, $N_{\text{comp}} = 6LK\log_2(K) + 9LK + 8L$ and $N_{\text{mult}} = LK$. Therefore, we can conclude by looking at the higher order terms of comparisons, $N_{\text{comp}}$, since it is higher than additions and multiplications, hence the overall complexity is $O(LK\log_2(K))$. We note that in both NDA and FDA algorithms we do not consider assignments in our complexity computations. We can see that the complexity of FDA is comparably larger than NDA but much lower than the MMSE-PIC, Slab-sphere, PDA and ML decoders, which have complexities of $O(LK^2)$, $O(LK^2)$, $O(L^2K^2)$, and $O(2^K)$, respectively, as shown in Table 8. The exact computation for the cases $L = 4$, $L = 8$ and $L = 16$ are shown in Table 9.

![](image)

**FIGURE 4.** Complexity for $L = 16$.

The complexity when $L = 8$ is $N_1 + 2N_4$ since the NDA will not execute block $B_2$. In the case when $L = 16$, the complexity of NDA is $4N_1 + N_2 + 4N_4$. It follows that when $L > 8$ the complexity of the recursive NDA can be represented as such

$$(i - 2)^2(i - 3)N_1 + ((i - 4)2(i - 3) + 1)N_2 + 2(i - 2)N_4,$$  

where $L = 2^i$ for $i \in \{4, 5, \ldots\}$. Based on this calculation, we can conclude that the complexity of NDA algorithm is $O(L\log_2(L))$.

**TABLE 8.** Complexity comparison of the Detectors

| Algorithms    | Complexity          | Main procedures  |
|---------------|---------------------|------------------|
| NDA           | $O(L\log_2(L))$     | Comparisons      |
| FDA           | $O(LK\log_2(K))$    | Comparisons      |
| MMSE-PIC      | $O(LK^2)$           | Inversion + multiplication |
| Slab-sphere   | $O(LK^2)$           | Multiplication + comparisons |
| PDA           | $O(L^2K^2)$         | Inversion + multiplication |
| ML            | $O(2^K)$            | Multiplication + addition |

As one would normally expect, the complexity of the decoder in noisy channels is much higher than in the noiseless channels. However, the complexity of the proposed FDA decoder in Section 6 is not more complex than the NDA in terms of big $O$ notation.
Therefore, we demonstrate the number of additions, comparisons and multiplications in Figs. 2, 3 and 4 by varying $\sigma^2$ in terms of $E_b/N_0$ in our simulations. For the case of $L = 4$ in Fig. 2, the number of additions, comparisons and multiplications does not depend on variance due to small overload factor 5/4. However, for $L = 8$ and $L = 16$ it stays high for up to 10 dB in $E_b/N_0$. Then it drops to constant numbers at around 20 dB in $E_b/N_0$, which is shown in Figs. 3 and 4 respectively.

**VII. SIMULATION RESULTS**

In this section, we evaluate the performance of the proposed antipodal UD code sequences generated by the seed matrix [20], which are portrayed in Figs. 6, 7 and 5. In our simulations, we compare the FDA with the MMSE-PIC [49], slab-sphere [63], PDA [53] and ML detectors for binary phase-shift keying (BPSK) modulation, which are characterized in Figs. 8, 9 and 10.

The BER performance of UD code sets are averaged over the different users for $C_{4 \times 5}$, $C_{8 \times 13}$ and $C_{16 \times 33}$, respectively. The performance of the proposed FDA is comparable to that of ML, as shown in Fig. 8.

For the larger values of $Ls$, the FDA has slightly inferior performance in terms of BER compared to ML. However, in practice the $E_b/N_o$ at the BER of $10^{-3}$ is considered to be the operating threshold since we can apply channel encoding to achieve even as low as $10^{-6}$ BER. At the BER of $10^{-3}$ the FDA achieves 1 dB and 4 dB gain compared to the Slab-sphere, 4 dB and 15 dB gain compared the PDA, as shown in Figs. 9 and 10. The significant increase of the BER performance gap between the FDA and Slab-sphere, PDA detectors for greater values of $Ls$ is due to error floor experienced by the Slab-sphere and the PDA detectors. The reason for this is since for larger values of

![Figure 6](image-url)  
**FIGURE 6.** UD code set $C$ with $L = 4$ and $K = 5$.

![Figure 7](image-url)  
**FIGURE 7.** UD code set $C$ with $L = 8$ and $K = 13$.

![Figure 5](image-url)  
**FIGURE 5.** UD code set $C$ with $L = 16$ and $K = 33$.

![Figure 8](image-url)  
**FIGURE 8.** UD code set $C_{4 \times 5}$.
the overload factor increases exponentially. This in turn results in degradation of the linear separability of the overall system.

VIII. CONCLUSION

In this paper, we introduced a novel fast (low-complexity) decoder algorithm (FDA) for antipodal uniquely decodable (UD) code sets. The proposed algorithm has a much lower computational complexity compared to the maximum likelihood (ML) decoder whose complexity may be prohibitive for even moderate code lengths. Simulation results show that the performance of the proposed decoder is almost as good as that of the ML decoder with only a 1–2 dB SNR degradation at a BER of $10^{-3}$. Moreover, we proved the minimum Manhattan distance of UD codes proposed in [31] and a number of propositions which collectively served to be the foundation of the formal proof for the maximum number of users, $K_{\text{max}}^a$, for the case of $L = 8$.

In our future research, we will conceive multiuser detection for higher-order constellations for transmission over dispersive fading channels.

APPENDIX A PROOF OF THE CONVERSION FROM $C^A$ TO $C^B$

The proof of the conversion from the antipodal overloaded UD code sets to binary UD code sets, coined as optical CDMA code sets in [27], are presented next.

**Theorem 2.** If there is an antipodal UD code set $C^a \in \{\pm 1\}^{L \times K}$, then there is an equivalent binary UD code set $C^b \in \{0, 1\}^{L \times K}$.

**Proof.** Suppose there is an antipodal UD code set $C^a_{L \times K}$. By corollary, if multiplying each row or column by $-1$ we can assume that the entries of the first row of $C^a_{L \times K}$ are all 1s. Let the conversion to the binary matrix $C^b_{L \times K} = (C^a_{L \times K} + J)/2$, where $J$ is the $L \times K$ all-one matrix. It is clear that $C^b_{L \times K} \in \{0, 1\}^{L \times K}$, therefore, we now need to prove the following Null statement. Assume that $C^b_{L \times K} z = 0_L$, which yields to

$\text{Null}()$ represents the nullspace of a matrix.
(C_{L×K} + J)z = 0_L and thus, C'_{L×K}z = -Jz, where z ∈ {0, ±1}^{K×1}. Since the entries of the first row of C'_{L×K} as well as the matrix J are all 1s, the first entry of C'_{L×K}z must be equal to the first entry of -Jz. It is only possible if the first entry of -Jz is 0. Thus, -Jz = 0_L which leads to z = 0_L. As a consequence of UD code set of C'_{L×K} that satisfies unique decodability condition, the expression C'_{L×K}z = 0_L implies that z = 0_L.

However, in general it is not necessarily true that any binary (optical) code sets C_b of any size can be converted to UD code sets C_a. The reason is because for a given UD code set C_b with K_b^max, the following is true K_b^max ≥ K_a^max since K_b^max = γ(L+1) and K_a^max = γ(L)+1 and obviously γ(L+1) ≥ γ(L)+1. It is equality, K_b^max = K_a^max, if L = 2^i where i ∈ {1, 2, . . .}. Therefore, for a given UD code set C_b with K_b^max if L ≠ 2^i we have K_b^max > K_a^max and there is no equivalent UD code set C_a.

**APPENDIX B PROOF OF ALL POSSIBLE COMBINATIONS**

Combination 1 (A, A, A, D, G): This group combination is such that distinct (A, A, A) produce 3 distinct D’s different from existing D, and G’s by A + A = D/Γ rule. Different A is also produced by A + D = A from existing ones. With this new A it can produce another distinct 3 = Γ’s. Total of six Γ’s are produced with (A, A, A, D) and existing G we have seven Γ’s. This completes all seven different Γ’s from each seven groups, hence we cannot add any other Γ to the existing combination. So three A’s with D produce six distinct A’s and the A + A + A = A, A + A + A + D = A produce another two distinct A’s. Therefore, we cannot add any more A to the existing combinations. Now three A’s will produce three distinct D’s, and three A’s with D, A + D = A, produce another three distinct D’s, plus the D that is in the combination that makes a total of seven D’s. Therefore, we cannot add any more D’s to the combination. Therefore, we proved that we cannot add any more A, D and G to the (A, A, A, D, G).

Combination 2 (A, A, D, D, D): This group combination is such that distinct (D, D, D) produce 3 = Γ’s by D + D = Γ rule, and those Γ’s must be different from created by (A, A) A + A = Γ rule. Also each A must be different from three distinct A’s created by D + A = A rule. Each existing A’s with 3 = Γ’s distinct D + D + A = A produce six distinct A’s and plus two existing A makes a total of eight A’s. This completes all distinct A’s and we cannot add any more A to the combination. From existing combination there are four distinct D’s produced by A + A + D = D and D + D + D = D rules as well as three existing D’s this make a total of seven distinct D’s. Hence, we cannot add any more D to the combination. We have already seen that with existing combinations we create six distinct Γ’s and with (A, A, D) we can create more Γ’s by A + A + D = Γ rule. This tells us that we cannot add any more Γ’s. Therefore, we proved that we cannot add any more A, D and G to the (A, A, D, D, D).

Combination 3 (A, A, D, D, Γ): This group combination is such that different A’s created by A + D = A, A + D + D = A, A + G = A, A + D + G = A, A + D + D + G = A rules are distinct from existing A’s. Similarly, different D’s created by A + A = D, A + A + G = D, A + D = G rules are distinct from existing D’s of and different Γ’s created by A + A = Γ, A + A + D = Γ, A + A + D + D = Γ rules are distinct from existing Γ’s. The combination produces six distinct A’s by A + D + D = A, A + G = A, A + A = D rules and with two existing A’s that makes a total of eight A’s, hence we cannot add any more A to the combination. There are five distinct D’s produced by A + A + D = D, G + D = D, A + A = D, D rules and with two existing D’s that makes a total of seven D’s, hence we cannot add any more D to the combination. Similarly, for the case of Γ the combination produces six distinct Γ’s by D + D = Γ, A + A = Γ, Γ + A + G = D, A + A + D + D = Γ rules and with the existing Γ’s that makes a total of seven Γ’s, hence we cannot add any more Γ to the combination. Therefore, we proved that we cannot add any more A, D and G to the (A, A, D, D, Γ).

Combination 4 (A, A, D, G, Γ): This group combination is such that different A’s created by A + D = A, A + G = A, A + G + Γ = A, A + D + G = A rules are distinct from existing A’s. Similarly, different D’s created by A + A = D, A + A + G = D, A + A + D + G = D rules are distinct from existing D’s of and different Γ’s created by A + A = Γ, A + A = Γ, Γ + A + G = D rules are distinct from existing Γ’s. The combination produces six distinct A’s by A + D = A, A + D + G = A rules and with two existing A’s that makes a total of eight A’s, therefore we cannot add any more A to the combination. There are six distinct D’s produced by A + A = D, Γ + D = D, A + A + G = D, A + A + G + Γ = D rules and with the existing D’s that makes a total of seven D’s, hence we cannot add any more D to the combination. Similarly, for the case of Γ the combination produces five distinct Γ’s by A + A = Γ, Γ + A + G = Γ, Γ + A + A = Γ rules and with the existing Γ’s that makes a total of seven Γ’s, hence we cannot add any more Γ to the combination. Therefore, we proved that we cannot add any more A, D and G to the (A, A, D, D, Γ).

Combination 5 (A, D, D, D, D): This group combination is such that all D’s are distinct and no D + D = D + D is satisfied. The combination produces seven distinct A’s by A + D = A, A + D + D = A rules and with the existing A that makes a total of eight A’s, hence we cannot add any more A to the combination. There are three distinct D’s produced by D + D + D = D rules and with the existing D’s that makes a total of seven D’s, hence we cannot add any more D to the combination. Similarly, for the case of Γ the combination produces seven distinct Γ’s by D + D = Γ, D + D + D = Γ rules and with the existing Γ’s that makes a total of seven Γ’s, hence we cannot add any more Γ to the combination. Therefore, we proved that we cannot...
add any more $A$, $D$ and $G$ to the $(A, D, D, D, D)$.

Combination 6 $(A, D, D, D, G)$: This group combination is such that different $D$'s created by $D + G = D$, $D + D + D = D$ rules are distinct from existing $D$'s and different $G$'s created by $D + D = G$, $A + D + D = G$ rules are distinct from existing $G$'s. The combination produces seven distinct $A$'s by $A + D = A$, $A + G = A$, $A + D + G = A$ rules and with the existing $A$ that makes a total of eight $A$'s, hence we cannot add any more $A$ to the combination. There are four distinct $D$'s produced by $D + G = D$, $D + D + G = D$ rules and with three existing $D$'s that makes a total of seven $D$'s, hence we cannot add any more $D$ to the combination. Similarly, for the case of $G$ the combination produces five distinct $G$'s by $D + D = G$, $D + D + G = G$ rules and with the existing $G$ that makes a total of seven $G$'s, hence we cannot add any more $G$ to the combination. Therefore, we proved that we cannot add any more $A$, $D$ and $G$ to the $(A, D, D, D, D, G)$.

Combination 7 $(A, D, D, G, G)$: This group combination is such that different $D$'s created by $D + G = D$, $G = G$ rules are distinct from existing $D$'s and different $G$'s created by $D + D + G = G$ rules are distinct from existing $G$'s. The combination produces seven distinct $A$'s by $A + D = A$, $A + G = A$, $A + D + G = A$ rules and with the existing $A$ that makes a total of eight $A$'s, hence we cannot add any more $A$ to the combination. There are five distinct $D$'s produced by $D + G = D$, $D + G + G = D$ rules and with two existing $D$'s that makes a total of seven $D$'s, hence we cannot add any more $D$ to the combination. Similarly, for the case of $G$ the combination produces five distinct $G$'s by $D + D = G$, $G = G$, $D + D + G = G$, $D + G + G = G$ rules and with the existing $G$ that makes a total of seven $G$'s, hence we cannot add any more $G$ to the combination. Therefore, we proved that we cannot add any more $A$, $D$ and $G$ to the $(A, D, D, G, G, G)$.

Combination 8 $(A, D, G, G, G)$: This group combination is such that different $D$'s created by $D + G = D$, $G = G$ rules are distinct from existing $D$'s and different $G$'s created by $G + G = G$ rules are distinct from existing $G$'s. The combination produces seven distinct $A$'s by $A + D = A$, $A + G = A$, $A + G + G = A$, $A + D + G = A$ rules and with the existing $A$ that makes a total of eight $A$'s, hence we cannot add any more $A$ to the combination. There are six distinct $D$'s produced by $D + G = D$, $G = G$, $D + B + G = D$, $D + G + G + D$ rules and with two existing $D$'s that makes a total of seven $D$'s, hence we cannot add any more $D$ to the combination. Similarly, for the case of $G$ the combination produces five distinct $G$'s by $G + G = G$, $G + G + G = G$ rules and with the existing $G$ that makes a total of seven $G$'s, hence we cannot add any more $G$ to the combination. Therefore, we proved that we cannot add any more $A$, $D$ and $G$ to the $(A, D, G, G, G, G)$.

REFERENCES
[1] A. J. Viterbi, CDMA: Principles of Spread Spectrum Communication, C. Sammut and G. I. Webb, Eds. Redwood City, CA, USA: Addison Wesley Longman Publishing Co., Inc., 1995.
[2] L. Dai, B. Wang, Z. Ding, Z. Wang, S. Chen, and L. Hanzo, “A survey of non-orthogonal multi-access for 5G,” IEEE Commun. Surveys & Tutorials, vol. 20, no. 3, pp. 2294–3232, Thirdquarter 2018.
[3] H. Sari, F. Vanhaverbeke, and M. Moeneclaey, “Extending the capacity of multiple access channels,” IEEE Commun. Mag., vol. 38, no. 1, pp. 74–82, Jan. 2000.
[4] A. J. Grant and P. D. Alexander, “Random sequence multisets for synchronous code-division multiple-access channels,” IEEE Trans. Inf. Theory, vol. 44, no. 7, pp. 2832–2836, Nov. 1988.
[5] S. Verdu and S. Shamai, “Spectral efficiency of CDMA with random spreading,” IEEE Trans. Inf. Theory, vol. 45, no. 2, pp. 622–640, Mar. 1999.
[6] F. Vanhaverbeke, M. Moeneclaey, and H. Sari, “DS/CDMA with two sets of orthogonal spreading sequences and iterative detection,” IEEE Commun. Lett., vol. 4, no. 9, pp. 289–291, Sep. 2000.
[7] H. Sari, F. Vanhaverbeke, and M. Moeneclaey, “Multiple access using two sets of orthogonal signal waveforms,” IEEE Commun. Lett., vol. 4, no. 1, pp. 4–6, Jan. 2000.
[8] F. Vanhaverbeke, M. Moeneclaey, and H. Sari, “Increasing CDMA capacity using multiple orthogonal spreading sequence sets and successive interference cancellation,” in Proc. IEEE Inter. Conf. on Commun. (ICC), vol. 3, New York, U.S.A., Apr. 2002, pp. 1516–1520.
[9] F. Vanhaverbeke and M. Moeneclaey, “An improved OCDMA/OCDMA scheme based on displaced orthogonal user sets,” IEEE Commun. Lett., vol. 8, no. 5, pp. 265–267, May 2004.
[10] M. Li, S. N. Batalama, D. A. Pados, and J. D. Mattyas, “Minimum total-squared-correlation quaternary signature sets: new bounds and optimal designs,” IEEE Trans. on Commun., vol. 57, no. 12, pp. 3662–3671, Dec. 2009.
[11] R. Erdős and A. Rényi, “On two problems of information theory,” Publ. Math. Inst. Hung. Acad. Sci., vol. 8, pp. 241–254, 1963.
[12] S. Söderberg and H. S. Shapiro, “A combinatorial detection problem,” The Amer. Math. Monthly, vol. 70, no. 10, pp. 1066–1070, Dec. 1963.
[13] D. G. Cantor, “Determining a set from the cardinalities of its intersections with other sets,” Can. J. Math., vol. 16, pp. 94–97, Feb. 1964.
[14] B. Lindström, “On a combinatorial detection problem,” Publ. Math. Inst. Hung. Acad. Sci., vol. 9, pp. 195–207, 1964.
[15] ———, “On a combinatorial problem in number theory,” Can. Math. Bull., vol. 8, no. 4, pp. 477–490, Mar. 1965.
[16] D. G. Cantor and W. H. Mills, “Determining a subset from certain combinatorial properties,” Can. J. Math., vol. 18, pp. 42–48, Feb. 1966.
[17] P. Erdős, “On a combinatorial problem in number theory,” Amer. Math. Monthly, vol. 77, no. 6, pp. 619–621, Jul. 1970.
[18] Shih-Chun Chang and E. Weldon, “Coding for T-user multiple-access channels,” IEEE Trans. Inf. Theory, vol. 25, no. 6, pp. 684–691, Nov. 1979.
[19] T. Ferguson, “Generalized T-user codes for multiple-access channels (Corresp.),” IEEE Trans. Inf. Theory, vol. 28, no. 5, pp. 775–778, Sep. 1982.
[20] Shih-Chun Chang, “Further results on coding for T-user multiple-access channels (Corresp.),” IEEE Trans. Inf. Theory, vol. 30, no. 2, pp. 411–415, Mar. 1984.
[21] G. H. Khachatrian and S. S. Martirosian, “Codes for T-user noiseless adder channel,” Prob. Control Inform. Theory, vol. 16, no. 3, pp. 187–192, 1987.
[22] S. S. Martirosian and G. H. Khachatrian, “Construction of signature codes and the coin weighing problem,” Prob. Peredachi Inform., vol. 25, no. 4, pp. 96–97, 1989.
[23] G. H. Khachatrian and S. S. Martirosian, “A new approach to the design of codes for synchronous-CDMA systems,” IEEE Trans. Inf. Theory, vol. 41, no. 5, pp. 1503–1506, Sep. 1995.
[24] ———, “Code construction for the T-user noiseless adder channel,” IEEE Trans. Inf. Theory, vol. 44, no. 5, pp. 1953–1957, Sep. 1998.
[25] Ying-Wah Wu and Shih-Chun Chang, “Coding scheme for synchronous-CDMA systems,” IEEE Trans. Inf. Theory, vol. 43, no. 3, pp. 1064–1067, May 1997.
[26] W. H. Mow, “Recursive constructions of detecting matrices for multuser coding: a unifying approach,” IEEE Trans. Inf. Theory, vol. 55, no. 1, pp. 93–98, Jan. 2009.
[27] P. Pad, F. Marvasti, K. Aliashahi, and S. Akbari, “Further results on coding for T-user multiple-access channels (Corresp.),” IEEE Trans. Inf. Theory, vol. 55, no. 6, pp. 2705–2715, Jun. 2009.
[28] M. Li and Q. Liu, “Fast code design for overloaded code-division multiplexing systems,” IEEE Trans. on Vehic. Tech., vol. 65, no. 1, pp. 447–452, Jan. 2016.
S. Morosi, R. Fantacci, E. Del Re, and A. Chiassai, “Design of turbo-MUD codes for overloaded synchronous CDMA systems,” *IEEE Trans. Commun.*, vol. 60, no. 11, pp. 3145–3149, Nov. 2012.

A. Singh, A. Armani, P. Singh, and F. Marvasti, “Set of uniquely decodable codes for overloaded synchronous CDMA,” *IET Commun.*, vol. 10, no. 10, pp. 1236–1245, Jul. 2016.

M. Kulhandjian and D. A. Pados, “Uniquely decodable code-division via augmented Sylvester-Hadamard matrices,” in *Proc. IEEE Wireless Commun. and Network. Conf. (WCNC)*, Paris, France, Apr. 2012, pp. 359–363.

M. Kulhandjian, C. D’Amours, and H. Kulhandjian, “Uniquely decodable ternary codes for synchronous CDMA systems,” in *Proc. IEEE Pers., Indoor, Mobile Radio Conf. (PIMRC)*, Bologna, Italy, Sep. 2018, pp. 1–6.

Multifarious physical-layer network coding via uniquely decodable codes,” *Wireless Commun. and Mobile Comp.*, vol. 2018, no. ID 2034870, pp. 1–9, Jun. 2018.

M. Kulhandjian, H. Kulhandjian, H. Yanikomeroglu, and G. Khachatrian, “Fast decoder for overloaded uniquely decodable synchronous optical CDMA,” in *Proc. IEEE Wireless Commun. and Network. Conf. (WCNC)*, Marrakech, Morocco, Apr. 2019, pp. 1–7.

M. Kulhandjian, H. Kulhandjian, and C. D’Amours, “Uniquely decodable ternary codes via augmented Sylvester-Hadamard matrices,” in *Proc. IEEE Int. Black Sea Conf. on Commun. and Networking (BlackSeaCom)*, Virtual, May 2021, pp. 1–5.

S. Lu, W. Hou, J. Cheng, and H. Kamabe, “Multi-user ud k-ary codes recursively constructed from short-length multiary codes for multiple-access adder channel,” in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Paris, France, Jul. 2019, pp. 425–429.

Q. Yu, Y. Li, W. Meng, and W. Xiang, “Uniquely decodable codes for physical-layer network coding in wireless cooperative communications,” *IEEE Systems Journal*, vol. 13, no. 4, pp. 3959–3967, Dec. 2019.

Y. Dai, B. Wang, Y. Yuan, S. Han, C. I, and Z. Wang, “Non-orthogonal multiple access for 5G: solutions, challenges, opportunities, and future research trends,” *IEEE Commun. Mag.*, vol. 53, no. 9, pp. 74–81, Sep. 2015.

R. Hoshyar, F. P. Wathan, and R. Tafazoli, “Novel low-density signature for synchronous CDMA systems over AWGN channel,” *IEEE Trans. Signal Process.*, vol. 56, no. 4, pp. 1616–1626, Apr. 2008.

M. Kulhandjian, C. D’Amours, and L. Hanzo, “Low-dimension spreading code design based on Gaussian separability,” *IEEE Access*, vol. 9, pp. 33 963–33 993, Mar. 2021.

H. Nikopour and H. Baligh, “Sparse code multiple access,” in *Proc. IEEE Pers., Indoor, Mobile Radio Conf. (PIMRC)*, London, U.K., Sep. 2013, pp. 332–336.

M. Kulhandjian and C. D’Amours, “Design of permutation-based sparse code multiple access system,” in *Proc. IEEE Pers., Indoor, Mobile Radio Conf. (PIMRC)*, Montreal, Canada, Oct. 2017, pp. 1–6.

J. van de Beek and B. M. Popovic, “Multiple access with low-density signatures,” in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Honolulu, Hawaii, U.S.A., Nov. 2009, pp. 1–6.

M. Kobayashi, J. Bottros, and G. Caire, “Successive interference cancellation with SISO decoding and EM channel estimation,” *IEEE J. Sel. Areas Commun.*, vol. 19, no. 8, pp. 1450–1460, Aug. 2001.

M. K. Varanasi and B. Aazhang, “Multistage detection in asynchronous code-division multiple-access communications,” *IEEE Trans. Commun.*, vol. 38, no. 4, pp. 509–519, Apr. 1990.

G. Xue, J. Weng, T. Le-Ngoc, and S. Tahar, “Adaptive multistage parallel interference cancellation for CDMA,” *IEEE J. Sel. Areas Commun.*, vol. 17, no. 10, pp. 1815–1827, Oct. 1999.

D. Guo, L. K. Rasmussen, S. Sun, and T. J. Lim, “A matrix-algebraic approach to parallel interference cancellation in high-speed CDMA,” *IEEE Trans. Commun.*, vol. 48, no. 1, pp. 152–161, Jan. 2000.

M. C. Reed, C. B. Schlegel, P. D. Alexander, and J. A. Asenstorfer, “Iterative multiuser detection for CDMA with FEC: near-single-user performance,” *IEEE Trans. on Commun.*, vol. 46, no. 12, pp. 1693–1699, Dec. 1998.

X. Wang and H. V. Poor, “Iterative (turbo) soft interference cancellation and decoding for coded CDMA,” *IEEE Trans. Commun.*, vol. 47, no. 7, pp. 1046–1061, Jul. 1999.

S. Morosi, R. Fantacci, E. Del Re, and A. Chiassai, “Design of turbo-MUD receivers for overloaded CDMA systems by density evolution technique,” *IEEE Trans. on Wirel. Commun.*, vol. 6, no. 10, pp. 3552–3557, Oct. 2007.

P. Kumar and S. Chakrabarti, “An analytical model of iterative interference cancellation receiver for orthogonal/orthogonal overloaded DS-CDMA system,” *Int. J. Wirel. Inf. Netw.*, vol. 17, no. 1-2, pp. 64–72, Jun. 2010.

M. Kulhandjian, H. Kulhandjian, and C. D’Amours, “Uniquely decodable ternary codes for synchronous CDMA systems,” in *Proc. IEEE Int. Conf. on Acoust., Speech, and Sig. Proc. (ICASSP)*, vol. 3, Philadelphia, PA, U.S.A., Mar. 2005, pp. 733–736.

A. Najkha, C. Roland, and E. Boutilier, “A near-optimal multiuser detector for MC-CDMA systems using geometrical approach,” in *Proc. IEEE Int. Conf. on Acoust., Speech, and Sig. Proc. (ICASSP)*, vol. 3, Philadelphia, PA, U.S.A., Mar. 2005, pp. 877–880.

G. Manglani and A. K. Chaturvedi, “Application of computational geometry to multiuser detection in CDMA,” *IEEE Trans. on Commun.*, vol. 54, no. 2, pp. 204–207, Feb. 2006.

S. Lu, W. Hou, J. Cheng, and H. Kamabe, “A New Kind of Non-binary Uniquely Decodable Codes with Arbitrary Code Length for Multiple-Access Adder Channel,” in *IEEE Inf. Theory Workshop (ITW)*, Guangzhou, China, Nov. 2018, pp. 1–5.

D. S. Dummitt and R. M. Foote, *Abstract Algebra*. Englewood Cliffs, N.J.: Prentice Hall, 1991.

T. K. Moon, *Error Correction Coding: Mathematical Methods and Algorithms*. New Jersey: Wiley-Interscience, 2005.

C. Olds, A. Lax, and G. Davidoff, *The Geometry of Numbers*, 1st ed. Mathematical Association of America, 2000.

S. Craw, “Manhattan Distance,” in *Encyclopedia of Machine Learning*, C. Sammut and G. I. Webb, Eds. Boston, MA: Springer US, 2010, pp. 639–639.

R. Lupas and S. Verdu, “Linear multiuser detectors for synchronous code-division multiple-access channels,” *IEEE Trans. Inf. Theory*, vol. 35, no. 1, pp. 123–136, Jan. 1989.

M. Fanzutti and C. D’Amours, “Low Complexity PIC-MMSE Detector for LDS Systems,” in *Proc. IEEE Canadian Conf. on Electrical and Computer Engineering (CECE)*, Edmonton, Canada, May 2019.

K. Wong, A. Pautraj, and R. D. Murch, “Efficient high-performance decoding for overloaded MIMO antenna systems,” *IEEE Trans. on Wirel. Commun.*, vol. 6, no. 5, pp. 1833–1843, May 2007.

MICHEL KULHANDJIAN (M’18–SM’20) received his M.S. and Ph.D. degrees in Electrical Engineering from the State University of New York at Buffalo in 2007 and 2012, respectively. He had previously received his B.S. degree in Electronics Engineering and Computer Science (Minor), with “Summa Cum Laude” from the American University in Cairo (AUC) in 2005. He was employed at Alcatel-Lucent, in Ottawa, Ontario, in 2012. In the same year he was appointed as a Research Associate at EION Inc. He received Natural Science and Engineering Research Council of Canada (NSERC) Industrial R&D Fellowship (IRDF). He is currently a Research Scientist at the School of Electrical Engineering and Computer Science at the University of Ottawa. He is also employed as a senior embedded software engineer at L3Harris Technologies. His research interests include wireless multiple access communications, adaptive coded modulation, waveform design for overloaded code-division multiplexing applications, channel coding, space-time coding, adaptive multiuser detection, statistical signal processing, covert communications, spread-spectrum steganography and steganalysis. He has served as a guest editor for Journal of Sensor and Actuator Networks (JSN). He actively serves as member of Technical Program Committee (TPC) of IEEE WCNC, IEEE GLOBECOM, IEEE ICC, and IEEE VTC.
HOVANNES KULHANDJIAN (S’14-M’15-SM’20) received the B.S. degree (magna cum laude) in electronics engineering from The American University in Cairo, Cairo, Egypt, in 2008, and the M.S. and Ph.D. degrees in electrical engineering from the State University of New York at Buffalo, Buffalo, NY, USA, in 2010 and 2014, respectively. From December 2014 to July 2015, he was an Associate Research Engineer with the Department of Electrical and Computer Engineering, Northeastern University, Boston, MA, USA. He is currently an Associate Professor with the Department of Electrical and Computer Engineering, California State University, Fresno, Fresno, CA, USA. His current research interests include wireless communications and networking, with applications to underwater acoustic communications, visible light communications and applied machine learning. He has served as a guest editor for IEEE Access - Special Section Journal on Underwater Communications and Networking. He has also served as a Session Co-editor for IEEE Access - Special Section Journal on Underwater Wireless Communications and Applied Machine Learning. He has served as a guest editor for IEEE Communications, IEEE Transactions on Wireless Communications, IEEE Transactions on Vehicular Technology, and IEEE Transactions on Cybernetics. He also serves as an Associate Editor for IEEE Access and IEEE Journal on Selected Areas in Communications. He has served as a member of the Technical Program Committee for ACM and IEEE conferences such as IEEE GLOBECOM, ICC, UComms, PIMRC, WD, ACM WUWNet, among others.

CLAUDE D’AMOURS received the degrees of B.A.Sc., M.A.Sc. and Ph.D. in Electrical Engineering from the University of Ottawa in 1990, 1992 and 1995 respectively. In 1992 he was employed as a Systems Engineer at Callian Communications Ltd. In 1995 he joined the Communications Research Centre in Ottawa, Ontario, Canada, as a Systems Engineer. Later in 1995, he joined the Department of Electrical and Computer Engineering at the Royal Military College of Canada in Kingston, Ontario, Canada, as an Assistant Professor. He joined the School of Information Technology and Engineering (SITE), which has since been renamed as the School of Electrical Engineering and Computer Science (EECS), at the University of Ottawa as an Assistant Professor in 1999. From 2007-2011, he served as Vice Dean of Undergraduate Studies for the Faculty of Engineering and has been serving as the Director of the School of EECS at the University of Ottawa since 2013. His research interests are in physical layer technologies for wireless communications systems, notably in multiple access techniques and interference cancellation.

HALIM YANIKOMEROGLU (Fellow, IEEE) is a Professor in the Department of Systems and Computer Engineering at Carleton University, Ottawa, Canada. His research group has made substantial contributions to 4G and 5G wireless technologies. His group’s current focus is the aerial and satellite networks for the 6G and beyond-6G era. His extensive collaboration with industry resulted in 39 granted patents. He is a Fellow of IEEE, EIC (Engineering Institute of Canada), and CAE (Canadian Academy of Engineering), and a Distinguished Speaker for both IEEE Communications Society and IEEE Vehicular Technology Society. Dr. Yanikomeroglu received several awards for his research, teaching, and service.

GURGEN KHATCHATRIXIAN is a well known and highly regarded applied scientist, scholar and technologist in the field of Error Control Coding and Cryptography. For his career achievements in these fields, he was elected as a full member of Armenian National Academy of Sciences in 1996 which is the highest honor awarded to scientists in Armenia. Dr. Khachatryan worked from 1999 through 2001 as Chief Cryptographer for Cylink Corporation (Nasdaq: CYLK) located in Sunnyvale, CA. Dr. Khachatryan then worked from 2001 through 2007 as Chief Cryptographer for Cylinda Corporation located in Marina Del Rey, CA. Since 2011 he is a Chief Cryptography Officer at Quantum Digital Solutions Corporation (QDSC) in Marina Del Rey, CA, USA and Professor on leave from AUA.
