

# Massive Coded-NOMA for Low-Capacity Channels: A Low-Complexity Recursive Approach

Mohammad Vahid Jamali and Hessam Mahdavifar, *Member, IEEE*

**Abstract**—In this paper, we present a low-complexity recursive approach for massive and scalable code-domain nonorthogonal multiple access (NOMA) with applications to emerging low-capacity scenarios. The problem definition in this paper is inspired by three major requirements of the next generations of wireless networks. Firstly, the proposed scheme is particularly beneficial in low-capacity regimes which is important in practical scenarios of utmost interest such as the Internet-of-Things (IoT) and massive machine-type communication (mMTC). Secondly, we employ code-domain NOMA to efficiently share the scarce common resources among the users. Finally, the proposed recursive approach enables code-domain NOMA with low-complexity detection algorithms that are scalable with the number of users to satisfy the requirements of massive connectivity. To this end, we propose a novel encoding and decoding scheme for code-domain NOMA based on factorizing the pattern matrix, for assigning the available resource elements to the users, as the Kronecker product of several smaller factor matrices. As a result, both the pattern matrix design at the transmitter side and the mixed symbols’ detection at the receiver side can be performed over matrices with dimensions that are much smaller than the overall pattern matrix. Consequently, this leads to significant reduction in both the complexity and the latency of the detection. We present the detection algorithm for the general case of factor matrices. The proposed algorithm involves several recursions each involving certain sets of equations corresponding to a certain factor matrix. We then characterize the system performance in terms of average sum rate, latency, and detection complexity. Our latency and complexity analysis confirm the superiority of our proposed scheme in enabling large pattern matrices. Moreover, our numerical results for the average sum rate show that the proposed scheme provides better performance compared to straightforward code-domain NOMA with comparable complexity, especially at low-capacity regimes.

**Index Terms**—Code-domain NOMA, low-capacity channels, massive communication, low-complexity recursive detection, low-latency communication, IoT, mMTC.

## I. INTRODUCTION

Low-capacity scenarios have become increasingly important in a variety of emerging applications such as the Internet-of-Things (IoT) and massive machine-type communication (mMTC) [2]. For example, the narrowband IoT (NB-IoT) feature, included in Release-13 of the 3rd generation partnership project (3GPP), is specifically meant for ultra-low-rate, wide-area, and low-power applications [3]. To ensure

The material in this paper was presented in part at the IEEE Global Communications Conference (GLOBECOM), Abu Dhabi, UAE, Dec. 2018 [1].

The authors are with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109, USA (e-mail: mvjamali@umich.edu, hessam@umich.edu).

This work was supported by the National Science Foundation under grants CCF-1763348, CCF-1909771, and CCF-1941633.

wide-area applications, NB-IoT is designed to support maximum coupling losses (MCLs) as large as 170 dB. Achieving such large MCLs requires reliable detection for signal-to-noise ratios (SNRs) as low as  $-13$  dB [4]. Consequently, one needs to carefully design the communication protocols aimed for massive communication applications, such as IoT and mMTC, with respect to the low-SNR constraints.

Recently, nonorthogonal multiple access (NOMA) techniques, that borrow ideas from solutions to traditional problems in network information theory including multiple access and broadcast channels, have gained significant attention. In NOMA schemes, multiple users are served in the same orthogonal resource element (RE) or, more generally speaking, a set of users are served in a smaller set of REs. The goal is to significantly increase the system throughput and reliability, improve the users’ fairness, reduce the latency, and support massive connectivity [5], [6]. NOMA is a general setup and, in principle, any multiple access scheme that attempts to *non-orthogonally* share the REs among the users, e.g., random multiple access [7]–[9] and opportunistic approaches [10], can be formulated in this setting. In general, NOMA techniques in the literature can be classified into two categories: power-domain NOMA and code-domain NOMA.

Power-domain NOMA serves multiple users in the same orthogonal RE by properly allocating different power levels to the users [11]. Power-domain NOMA has attracted significant attention in recent years and several problems have been explored in this context. This includes cooperative communication [12]–[14], simultaneous wireless information and power transfer (SWIPT) [15], multiple-input multiple-output (MIMO) systems [16], mmWave communications [17], mixed radio frequency and free-space optics (RF-FSO) systems [18], [19], unmanned aerial vehicles (UAVs) communications [20], and cache-aided systems [21]. The detection procedure in power-domain NOMA highly relies on successive interference cancellation (SIC) which works well provided that the channel conditions of the users paired together are not close to each other.

Code-domain NOMA, on the other hand, aims at serving a set of users, say  $K$ , in a set of  $M$  orthogonal REs, with  $M \leq K$ , using a certain code/pattern matrix. The pattern matrix comprises  $K$  pattern vectors each assigned to a user specifying the set of available REs to that user. Unlike power-domain NOMA, code-domain NOMA works well even in power-balanced scenarios provided that each user is served by a unique pattern vector. Nevertheless, code-domain NOMA has received relatively less attention in the literature compared to the power-domain NOMA. This is mainly because its gain usually comes at the expense of complex multiuser detection

(MUD) algorithms, such as maximum *a posteriori* (MAP) detection, message passing algorithm (MPA), and maximum likelihood (ML) detection. In this context, sparse code multiple access (SCMA) is proposed in [22] based on directly mapping the incoming bits to multidimensional codewords of a certain SCMA codebook set. Also, a low-complexity SCMA decoding algorithm is proposed in [23] based on list sphere decoding. Another efficient SCMA decoder is discussed in [24] based on deterministic message passing algorithms. Lattice partition multiple access (LPMA) is proposed in [25]–[27] based on multilevel lattice codes for multiple users. Interleave-grid multiple access (IGMA) is proposed in [28], [29] in order to increase the user multiplexing capability and to improve the performance. Moreover, pattern division multiple access (PDMA) is introduced in [30], [31], where the pattern vectors are designed with disparate orders of transmission diversity to mitigate the error propagation problem in SIC receivers. As opposed to most of the aforementioned prior works in code-domain NOMA, the pattern matrix in PDMA is not necessarily sparse. In other words, the number of REs allocated to a particular user can potentially be comparable to the total number of REs. Very recently, a low-complexity on-off division multiple access (ODMA) scheme has been proposed in [32] for code-domain NOMA systems. In the ODMA scheme proposed in [32], each user employs the same channel code whose coded bits, after modulation, are sent in a random time-hopping manner over much larger number of time slots than the length of the channel code by keeping the remaining time slots idle, resulting in a super-sparse multiple access with a very low-complexity iterative multi-user decoding method. Also, a coded MIMO-NOMA system with capacity-approaching performance and low implementation complexity has been proposed in [33] which consists of a linear minimum mean-square error (LMMSE) multi-user detector and a bank of single-user message-passing decoders to decompose the overall NOMA signal recovery into distributed low-complexity computations with iterative processing. Additionally, the achievable rates of approximate message passing (AMP) algorithm for coded random linear systems has been analyzed in [34], proving that the low-complexity AMP algorithm achieves the constrained capacity based on matched forward error control (FEC) coding.

Design of the pattern matrix plays a critical role in code-domain NOMA to balance the trade-off between the system performance and the complexity. Impact of the pattern matrix on the average sum-rate of SCMA systems is explored in [35], where a low-complexity iterative algorithm is proposed to facilitate the design of the pattern matrix. Moreover, the total throughput of low-density code-domain (LCD) NOMA is characterized in [36] for regular random pattern matrices with large dimensions. It is well understood that, for a given overload factor  $\beta \triangleq K/M$ , expanding the dimension of the pattern matrix improves the system performance [31]. However, increasing the pattern matrix dimension significantly increases the detection complexity.

Inspired by the aforementioned trade-off between the system performance and the detection complexity, we propose a novel encoding and decoding approach toward code-domain NOMA

which factorizes the pattern matrix as the Kronecker product of several smaller factor matrices. Consequently, as we show, both the pattern matrix design at the transmitter side and the mixed symbols' detection at the receiver side can be performed over much smaller dimensions and with significantly reduced complexity, through recursive detection. In other words, our scheme enables application of pattern matrices with large dimensions while keeping the overall detection complexity and latency manageable low. As we establish later, this provides the possibility to significantly improve the system performance at a given, reasonably low, complexity and latency level.

The low complexity of the proposed detection algorithm for pattern matrices with large dimensions enables grouping a massive number of users together, referred to as “massive coded-NOMA” in this paper. Moreover, the proposed multiple access technique is particularly advantageous in scenarios where there is a stringent constraint on the maximum power of the users' symbols. Therefore, even full-power transmission of the users' symbols (up to a predefined maximum allowed power) over only one RE (or few REs) does not meet the users' desired data rates. In such circumstances, users are inevitable to transmit over several REs, possibly with the maximum allowed power per symbol, to achieve the desired data rates. For instance, a large number of repetitions are allowed in low-capacity scenarios such as NB-IoT and mMTC in order to enable reliable communications in these emerging applications [4]. Our proposed protocol, through facilitating coded-NOMA over large-dimension pattern matrices, enables spreading the symbols of the low-capacity users over several REs and then detecting them with a low complexity. We start our studies with the traditional Gaussian multiple access channel (GMAC) model, and then clarify how the results can be extended to more practical scenarios such as fading channels.

The main contributions of the paper are summarized as follows.

- We propose a low-complexity and scalable approach toward code-domain NOMA by constructing the overall pattern matrix as the Kronecker product of several factor matrices. We show that the proposed scheme significantly reduces the detection complexity and also facilitates the design of good pattern matrices; hence, it allows incorporating large pattern matrices that are of particular interest for massive communication and low-capacity channels.
- For the Kronecker product of square factor matrices we propose a systematic way of choosing the factor matrices that enables a remarkably low-complexity detection algorithm involving only few linear operations at each recursion. We show that our design scheme and the proposed detection algorithm for the Kronecker product of square factor matrices effectively increases the SNRs of data symbols after each recursion that is of particular importance for low-capacity channels.
- We provide a generic recursive detection algorithm for the Kronecker product of rectangular factor matrices that can work on the general case of factor matrices.
- We derive useful expressions for the characterization of important system performance metrics such as the average sum-rate per RE, latency, and detection complexity.

- We demonstrate how the results of the paper, derived for the Gaussian MAC model, can be extended to practical scenarios such as uplink and downlink fading channels. We also highlight how the proposed code-domain NOMA scheme in this paper can be combined with power-domain NOMA to boost the system performance in various aspects.
- We provide extensive numerical analysis to study the system performance in various scenarios.

The rest of the paper is organized as follows. In Section II, we briefly describe the system and channel models. In Section III, we focus on the design procedure and detection algorithm of the proposed protocol over GMAC. We then characterize the average sum rate, latency, and detection complexity in Section IV. We devote Section V to more realistic scenarios, and Section VI to numerical results. Finally, we conclude the paper and highlight several future research directions in Section VII.

## II. PRELIMINARIES AND SYSTEM MODEL

In this section, we briefly review the basics of code-domain NOMA, specifically PDMA [31], which is relevant to the system model of our proposed scheme. We consider a collection of  $K$  users communicating using  $M$  REs, and define the overload factor as  $\beta \triangleq K/M \geq 1$  implying the multiplexing gain of PDMA compared to the orthogonal multiple access (OMA). The modulation symbol  $x_k$  of the  $k$ -th user is spread over  $M$  orthogonal REs using the pattern vector  $\mathbf{g}_k$  as  $\mathbf{v}_k \triangleq \mathbf{g}_k x_k$ ,  $1 \leq k \leq K$ , where  $\mathbf{g}_k \in \mathcal{B}^{M \times 1}$ , with  $\mathcal{B} \triangleq \{0, 1\}$ , is an  $M \times 1$  binary vector defining the set of REs available to the  $k$ -th user; the  $k$ -th user can use the  $i$ -th RE,  $1 \leq i \leq M$ , if the  $i$ -th element of  $\mathbf{g}_k$  is “1”, i.e., if  $g_{i,k} \triangleq \mathbf{g}_k(i) = 1$ . Otherwise, if  $g_{i,k} = 0$ , then the  $k$ -th user does not use the  $i$ -th RE. Then the overall  $(M \times K)$ -dimensional pattern matrix  $\mathbf{G} \in \mathcal{B}^{M \times K}$  is specified as follows [31]:

$$\mathbf{G}_{M \times K} \triangleq [\mathbf{g}_1 \ \mathbf{g}_2 \ \cdots \ \mathbf{g}_K] = [g_{i,k}]_{M \times K}. \quad (1)$$

For the uplink transmission each user transmits its spread symbol to the base station (BS). Therefore, assuming perfect synchronization at the BS, the vector  $\mathbf{y}$  comprising the received signals at all  $M$  REs can be modeled as

$$\mathbf{y} = \sum_{k=1}^K \text{diag}(\mathbf{h}_k) \mathbf{v}_k + \mathbf{n}, \quad (2)$$

where  $\mathbf{h}_k$  is the vector modeling the uplink channel response of the  $k$ -th user at all of  $M$  REs, and  $\mathbf{n}$  is the noise vector at the BS with length  $M$ . Furthermore,  $\text{diag}(\mathbf{h}_k)$  is a diagonal matrix consisting of the elements of  $\mathbf{h}_k$ . Then the uplink transmission model can be reformulated as

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}, \quad (3)$$

where  $\mathbf{x} \triangleq [x_1, x_2, \dots, x_K]^T$ ,  $\mathbf{H} \triangleq \mathbf{H} \odot \mathbf{G}_{M \times K}$  is the PDMA equivalent uplink channel response,  $\mathbf{H} \triangleq [\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_K]$ ,  $\mathbf{A}^T$  is the transpose of the matrix  $\mathbf{A}$ , and  $\odot$  denotes the element-wise product [31].

Moreover, for the downlink transmission, the BS first encodes the data symbol of each user according to its pattern vector and then transmits the superimposed encoded symbols

$\sum_{j=1}^K \mathbf{v}_j$  through the channel. Therefore, the received signal  $\mathbf{y}_k$  at the  $k$ -th user can be expressed as

$$\mathbf{y}_k = \text{diag}(\mathbf{h}_k) \sum_{j=1}^K \mathbf{g}_j x_j + \mathbf{n}_k = \mathbf{H}_k \mathbf{x} + \mathbf{n}_k, \quad (4)$$

where  $\mathbf{H}_k \triangleq \text{diag}(\mathbf{h}_k) \mathbf{G}_{M \times K}$  is the PDMA equivalent downlink channel response of the  $k$ -th user. Moreover,  $\mathbf{h}_k$  and  $\mathbf{n}_k$  are the downlink channel response and the noise vector, both with length  $M$ , at the  $k$ -th user, respectively [31].

As a standard MUD algorithm in NOMA systems SIC provides a proper trade-off between the system performance and the complexity. However, SIC receivers often suffer from error propagation problems as the system performance is highly dependent on the correctness of early-detected symbols. In order to resolve this issue, one can either improve the reliability of the initially-decoded users or employ more advanced detection algorithms including ML and MAP. In [31], disparate diversity orders are adopted for different users by assigning patterns with heavier weights to those early-detected users in order to increase their transmission reliability. Furthermore, employing more advanced detection algorithms severely increases the system complexity especially for larger pattern matrices; this may hinder their practical implementation particularly for downlink transmission where the users are supposed to have lower computational resources than the BS.

In the next section, we elaborate how the proposed scheme scales with the dimension of the pattern matrix, even with rather heavy pattern weights, to boost massive connectivity without a significant increase on the overall system complexity. In Sections III and IV, we consider the conventional model of GMAC, i.e.,  $\mathbf{y} = \mathbf{G}\mathbf{x} + \mathbf{n}$ , that corresponds to the case where  $\mathbf{H}$  defined after (3) is replaced by an all-one matrix (or  $\mathbf{h}_k$  defined after (4) is replaced by an all-one vector). The simple channel model of GMAC, although is conventional in the literature, has several limitations in real practice, e.g., it ignores the information about the channel gains. However, since this is an initial research on this topic, we start with the simplest channel model to streamline the presentation of the underlying design strategies and detection algorithms for the proposed low-complexity code-domain NOMA. We then demonstrate, in Section V-A, how the results can be extended to the cases of uplink and downlink fading channels, as in (3) and (4), respectively.

## III. DESIGN AND DETECTION OVER GAUSSIAN MAC

### A. Design Principles

Depending on the number of users and the available REs to them, we propose to consider pattern matrices that are factorized as the Kronecker product of a certain number, denoted by  $L$ , of smaller factor matrices as follows:

$$\mathbf{G}_{M \times K} = \mathbf{G}_{m_1 \times k_1}^{(1)} \otimes \mathbf{G}_{m_2 \times k_2}^{(2)} \otimes \cdots \otimes \mathbf{G}_{m_L \times k_L}^{(L)}, \quad (5)$$

where  $L$  is a design parameter and  $\otimes$  denotes the Kronecker product defined as

$$\mathbf{A}_{m \times k} \otimes \mathbf{B} \triangleq \begin{bmatrix} a_{11}\mathbf{B} & \cdots & a_{1k}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{m1}\mathbf{B} & \cdots & a_{mk}\mathbf{B} \end{bmatrix}, \quad (6)$$

for any two matrices  $\mathbf{A}$  and  $\mathbf{B}$ . The dimensions of the resulting pattern matrix in (5) relate to the dimensions of the factor matrices as  $M = \prod_{l=1}^L m_l$  and  $K = \prod_{l=1}^L k_l$ . In general, if at least one of  $M$  or  $K$  is not a prime number, we can find some  $m_l > 1$  or  $k_l > 1$  to construct the pattern matrix as the Kronecker product of some smaller factor matrices. On the other hand, if both  $M$  and  $K$  are prime numbers,  $L$  is equal to one and the design procedure simplifies to that of conventional pattern matrix design (such as PDMA [31]). Note that both  $M$  and  $K$  are design parameters and we can always properly group a certain number of users over a desired number of REs to optimize the system performance and the complexity.

The main inspiration behind factorizing the overall pattern matrix as the Kronecker product of smaller factor matrices is to come up with a low-complexity recursive detection algorithm that can be applied to large-scale code-domain NOMA settings. As it will be elaborated in the next subsections, the proposed structure enables a recursive detection algorithm that at each recursion divides the set of equations to disjoint subsets of equations and works on them in parallel. In that sense, the factorization helps us to devise a “divide-and-conquer” type of algorithm that recursively breaks down the detection problem into sub-problems that can be executed in parallel. This then significantly lowers the complexity and latency compared to the case where we directly solve the system of equations defined according to the overall pattern matrix.

The proposed structure not only alleviates the detection complexity and latency at the receiver side but also significantly reduces the search space at the transmitting party, enabling the usage of large pattern matrices with a reasonable complexity for massive connectivity. With the search space, we mean the number of possible pattern matrices for which one should perform a (possibly brute-force) search over them to obtain a *good* pattern matrix. The notion of a *good* pattern matrix depends on the context and the objective function for which we want to optimize. With that being said, a regular pattern matrix design (without factorization) requires a comprehensive search over all  $\binom{2^M-1}{K}$  possible  $(M \times K)$ -dimensional binary matrices with distinct nonzero columns (patterns assigned to each user) to find an optimal pattern matrix [31]. On the other hand, it is easy to show that satisfying distinct nonzero columns for the overall pattern matrix  $\mathbf{G}_{M \times K}$  of the form given by (5) requires distinct nonzero columns for all of the factor matrices  $\mathbf{G}_{m_l \times k_l}^{(l)}$ ,  $l = 1, 2, \dots, L$ . Otherwise, if any of the factor matrices has a repeated column, many of the pattern vectors in the overall pattern matrix will be the same, i.e., many of the users will be served with a same pattern vector which itself requires more advanced detection algorithms to distinguish them at the receiver (see also Section V-B). Therefore, the search space for our proposed design method reduces to  $\prod_{l=1}^L \binom{2^{m_l}-1}{k_l}$  which is significantly smaller than  $\binom{2^M-1}{K}$ . For example, for  $M = 6$  and  $K = 9$ , the regular pattern matrix design requires searching over  $\binom{2^6-1}{9} = 2.36 \times 10^{10}$  possible matrices, while our design method with the factorization of  $\mathbf{G}_{2 \times 3}^{(1)} \otimes \mathbf{G}_{3 \times 3}^{(2)}$  only needs to search over  $\binom{2^2-1}{3} \cdot \binom{2^3-1}{3} = 35$  matrices. Note that recursive construction of large matrices based on the

Kronecker product of some smaller matrices has been used in different contexts such as polar coding [37] and its extended versions such as compound polar coding [38], [39].

In the following subsections, we first describe the detection algorithm over the Kronecker product of square and rectangular factor matrices, and then summarize the overall detection algorithm over the general case of the pattern matrix. It is worth mentioning at this point that, as it will be clarified from the extensive characterizations in the following subsections, our proposed detection algorithm for the case of square factor matrices significantly differs from that of the rectangular factor matrices and cannot be obtained as a special case of that.

### B. Square Factor Matrices

In this subsection, we explore the design of square factor matrices and describe the corresponding detection algorithm over GMAC. In particular, we assume that the overall pattern matrix is represented as  $\mathbf{G}_{M \times K} = \mathbf{P}_{m_1 \times m_1}^{(1)} \otimes \mathbf{P}_{m_2 \times m_2}^{(2)} \otimes \dots \otimes \mathbf{P}_{m_L \times m_L}^{(L)}$ , where  $\mathbf{P}_{m_l \times m_l}^{(l)}$ ,  $l = 1, 2, \dots, L$ , is an  $m_l \times m_l$  binary square matrix. In this case,  $M = K = \prod_{l=1}^L m_l$  and the overload factor  $\beta$  is equal to 1. However, we will observe that with a careful design of the square factor matrices one can improve the effective SNR of individual data symbols to a desired level that can guarantee a predetermined data rate. As it will be clarified in the next example, we define the effective SNR as the SNR of the individual data symbols at the end of the detection algorithm after several rounds of recursive combining. The aforementioned gain is obtained by a low-complexity detection algorithm involving only few linear operations (additions/subtractions) at each recursion, as detailed in Section III-B2. Therefore, this scheme is useful especially when the transmission of each symbol over each RE is constrained by a maximum power limit which is often the case in low-capacity scenarios as discussed in Section I. To proceed, we begin with the following illustrative example that helps clarifying the design procedure and the recursive detection algorithm provided afterward.

**Example 1.** Consider the transmission of  $K = 12$  users (symbols) over  $M = 12$  REs realized using the Kronecker product of the following two square factor matrices

$$\mathbf{P}_{3 \times 3}^{(1)} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}, \quad \mathbf{P}_{4 \times 4}^{(2)} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}. \quad (7)$$

Let us further define the following square matrices  $\boldsymbol{\alpha}^{(l)}$ ,  $l = 1, 2$ , such that the matrix product  $\boldsymbol{\alpha}^{(l)} \mathbf{P}^{(l)}$  is diagonal. As it will be clarified after this example and the detailed detection algorithm in Section III-B2, combining the received signals according to the rows of matrices  $\boldsymbol{\alpha}^{(l)}$ ’s significantly reduces the number of unknown data symbols after each recursion.

$$\boldsymbol{\alpha}_{3 \times 3}^{(1)} = \begin{bmatrix} 1 & 1 & -1 \\ 1 & -1 & 1 \\ -1 & 1 & 1 \end{bmatrix}, \quad \boldsymbol{\alpha}_{4 \times 4}^{(2)} = \begin{bmatrix} 0 & -1 & 1 & 1 \\ 0 & 1 & -1 & 1 \\ 0 & 1 & 1 & -1 \\ 1 & 0 & 0 & 0 \end{bmatrix}. \quad (8)$$

Then  $\mathbf{y}_{12 \times 1} = \mathbf{G}_{12 \times 12} \mathbf{x}_{12 \times 1} + \mathbf{n}_{12 \times 1}$  defines the received signals vector over all 12 REs, where  $\mathbf{y}_{12 \times 1} \triangleq [y_1, y_2, \dots, y_{12}]^T$ ,

$\mathbf{G}_{12 \times 12} = \mathbf{P}_{3 \times 3}^{(1)} \otimes \mathbf{P}_{4 \times 4}^{(2)}$ ,  $\mathbf{x}_{12 \times 1} \triangleq [x_1, x_2, \dots, x_{12}]^T$ , and  $\mathbf{n}_{12 \times 1} \triangleq [n_1, n_2, \dots, n_{12}]^T$ . Using the definition of the Kronecker product in (6), we have, for example, for the received signals over the first four REs  $y_1 = x_4 + x_8 + n_1$ ,  $y_2 = x_2 + x_3 + x_6 + x_7 + n_2$ ,  $y_3 = x_1 + x_3 + x_5 + x_7 + n_3$ , and  $y_4 = x_1 + x_2 + x_5 + x_6 + n_4$ .

The resulting set of 12 equations for the received signals  $y_i$ 's over all REs can be analyzed using different MUD methods to detect the data symbols  $x_i$ 's,  $i = 1, 2, \dots, 12$ . However, the proposed design method in this paper enables recursive detection of the data symbols with a significantly lower complexity. Here, for the sake of brevity, we focus on the detection of  $x_1$ ,  $x_5$ , and  $x_9$ . The rest of the symbols can be detected in a similar fashion. Let us define  $y_1^{(1)}$ ,  $y_5^{(1)}$ , and  $y_9^{(1)}$  each as the linear combination of four consecutive received signals according to the first row of  $\alpha^{(2)}$ , i.e.,

$$\begin{aligned} y_1^{(1)} &\triangleq y_3 + y_4 - y_2 = 2x_1 + 2x_5 + n_1^{(1)}, \\ y_5^{(1)} &\triangleq y_7 + y_8 - y_6 = 2x_1 + 2x_9 + n_5^{(1)}, \\ y_9^{(1)} &\triangleq y_{11} + y_{12} - y_{10} = 2x_5 + 2x_9 + n_9^{(1)}, \end{aligned} \quad (9)$$

where  $n_1^{(1)} \triangleq n_3 + n_4 - n_2$ ,  $n_5^{(1)} \triangleq n_7 + n_8 - n_6$ , and  $n_9^{(1)} \triangleq n_{11} + n_{12} - n_{10}$ . In a given equation involving a certain number of data symbols and a noise term, we refer to the ratio of the power of each data symbol to the noise variance as the effective SNR of that data symbol. For instance, the effective SNR of each of the symbols  $x_1$ ,  $x_5$ , and  $x_9$  in (9) is  $4/3$  times the original SNR. This is because the noise terms  $n_i$ 's in different REs are independent, i.e., each of  $n_j^{(1)}$ 's, for  $j = 1, 5, 9$ , has a mean equal to zero and variance  $\sigma_1^2 = 3\sigma^2$ , where  $\sigma^2$  is the variance of the original noise terms  $n_i$ 's. In this case, roughly speaking, we say that the effective SNR is increased by a factor of  $4/3$  through this recursion<sup>1</sup>.

Note that the set of three equations in (9) is defined according to the factor matrix  $\mathbf{P}_{3 \times 3}^{(1)}$  as follows:

$$\begin{bmatrix} y_1^{(1)} \\ y_5^{(1)} \\ y_9^{(1)} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 2x_1 \\ 2x_5 \\ 2x_9 \end{bmatrix} + \begin{bmatrix} n_1^{(1)} \\ n_5^{(1)} \\ n_9^{(1)} \end{bmatrix}. \quad (10)$$

Therefore, combining the three new symbols according to the rows of  $\alpha^{(1)}$  results in the following set of equations involving only one data symbol, also referred to as singleton equations,

$$\begin{aligned} y_1^{(2)} &\triangleq y_1^{(1)} + y_5^{(1)} - y_9^{(1)} = 4x_1 + n_1^{(2)}, \\ y_5^{(2)} &\triangleq y_1^{(1)} + y_9^{(1)} - y_5^{(1)} = 4x_5 + n_5^{(2)}, \\ y_9^{(2)} &\triangleq y_5^{(1)} + y_9^{(1)} - y_1^{(1)} = 4x_9 + n_9^{(2)}, \end{aligned} \quad (11)$$

in which  $n_1^{(2)} \triangleq n_1^{(1)} + n_5^{(1)} - n_9^{(1)}$ ,  $n_5^{(2)} \triangleq n_1^{(1)} + n_9^{(1)} - n_5^{(1)}$ , and  $n_9^{(2)} \triangleq n_5^{(1)} + n_9^{(1)} - n_1^{(1)}$ . Hence, the effective SNR is increased again by a factor of  $4/3$  since the noise components

<sup>1</sup>Note that, as it will be seen at the end of Example 1 and also will be demonstrated in Section III-B2 (see also Fig. 1), the proposed detection algorithm for the Kronecker product of  $L$  square factor matrices reduces (at the end of the  $L$ -th recursion) to singleton equations each containing a single data symbol mixed with an additive noise term. Therefore, the notion of “effective SNR”, defined here, is more relevant than the common notion of signal-to-interference-plus-noise ratio (SINR).

$n_j^{(2)}$ 's have mean zero and variance  $\sigma_2^2 = 3\sigma_1^2 = 3^2\sigma^2$  due to the independence of  $n_j^{(1)}$ 's. Finally, (11) can be used to decode the original data symbols  $x_j$ 's though with the effective SNRs increased by a factor of  $(4/3)^2$ . Applying a similar method demonstrates that the effective SNR of the data symbols  $x_4$ ,  $x_8$ , and  $x_{12}$  is improved by a factor of  $4/3$  while the gain on all other data symbols is  $(4/3)^2$  (please refer to Section III-B3 and Fig. 2 for the general characterization of the effective SNR development through the proposed recursive detection algorithm in Section III-B2).  $\square$

The above example illustrates the idea behind the proposed structure for the square pattern matrix and shows its potential advantages by enabling a low-complexity recursive detection process. However, there are several important questions that need to be carefully addressed. In particular, what is the criteria for selecting the factor matrices? How can the received signals in different REs be combined to get smaller dimensions and simpler sets of equations (e.g., converting the original equations for  $y_i$ 's to (9) and then (9) to (11))? What exactly will be the gain of such a combining in terms of increasing the effective SNRs? And, how many equations and with what dimensions will be left at the end to perform advanced detection algorithms? Next, we aim at properly answering these questions with respect to a square pattern matrix factorized as the Kronecker product of smaller square factor matrices.

1) *Pattern Matrix Design and Recursive Combining:* With the pattern matrix structure,  $\mathbf{G}_{M \times K} = \mathbf{P}_{m_1 \times m_1}^{(1)} \otimes \mathbf{P}_{m_2 \times m_2}^{(2)} \otimes \dots \otimes \mathbf{P}_{m_L \times m_L}^{(L)}$ , both the combining procedure of the received signals over different REs and the resulting SNR gains, in each recursion, directly relate to the underlying square factor matrices  $\mathbf{P}^{(l)}$ 's. As it will be elaborated in Section III-B2, our proposed detection algorithm starts from the rightmost factor matrix  $\mathbf{P}^{(L)}$  and involves  $L$  recursions. We will further establish that the  $l'$ -th recursion, for  $l' = 1, 2, \dots, L$ , involves equations defined according to the factor matrix  $\mathbf{P}^{(l')}$  with  $l = L - l' + 1$ . Now, Let  $\mathbf{P}^{(l,v)}$ , for  $v = 1, 2, \dots, \binom{2^{m_l}-1}{m_l}$ , denote the  $v$ -th possible matrix for  $\mathbf{P}^{(l)}$ . Then at the beginning of the  $l'$ -th recursion we have certain sets of auxiliary equations of the following form (see, e.g., (10)):

$$\begin{bmatrix} y_{i_1}^{(l'-1)} \\ y_{i_2}^{(l'-1)} \\ \vdots \\ y_{i_{m_l}}^{(l'-1)} \end{bmatrix} = \begin{bmatrix} p_{1,1}^{(l,v)} & p_{1,2}^{(l,v)} & \dots & p_{1,m_l}^{(l,v)} \\ p_{2,1}^{(l,v)} & p_{2,2}^{(l,v)} & \dots & p_{2,m_l}^{(l,v)} \\ \vdots & \vdots & \ddots & \vdots \\ p_{m_l,1}^{(l,v)} & p_{m_l,2}^{(l,v)} & \dots & p_{m_l,m_l}^{(l,v)} \end{bmatrix} \begin{bmatrix} x_{i_1}^{(l'-1)} \\ x_{i_2}^{(l'-1)} \\ \vdots \\ x_{i_{m_l}}^{(l'-1)} \end{bmatrix} + \begin{bmatrix} n_{i_1}^{(l'-1)} \\ n_{i_2}^{(l'-1)} \\ \vdots \\ n_{i_{m_l}}^{(l'-1)} \end{bmatrix}, \quad (12)$$

where  $p_{i,j}^{(l,v)}$  is the  $(i, j)$ -th element of the factor matrix  $\mathbf{P}^{(l,v)}$ ,  $\{i_1, i_2, \dots, i_{m_l}\}$  is a length- $m_l$  subset of  $\{1, 2, \dots, M\}$ , and  $y_i^{(l'-1)}$ ,  $x_i^{(l'-1)}$ , and  $n_i^{(l'-1)}$  are the  $i$ -th auxiliary received signal, the data symbol, and the noise component, respectively, at the beginning of the  $l'$ -th (end of the  $(l' - 1)$ -st) recursion.

In order to facilitate a low-complexity recursive detection while providing the maximum increase in the effective SNRs, we obtain the effective combining coefficients (such as combining  $y_i$ 's in (9)) as follows. Given the  $m_l \times m_l$  square factor matrix  $\mathbf{P}^{(l,v)}$ , as in (12), we find all  $T$  possible combining

**Algorithm 1** Calculation of the combining coefficients and the corresponding SNR gains.

---

1: Input: dimension of the square factor matrix  $m_l$   
 2: Output: combining matrices  $\alpha^{(l,v)} = [\alpha_{i,j}^{(l,v)}]_{m_l \times m_l}$  and the corresponding sets  $\{\gamma_i^{(l,v)}\}_{i=1}^{m_l}$  of SNR gains for all  $v$ 's  
 3: **for**  $v = 1 : (2^{m_l} - 1)$  **do**  
 4:    $\mathbf{P}^{(l,v)} = [p_{i,j}^{(l,v)}]_{m_l \times m_l}$   
 5:   **for**  $i = 1 : m_l$  **do**  
 6:     **Find** all  $T$  possible sets of coefficients  $\{\alpha_{i,j}^{(l,v,t)}\}_{j=1}^{m_l}$  such that:  
 7:       **C1:**  $\alpha_{i,j}^{(l,v,t)} \in \{-1, 0, 1\}$   
 8:       **C2:**  $\sum_{j=1}^{m_l} \alpha_{i,j}^{(l,v,t)} p_{j,i}^{(l,v)} = w_i^{(l,v,t)} \neq 0$   
 9:       **C3:**  $\sum_{j=1}^{m_l} \alpha_{i,j}^{(l,v,t)} p_{j,i'}^{(l,v)} = 0, i' \neq i = 1, 2, \dots, m_l$   
 10:      **Calculate**  $\gamma_i^{(l,v,t)} \triangleq [w_i^{(l,v,t)}]^2 / \sum_{j=1}^{m_l} [\alpha_{i,j}^{(l,v,t)}]^2$   
 11:      **Calculate**  $t^* = \operatorname{argmax}_{t=1:T} \gamma_i^{(l,v,t)}$   
 12:      **Output**  $\{\alpha_{i,j}^{(l,v)}\}_{j=1}^{m_l} = \{\alpha_{i,j}^{(l,v,t^*)}\}_{j=1}^{m_l}$   
 13:      **Output**  $\gamma_i^{(l,v)} = \gamma_i^{(l,v,t^*)}$   
 14:   **end for**  
 15: **end for**

---

matrices of size  $m_l \times m_l$ , with the  $t$ -th such matrix denoted by  $\alpha^{(l,v,t)}$ , for  $t = 1, 2, \dots, T$ , such that  $\alpha^{(l,v,t)} \mathbf{P}^{(l,v)}$  is a diagonal matrix with the nonzero diagonal entries equal to  $w_i^{(l,v,t)}$ , for  $i = 1, 2, \dots, m_l$ .<sup>2</sup> The  $(i, j)$ -th entry of the matrix  $\alpha^{(l,v,t)}$  is denoted by  $\alpha_{i,j}^{(l,v,t)} \in \{-1, 0, +1\}$ . Now, by linearly combining the  $m_l$  auxiliary received signals in (12) according to the rows of  $\alpha^{(l,v,t)}$  we get  $m_l$  new equations as

$$\sum_{j=1}^{m_l} \alpha_{i',j}^{(l,v,t)} y_{i,j}^{(l'-1)} = w_{i'}^{(l,v,t)} x_{i'}^{(l'-1)} + \sum_{j=1}^{m_l} \alpha_{i',j}^{(l,v,t)} n_{i,j}^{(l'-1)}, \quad (13)$$

for  $i' = 1, 2, \dots, m_l$ , where  $w_{i'}^{(l,v,t)} \triangleq \sum_{j=1}^{m_l} \alpha_{i',j}^{(l,v,t)} p_{j,i'}^{(l,v)}$ .

The process for selecting the combining coefficients that yield the maximum SNR gain for each auxiliary data symbol is summarized in Algorithm 1. Condition **C1** in Algorithm 1 defines the set of possible values for the combining coefficients. This implies that a specific row in  $\mathbf{P}^{(l,v)}$  is included in the combining process with a positive/negative sign or it is not included at all. Note that scaling the set of possible values  $\{-1, 0, +1\}$  by a constant factor does not change the SNR gains and the performance since it scales both the data symbols and the noise coefficients in (13) by the same multiplicative factor. Moreover, conditions **C2** and **C3** are included to make sure that the  $t$ -th possible combining matrix  $\alpha^{(l,v,t)}$  results in a diagonal matrix with the nonzero diagonal

<sup>2</sup>If for a given  $\mathbf{P}^{(l,v)}$  there is no any combining matrix that results in a diagonal form, with nonzero diagonal entries, for the matrix multiplication of  $\alpha^{(l,v)} \mathbf{P}^{(l,v)}$ , we skip that matrix and do not save its attributes. Additionally, if for a given dimension  $m_l$  none of  $(2^{m_l} - 1)$  candidates satisfy the aforementioned property, we do not consider that dimension (i.e.,  $m_l \times m_l$  square factor matrices) in the design of the overall pattern matrix.

entries  $w_i^{(l,v,t)}$ 's in  $\alpha^{(l,v,t)} \mathbf{P}^{(l,v)}$ . This is required to guarantee singleton equations, such as (13), in terms of the auxiliary data symbols  $x_{i'}^{(l'-1)}$ 's at the end of the  $l'$ -th recursion. With these constraints we can ensure that after the  $l'$ -th recursion the maximum number of data symbols in each equation is reduced by a factor of  $m_{L-l'+1}$ , resulting in much simpler sets of equations. This way we come up with a very low-complexity recursive detection algorithm described in Section III-B2. Note that, except for the last recursion ( $l' = L$ ), each auxiliary data symbol  $x_{i'}^{(l'-1)}$  involves a combination of several original data symbols  $x_i$ 's defined according to  $\mathbf{P}_{m_1 \times m_1}^{(1)} \otimes \mathbf{P}_{m_2 \times m_2}^{(2)} \otimes \dots \otimes \mathbf{P}_{m_{L-l'} \times m_{L-l'}}^{(L-l')}$ , i.e., the Kronecker product of the  $L - l'$  leftmost factor matrices. For instance, in Example 1, the auxiliary data symbol  $x_1^{(0)}$  at the beginning of the first recursion is equal to  $x_1 + x_5$  (see (9)), which is defined according to  $\mathbf{P}_{3 \times 3}^{(1)}$  as specified in (10).

After combining the auxiliary received signals in (12) according to the combining coefficients  $\{\alpha_{i',j}^{(l,v,t)}\}_{j=1}^{m_l}$ , it can be observed from (13) that the effective SNR of the  $i_{i'}$ -th auxiliary data symbol  $x_{i'}^{(l'-1)}$  is increased by a factor of  $\gamma_i^{(l,v,t)}$ , defined in line 10 of Algorithm 1. This is because the auxiliary noise components in (12) are independent, which is clarified when we describe the detection algorithm in Section III-B2. Finally, Algorithm 1, among all  $T$  possible sets of coefficients  $\{\alpha_{i,j}^{(l,v,t)}\}_{j=1}^{m_l}$  that satisfy the three constraints **C1-3** for each  $i$ ,  $v$ , and  $l$ , picks the one that maximizes the corresponding SNR gain  $\gamma_i^{(l,v,t)}$  as the  $i$ -th row of the combining matrix  $\alpha^{(l,v)}$ . By repeating this procedure for all  $i$ 's, for  $i = 1, 2, \dots, m_l$ , we get the whole combining matrix  $\alpha^{(l,v)} = [\alpha_{i,j}^{(l,v)}]_{m_l \times m_l}$  and the corresponding sets  $\{\gamma_i^{(l,v)}\}_{i=1}^{m_l}$  of SNR gains given the  $v$ -th possible square factor matrix  $\mathbf{P}^{(l,v)}$  for the  $l$ -th leftmost square factor matrix used in the Kronecker product to form the overall pattern matrix.

**Remark 1.** Algorithm 1 does not output a specific square factor matrix for the  $l$ -th position in the Kronecker product. Instead, it finds the best combining coefficients (i.e., the combining matrix  $\alpha^{(l,v)}$ ) and the resulting sets of SNR gains for all  $(2^{m_l} - 1)$  possible square factor matrices  $\mathbf{P}_{m_l \times m_l}^{(l,v)}$ . Among all these possibilities for  $v$ , the *best* one represented by the index  $v^*$  that results in the *optimal* square factor matrix  $\mathbf{P}^{(l)}$  with the corresponding set  $\{\gamma_1^{(l)}, \gamma_2^{(l)}, \dots, \gamma_{m_l}^{(l)}\}$  of SNR gains and the optimal combining matrix  $\alpha^{(l)}$  can be obtained by choosing the best answer set satisfying further constraints such as maximizing the average sum rate (see also Remark 3). Note that, in the case of GMAC model considered throughout Sections III and IV, Algorithm 1 needs to run only once per dimension  $m_l$  of the square factor matrix (please see Remark 9 for the clarifications on the case of fading channels). Once the *best*  $\mathbf{P}^{(l)}$  and the corresponding  $\alpha^{(l)}$  are obtained for a given dimension  $m_l$ , they will be known to all users and the BS. Note also that this is a part of the system design and not the detection algorithm, and one needs to also take into account the number of users  $K$  and available REs  $M$  in the design of the overall pattern matrix and choosing the dimensions of the underlying factor matrices.

Finally, the following lemma is useful in the process of finding the combining coefficients described in Algorithm 1.

**Lemma 1.** *For a binary square factor matrix  $\mathbf{P}$  with linearly-independent rows, we have  $T \leq 1$ . In other words, there exists at most one combining matrix  $\alpha = [\alpha_{i,j}]$ ,  $\alpha_{i,j} \in \{-1, 0, 1\}$ , that results in a diagonal form for the matrix multiplication  $\alpha\mathbf{P}$  with nonzero diagonal entries.*

*Proof:* Please refer to Appendix A.  $\blacksquare$

2) *Recursive Detection Algorithm:* Suppose that the communication protocol is established between the transmitters and receivers, i.e., the design parameters such as the optimal square factor matrices  $\mathbf{P}_{m_l \times m_l}^{(l)}$ , for  $l = 1, 2, \dots, L$ , and their corresponding combining matrices  $\alpha_{m_l \times m_l}^{(l)}$ 's from Algorithm 1 (also recall Remark 1) are known to the receiver. Next, we describe the low-complexity recursive detection algorithm thanks to the underlying structure of the overall pattern matrix. The proposed detection algorithm, which is schematically shown through a tree diagram in Fig. 1, proceeds by combining the received signals according to the combining matrix of the rightmost factor matrix and involves  $L$  recursions.

**First Recursion:** The receiver in the first recursion takes the vector of received signals over  $M = \prod_{l=1}^L m_l$  REs, i.e.,  $y_i$ 's, for  $i = 1, 2, \dots, M$ , and divides them into  $M_{L-1} \triangleq M/m_L = \prod_{l=1}^{L-1} m_l$  groups of  $m_L$  received signals; therefore, the  $i_L$ -th group, for  $i_L = 1, 2, \dots, M_{L-1}$ , includes  $\{y_{(i_L-1)m_L+1}, y_{(i_L-1)m_L+2}, \dots, y_{i_L m_L}\}$ . Each of these  $y_i$ 's contains at most  $K = \prod_{l=1}^L m_l$  different transmitted symbols  $x_k$ 's,  $k = 1, 2, \dots, K$ , since each one is constructed as a linear combination of  $x_k$ 's, defined with respect to the  $i$ -th row of the overall pattern matrix  $\mathbf{G}_{M \times K}$ , and the noise component  $n_i$  over the  $i$ -th RE. Note that our recursive detection algorithm attempts to reduce the maximum number of different symbols by a factor of  $m_{l''}$ ,  $l'' \triangleq L - l' + 1$ , after each  $l'$ -th recursion, such that after all  $L$  recursions we have  $M$  equations each containing only a single data symbol. The receiver now combines the elements of each group using the combining matrix  $\alpha_{m_L \times m_L}^{(L)}$  to form new  $m_L$  symbols at each group; the first new symbol is constructed by combining the previous symbols using the first row of  $\alpha^{(L)}$  and so on. Note that the impact of such a combining on the involved data symbols can be expressed in a matrix form through multiplying  $\alpha^{(L)}$  by a new matrix comprising the  $m_L$  rows of  $\mathbf{G}$  corresponding to the indices of each group. Then it is easy to verify that the  $j_L$ -th new equation of each group (which is constructed through the  $j_L$ -th row of  $\alpha^{(L)}$ ), for  $j_L = 1, 2, \dots, m_L$ , contains at most  $K_{L-1} \triangleq K/m_L = \prod_{l=1}^{L-1} m_l$  different symbols from the set  $\{x_{j_L}, x_{j_L+m_L}, \dots, x_{j_L+(K_{L-1}-1)m_L}\}$  (recall that the product of  $\alpha^{(L)}\mathbf{P}^{(L)}$  is a diagonal matrix), i.e., the number of unknown variables is reduced by a factor of  $m_L$  from  $K$  to  $K_{L-1}$ . Also, based on the detailed discussion in Section III-B1 (see, e.g., Eq. (13)), the effective SNR of each symbol in the  $j_L$ -th new equation is increased by a factor of  $\gamma_{j_L}^{(L)}$ .

**Second Recursion:** Note that after the first recursion the  $j_L$ -th new equation of each group can only contain data symbols from the set  $\{x_{j_L}, x_{j_L+m_L}, \dots, x_{j_L+(K_{L-1}-1)m_L}\}$ . This means different equations of a given group contain

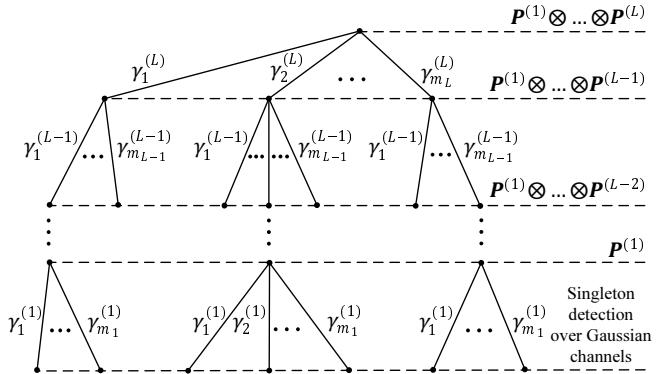


Fig. 1. Tree diagram representation of the recursive detection algorithm.

disjoint sets of data symbols while equations with the same index of different groups (e.g., the  $j_L$ -th equation of all groups) contain symbols from the same set. Therefore, the receiver in the second recursion forms  $m_L$  super-groups of  $M_{L-1}$  equations/signals by consecutively, placing the  $j_L$ -th new equation of each of those  $M_{L-1}$  groups of the first recursion into the  $j_L$ -th super-group. Note that the combination of the original data symbols over each super-group (in Section III-B1, we denoted each of those combinations by an auxiliary data symbol obtained after the first recursion) is defined according to the Kronecker product  $\mathbf{P}_{m_1 \times m_1}^{(1)} \otimes \mathbf{P}_{m_2 \times m_2}^{(2)} \otimes \dots \otimes \mathbf{P}_{m_{L-1} \times m_{L-1}}^{(L-1)}$ . Now, the receiver follows exactly the same procedure as the first recursion over each of these disjoint  $m_L$  super-groups of the size  $M_{L-1}$ . In other words, it divides the equations in each of the super-groups into  $M_{L-2} \triangleq M_{L-1}/m_{L-1} = \prod_{l=1}^{L-2} m_l$  groups of  $m_{L-1}$  equations each and combines the signals within each group using the combining matrix  $\alpha_{m_{L-1} \times m_{L-1}}^{(L-1)}$ . Following the same logic, we argue that the maximum number of unknown variables at each of the new equations is reduced by a factor of  $m_{L-1}$  from  $K_{L-1}$  to  $K_{L-2} \triangleq K_{L-1}/m_{L-1} = \prod_{l=1}^{L-2} m_l$ , and the SNR of the symbols in the  $j_{L-1}$ -st new equation, for  $j_{L-1} = 1, 2, \dots, m_{L-1}$ , of each of the groups in the  $j_L$ -th super-group is increased by a factor of  $\gamma_{j_{L-1}}^{(L-1)}$  from  $\gamma_{j_L}^{(L)}$  in the first recursion to  $\gamma_{j_{L-1}}^{(L-1)}\gamma_{j_L}^{(L)}$ .

**Final Recursion:** It can be verified by induction that in the  $L$ -th recursion we have  $\prod_{l=2}^L m_l$  super-groups of size  $m_1$ . The sets of  $m_1$  equations in each of these  $\prod_{l=2}^L m_l$  super-groups is formed according to the square factor matrix  $\mathbf{P}_{m_1 \times m_1}^{(1)}$ . Therefore, the receiver in the  $L$ -th recursion combines the  $m_1$  symbols of each super-group using  $\alpha_{m_1 \times m_1}^{(1)}$  to get equations containing a single unknown variable. Finally, the receiver detects the original  $K = M$  data symbols according to the  $M$  singleton equations in the form of a single-user additive white Gaussian noise (AWGN) channel. However, the effective SNR of each of these  $K$  data symbols is increased to a desired level that can guarantee a predetermined data rate, particularly for low-capacity applications.

3) *Detection and SNR Evolution Trees:* For the sake of clarity, we have separately explained the second and the final recursions in Section III-B2. Note that the same procedure is applied in all of the recursion steps for  $l' = 2, \dots, L$ . In particular, the receiver at the beginning of the  $l'$ -th recursion, first, forms  $\prod_{l=l''+1}^L m_l$  super-groups each containing

$M_{l''} \triangleq \prod_{l=1}^{l''} m_l$  equations/signals defined according to the pattern matrix  $\mathbf{P}_{m_1 \times m_1}^{(1)} \otimes \mathbf{P}_{m_2 \times m_2}^{(2)} \otimes \cdots \otimes \mathbf{P}_{m_{l''} \times m_{l''}}^{(l'')}$ , where  $l'' \triangleq L - l' + 1$ . The receiver then applies similar steps to the first recursion, i.e., it combines the equations/signals inside each super-group according to the rightmost combining matrix  $\alpha_{m_{l''} \times m_{l''}}^{(l'')}$  to reduce the (maximum) number of unknown data symbols included in each equation/signal by a factor of  $m_{l''}$  from  $K_{l''} \triangleq \prod_{l=1}^{l''} m_l$  to  $K_{l''-1}$ . Finally, as a result of this combining, the effective SNR of each data symbol involved in the  $j_{l''}$ -th combined signal,  $j_{l''} = 1, 2, \dots, m_{l''}$ , is increased by a factor of  $\gamma_{j_{l''}}^{(l'')}$ . The whole process is, schematically, shown in Fig. 1. Each node in the tree diagram of Fig. 1 represents a super-group while the weights of the edges characterize the SNR gains after combining the signals inside each super-group. Moreover, the pattern matrix governing the way data symbols involved in each super-group, at the beginning and end of each recursion, are merged together is also specified in Fig. 1 which further clarifies how after  $L$  recursions we end up with  $M$  singleton equations over single-user AWGN channels with increased effective SNRs.

It is worth mentioning that the  $M$  overall SNR gains obtained by multiplying the edge weights involved from the topmost node to each of the  $M$  bottommost nodes in Fig. 1 specify the set of  $M$  overall SNR gains for the  $M$  data symbols in an *unsorted* fashion. In order to get the overall SNR gains *sorted*, we need to shuffle the edge weights, i.e., the SNR gains, according to Fig. 2. This way we can assure that the overall SNR gain obtained by multiplying the edge weights involved from the topmost node to each  $i$ -th bottommost node in Fig. 2, for  $i = 1, 2, \dots, M$ , exactly specifies the overall SNR gain on the  $i$ -th data symbol  $x_i$  after  $L$  layers of combining. This can be understood from the detection algorithm elaborated in Section III-B2, especially the parts emphasizing on the data symbols that remain in each new combined equation/signal and the gain those data symbols attain after each combining. Now, according to Fig. 2, we have the following lemma for the overall SNR gain on the  $i$ -th data symbol, denoted by  $\gamma_{t,i}$ .

**Lemma 2.** *The overall SNR gain  $\gamma_{t,i}$  on the  $i$ -th data symbol, for  $i = 1, 2, \dots, M$ , can be obtained as*

$$\gamma_{t,i} = \prod_{l=1}^L \gamma_{s_l}^{(l)}, \quad (14)$$

where the integers  $s_l \in \{1, 2, \dots, m_l\}$ , for  $l = 1, 2, \dots, L$ , constitute a unique representation of  $i$  as follows:

$$i = s_L + \sum_{l=1}^{L-1} (s_{L-l} - 1) \times \prod_{l=L-l+1}^L m_{l_1}. \quad (15)$$

Equivalently,  $x_i$  is the single data symbol remained in the singleton equation of the final recursion indexed by the path  $(s_L, s_{L-1}, \dots, s_2, s_1)$  of super-groups in Fig. 1.

**Example 2.** For the factor matrices considered in Example 1 and their corresponding combining matrices, according to the detailed discussions in Section III-B1, we have  $\gamma_1^{(1)} = \gamma_2^{(1)} = \gamma_3^{(1)} = \gamma_1^{(2)} = \gamma_2^{(2)} = \gamma_3^{(2)} = 4/3$ , and  $\gamma_4^{(2)} = 1$ . Then,

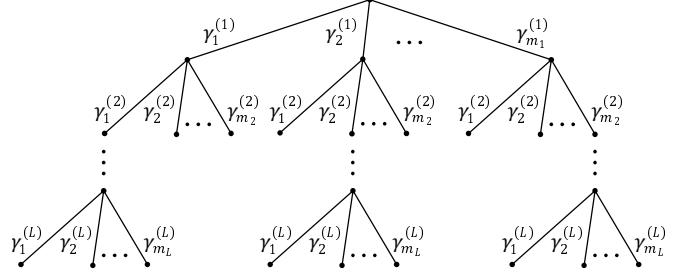


Fig. 2. Tree diagram illustrating the progressive development of SNR gains.

according to Lemma 2, we have, for instance,  $s_1 = 2$  and  $s_2 = 4$  for  $i = 8$ . Hence,  $\gamma_{t,8} = \gamma_2^{(1)} \gamma_4^{(2)} = 4/3$ . Similarly,  $\gamma_{t,4} = \gamma_{t,12} = 4/3$ , and  $\gamma_{t,i} = (4/3)^2$ ,  $i \notin \{4, 8, 12\}$ .  $\square$

### C. Rectangular Factor Matrices

In this subsection, we describe the detection algorithm when the data symbols are mixed using the Kronecker product of rectangular factor matrices over GMAC. Specifically, we consider the overall pattern matrix as  $\mathbf{G}_{M \times K} = \mathbf{F}_{m_1 \times k_1}^{(1)} \otimes \mathbf{F}_{m_2 \times k_2}^{(2)} \otimes \cdots \otimes \mathbf{F}_{m_L \times k_L}^{(L)}$ , where  $\mathbf{F}_{m_l \times k_l}^{(l)}$ , for  $l = 1, 2, \dots, L$ , is an  $m_l \times k_l$  binary matrix with  $k_l > m_l$ . In this case, the overload factor  $\beta = K/M$  is greater than 1 resulting in an improved spectral efficiency compared to the case of square factor matrices considered in Section III-B. We begin with the following illustrative example that helps better understand the recursive detection algorithm provided afterward.

**Example 3.** Consider the transmission of  $K = 18$  users (symbols) over  $M = 4$  REs realized using the Kronecker product of the following rectangular factor matrices

$$\mathbf{F}_{1 \times 2}^{(1)} = [1 \ 1], \quad \mathbf{F}_{2 \times 3}^{(2)} = \mathbf{F}_{2 \times 3}^{(3)} = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}. \quad (16)$$

Hence,  $\mathbf{y}_{4 \times 1} = \mathbf{G}_{4 \times 18} \mathbf{x}_{18 \times 1} + \mathbf{n}_{4 \times 1}$ , with  $\mathbf{G}_{4 \times 18} = \mathbf{F}_{1 \times 2}^{(1)} \otimes \mathbf{F}_{2 \times 3}^{(2)} \otimes \mathbf{F}_{2 \times 3}^{(3)}$ , represents the received signals vector.

The detection starts by defining 6 auxiliary symbols  $Z_{3i+j}^{(3)}$ , for  $i = 0, 1$  and  $j = 1, 2, 3$ , each obtained by spreading the  $j$ -th data symbol of  $\mathbf{F}^{(3)}$  according to the  $(i+1)$ -st row of  $\mathbf{R}^{(2)} \triangleq \mathbf{F}^{(1)} \otimes \mathbf{F}^{(2)}$ , i.e.,

$$\begin{aligned} Z_j^{(3)} &= x_j + x_{j+6} + x_{j+9} + x_{j+15}, \\ Z_{3+j}^{(3)} &= x_j + x_{j+3} + x_{j+9} + x_{j+12}. \end{aligned} \quad (17)$$

Now, it is easy to observe that the original set of equations can be rewritten as the following two sets of equations each defined according to  $\mathbf{F}^{(3)}$  as

$$\begin{aligned} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} &= \mathbf{F}_{2 \times 3}^{(3)} \begin{bmatrix} Z_1^{(3)} & Z_2^{(3)} & Z_3^{(3)} \end{bmatrix}^T + \begin{bmatrix} n_1 \\ n_2 \end{bmatrix}, \\ \begin{bmatrix} y_3 \\ y_4 \end{bmatrix} &= \mathbf{F}_{2 \times 3}^{(3)} \begin{bmatrix} Z_4^{(3)} & Z_5^{(3)} & Z_6^{(3)} \end{bmatrix}^T + \begin{bmatrix} n_3 \\ n_4 \end{bmatrix}. \end{aligned} \quad (18)$$

Therefore, at the beginning of the first recursion in the receiver, the sets of equations are formed according to  $\mathbf{F}_{2 \times 3}^{(3)}$  as (18). The receiver first detects  $Z_1^{(3)}, Z_2^{(3)}, \dots, Z_6^{(3)}$ , using any MUD algorithm performed over the set of equations in (18), and then groups  $Z_j^{(3)}$  with  $Z_{3+j}^{(3)}$  to obtain three sets of equations

each defined according to  $\mathbf{R}^{(2)}$ . Let us consider the first group containing  $Z_1^{(3)}$  and  $Z_4^{(3)}$ . According to (17),

$$\begin{bmatrix} Z_1^{(3)} \\ Z_4^{(3)} \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}}_{\mathbf{F}_{2 \times 3}^{(2)}} \begin{bmatrix} Z_1^{(2)} \\ Z_2^{(2)} \\ Z_3^{(2)} \end{bmatrix}, \quad (19)$$

after setting  $Z_1^{(2)} \triangleq x_1 + x_{10}$ ,  $Z_2^{(2)} \triangleq x_4 + x_{13}$ ,  $Z_3^{(2)} \triangleq x_7 + x_{16}$  each according to  $\mathbf{R}^{(1)} \triangleq \mathbf{F}^{(1)}$ . Now, the receiver first detects  $Z_1^{(2)}$ ,  $Z_2^{(2)}$ , and  $Z_3^{(2)}$ , using any arbitrary MUD algorithm, from the system of equations in (19) defined according to  $\mathbf{F}_{2 \times 3}^{(2)}$ , and then detects the corresponding 6 original data symbols from the three subsequent sets of equations each defined according to  $\mathbf{F}^{(1)}$ . Finally, by applying the same procedure to the other two groups, we detect all of the 18 original data symbols.  $\square$

The above example illustrates how the detection proceeds recursively by processing, at each recursion, sets of equations with smaller number of variables each formed according to one of the rectangular factor matrices. In the following, we present the detailed description of the recursive detection algorithm over rectangular factor matrices. Same as in Section III-B2, we assume that the rectangular factor matrices  $\mathbf{F}_{m_l \times k_l}^{(l)}$ , for  $l = 1, 2, \dots, L$ , are known to the receiver. The proposed detection algorithm entails  $L$  recursions while the  $l'$ -th recursion, for  $l' = 1, 2, \dots, L$ , involves sets of equations formed by the factor matrix  $\mathbf{F}_{m_{l''} \times k_{l''}}^{(l'')}$  with  $l'' \triangleq L - l' + 1$ . Note that when  $L = 1$  we directly detect the  $K$  data symbols mixed over  $M$  REs using the single factor matrix. Hence, in the algorithm it is assumed that  $L > 1$ . Also, no specific structure is imposed here on the rectangular factor matrices. Instead, we assume a generic form for the factor matrices such that a given advanced MUD algorithm can properly work to detect the unknown variables of the  $l'$ -th recursion defined according to  $\mathbf{F}_{m_{l''} \times k_{l''}}^{(l')}$ . However, one can impose further constraints on the rectangular factor matrices to improve the detection performance of the scheme at each recursion. For example, one can potentially apply the results of [32] to more efficiently design each of the middle rectangular factor matrices and improve the detection performance of the MUDs performed over each of them at each recursion. Further investigation on this matter is left for future studies. Throughout this subsection, we define  $\mathbf{R}_{M_l \times K_l}^{(l)} \triangleq \mathbf{F}_{m_1 \times k_1}^{(1)} \otimes \mathbf{F}_{m_2 \times k_2}^{(2)} \otimes \dots \otimes \mathbf{F}_{m_l \times k_l}^{(l)}$  with  $M_l \triangleq \prod_{i=1}^l m_i$  and  $K_l \triangleq \prod_{i=1}^l k_i$ ,  $l = 1, 2, \dots, L$ .

**First Recursion:** Given that  $\mathbf{G}_{M \times K} = \mathbf{R}_{M_{L-1} \times K_{L-1}}^{(L-1)} \otimes \mathbf{F}_{m_L \times k_L}^{(L)}$ , in the first recursion, the receiver forms sets of equations each defined according to  $\mathbf{F}_{m_L \times k_L}^{(L)}$ . To this end, it detects  $Q_L \triangleq k_L M_{L-1}$  new/auxiliary symbols  $Z_{k_L i_L + j_L}^{(L)}$ , for  $i_L = 0, 1, \dots, M_{L-1} - 1$  and  $j_L = 1, 2, \dots, k_L$ , each defined as the expansion of the  $j_L$ -th data symbol of  $\mathbf{F}_{m_L \times k_L}^{(L)}$  according to the  $(i_L + 1)$ -st row of  $\mathbf{R}^{(L-1)}$  as

$$Z_{k_L i_L + j_L}^{(L)} = \mathbf{r}_{i_L+1}^{(L-1)} [x_{j_L} \ x_{j_L+k_L} \ \dots \ x_{j_L+(K_{L-1}-1)k_L}]^T, \quad (20)$$

where  $\mathbf{r}_{i_L+1}^{(L-1)}$  is the  $(i_L + 1)$ -st row of  $\mathbf{R}_{M_{L-1} \times K_{L-1}}^{(L-1)}$ . Using (20), the original set of equations  $\mathbf{y}_{M \times 1} = \mathbf{G}_{M \times K} \mathbf{x}_{K \times 1} +$

$\mathbf{n}_{M \times 1}$  can be decomposed into  $M_{L-1}$  sets of equations each having  $m_L$  consecutive  $y_i$ 's as the input and  $k_L$  consecutive auxiliary symbols  $Z_{k_L i_L + j_L}^{(L)}$ 's as the unknown variables. The  $(i_L + 1)$ -st such set of equations is expressed as

$$\begin{bmatrix} y_{i_L m_L + 1} \\ y_{i_L m_L + 2} \\ \vdots \\ y_{(i_L+1)m_L} \end{bmatrix} = \mathbf{F}_{m_L \times k_L}^{(L)} \begin{bmatrix} Z_{i_L k_L + 1}^{(L)} \\ Z_{i_L k_L + 2}^{(L)} \\ \vdots \\ Z_{(i_L+1)k_L}^{(L)} \end{bmatrix} + \begin{bmatrix} n_{i_L m_L + 1} \\ n_{i_L m_L + 2} \\ \vdots \\ n_{(i_L+1)m_L} \end{bmatrix}. \quad (21)$$

Therefore, the first recursion involves  $M_{L-1}$  separate sets of equations, each defined according to the rectangular factor matrix  $\mathbf{F}_{m_L \times k_L}^{(L)}$ , as specified in (21), such that  $Q_L$  auxiliary symbols are detected. Next, the receiver forms  $k_L$  disjoint super-groups, given that all of the  $M_{L-1}$  auxiliary symbols  $Z_{k_L i_L + j_L}^{(L)}$ , for  $i_L = 0, 1, \dots, M_{L-1} - 1$ , contain the same disjoint set of data symbols for a fixed  $j_L$  (see (20)). The  $j_L$ -th super-group contains  $M_{L-1}$  auxiliary symbols  $Z_{j_L}^{(L)}, Z_{k_L + j_L}^{(L)}, \dots, Z_{(M_{L-1}-1)k_L + j_L}^{(L)}$ . Therefore, the set of equations in the  $j_L$ -th super-group can be expressed as

$$\begin{bmatrix} Z_{j_L}^{(L)} \\ Z_{k_L + j_L}^{(L)} \\ \vdots \\ Z_{(M_{L-1}-1)k_L + j_L}^{(L)} \end{bmatrix} = \mathbf{R}_{M_{L-1} \times K_{L-1}}^{(L-1)} \begin{bmatrix} x_{j_L} \\ x_{k_L + j_L} \\ \vdots \\ x_{(K_{L-1}-1)k_L + j_L} \end{bmatrix}. \quad (22)$$

**Middle Recursions ( $1 < l' < L$ ):** When  $L > 2$ , the detection algorithm involves  $L - 2$  middle recursions excluding the first and the last recursions. Each of these middle recursions involves a procedure that is somewhat similar to the first recursion. For example, in the second recursion we start with  $k_L$  disjoint super-groups each containing a set of equations as (22) and then we apply similar steps as the first recursion to this set of equations. However, since the indices of the data and the auxiliary symbols matter in the detection done at each recursion, here, we carefully describe these middle recursions. In general, at the  $l'$ -th recursion, for  $1 < l' < L$ , the algorithm processes the  $\kappa_{l'-1}^S \triangleq \prod_{i=L-l'+2}^L k_i$  disjoint super-groups of the  $(l' - 1)$ -st recursion separately. It then comes up with  $k_{L-l'+1}$  smaller size super-groups for each of those  $\kappa_{l'-1}^S$  starting super-groups, resulting in  $k_{L-l'+1} \kappa_{l'-1}^S = \kappa_{l'}^S$  super-groups at the end. Particularly, consider the path  $(j_L, j_{L-1}, \dots, j_{L-l'+2})$  of the super-groups from the previous recursions, with  $j_l = 1, 2, \dots, k_l$  for  $l = 1, 2, \dots, L$ , (a similar tree diagram to Fig. 1 can be considered for the recursive detection here). The indices in the path correspond to the super-groups considered from the previous recursions while there are total of  $\kappa_{l'-1}^S$  such paths, i.e., starting super-groups. By induction, the set of equations in the super-group indexed by the path  $(j_L, j_{L-1}, \dots, j_{L-l'+2})$  can be expressed as follows:

$$\begin{bmatrix} Z_{\psi_{l'-1},1}^{(l'+1)} \\ Z_{\psi_{l'-1},2}^{(l'+1)} \\ \vdots \\ Z_{\psi_{l'-1},M_{l'}}^{(l'+1)} \end{bmatrix} = \mathbf{R}_{M_{l'} \times K_{l'}}^{(l')} \begin{bmatrix} x_{\tau_{l'-1}} \\ x_{\kappa_{l'-1}^S + \tau_{l'-1}} \\ \vdots \\ x_{(K_{l'}-1)\kappa_{l'-1}^S + \tau_{l'-1}} \end{bmatrix}, \quad (23)$$

where  $\psi_{l'-1} \triangleq j_{L-l'+2} + \sum_{i=1}^{l'-2} (j_{L-i+1} - 1) \prod_{i'=L-l'+2}^{L-i} k_{i'}$  is the index of the super-group represented by the path  $(j_L, j_{L-1}, \dots, j_{L-l'+2})$ , and  $\tau_{l'-1} \triangleq j_L + \sum_{i=1}^{l'-2} (j_{L-i} - 1) \kappa_i^S$ . Moreover,  $Z_{\psi_{l'-1}, 1}^{(l'+1)}, \dots, Z_{\psi_{l'-1}, M_{l''}}^{(l'+1)}$  are  $M_{l''}$  auxiliary symbols defined in the  $(l'-1)$ -st recursion, similar to (20), corresponding to the  $\psi_{l'-1}$ -st super-group. Note that tracking the indices of the auxiliary symbols is not needed since the algorithm processes each super-group separately. More specifically, these symbols are defined at the beginning of a recursion and they are known when the detection is done at the end of that recursion. However, the indices of the data symbols matters since we need to know the data symbols in each super-group/path and the order they appeared in the corresponding equations.

The algorithm involves  $\kappa_{l'-1}^S$  sets of equations at the beginning of the  $l'$ -th recursion. Each such a set of equations is processed separately and simpler sets of equations defined according to  $\mathbf{F}_{m_{l''} \times k_{l''}}$  are derived. Let us consider the  $\psi_{l'-1}$ -st super-group with the set of equations given by (23). Similar to the first recursion, the receiver assumes  $Q_{l''} \triangleq k_{l''} M_{L-l'}$  auxiliary symbols  $Z_{k_{l''} i_{l''} + j_{l''}}^{(l'')}$ , for  $i_{l''} = 0, 1, \dots, M_{L-l'} - 1$  and  $j_{l''} = 1, 2, \dots, k_{l''}$ , each defined as the expansion of the  $j_{l''}$ -th data symbol according to  $r_{i_{l''}+1}^{(L-l')}$ , i.e., the  $(i_{l''} + 1)$ -st row of  $\mathbf{R}_{M_{L-l'} \times K_{L-l'}}$ , as

$$Z_{k_{l''} i_{l''} + j_{l''}}^{(l'')} = \mathbf{r}_{i_{l''}+1}^{(L-l')} \left[ x_{\tau_{l'}} \ x_{\tau_{l'} + \kappa_{l'}^S} \ \dots \ x_{\tau_{l'} + (K_{L-l'} - 1) \kappa_{l'}^S} \right]^T. \quad (24)$$

Note that the  $j_{l''}$ -th data symbol of (23) is  $x_{\tau_{l'}}$  since  $\tau_{l'-1} + (j_{l''} - 1) \kappa_{l'-1}^S \triangleq \tau_{l'}$ . Now, instead of detecting the original data symbols from rather complex equations such as (23), the receiver forms  $M_{L-l'}$  simpler sets of equations each defined according to the rectangular factor matrix  $\mathbf{F}_{m_{l''} \times k_{l''}}$ . The  $(i_{l''} + 1)$ -st such set of equations can be expressed as

$$\begin{bmatrix} Z_{\psi_{l'-1}, i_{l''} m_{l''} + 1}^{(l'+1)} \\ Z_{\psi_{l'-1}, i_{l''} m_{l''} + 2}^{(l'+1)} \\ \vdots \\ Z_{\psi_{l'-1}, (i_{l''} + 1) m_{l''}}^{(l'+1)} \end{bmatrix} = \mathbf{F}_{m_{l''} \times k_{l''}}^{(l'')} \begin{bmatrix} Z_{i_{l''} k_{l''} + 1}^{(l'')} \\ Z_{i_{l''} k_{l''} + 2}^{(l'')} \\ \vdots \\ Z_{(i_{l''} + 1) k_{l''}}^{(l'')} \end{bmatrix}. \quad (25)$$

After detecting the  $Q_{l''}$  auxiliary symbols  $Z_{k_{l''} i_{l''} + j_{l''}}^{(l'')}$ 's from the sets of equations as (25), the receiver then forms  $k_{l''}$  disjoint super-groups, given that all of the  $M_{L-l'}$  auxiliary symbols  $Z_{k_{l''} i_{l''} + j_{l''}}^{(l'')}$ ,  $i_{l''} = 0, 1, \dots, M_{L-l'} - 1$ , contain the same disjoint set of data symbols for a fixed  $j_{l''}$ . The set of equations in the  $j_{l''}$ -th super-group can be expressed as

$$\begin{bmatrix} Z_{\psi_{l'}, 1}^{(l'')} \\ Z_{\psi_{l'}, 2}^{(l'')} \\ \vdots \\ Z_{\psi_{l'}, M_{l''}-1}^{(l'')} \end{bmatrix} = \mathbf{R}_{M_{l''}-1 \times K_{l''}-1}^{(l'-1)} \begin{bmatrix} x_{\tau_{l'}} \\ x_{\kappa_{l'}^S + \tau_{l'}} \\ \vdots \\ x_{(K_{l''}-1) \kappa_{l'}^S + \tau_{l'}} \end{bmatrix}, \quad (26)$$

given that  $l'' - 1 = L - l'$ . Note that in (26)  $Z_{k_{l''} i_{l''} + j_{l''}}^{(l'')}$  is denoted by  $Z_{\psi_{l'}, i_{l''}+1}^{(l'')}$  in order to be consistent with (23) given that the index of the super-group represented by the

path  $(j_L, j_{L-1}, \dots, j_{L-l'+2}, j_{L-l'+1})$  is  $\psi_{l'} \triangleq j_{L-l'+1} + \sum_{i=1}^{l'-1} (j_{L-i+1} - 1) \prod_{i'=L-l'+2}^{L-i} k_{i'}$ . Finally, repeating the same procedure for all of the  $\kappa_{l'-1}^S$  starting super-groups results in  $k_{L-l'+1} \kappa_{l'-1}^S = \kappa_{l'}^S$  super-groups, each of the form given by (26), at the end of the  $l'$ -th recursion.

**Final Recursion:** The receiver in the  $L$ -th recursion starts with  $\kappa_{L-1}^S \triangleq \prod_{i=2}^L k_i$  super-groups each containing a disjoint subset of the original data symbols of size  $k_1$ . The set of equations in the  $\psi_{L-1}$ -st such super-group, where  $\psi_{L-1} \triangleq j_2 + \sum_{i=1}^{L-2} (j_{L-i+1} - 1) \prod_{i'=2}^{L-i} k_{i'}$ , indexed by the path  $(j_L, j_{L-1}, \dots, j_2)$  can be expressed as (see, e.g., (23)):

$$\begin{bmatrix} Z_{\psi_{L-1}, 1}^{(2)} \\ Z_{\psi_{L-1}, 2}^{(2)} \\ \vdots \\ Z_{\psi_{L-1}, m_1}^{(2)} \end{bmatrix} = \mathbf{F}_{m_1 \times k_1}^{(1)} \begin{bmatrix} x_{\tau_{L-1}} \\ x_{\kappa_{L-1}^S + \tau_{L-1}} \\ \vdots \\ x_{(k_1 - 1) \kappa_{L-1}^S + \tau_{L-1}} \end{bmatrix}, \quad (27)$$

where  $\tau_{L-1} \triangleq j_L + \sum_{i=1}^{L-2} (j_{L-i} - 1) \kappa_i^S$ , and  $Z_{\psi_{L-1}, 1}^{(2)}, \dots, Z_{\psi_{L-1}, m_1}^{(2)}$  are  $m_1$  auxiliary symbols defined in the  $(L-1)$ -st recursion and their values are known at the beginning of the  $L$ -th recursion. Finally, after detecting the  $k_1$  disjoint data symbols of each of the  $\kappa_{L-1}^S$  super-groups of the form (27), the receiver obtains the values of all  $K$  data symbols. Note that the indices of the data symbols contained in each super-group is fully known according to (27).

**Remark 2.** The recursive algorithm proposed in this subsection can readily be applied to the Kronecker product of any general square factor matrices by inserting  $k_l = m_l$ ,  $l = 1, 2, \dots, L$ . In that case, any given MUD scheme can be applied to sets of equations defined according to the corresponding factor matrix at each recursion. This is while the algorithm presented in Section III-B2 provides a very low-complexity detection.

#### D. General Pattern Matrices

In this subsection, we turn our attention to the general case of pattern matrices by building upon the analysis in Sections III-B and III-C specifically on square and rectangular factor matrices, respectively. In particular, the general case of the pattern matrix is considered as

$$\mathbf{G}_{M \times K} = \mathbf{F}_{m_1 \times k_1}^{(1)} \otimes \mathbf{F}_{m_2 \times k_2}^{(2)} \otimes \dots \otimes \mathbf{F}_{m_{L_r} \times k_{L_r}}^{(L_r)} \otimes \mathbf{P}_{m'_1 \times m'_1}^{(1)} \otimes \mathbf{P}_{m'_2 \times m'_2}^{(2)} \otimes \dots \otimes \mathbf{P}_{m'_{L_s} \times m'_{L_s}}^{(L_s)}, \quad (28)$$

where  $L_r$  is the number of rectangular factor matrices  $\mathbf{F}_{m_{l_r} \times k_{l_r}}$ 's, for  $l_r = 1, 2, \dots, L_r$ , and  $L_s$  is the number of square factor matrices  $\mathbf{P}_{m'_{l_s} \times m'_{l_s}}$ 's, for  $l_s = 1, 2, \dots, L_s$ , in the scheme. Moreover, let  $M_r \triangleq \prod_{l_r=1}^{L_r} m_{l_r}$ ,  $K_r \triangleq \prod_{l_r=1}^{L_r} k_{l_r}$ , and  $M_s \triangleq \prod_{l_s=1}^{L_s} m'_{l_s}$ . Then  $M = M_r M_s$ ,  $K = K_r M_s$ , and the overall overload factor  $\beta$  is  $K_r/M_r > 1$ . This is because  $k_{l_r} > m_{l_r}$  for  $l_r = 1, 2, \dots, L_r$ . In addition to the structure of the overall pattern matrix given by (28), it is assumed that the combining matrices  $\alpha_{m'_{l_s} \times m'_{l_s}}$ 's corresponding to each  $\mathbf{P}_{m'_{l_s} \times m'_{l_s}}$  are also known to the receiver. As discussed in Section III-B2, our proposed detection algorithm with only the Kronecker product of  $L_s$  square factor matrices as the

pattern matrix results in  $M_s$  singleton equations with the SNR gain on each equation obtained using Lemma 2.<sup>3</sup> The detection algorithm for the general case of the pattern matrix is then summarized as the following two phases each incurring  $L_s$  and  $L_r$  recursions, respectively.

**Phase I:** In the first phase, the receiver applies  $L_s$  recursions, by following combining and (super-) grouping as described in Section III-B2, to the system of equation  $\mathbf{y}_{M \times 1} = \mathbf{G}_{M \times K} \mathbf{x}_{K \times 1} + \mathbf{n}_{M \times 1}$  with  $\mathbf{G}_{M \times K}$  defined in (28). Then the receiver obtains  $M_s$  super-groups each containing a set of equations defined according to the pattern matrix  $\mathbf{F}_{M_r \times K_r} \triangleq \mathbf{F}_{m_1 \times k_1}^{(1)} \otimes \mathbf{F}_{m_2 \times k_2}^{(2)} \otimes \cdots \otimes \mathbf{F}_{m_{L_r} \times k_{L_r}}^{(L_r)}$ . The set of equations in the  $\psi'_{L_s}$ -th super-group, where  $\psi'_{L_s} \triangleq j_1 + \sum_{i=1}^{L_s-1} (j_{L_s-i+1} - 1) \prod_{i'=1}^{L_s-i} m'_{i'}$ , indexed by the path of super-groups  $(j_{L_s}, j_{L_s-1}, \dots, j_2, j_1)$ , for  $j_{l_s} = 1, 2, \dots, m'_{l_s}$ , is expressed as

$$\begin{bmatrix} \tilde{y}_{\psi'_{L_s}, 1} \\ \tilde{y}_{\psi'_{L_s}, 2} \\ \vdots \\ \tilde{y}_{\psi'_{L_s}, M_r} \end{bmatrix} = \mathbf{F}_{M_r \times K_r} \begin{bmatrix} x_{\tau'_{L_s}} \\ x_{M_s + \tau'_{L_s}} \\ \vdots \\ x_{(K_r-1)M_s + \tau'_{L_s}} \end{bmatrix} + \begin{bmatrix} \tilde{n}_{\psi'_{L_s}, 1} \\ \tilde{n}_{\psi'_{L_s}, 2} \\ \vdots \\ \tilde{n}_{\psi'_{L_s}, M_r} \end{bmatrix}, \quad (29)$$

where  $\tau'_{L_s} \triangleq j_{L_s} + \sum_{i=1}^{L_s-1} (j_{L_s-i} - 1) \prod_{i'=L_s-i+1}^{L_s} m'_{i'}$ . Moreover,  $\tilde{y}_{\psi'_{L_s}, 1}, \tilde{y}_{\psi'_{L_s}, 2}, \dots, \tilde{y}_{\psi'_{L_s}, M_r}$  are the combined versions of the channel outputs in  $\mathbf{y}_{M \times 1}$ , after  $L_s$  rounds of combining, and their values are fully known to the receiver at the end of Phase I. Furthermore,  $\tilde{n}_{\psi'_{L_s}, 1}, \tilde{n}_{\psi'_{L_s}, 2}, \dots, \tilde{n}_{\psi'_{L_s}, M_r}$  are independent noise terms each having a variance equal to  $\sigma^2 / \gamma_{t, \tau'_{L_s}}$ . Note that, as a result of  $L_s$  square factor matrices, the effective SNR of the equations in the  $\psi'_{L_s}$ -th super-group in (29) is increased by a factor of  $\gamma_{t, \tau'_{L_s}}$ , where, by Lemma 2,<sup>4</sup>  $\gamma_{t, \tau'_{L_s}} \triangleq \prod_{l_s=1}^{L_s} \gamma_{j_{l_s}}^{(l_s)}$  with  $\gamma_{j_{l_s}}^{(l_s)}$  defined the same way as in Section III-B for  $\mathbf{P}_{m'_{l_s} \times m'_{l_s}}$ .

**Phase II:** The receiver in the second phase separately processes each of the  $M_s$  disjoint sets of equations of the form given by (29). According to the detailed description of the detection algorithm in Section III-C, the receiver, after applying  $L_r$  recursions, detects the  $K_r$  symbols of each of the aforementioned  $M_s$  sets of equations resulting in  $K = K_r M_s$  data symbols in total.

#### IV. PERFORMANCE CHARACTERIZATION

##### A. Average Sum-Rate

Based on the detailed analysis provided in Section III, we have the following theorem on the average sum-rate of the proposed scheme over GMAC<sup>5</sup>.

<sup>3</sup>Note that the proposed detection algorithm for the Kronecker product of square factor matrices in Section III-B is different than the proposed detection algorithm in Section III-C for the Kronecker product of rectangular factor matrices. Indeed, the former cannot be obtained as a special case of the latter simply by inserting  $k_l = m_l$ . In particular, we carefully designed the square factor matrices and the corresponding combining matrices to enable a much more efficient detection algorithm for the Kronecker product of square factor matrices.

<sup>4</sup>Observe that  $j_{l_s}$  here has the same meaning as  $s_{l_s}$  defined in Lemma 2 and  $\tau'_{L_s}$  has the same value as the variable  $i$  defined in Eq. (15).

<sup>5</sup>Note that, in general, the average sum-rate should depend on the statistical properties of the channel. One needs to account on our explanations in Section V-A to extend Theorem 3 to the case of fading channels.

**Theorem 3.** *The per-RE average sum-rate  $C_M$  of the proposed code-domain NOMA with the general pattern matrix  $\mathbf{G}_{M \times K}$ , specified in (28), and the proposed recursive detection algorithm is expressed as*

$$C_M = \frac{1}{2M} \sum_{j_{L_s}=1}^{m'_{L_s}} \sum_{j_{L_s-1}=1}^{m'_{L_s-1}} \cdots \sum_{j_2=1}^{m'_2} \sum_{j_1=1}^{m'_1} \log_2 \det \left( \mathbf{I}_{M_r} + \rho \prod_{l_s=1}^{L_s} \gamma_{j_{l_s}}^{(l_s)} \mathbf{F} \mathbf{F}^T \right), \quad (30)$$

where  $\mathbf{I}_{M_r}$  is an  $M_r \times M_r$  identity matrix,  $\mathbf{F}_{M_r \times K_r} \triangleq \mathbf{F}_{m_1 \times k_1}^{(1)} \otimes \mathbf{F}_{m_2 \times k_2}^{(2)} \otimes \cdots \otimes \mathbf{F}_{m_{L_r} \times k_{L_r}}^{(L_r)}$ ,  $\rho \triangleq P_x / \sigma^2$  with  $P_x \triangleq \mathbb{E}[x_j^2]$ , for  $j = 1, 2, \dots, K$ , and  $\sigma^2 = \mathbb{E}[n_i^2]$ , for  $i = 1, 2, \dots, M$ .

*Proof:* It is well known that for a PDMA system with the pattern matrix  $\mathbf{A}_{M \times K}$  and the SNR of  $\rho$  for all of the received original data symbols, the per-RE average sum rate is given by  $C_M^{\text{PDMA}} = \frac{1}{2M} \log_2 \det \left( \mathbf{I}_M + \rho \mathbf{A} \mathbf{A}^T \right)$  for the optimal MAP detection [36]. As elaborated in Phase I of the detection algorithm in Section III-D, we end up with  $M_s$  sets of equation each defined according to (29). Therefore, the per-RE average sum-rate  $C_{M_r}(\psi'_{L_s})$  of the  $\psi'_{L_s}$ -th super-group, indexed by the path  $(j_{L_s}, j_{L_s-1}, \dots, j_2, j_1)$ , is obtained as

$$C_{M_r}(\psi'_{L_s}) = \frac{1}{2M_r} \log_2 \det \left( \mathbf{I}_{M_r} + \rho \gamma_{t, \tau'_{L_s}} \mathbf{F} \mathbf{F}^T \right). \quad (31)$$

Averaging (31) over all  $M_s$  paths completes the proof. ■

In the special case with all the square factor matrices being identical, i.e.,  $\mathbf{P}_{m'_{l_s} \times m'_{l_s}}^{(l_s)} = \mathbf{P}_{m_p \times m_p}$ , for  $l_s = 1, 2, \dots, L_s$ , we have the following corollary given that  $\gamma_j^{(l_s)} = \gamma_j$ , for  $j = 1, 2, \dots, m_p$ .

**Corollary 4.** *The per-RE average sum-rate of the proposed code-domain NOMA with the pattern matrix  $\mathbf{G}_{M \times K} = \mathbf{F}_{M_r \times K_r} \otimes \mathbf{P}_{m_p \times m_p}^{\otimes L_s}$ , where  $\mathbf{P}^{\otimes L_s}$  denotes the  $L_s$ -times Kronecker product of  $\mathbf{P}$  with itself, is given by*

$$C_M = \frac{1}{2M} \sum_{r_1+r_2+\cdots+r_{m_p}=L_s} \frac{L_s!}{r_1!r_2!\cdots r_{m_p}!} \times \log_2 \det \left( \mathbf{I}_{M_r} + \rho \gamma_1^{r_1} \gamma_2^{r_2} \cdots \gamma_{m_p}^{r_{m_p}} \mathbf{F} \mathbf{F}^T \right), \quad (32)$$

where the summation is over all disjoint sets  $\{r_1, r_2, \dots, r_{m_p}\}$ :  $r'_i \in \{0, 1, \dots, L_s\}$ ,  $\sum_{i'=1}^{m_p} r'_{i'} = L_s$ .

**Remark 3 (Optimal Factor Matrices).** When the design target is to maximize the average sum rate, the design of the *optimal pattern matrix* can be formulated as the selection of an appropriate rectangular matrix  $\mathbf{F}^*$  (that can properly trade off between the performance and complexity) together with the  $v_{l_s}^*$ -th square matrices,  $l_s = 1, 2, \dots, L_s$ , from Algorithm 1 with the corresponding set of SNRs  $\{\gamma_{v_{l_s}^*, 1}^{(l_s)}, \gamma_{v_{l_s}^*, 2}^{(l_s)}, \dots, \gamma_{v_{l_s}^*, m_{l_s}}^{(l_s)}\}$ , such that they result in the maximum sum rate in (30).

**Remark 4 (Rate-Reliability Trade-off).** After the detection of a properly chosen subset of data symbols, one can successively cancel the interference on some detection equations such that higher SNR gains are achieved for the detection of some

remained symbols (see Example 4). This, in addition to increasing the average sum rate in (30), may also increase the error probability due to the error propagation issue in SIC detection, resulting in a trade-off between the rate and reliability. This mechanism also increases the latency because we should wait for the detection of some symbols before starting the detection of the others.

**Example 4.** For the system parameters in Example 1, by using Theorem 3 and Example 2 the average sum rate is  $C_{12} = (1/8) \log_2 (1 + (4/3)\rho) + (3/8) \log_2 (1 + (4/3)^2\rho)$ . However, after detecting  $x_1$  and  $x_5$  from the first two equations in (11), we can apply SIC to form  $\tilde{y}_9^{(2)} \triangleq y_5^{(1)} + y_9^{(1)} - 2x_1 - 2x_5 = 4x_9 + n_5^{(1)} + n_9^{(1)}$ , instead of the third equation in (11). Applying the same procedure to all of the four sets of equations defined according to  $\mathbf{P}_{3 \times 3}^{(1)}$ , increases the SNR gain  $\gamma_3^{(1)}$  on the third data symbols from  $4/3$  to  $2$ . Therefore, the average sum rate is increased to  $C_{12} = (1/12) \log_2 (1 + (4/3)\rho) + (1/4) \log_2 (1 + (4/3)^2\rho) + (1/8) \log_2 (1 + (8/3)\rho) + (1/24) \log_2 (1 + 2\rho)$ .  $\square$

### B. Latency Analysis

Given our proposed fully parallel detection algorithm, we have the following theorem for the system latency.

**Theorem 5.** *The maximum execution time  $T_{\text{exe}}^{\max}$  for the detection of  $K$  data symbols in the proposed code-domain NOMA with the general pattern matrix  $\mathbf{G}_{M \times K}$ , specified in (28), using the detection algorithm in Section III is as follows:*

$$T_{\text{exe}}^{\max} = t_a \sum_{l_s=1}^{L_s} (m'_{l_s} - 1) + T_{\text{MUD}}^{\text{noisy}}(\mathbf{F}_{m_{L_r} \times k_{L_r}}^{(L_r)}, \mathcal{C}(K_{L_r-1}, \mathcal{C}_0)) + \sum_{l_r=2}^{L_r-1} T_{\text{MUD}}^{\text{noiseless}}(\mathbf{F}^{(l_r)}, \mathcal{C}(K_{l_r-1}, \mathcal{C}_0)) + T_{\text{MUD}}^{\text{noiseless}}(\mathbf{F}^{(1)}, \mathcal{C}_0), \quad (33)$$

where  $t_a$  is the required time for an addition/subtraction,  $\mathcal{C}(K_{l_r}, \mathcal{C}_0)$  denotes the constellation space of the sum of (at most)  $K_{l_r}$  original data symbols each from a modulation with the constellation space  $\mathcal{C}_0$ ,  $T_{\text{MUD}}^{\text{noisy}}(\mathbf{F}_{m_{l_r} \times k_{l_r}}^{(l_r)}, \mathcal{C})$  is the required time for a given MUD algorithm to detect the  $k_{l_r}$  symbols with the constellation space  $\mathcal{C}$  from a set of  $m_{l_r}$  equations defined according to the pattern matrix  $\mathbf{F}_{m_{l_r} \times k_{l_r}}^{(l_r)}$  in the presence of additive noise, and  $T_{\text{MUD}}^{\text{noiseless}}(\mathbf{F}_{m_{l_r} \times k_{l_r}}^{(l_r)}, \mathcal{C})$  is defined similar to  $T_{\text{MUD}}^{\text{noisy}}(\mathbf{F}_{m_{l_r} \times k_{l_r}}^{(l_r)}, \mathcal{C})$  in the absence of noise.

*Proof:* The detection algorithm processes  $L_s$  square factor matrices through  $L_s$  recursions. The  $l'_s$ -th recursion, for  $l'_s = 1, 2, \dots, L_s$ , involves combining sets of  $m'_{l'_s}$  equations, where  $l''_s \triangleq L_s - l'_s + 1$ , according to the rows of  $\alpha_{m'_{l'_s} \times m'_{l'_s}}^{(l''_s)}$  requiring at most  $m'_{l'_s} - 1$  additions/subtractions. Thus, assuming all such operations are done in parallel, the  $l'_s$ -th recursion requires at most  $t_a(m'_{l'_s} - 1)$  units of time. Now, after at most  $t_a \sum_{l_s=1}^{L_s} (m'_{l_s} - 1)$  units of time, we get sets of equations defined according to the Kronecker product of  $L_r$  rectangular factor matrices, as (29), requiring  $L_r$  recursions. According to the analysis in Section III-C, the first recursion involves processing sets of equations in the form of

(21) each requiring at most  $T_{\text{MUD}}^{\text{noisy}}(\mathbf{F}_{m_{L_r} \times k_{L_r}}^{(L_r)}, \mathcal{C}(K_{L_r-1}, \mathcal{C}_0))$  units of time. Furthermore, the middle recursions require processing sets of equations in the form of (25) each incurring at most  $T_{\text{MUD}}^{\text{noiseless}}(\mathbf{F}_{m_{l''_r} \times k_{l''_r}}^{(l''_r)}, \mathcal{C}(K_{l''_r-1}, \mathcal{C}_0))$  time<sup>6</sup>, where  $l''_r \triangleq L_r - l'_r + 1$ . Finally, the last recursion contains sets of equations defined according to (27) each necessitating  $T_{\text{MUD}}^{\text{noiseless}}(\mathbf{F}_{m_1 \times k_1}^{(1)}, \mathcal{C}_0)$  time. Summing up the execution times of all  $L_s + L_r$  recursions completes the proof.  $\blacksquare$

**Remark 5.** Theorem 5 characterizes the latency in the worst-case scenario. However, the actual execution time might be smaller. For example, the unknown variables of the middle recursions of rectangular factor matrices are obtained according to (24). Therefore, if, for instance, half of the elements of  $r_{i_{l''_r}+1}^{(L_r-l'_r)}$  are zero, the constellation space is  $\mathcal{C}(1/2K_{l''_r-1}, \mathcal{C}_0)$ . Hence, the  $l'_r$ -th recursion will require  $T_{\text{MUD}}^{\text{noiseless}}(\mathbf{F}_{m_{l''_r} \times k_{l''_r}}^{(l''_r)}, \mathcal{C}(1/2K_{l''_r-1}, \mathcal{C}_0))$  time which is, in general, smaller than  $T_{\text{MUD}}^{\text{noiseless}}(\mathbf{F}_{m_{l''_r} \times k_{l''_r}}^{(l''_r)}, \mathcal{C}(K_{l''_r-1}, \mathcal{C}_0))$ . Similar arguments hold for all other recursions.

**Remark 6.** Note that (33) holds for  $L_r > 2$ . It can be observed that with  $L_r = 1$  the last three terms of (33) should be replaced by  $T_{\text{MUD}}^{\text{noisy}}(\mathbf{F}_{m_1 \times k_1}^{(1)}, \mathcal{C}_0)$ . Moreover, for  $L_r = 2$ , the third term in (33) becomes zero.

**Remark 7.** As shown throughout this section, the proposed detection algorithm leads to a latency that grows logarithmically with the total number of users/REs. Thanks to the proposed parallel structure of the detection algorithm, a powerful BS can simultaneously process all the branches of the detection tree in uplink. Also, a given user with a typical processing power during the downlink phase can focus only on the detection of the branch containing its data symbol to avoid unnecessary detections.

### C. Detection Complexity

Recall that the proposed recursive approach results in a relatively small size for each factor matrix, even for a large  $M$  and  $K$ ; hence, it is highly desired to exactly characterize the maximum number of required operations.

**Theorem 6.** *The maximum number of additions/subtractions  $N_{\text{add}}^{\max}$  and multiplications  $N_{\text{mul}}^{\max}$  for the detection of  $K$  data symbols in the proposed code-domain NOMA system with the general pattern matrix  $\mathbf{G}_{M \times K}$ , specified in (28), using the recursive detection algorithm in Section III is as follows:*

$$N_{\text{add}}^{\max} = M \sum_{l_s=1}^{L_s} (m'_{l_s} - 1) + M_s N_{\text{add}}^{\max}(\mathbf{F}_{M_r \times K_r}), \\ N_{\text{mul}}^{\max} = M_s N_{\text{mul}}^{\max}(\mathbf{F}_{M_r \times K_r}), \quad (34)$$

where  $N_{\text{add}}^{\max}(\mathbf{F}_{M_r \times K_r})$  ( $N_{\text{mul}}^{\max}(\mathbf{F}_{M_r \times K_r})$ ) is the maximum number of additions/subtractions (multiplications) for a NOMA system with the pattern matrix  $\mathbf{F}_{M_r \times K_r} \triangleq \mathbf{F}^{(1)} \otimes$

<sup>6</sup>Observe that one might be able to employ much simpler MUD algorithms in the absence of noise. Therefore, the current characterization which accounts for any difference on the detection latency (or complexity in Theorem 6), given the presence or absence of noise, provides a more general formulation.

TABLE I  
COMPUTATIONAL COMPLEXITY FOR THE DETECTION OF ALL  $K$  DATA SYMBOLS.

Algorithm	Maximum number of additions/subtractions	Maximum number of multiplications
SIC over $\mathbf{G}_{M \times K}$	$\mathcal{O}(K^2 M^3)$	$\mathcal{O}(K^2 M^3)$
Ours with SIC over $\mathbf{F}_{M_r \times K_r}$	$\mathcal{O}(M + K^2 M^3 / M_s^4)$	$\mathcal{O}(K^2 M^3 / M_s^4)$
BP over $\mathbf{G}_{M \times K}$	$\mathcal{O}(T_{\text{in}} d_g M K  C_0 ^{d_g} \log_2  C_0 )$	$\mathcal{O}(d_g M K  C_0 ^{d_g})$
Ours with BP over $\mathbf{F}_{M_r \times K_r}$	$\mathcal{O}(M + T_{\text{in}} d_f M K  C_0 ^{d_f} \log_2  C_0  / M_s)$	$\mathcal{O}(d_f M K  C_0 ^{d_f} / M_s)$
BP-IDD over $\mathbf{G}_{M \times K}$	$\mathcal{O}(T_{\text{in}} T_{\text{out}} d_g M K  C_0 ^{d_g} \log_2  C_0 )$	$\mathcal{O}(d_g M K  C_0 ^{d_g})$
Ours with BP-IDD over $\mathbf{F}_{M_r \times K_r}$	$\mathcal{O}(M + T_{\text{in}} T_{\text{out}} d_f M K  C_0 ^{d_f} \log_2  C_0  / M_s)$	$\mathcal{O}(d_f M K  C_0 ^{d_f} / M_s)$

$\dots \otimes \mathbf{F}^{(L_r)}$  and the recursive detection algorithm in Section III-C.  $N_{\mathcal{X}}^{\max}(\mathbf{F}_{M_r \times K_r})$ ,  $\mathcal{X} \in \{\text{add, mul}\}$  is given by

$$\begin{aligned} N_{\mathcal{X}}^{\max}(\mathbf{F}_{M_r \times K_r}) &= M_{L_r-1} N_{\mathcal{X}}^{\text{noisy}}(\mathbf{F}_{m_{L_r} \times k_{L_r}}^{(L_r)}, \mathcal{C}(K_{L_r-1}, \mathcal{C}_0)) \\ &+ \sum_{l_r'=2}^{L_r-1} \kappa_{l_r'-1}^{\mathcal{S}} M_{L_r-l_r'} N_{\mathcal{X}}^{\text{noiseless}}(\mathbf{F}^{(L_r-l_r'+1)}, \mathcal{C}(K_{L_r-l_r'}, \mathcal{C}_0)) \\ &+ \kappa_{L_r-1}^{\mathcal{S}} N_{\mathcal{X}}^{\text{noiseless}}(\mathbf{F}_{m_1 \times k_1}^{(1)}, \mathcal{C}_0), \end{aligned} \quad (35)$$

where  $M_{l_r} \triangleq \prod_{i=1}^{l_r} m_i$ ,  $K_{l_r} \triangleq \prod_{i=1}^{l_r} k_i$ ,  $\kappa_{l_r'-1}^{\mathcal{S}} \triangleq \prod_{i=L_r-l_r'+2}^{L_r} k_i$ ,  $N_{\mathcal{X}}^{\text{noisy}}(\mathbf{F}_{m_{l_r} \times k_{l_r}}^{(l_r)}, \mathcal{C})$  is the required number of operations for a given MUD algorithm to detect the  $k_{l_r}$  symbols with the constellation space  $\mathcal{C}$  from a set of  $m_{l_r}$  equations defined according to the pattern matrix  $\mathbf{F}_{m_{l_r} \times k_{l_r}}^{(l_r)}$  in the presence of additive noise, and  $N_{\mathcal{X}}^{\text{noiseless}}(\mathbf{F}_{m_1 \times k_1}^{(1)}, \mathcal{C})$  is defined similarly in the absence of noise.

*Proof:* The proof is based on the details of the detection algorithm presented in Section III and following similar steps to the proof of Theorem 5 while counting the maximum number of operations at each recursion. ■

**Remark 8.** Note that (35) is obtained assuming  $L_r > 2$ . For  $L_r = 1$ , (35) should be replaced by  $N_{\mathcal{X}}^{\text{noisy}}(\mathbf{F}_{m_1 \times k_1}^{(1)}, \mathcal{C}_0)$ . Moreover, the second term in (35) is zero for  $L_r = 2$ .

Note that (35) provides the full flexibility to characterize the overall complexity of the system with respect to any particular MUD algorithm, such as MAP, ML, AMP, belief propagation (BP), etc. (see, e.g., [31]–[34]), applied to the sets of equations defined according to each of  $L_r$  rectangular factor matrices at each recursion (see Section III-C). In order to gain further insight into the complexity of the proposed detection algorithm, in the following, we consider the case where the overall pattern matrix is  $\mathbf{G}_{M \times K} = \mathbf{F}_{M_r \times K_r} \otimes \mathbf{P}_{m_1' \times m_1'}^{(1)} \otimes \mathbf{P}_{m_2' \times m_2'}^{(2)} \otimes \dots \otimes \mathbf{P}_{m_{L_s}' \times m_{L_s}'}^{(L_s)}$ , i.e., the Kronecker product of  $L_s$  square factor matrices with a large  $M_r \times K_r$  rectangular factor matrix  $\mathbf{F}_{M_r \times K_r}$ . Therefore, the overall complexity of the system is given as (34). In Section VI, we provide further numerical results to demonstrate that recursion over rectangular factor matrices also helps in reducing the detection complexity, as formulated in (35).

Assuming  $\mathbf{G}_{M \times K} = \mathbf{F}_{M_r \times K_r} \otimes \mathbf{P}_{m_1' \times m_1'}^{(1)} \otimes \dots \otimes \mathbf{P}_{m_{L_s}' \times m_{L_s}'}^{(L_s)}$ , we have  $M_s \triangleq \prod_{l_s=1}^{L_s} m_{l_s}'$ ,  $M = M_s M_r$ , and  $K = M_s K_r$ . Note that in (34),  $M \sum_{l_s=1}^{L_s} (m_{l_s}' - 1) = \mathcal{O}(M)$

since  $\sum_{l_s=1}^{L_s} (m_{l_s}' - 1)$  is a constant much smaller than  $M_s \triangleq \prod_{l_s=1}^{L_s} m_{l_s}'$  and hence than  $M$  and  $K$ . Table I then compares the complexity of our scheme, for the case of the pattern matrix considered here, with some other well-known detection algorithms, i.e., SIC, BP, and BP-based iterative detection and decoding (BP-IDD) [31], [40], [41]. Note that the computational complexities in Table I are for the detection of all  $K$  data symbols (per symbol of all users) using our proposed scheme or by directly applying the aforementioned algorithms to the overall pattern matrix  $\mathbf{G}_{M \times K}$ . In Table I,  $|\mathcal{C}_0|$  denotes the size of the modulation constellation for the original data symbols. Note that our detection algorithm for the Kronecker product of square factor matrices results in sets of equations defined according to  $\mathbf{F}_{M_r \times K_r}$  over subsets of size  $K_r$  of original data symbols. Therefore, the constellation space of detections performed over both  $\mathbf{F}_{M_r \times K_r}$  and  $\mathbf{G}_{M \times K}$  is  $\mathcal{C}_0$ . Moreover,  $T_{\text{in}}$  and  $T_{\text{out}}$  in Table I represent the BP-IDD inner and outer iteration numbers, respectively. Additionally,  $d_f$  and  $d_g$  denote the maximum row weights of the pattern matrices  $\mathbf{F}_{M_r \times K_r}$  and  $\mathbf{G}_{M \times K}$ , respectively. Note that  $d_f \ll d_g$  since  $K_r = K/M_s \ll K$ . As seen, applying our recursive detection algorithm in conjunction with other MUD algorithms results in significantly lower complexities compared to directly applying an MUD algorithm. It is worth mentioning that the results in Table I are without applying recursion over the overall rectangular pattern matrix  $\mathbf{F}_{M_r \times K_r}$ . As numerically shown in Section VI, applying our proposed recursive detection algorithm for the Kronecker product of rectangular factor matrices further lowers the complexity. Therefore, our proposed recursive detection algorithm not only significantly reduces the latency, thanks to fully parallel detection, but also noticeably lowers the overall detection complexity.

## V. EXTENSIONS TO PRACTICAL SCENARIOS

In this section, we discuss how the results of the paper can be applied to several other practical scenarios.

### A. Downlink and Uplink Fading Channels

Throughout the paper, we focused on the case of GMAC where unit/equal gains are assumed for the channels between the users and the BS in different REs. In practice, the channel gain  $h_{mk}$  between the  $k$ -th user, for  $k = 1, 2, \dots, K$ , and the BS at the  $m$ -th RE, for  $m = 1, 2, \dots, M$ , can be different. In

this subsection we discuss how the results of the paper can be readily extended to downlink and uplink fading channels.

1) *Downlink Transmission*: According to (4), the received signal vector at the  $k$ -th user is given by  $\mathbf{y}_k = \text{diag}(h_{1k}, h_{2k}, \dots, h_{Mk}) \mathbf{G}_{M \times K} \mathbf{x} + \mathbf{n}_k$ . By dividing the  $m$ -th equation in  $\mathbf{y}_k$  by  $h_{mk}$  we get a new set of equations as  $\tilde{\mathbf{y}}_k = \mathbf{G}_{M \times K} \mathbf{x} + \tilde{\mathbf{n}}_k$ , where  $\tilde{\mathbf{n}}_k(m)$  is Gaussian with mean zero and variance  $\tilde{\sigma}_m^2 \triangleq \sigma^2 / h_{mk}^2$ . Therefore, all the results can be extended to the case of downlink fading channels with the slight change that the noise components over different REs do not have identical variances though they are still independent.

2) *Uplink Transmission*: Here, we assume in this paper that for a given user the channel gains over all  $M$  REs are equal, i.e.,  $h_{mk} = h_k, \forall m, k$ . This is a reasonable (and common) assumption, e.g., if the  $M$  REs are the adjacent sub-channels of a slowly changing channel, they have almost equal gains. However, if the channel gains are different, the users can adjust their power at each RE in such a way that equal gain is observed on the received signals over different REs. Note that this assumption for the case of downlink transmission yields exactly the same channel model  $\mathbf{y}_k / h_k = \mathbf{G}_{M \times K} \mathbf{x} + \mathbf{n}_k$  as GMAC with identical noise variances over different REs.

Recall, based on (3), that the uplink signal received by the BS can be expressed as  $\mathbf{y} = \mathbf{H} \mathbf{x} + \mathbf{n}$ , where  $\mathbf{H} \triangleq \mathcal{H} \odot \mathbf{G}_{M \times K}$  and  $\mathcal{H} \triangleq [\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_K]$ . With the above assumption, we have  $\mathbf{h}_k = h_k \mathbf{1}_{M \times 1}, \forall k$ . The received signal vector can then be rewritten as  $\mathbf{y} = \mathbf{G}_{M \times K} \tilde{\mathbf{x}} + \mathbf{n}$ , where  $\tilde{\mathbf{x}}_k \triangleq \tilde{\mathbf{x}}(k) = h_k x_k$ . Therefore, all of our earlier results can be readily extended to the case of uplink fading channels with the slight change that each  $k$ -th data symbol is scaled by the channel gain  $h_k$ .

**Remark 9.** As clarified in this section, one can readily extend the results of the paper from the case of GMAC model to fading channels. One of the important extensions is the extension of Algorithm 1 to the case of fading channels. The discussions in this section demonstrate that, in the case of downlink fading channels, given a square factor matrix  $\mathbf{P}$ , the corresponding optimal combining matrix  $\boldsymbol{\alpha}$  and SNR gains are dependent to the statistical properties of the channel and should update every time the downlink channel gains change. This is because for the case of downlink fading channels the noise components have different variances (inversely proportional with the square of the channel gains); hence, we need to revise the definition of the SNR gains with respect to the noise variances which intuitively suggests assigning higher weights to the stronger channels to maximize the SNR gains. Therefore, it is more desirable to apply the proposed algorithm in this paper to slow fading downlink channels. Note that this is also the case with other code-domain NOMA schemes where the overall pattern matrix should be updated after each change of the channel matrix. However, as discussed in Section III-A, the complexity of such updates is much smaller here due to the factorization of the overall pattern matrix which significantly reduces the search space for our proposed algorithm compared to conventional PDMA algorithms. Additionally, in the case of downlink fading channels, the BS potentially has the computing power to run Algorithm 1 every time the downlink channel matrix changes. On the other hand, in the case of uplink fading channels, we have  $\mathbf{y} = \mathbf{G}_{M \times K} \tilde{\mathbf{x}} + \mathbf{n}$ . Although

this changes the SNRs of different users according to their channel gains, it does not require Algorithm 1 to run again. Therefore, Algorithm 1, in its current form, can be applied to find the corresponding combining matrices and the SNR gains, and one can use the pre-stored results from Algorithm 1 for different factor matrices.

### B. Joint Power- and Code-Domain NOMA (JPC-NOMA)

The code-domain NOMA scheme proposed in this paper can be combined in many ways with power-domain NOMA. For instance, if, for some  $l$ ,  $2^{m_l} - 1 < k_l$ , then we cannot satisfy distinct nonzero columns for the factor matrix  $\mathbf{G}_{m_l \times k_l}^{(l)}$ . Consequently, the overall pattern matrix  $\mathbf{G}_{M \times K}$  will have some repeated columns, i.e., same pattern vectors for some of the users (see, e.g., [1, Example 1]). Therefore, one cannot distinguish between the data of the users with an equal pattern vector. In other words, the data symbols of the users with an equal pattern vector will be paired together, i.e., they coexist in all sets of equations containing any of those symbols. In such a case, we propose to assign different power coefficients to the users having the same pattern vector in order to differentiate between them using power-domain NOMA techniques. Then, the receiver after detecting the paired symbols using our proposed detection algorithm, detects the individual data symbols contained in each paired symbol according to the principles of power-domain NOMA. In this case, users with the most disparate channel qualities should be paired together.

More importantly, we can incorporate power-domain NOMA principles to increase the reliability of the detection performed over sets of equations at each layer of the detection algorithm proposed in Section III-C. Recall that each of those sets of equations contains a disjoint subset of symbols. For example, considering the general case of pattern matrix, specified in (28), the final recursion involves  $M_s \prod_{l_1=2}^{L_r} k_{l_r}$  sets of equations defined similar to (27), each containing  $k_1$  disjoint symbols while the indices of the symbols are known. As shown in [31], assigning different power scaling and phase shifting factors to the data symbols in a PDMA system improves the detection reliability. Therefore, the transmitter can group each subset of  $k_1$  users together. Then different power levels (as well as, possibly, phase shifts) are assigned to the data symbols of the users in each group according to the principles of power-domain NOMA in order to increase the reliability of the detection over the aforementioned  $M_s \prod_{l_1=2}^{L_r} k_{l_r}$  sets of equations.

## VI. NUMERICAL RESULTS

In this section, we numerically compare the average sum-rate of various configurations, including the traditional OMA, regular PDMA with the reported optimal pattern matrices in [31] such as  $3 \times 6$  and  $4 \times 8$  matrices with the maximum row weight  $d_f = 4$ , and our proposed recursive code-domain NOMA with the square factor matrices obtained from Algorithm 1. For our numerical analysis in this section, we consider the GMAC model that corresponds to the case where all channel gains are equal to one. Additionally, we assume that each user transmits with power  $P_x$ , and the noise variance

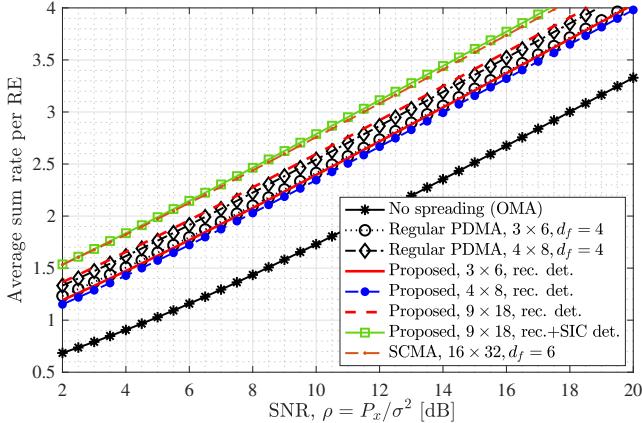


Fig. 3. Average sum-rate of various multiple access mechanisms including the traditional OMA, regular PDMA, and our proposed recursive method.

of the receiver at each RE is  $\sigma^2$ . By running Algorithm 1 for  $m_l = 3$  and 4, we get the optimal square factor matrices  $\mathbf{P}_{3 \times 3}^{(1)}$  and  $\mathbf{P}_{4 \times 4}^{(2)}$  in (7) with the combining matrices as (8) and individual SNR gains reported in Example 2.

Fig. 3 shows the per-RE average sum-rate of various multiple access techniques. Here, for our proposed scheme, we considered  $\mathbf{G}_{M \times K} = \mathbf{F} \otimes \mathbf{P}^{\otimes r}$  with  $\mathbf{F} = [1 \ 1]$ ,  $\mathbf{P}$  being either  $\mathbf{P}_{3 \times 3}^{(1)}$  or  $\mathbf{P}_{4 \times 4}^{(2)}$ , and  $r = 1$  or 2. It is observed that both the regular PDMA and our proposed scheme significantly outperform the traditional OMA method. Moreover, for relatively similar system parameters, our proposed method with the recursive detection performs very close to the capacity of regular PDMA with the optimal pattern matrix  $\mathbf{A}_{M \times K}$  and optimal MAP detection, i.e.,  $C_M^{\text{PDMA}} = \frac{1}{2M} \log_2 \det(\mathbf{I}_M + \rho \mathbf{A} \mathbf{A}^T)$ . Note that the negligible loss in the sum-rate performance of our scheme compared to that of PDMA, for a given dimension  $M$  and  $K$  of the pattern matrix, is because of two major reasons. First, for PDMA we assumed optimal MAP detection over the overall pattern matrix and plotted the capacity formula given the pattern matrix. However, for our scheme we are applying our low-complexity detection scheme which has a much lower complexity than MAP. Second, the design of the pattern matrix for PDMA is more complex. Indeed, it searches for the best  $M \times K$  pattern matrix to maximize the sum rate. Our scheme, on the other hand, designs the pattern matrix as the Kronecker product of smaller factor matrices such that each factor matrix has some properties to render a low-complexity detection. On the positive side, due to significantly lower complexity of our scheme, we can utilize pattern matrices with higher dimensions (e.g., by increasing  $r$ ) and, also, apply SIC detection at some recursions similar to Example 4 (see also [1, Example 4]) to get larger sum rates. This can significantly boost the performance of our scheme for a fixed overload factor.

Fig. 3 also shows the capacity formula curve (i.e., the best performance one can achieve) for the SCMA method having a  $16 \times 32$  pattern matrix  $\mathbf{H}_{16 \times 32}$  with a row weight of 6 and a column weight of 3. This sparse pattern matrix is formed by picking the parity-check matrix of a regular low-density parity-check (LDPC) code according to [42]. It is

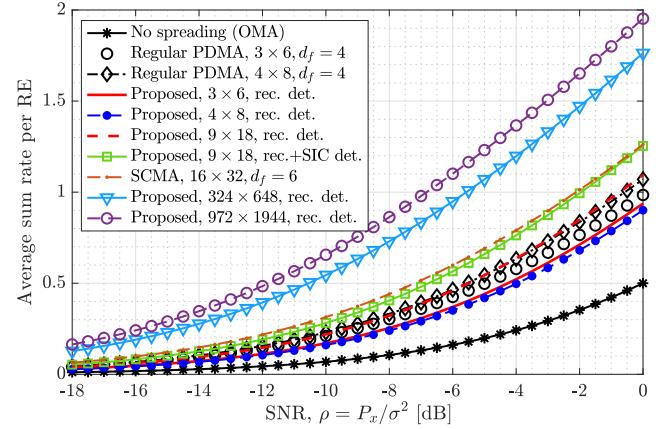


Fig. 4. Average sum-rate of various multiple access schemes at low SNRs.

observed that the performance of the SCMA method with this pattern matrix is almost the same as the green curve, i.e., our method with a  $9 \times 18$  pattern matrix decoded recursively (with SIC applied at some recursions), according to our proposed detection algorithm. This further highlights the potentials of our proposed scheme given that it is capable of achieving almost the same performance as SCMA with a twice smaller dimension and a low-complexity decoder. Additionally, the performance of our proposed scheme is significantly improved by increasing the dimensions of the overall pattern matrix (as shown in Fig. 4) while the SCMA performance does not change much by increasing the pattern matrix dimension (given the sparsity constraint on the pattern matrix which does not allow transmitting the symbols of the users over many REs).

In Fig. 4, the focus is specifically on low-SNR regimes. Here, we consider  $\mathbf{G}_{M \times K} = \mathbf{F}_{4 \times 8} \otimes \mathbf{P}_{3 \times 3}^{(1) \otimes r}$  for the proposed scheme with  $\mathbf{F}_{4 \times 8}$  being the  $4 \times 8$  PDMA matrix with  $d_f = 4$  from [31]. In this case,  $M = 4 \times 3^r$ ,  $K = 8 \times 3^r$ , and the overload factor  $\beta = 2$ . It can be observed in Fig. 4 that having large pattern matrices are essential at low SNRs to avoid getting close-to-zero sum rates. While it is not practically feasible to implement conventional code-domain NOMA schemes at such large dimensions considered in this setting, our low-complexity recursive approach makes it possible to spread the data symbols of the users using desirably large pattern vectors. As a consequence, 2.74 and 3.46 times larger average sum rates are achieved for our recursive scheme with  $\mathbf{G}_{324 \times 648}$  (i.e.,  $r = 4$ ) and  $\mathbf{G}_{972 \times 1944}$  (i.e.,  $r = 5$ ), respectively, compared to  $4 \times 8$  PDMA scheme at the SNR of  $-15$  dB. These gains are as large as  $10.25 \times$  and  $12.91 \times$ , respectively, when compared to OMA. One may deduce that such gains are directly as a result of the development of the SNR gains through the proposed recursive detection algorithm in Section III-B2. Indeed, using Corollary 4, the average sum-rate per RE for the pattern matrix design of  $\mathbf{G}_{M \times K} = \mathbf{F}_{4 \times 8} \otimes \mathbf{P}_{3 \times 3}^{(1) \otimes r}$  is  $C_M = \frac{1}{8} \log_2 \det(\mathbf{I}_4 + \rho(4/3)^r \mathbf{F}_{4 \times 8} \mathbf{F}_{4 \times 8}^T)$ . Therefore, starting from  $r = 0$ , each unit increase of  $r$  improves the average sum-rate by  $10 \log_{10}(4/3) \approx 1.25$  dB.

In order to analyze the bit error rate (BER) and also observe

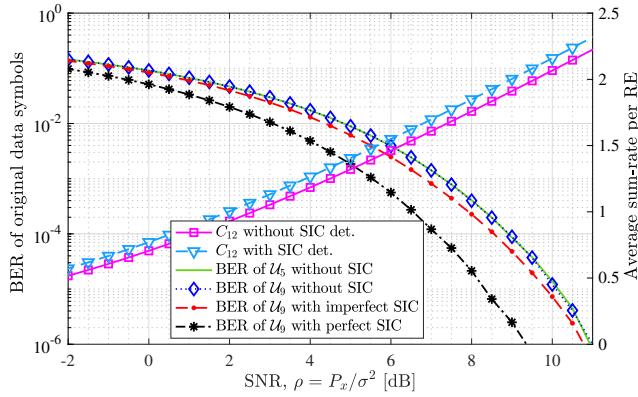


Fig. 5. Average sum-rate and BER comparison for the detection with and without SIC.

the impact of SIC detection on the BER, we consider the setup in Example 1. The average sum-rate  $C_{12}$  of the system is characterized in Example 4. Additionally, we have considered a sample way of incorporating SIC detection in the middle of the detection process, in Example 4. The increase on the average sum-rate by incorporating SIC detection is evident from Example 4 and is also confirmed in Fig. 5. For the BER performance we assume that all users employ binary phase shift keying (BPSK) modulation and, similar to Example 1, we only focus on the detection of the users  $\mathcal{U}_1$ ,  $\mathcal{U}_5$ , and  $\mathcal{U}_9$ . Using Monte-Carlo simulations with  $10^7$  trials, it is confirmed that all three users have the same BER performance (without SIC detection) since the overall SNR gain for all of them is  $\gamma_{t,i} = (4/3)^2$ ,  $i = 1, 5, 9$  (see Example 2). Additionally, as shown in Example 4, incorporating the SIC detection, according to Example 4, increases the overall SNR gain of  $\mathcal{U}_9$  from  $(4/3)^2$  to  $8/3$ . This is equivalent to  $10 \log_{10}(3/2) \approx 1.76$  dB gain on the BER performance of  $\mathcal{U}_9$ , assuming perfect SIC detection, which is also verified in Fig. 5. However, due to the error propagation on the SIC detection, the gain on the BER of  $\mathcal{U}_9$  with SIC detection is much smaller than 1.76 dB, demonstrating the impact of imperfect SIC detection. In particular, there is around 1.3 – 1.5 dB gap between perfect and imperfect SIC detection over almost all ranges of SNR. Nevertheless, interestingly, the performance of SIC detection (according to the sample way of incorporation as explained in Example 4) improves both average sum-rate and the BER performance. This is mainly because we only employed SIC detection over the last step of the detection, and once the detection of the earlier symbols is reliable, the gain of the SNR is dominant to the imperfection of the SIC detection. In general, the the BER performance may degrade if we incorporate SIC detection in many middle steps, calling for an interesting trade-off between the rate and reliability (see also Remark 4).

In Section IV-C, by considering the overall pattern matrix as  $\mathbf{G}_{M \times K} = \mathbf{F}_{M_r \times K_r} \otimes \mathbf{P}_{m'_1 \times m'_1}^{(1)} \otimes \cdots \otimes \mathbf{P}_{m'_{L_s} \times m'_{L_s}}^{(L_s)}$ , we have shown that applying our recursive detection algorithm in conjunction with other MUD algorithms significantly lowers the detection complexity compared to directly applying an MUD algorithm. Here, we provide a numerical example to show

that applying our proposed recursive detection algorithm for the Kronecker product of rectangular factor matrices further lowers the complexity. To see this, let us consider the overall pattern matrix as  $\mathbf{F}_{2 \times 6} = \mathbf{F}_{1 \times 2}^{(1)} \otimes \mathbf{F}_{2 \times 3}^{(2)}$ , where  $\mathbf{F}_{1 \times 2}^{(1)}$  and  $\mathbf{F}_{2 \times 3}^{(2)}$  are given in Example 3. Let us further assume that we want to apply MAP for the detection over the system of equations defined as  $\mathbf{y}_{2 \times 1} = \mathbf{F}_{2 \times 6} \mathbf{x}_{6 \times 1} + \mathbf{n}_{2 \times 1}$ , and we consider the number of possibilities that a MAP detector has to search for as a rough approximate of the detection complexity. In this case, directly applying MAP to this system of equations requires searching over all  $|\mathcal{C}_0|^6$  possibilities. On the other hand, applying our recursive detection algorithm requires  $|\mathcal{C}(2, \mathcal{C}_0)|^3 + 3|\mathcal{C}_0|^2$  searches, where  $\mathcal{C}(2, \mathcal{C}_0)$  is the constellation space of the sum of 2 original data symbols each from a modulation with the constellation space  $\mathcal{C}_0$  (see (35) for the details). Assuming quadrature phase shift keying (QPSK) modulation as an example,  $|\mathcal{C}_0| = 4$  and  $|\mathcal{C}(2, \mathcal{C}_0)| = 9$ . Therefore, our scheme requires computations of 777 cases which is much smaller than  $4^6 = 4096$ , i.e., direct application of MAP to  $\mathbf{F}_{2 \times 6} \mathbf{x}_{6 \times 1}$ .

## VII. CONCLUSIONS AND FUTURE DIRECTIONS

We proposed a low-complexity and scalable recursive approach toward code-domain NOMA by constructing the overall pattern matrix as the Kronecker product of several factor matrices. We then developed detection algorithms that reduce to several layers/recursions, each dealing with disjoint subsets of equations corresponding to certain factor matrices, that can be executed in parallel. For the Kronecker product of square factor matrices we proposed a systematic way of choosing the factor matrices that enables a remarkably low-complexity detection involving only few linear operations (additions/subtractions). Furthermore, for the Kronecker product of rectangular factor matrices we provided a recursive detection algorithm that can work on the general case of factor matrices. Given the proposed schemes and the detection algorithm we characterized the system performance in terms of average sum rate, latency, and detection complexity. We further discussed possible extensions of the work to the case of fading channels and joint power- and code-domain NOMA. We showed that the proposed scheme has significantly lower complexity and latency compared to straightforward code-domain NOMA schemes. Moreover, it is numerically verified that by utilizing large pattern matrices the proposed scheme significantly improves the average sum rate.

The proposed approach in this paper can be extended in various directions. Here, we highlight several directions for the future research.

1) *Optimal design of factor matrices:* Theorem 3 and Corollary 4 naturally lead to a design strategy for optimal factor matrices that maximize the average sum rate (see Remark 3). The characterization of other performance metrics, such as individual rates [43], and then finding the optimal factor matrices with respect to these metrics is a direction for future research.

2) *Rate-Reliability-Latency Trade-offs:* Given the set of factor matrices and the overall recursive approach discussed in

Section III, various detection schemes can be applied at each layer/recursion trading off between the rate, reliability, and latency, as exemplified in Remark 4 and Example 4. Detailed characterizations of such trade-offs is another direction for future research.

### 3) Joint Power- and Code-Domain NOMA (JPC-NOMA):

As discussed in Section V-B, power-domain NOMA can be combined with the proposed code-domain NOMA in several ways to improve the system design in terms of various performance metrics including reliability, throughput, fairness, etc. Designing such joint networks and analytically characterizing their performance metrics to show the advantages of the joint design is another important future research direction.

4) *Grant-Free Transmission*: In many use cases of massive communication, it is essential to reduce the coordination overhead of the communication protocol. Grant-free protocols attempt to serve a massive number of users with minimal (or even without any) coordination. Incorporating the proposed low-complexity code-domain NOMA scheme in the context of grant-free transmission is a vital future direction.

## APPENDIX A PROOF OF LEMMA 1

Let  $\mathbf{P} = [p_{i,j}]_{m \times m}$ . If there exists no such a combining matrix  $\boldsymbol{\alpha} = [\alpha_{i,j}]_{m \times m}$  with  $\alpha_{i,j} \in \{-1, 0, +1\}$ , then we are done. Otherwise, if such a matrix exists, then we prove the uniqueness of the  $i$ -th row of  $\boldsymbol{\alpha}$ , for  $i = 1, 2, \dots, m$ , by contradiction.

Assume to the contrary that there are  $T = 2$  distinct sets of coefficients  $\{\alpha_{i,j}^{(1)}\}_{j=1}^m$  and  $\{\alpha_{i,j}^{(2)}\}_{j=1}^m$  for the  $i$ -th row of  $\boldsymbol{\alpha}$  that result in singleton vectors with nonzero elements at the  $i$ -th position of the  $i$ -th row of  $\boldsymbol{\alpha}\mathbf{P}$ . In other words, we have

$$\sum_{j=1}^m \alpha_{i,j}^{(t)} p_{j,i} = w_t \neq 0, \quad (36)$$

$$\sum_{j=1}^m \alpha_{i,j}^{(t)} p_{j,i'} = 0, \quad i' \neq i = 1, 2, \dots, m, \quad (37)$$

for  $t = 1, 2$ , where  $w_1$  and  $w_2$  are two nonzero integers. Then we can define a new set of combining coefficients  $\{\alpha_{i,j}^{(3)}\}_{j=1}^m$  with  $\alpha_{i,j}^{(3)} \triangleq w_1 \alpha_{i,j}^{(2)} - w_2 \alpha_{i,j}^{(1)}$ . Note that

$$\begin{aligned} \sum_{j=1}^m \alpha_{i,j}^{(3)} p_{j,i} &= w_1 \sum_{j=1}^m \alpha_{i,j}^{(2)} p_{j,i} - w_2 \sum_{j=1}^m \alpha_{i,j}^{(1)} p_{j,i} \\ &= w_1 w_2 - w_2 w_1 = 0, \end{aligned} \quad (38)$$

$$\begin{aligned} \sum_{j=1}^m \alpha_{i,j}^{(3)} p_{j,i'} &= w_1 \sum_{j=1}^m \alpha_{i,j}^{(2)} p_{j,i'} - w_2 \sum_{j=1}^m \alpha_{i,j}^{(1)} p_{j,i'} \\ &= 0 - 0 = 0, \quad i' \neq i = 1, 2, \dots, m. \end{aligned} \quad (39)$$

Now, two cases are possible:

- $\alpha_{i,j}^{(3)} = 0$ , i.e.,  $\alpha_{i,j}^{(2)} = (w_2/w_1) \times \alpha_{i,j}^{(1)}$  for all  $j = 1, 2, \dots, m$ . This implies that there is only one set of coefficients  $\{\alpha_{i,j}^{(1)}\}_{j=1}^m$  for the  $i$ -th row of  $\boldsymbol{\alpha}$  since scaling the combining coefficients by a constant factor (here,  $w_2/w_1$ ) does not change the performance and the SNR gains as explained for the condition **C1** in Section III-B1.

- At least for one  $j$ ,  $\alpha_{i,j}^{(3)} \neq 0$ . This is in contradiction with the linear independence of the rows of  $\mathbf{P}$  since the linear combination of the rows with nonzero combining coefficients is equal to zero as shown by (38) and (39).

This proves that the  $i$ -th row of  $\boldsymbol{\alpha}$  is unique. Repeating the same procedure for  $i = 1, 2, \dots, m$  completes the proof. ■

## REFERENCES

- [1] M. V. Jamali and H. Mahdavifar, "A low-complexity recursive approach toward code-domain NOMA for massive communications," in *Proc. IEEE Global Communications Conference (GLOBECOM), Abu Dhabi, UAE, Dec. 2018*, pp. 1–6.
- [2] M. Fereydounian, M. V. Jamali, H. Hassani, and H. Mahdavifar, "Channel coding at low capacity," in *IEEE Information Theory Workshop (ITW)*, IEEE, 2019, pp. 1–5.
- [3] Y.-P. E. Wang, X. Lin, A. Adhikary, A. Grovlen, Y. Sui, Y. Blankenship, J. Bergman, and H. S. Razaghi, "A primer on 3GPP narrowband Internet of Things," *IEEE Commun. Mag.*, vol. 55, no. 3, pp. 117–123, 2017.
- [4] R. Ratasuk, N. Mangalvedhe, Y. Zhang, M. Robert, and J.-P. Koskinen, "Overview of narrowband IoT in LTE Rel-13," in *Proc. IEEE Conf. Standard Commun. Netw. (CSCN)*, 2016, pp. 1–7.
- [5] Z. Ding, X. Lei, G. K. Karagiannidis, R. Schober, J. Yuan, and V. K. Bhargava, "A survey on non-orthogonal multiple access for 5G networks: Research challenges and future trends," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 10, pp. 2181–2195, 2017.
- [6] D. Wan, M. Wen, X. Cheng, S. Mumtaz, and M. Guizani, "A promising non-orthogonal multiple access based networking architecture: Motivation, conception, and evolution," *IEEE Wireless Commun.*, vol. 26, no. 5, pp. 152–159, 2019.
- [7] O. Ordentlich and Y. Polyanskiy, "Low complexity schemes for the random access Gaussian channel," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2017, pp. 2528–2532.
- [8] L. Liu, E. G. Larsson, W. Yu, P. Popovski, C. Stefanovic, and E. De Carvalho, "Sparse signal processing for grant-free massive connectivity: A future paradigm for random access protocols in the Internet of Things," *IEEE Signal Process. Mag.*, vol. 35, no. 5, pp. 88–99, 2018.
- [9] W. Yuan, N. Wu, Q. Guo, D. W. K. Ng, J. Yuan, and L. Hanzo, "Iterative joint channel estimation, user activity tracking, and data detection for FTN-NOMA systems supporting random access," *IEEE Trans. Commun.*, vol. 68, no. 5, pp. 2963–2977, 2020.
- [10] S. Shahsavari, F. Shirani, and E. Erkip, "Opportunistic temporal fair scheduling for non-orthogonal multiple access," in *2018 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, IEEE, 2018, pp. 391–398.
- [11] S. R. Islam, N. Avazov, O. A. Dobre, and K.-S. Kwak, "Power-domain non-orthogonal multiple access (NOMA) in 5G systems: Potentials and challenges," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 2, pp. 721–742, 2016.
- [12] Z. Ding, M. Peng, and H. V. Poor, "Cooperative non-orthogonal multiple access in 5G systems," *IEEE Commun. Lett.*, vol. 19, no. 8, pp. 1462–1465, 2015.
- [13] X. Li, J. Li, Y. Liu, Z. Ding, and A. Nallanathan, "Residual transceiver hardware impairments on cooperative NOMA networks," *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 680–695, 2020.
- [14] X. Pei, H. Yu, M. Wen, Q. Li, and Z. Ding, "Secure outage analysis for cooperative NOMA systems with antenna selection," *IEEE Trans. Veh. Technol.*, vol. 69, no. 4, pp. 4503–4507, 2020.
- [15] Y. Liu, Z. Ding, M. Elkashlan, and H. V. Poor, "Cooperative non-orthogonal multiple access with simultaneous wireless information and power transfer," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 4, pp. 938–953, 2016.
- [16] Q. Sun, S. Han, I. Chin-Lin, and Z. Pan, "On the ergodic capacity of MIMO NOMA systems," *IEEE Wireless Commun. Lett.*, vol. 4, no. 4, pp. 405–408, 2015.
- [17] Z. Ding, P. Fan, and H. V. Poor, "Random beamforming in millimeter-wave noma networks," *IEEE Access*, 2017.
- [18] M. V. Jamali, S. M. Azimi-Abarghouyi, and H. Mahdavifar, "Outage probability analysis of uplink NOMA over ultra-high-speed FSO-backhauled systems," in *Proc. IEEE Globecom Workshops (GC Wkshps), Abu Dhabi, UAE, Dec. 2018*, pp. 1–6.
- [19] M. V. Jamali and H. Mahdavifar, "Uplink non-orthogonal multiple access over mixed RF-FSO systems," *IEEE Trans. Wireless Commun.*, vol. 19, no. 5, pp. 3558–3574, 2020.

- [20] T. Hou, Y. Liu, Z. Song, X. Sun, and Y. Chen, "Exploiting NOMA for UAV communications in large-scale cellular networks," *IEEE Trans. Commun.*, vol. 67, no. 10, pp. 6897–6911, 2019.
- [21] K. N. Doan, M. Vaezi, W. Shin, H. V. Poor, H. Shin, and T. Q. Quek, "Power allocation in cache-aided NOMA systems: Optimization and deep reinforcement learning approaches," *IEEE Trans. Commun.*, vol. 68, no. 1, pp. 630–644, 2020.
- [22] H. Nikopour and H. Baligh, "Sparse code multiple access," in *Proc. IEEE Int. Symp. Pers. Indoor Mobile Radio Commun., London, U.K., Sep. 2013*, pp. 332–336.
- [23] F. Wei and W. Chen, "A low complexity SCMA decoder based on list sphere decoding," in *Proc. IEEE Global Communications Conference (GLOBECOM), Washington, D.C., USA, Dec. 2016*, pp. 1–6.
- [24] C. Zhang, C. Yang, X. Pang, W. Song, W. Xu, S. Zhang, Z. Zhang, and X. You, "Efficient sparse code multiple access decoder based on deterministic message passing algorithm," *IEEE Trans. Veh. Technol.*, vol. 69, no. 4, pp. 3562–3574, 2020.
- [25] D. Fang, Y.-C. Huang, Z. Ding, G. Geraci, S.-L. Shieh, and H. Claussen, "Lattice partition multiple access: A new method of downlink non-orthogonal multiuser transmissions," in *Proc. IEEE Global Communications Conference (GLOBECOM), Washington, D.C., USA, Dec. 2016*, pp. 1–6.
- [26] M. Qiu, Y.-C. Huang, S.-L. Shieh, and J. Yuan, "A lattice-partition framework of downlink non-orthogonal multiple access without SIC," *IEEE Trans. Commun.*, vol. 66, no. 6, pp. 2532–2546, 2018.
- [27] M. Qiu, Y.-C. Huang, J. Yuan, and C.-L. Wang, "Lattice-partition-based downlink non-orthogonal multiple access without SIC for slow fading channels," *IEEE Trans. Commun.*, vol. 67, no. 2, pp. 1166–1181, 2019.
- [28] Q. Xiong, C. Qian, B. Yu, and C. Sun, "Advanced NoMA scheme for 5G cellular network: Interleave-grid multiple access," in *Proc. IEEE Globecom Workshops (GC Wkshps), Singapore, Dec. 2017*, pp. 1–5.
- [29] S. Hu, B. Yu, C. Qian, Y. Xiao, Q. Xiong, C. Sun, and Y. Gao, "Nonorthogonal interleave-grid multiple access scheme for industrial Internet of Things in 5G network," *IEEE Trans. Ind. Inform.*, vol. 14, no. 12, pp. 5436–5446, 2018.
- [30] X. Dai, S. Chen, S. Sun, S. Kang, Y. Wang, Z. Shen, and J. Xu, "Successive interference cancelation amenable multiple access (SAMA) for future wireless communications," in *Proc. IEEE Int. Conf. Commun. Syst. (ICC), 2014*, pp. 222–226.
- [31] S. Chen, B. Ren, Q. Gao, S. Kang, S. Sun, and K. Niu, "Pattern division multiple access—a novel nonorthogonal multiple access for fifth-generation radio networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 4, pp. 3185–3196, 2017.
- [32] G. Song, K. Cai, Y. Chi, J. Guo, and J. Cheng, "Super-sparse on-off division multiple access: Replacing repetition with idling," *IEEE Trans. Commun.*, vol. 68, no. 4, pp. 2251–2263, 2020.
- [33] Y. Chi, L. Liu, G. Song, C. Yuen, Y. L. Guan, and Y. Li, "Practical MIMO-NOMA: low complexity and capacity-approaching solution," *IEEE Trans. Wireless Commun.*, vol. 17, no. 9, pp. 6251–6264, 2018.
- [34] L. Liu, C. Liang, J. Ma, and L. Ping, "Capacity optimality of AMP in coded systems," *arXiv preprint arXiv:1901.09559*, 2019.
- [35] Z. Yang, J. Cui, X. Lei, Z. Ding, P. Fan, and D. Chen, "Impact of factor graph on average sum rate for uplink sparse code multiple access systems," *IEEE Access*, vol. 4, pp. 6585–6590, 2016.
- [36] O. Shental, B. M. Zaidel, and S. Shamai, "Low-density code-domain NOMA: Better be regular," in *Proc. IEEE Int. Symp. Inf. Theory, Aachen, Germany, Jul. 2017*, pp. 2628–2632.
- [37] E. Arikan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Trans. Inf. Theory*, vol. 55, no. 7, pp. 3051–3073, 2009.
- [38] H. Mahdavifar, M. El-Khamy, J. Lee, and I. Kang, "Compound polar codes," in *Information Theory and Applications Workshop (ITA)*. IEEE, 2013, pp. 1–6.
- [39] —, "Polar coding for bit-interleaved coded modulation," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 5, pp. 3115–3127, 2015.
- [40] D. N. Liu and M. P. Fitz, "Low complexity affine MMSE detector for iterative detection-decoding MIMO OFDM systems," *IEEE Trans. Commun.*, vol. 56, no. 1, pp. 150–158, 2008.
- [41] B. Ren, X. Yue, W. Tang, Y. Wang, S. Kang, X. Dai, and S. Sun, "Advanced IDD receiver for PDMA uplink system," in *Proc. IEEE/CIC Int. Conf. Commun. China (ICCC)*. IEEE, 2016, pp. 1–6.
- [42] M. M. Mansour and N. R. Shanbhag, "A 640-Mb/s 2048-bit programmable LDPC decoder chip," *IEEE J. Solid-State Circuits*, vol. 41, no. 3, pp. 684–698, 2006.
- [43] P. Xu, Z. Ding, X. Dai, and H. V. Poor, "A new evaluation criterion for non-orthogonal multiple access in 5G software defined networks," *IEEE Access*, vol. 3, pp. 1633–1639, 2015.



**Mohammad Vahid Jamali** is a Ph.D. candidate in the Department of Electrical Engineering and Computer Science at the University of Michigan, Ann Arbor, MI, USA. He received the B.Sc. degree from K.N. Toosi University of Technology, Tehran, Iran, in 2013, and the M.Sc. degree from Sharif University of Technology (SUT), Tehran, Iran, in 2015, both in electrical engineering. From 2013 to 2017, he was a member of the technical staff of the Optical Networks Research Laboratory (ONRL) at SUT. He received several awards and honors including The 2017 Best M.Sc. Thesis Award by the IEEE Iran Section, Rackham Graduate Fellowship by the University of Michigan in 2017–2018, Minuchehr Kashef Scholarship Award by the University of Michigan in 2018, Best Paper Award of IEEE GLOBECOM Workshops on UHS5G, Abu Dhabi, UAE, 2018, and Exemplary Reviewer of the IEEE Transactions on Communications in 2019. His general research interests are in coding, information theory, wireless communications, machine learning, and optics.



**Hessam Mahdavifar** (S'10, M'12) is an Assistant Professor in the Department of Electrical Engineering and Computer Science at the University of Michigan Ann Arbor. He received the B.Sc. degree from the Sharif University of Technology, Tehran, Iran, in 2007, and the M.Sc. and the Ph.D. degrees from the University of California San Diego (UCSD), La Jolla, in 2009, and 2012, respectively, all in electrical engineering. He was with the Samsung US R&D between 2012 and 2016, in San Diego, US, as a staff research engineer.

He received the NSF career award in 2020. He also received Best Paper Award in 2015 IEEE International Conference on RFID, and the 2013 Samsung Best Paper Award. He also received two Silver Medals at International Mathematical Olympiad in 2002 and 2003, and two Gold Medals at Iran National Mathematical Olympiad in 2001 and 2002. His main area of research is coding and information theory with applications to wireless communications, storage systems, security, and privacy.