A User-Independent Serial Interference Cancellation Based Coding Scheme for the Unsourced Random Access Gaussian Channel

†Avinash Vem, †Krishna R. Narayanan, §Jun Cheng and †Jean-Francois Chamberland
†Department of Electrical and Computer Engineering, Texas A&M University
§Department of Electrical Engineering, Doshisha University
{vem@tamu.edu,krn@tamu.edu, jcheng@doshisha.ac.jp,chmbrlnd@tamu.edu}

Abstract

We propose a novel coding scheme for the unsourced multiple access channel model introduced by Polyanskiy [1]. This new paradigm is composed of four main ingredients: (i) the transmission period is partitioned into sub-blocks, thereby instituting a slotted framework; (ii) The message (data) is split into two parts and one part chooses an interleaver for a low density parity check (LDPC) type code. This part of the message is encoded using spreading sequences or codewords that are designed to be decoded by a compressed sensing type decoder; (iii) The other part of the message is encoded using a low density parity check (LDPC) type code and decoded using a joint message passing decoding algorithm designed for the T-user binary input real adder channel; (iv) users repeat their codeword in multiple sub-blocks, with the transmission pattern being a deterministic function of message content and independent of the identity of the user. When this coding scheme is combined with successive interference cancellation, the ensuing communication infrastructure can offer significant performance improvements compared to the recently proposed coding scheme in [2] and results in the best performing coding scheme to date.

Index Terms

T-user adder channel, Slotted ALOHA, Interference Cancellation

I. Introduction

In [1], Polyanskiy introduced an interesting and timely multiple access problem; throughout, we refer to this new formulation as the unsourced multiple access channel model (MAC). In this setting, a very large number, K_{tot}, of users in a wireless network operate in an uncoordinated fashion. Out of the K_{tot} users, a subset of K_{a} users are active at any time; and each of them wishes to communicate a B-bit message to a central base station. The base station is interested only in recovering the list of messages without regard to the identity of the user who transmitted a particular message. In addition to this, the interest is typically in the case when B is small.

The unsourced, uncoordinated nature of the problem and the small block lengths represent a substantial departure from the traditional multiple access channel and, consequently, has important implications both on the fundamental limits as well as the design of pragmatic low-complexity coding schemes. Due to small block lengths, information rates do not provide reasonable benchmarks and finite block length bounds are more meaningful. In [1], Polyanskiy provides bounds on the performance of finite-length codes for this channel model. The design of coding schemes is also very challenging for this setting. Almost all well-known low-complexity coding solutions for the traditional MAC channel such as code-division multiple access, rate-splitting [3], and interleave-division multiple access [4], implicitly assume some form of coordination between the users and that some parameters of the coding scheme such as the spreading sequence, code rates, time sharing parameters, Tanner graph of the code, etc., are user dependent. When the message length is small, establishing such coordination becomes inefficient; this

The work of K. R. Narayanan and J.-F. Chamberland was funded in part by the National Science Foundation under Grant CCF 16.

renders well-known coding solutions tailored to the traditional MAC inadequate for the unsourced MAC. Ordentlich and Polyanskiy describe the first low-complexity coding paradigm for the unsourced MAC [2]. In their scheme, a transmission period is partitioned into smaller sub-blocks and users randomly pick one sub-block to transmit in. The encoding structure employed by each user is a concatenated code where the inner code is designed to recover the modulo-p sum of codewords transmitted by users and the outer code is designed to decode multiple users given the modulo-p sum of their codewords. Succinctly, the inner code operates in the spirit of integer-forcing [5], whereas the outer code is an optimal code for the T-user modulo-p multiple access channel [6].

While Ordentlich and Polyanskiy have contributed an important first step in finding practical schemes for the unsourced MAC, there remains a substantial gap between the performance of their proposed scheme and the capacity limit derived in [1]. Indeed, they point to this gap and discuss possibilities for improving its performance. In [2, Section III.A], they discuss the possibility of improving their scheme by decoding the T messages using the real sum from the channel output instead of first reducing the output of the channel to modulo-p operations. However, in the unsourced MAC, each user is forced to use the same codebook and they remark that "the task of designing low complexity capacity approaching same-codebook schemes for the real binary adder seems quite challenging." Another important limitation that is not discussed in [2] is that their scheme does not admit iterative cancellation and, hence, successive interference cancellation is not considered. Therefore, when more than T-users transmit in a slot, this slot is not utilized in the decoding process. As a result, their scheme uses a large number of slots in order to ensure that every user is received in a time slot that contains at most T-users, resulting in poor spectral efficiency.

The main contribution of this paper is to propose, analyze and optimize a new coding architecture that overcomes these drawbacks and substantially improves performance when compared to the state-of-the-art. Key features of our scheme are summarized as follows.

- **User Symmetry:** Active users employ the same coding scheme, with transmitted signals determined solely by the message to be transmitted and is independent of the identity of the user. To be precise, no parameter of the encoding scheme such as the interleaver and spreading sequence are unique to a transmitter.
- **Binary-input, real-adder channel:** The proposed coding scheme is tailored to the binary-input real-adder channel. The information message is split into two parts. The first portion picks an interleaver for an LDPC code, and the second part is encoded using this LDPC code. Bits associated with the first portion are communicated using a compressed sensing scheme. The second part is decoded using a message passing decoder that jointly recovers up to T messages within a slot.
- **Successive interference cancellation:** Active users repeat their codewords in several slots. The repetition patterns are selected based on message bits. This scheme facilitates interference cancellation within the slotted structure, and therefore renders obsolete the over-provisioning of slots to avoid undue collisions with more than T users.

While [2] also incorporates the user symmetry aspect described above, our scheme differs from theirs in the other features highlighted above.

II. System Model and Notation

Let K_{tot} and K_{a} denote the total number of users in the network and the number of active users, respectively. Each user has B bits of information (or, one of $M := 2^B$ indices) to be encoded and transmitted within a block of N uses of the channel. Let $W_i \in [1 : M]$ be a random variable that represents the message index of the i-th user and let w_i be a realization of the random variable W_i. Throughout this paper $[a : b]$ denotes the set of integers from a to b, both inclusive. We first assume that W_i is uniformly distributed over the set $[1 : M]$ and that for any pair of users i and j, W_i and W_j are independent of each other. Without loss of generality we assume that the set of indices of the active users is $[1 : K_{\text{a}}]$ and let $\mathcal{W} := \{w_1, w_2, \ldots, w_{K_{\text{a}}}\}$ denote the set of transmitted messages.

The observed signal vector at the receiver corresponding to the N channel uses can be written as

$$\vec{\mathrm{y}} = \sum_{i=1}^{K_{\text{tot}}} \vec{\mathrm{x}}_i + \vec{\mathrm{z}}, \tag{1}$$

where $\vec{\mathrm{x}}_i$ is a signal of dimension N transmitted by the user i, and the additive noise is characterized by $\vec{\mathrm{z}} \sim \mathcal{N}(0, \mathbf{I}_N)$. For convenience, we also introduce boolean indicators indexed by i, where $s_i = 1$ if user i is active and $s_i = 0$ otherwise. We impose an average power constraint on the transmitted vectors when averaged over all possible message indices, i.e., $\frac{1}{M}\sum_w ||\vec{\mathrm{x}}(w)||^2 \leq NP$. The receiver produces a list of messages $\mathcal{L}(\vec{y}) = \{\hat{w}_1, \hat{w}_2, \ldots, \hat{w}_{K_\mathrm{a}}\}$. As in [2], the probability of error is defined as

$$P_e = \max_{|(s_1,\ldots,s_{K_{\text{tot}}})|=K_\mathrm{a}} \frac{1}{K_\mathrm{a}} \sum_{i=1}^{K_{\text{tot}}} s_i \Pr\left(w_i \notin \mathcal{L}(\vec{y})\right) \tag{2}$$

where $|\cdot|$ denotes the Hamming weight. The objective is to design low-complexity encoding and decoding schemes such that $P_e \leq \varepsilon$, where ε is a target error probability.

III. Description of the Proposed Scheme

The overall schematic of the proposed scheme is shown in Fig. 1. In our proposed scheme, the N channel uses which are available for communication are split into V sub-blocks (also referred to as slots throughout the paper), each of length $\tilde{N} = N/V$ channel uses. The encoding operation at the i-th user takes place in two steps.

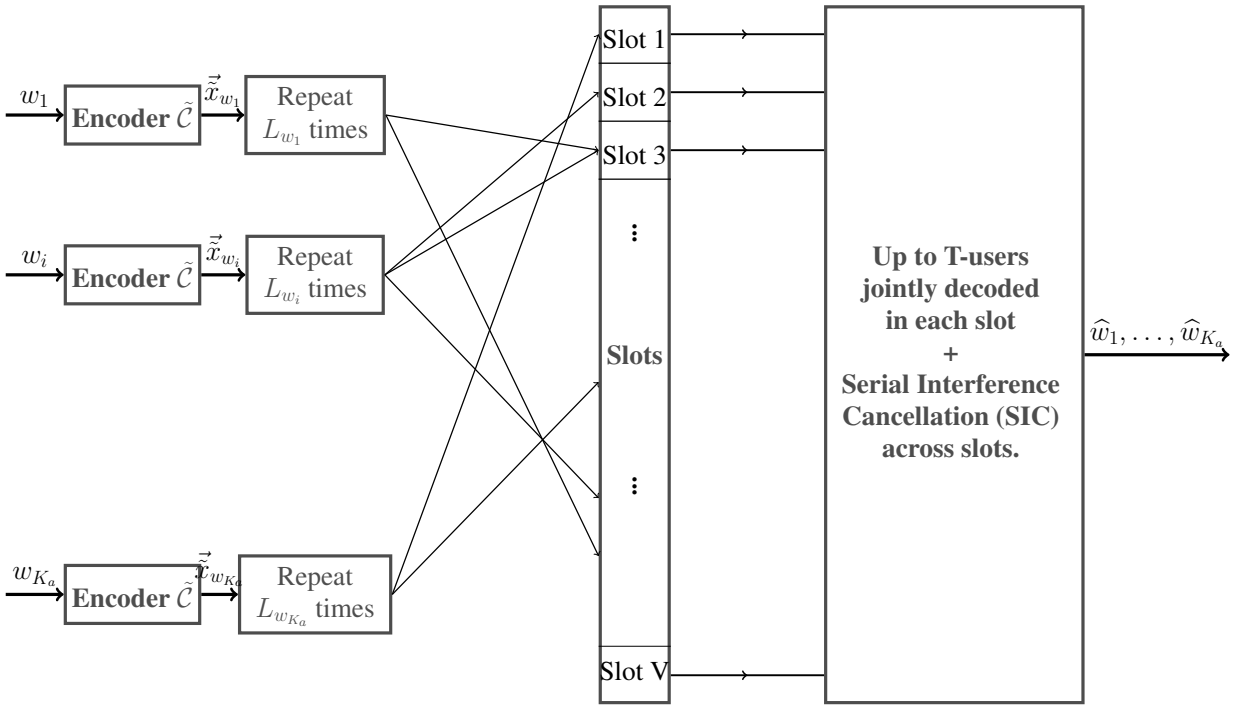

Fig. 1: Schematic of the proposed scheme

A. Transmission policy across sub-blocks - message based repetition

Each user uses an identical codebook $\tilde{\mathcal{C}}$ of rate $\tilde{R}$ and length $\tilde{N}$. The user encodes the message index w_i into a codeword $\vec{\tilde{c}}_{w_i} \in \tilde{\mathcal{C}}$ and modulates $\vec{\tilde{c}}_{w_i}$ into $\vec{\tilde{x}}_{w_i}$. In the following discussion, we will refer to $\vec{\tilde{c}}_{w_i}$ as the transmitted codeword and the reader should assume that the codeword is modulated appropriately and

transmitted. Each user also chooses a repetition parameter $\ell_{w_i} = g(w_i)$ using a function $g : [1 : M] \to [1 : V]$ and repeats their codeword $\vec{\tilde{c}}_{w_i}$, ℓ_{w_i} times by choosing ℓ_{w_i} sub blocks from $[1 : V]$ based on the message w_i and transmits during these sub blocks. It is important to note that ℓ_{w_i} as well as the slots where the codeword is repeated are deterministic functions of the message index and do not depend on the identity of the user. As shown in Fig. 1, a Tanner graph G can be used to visualize the repetition of the codewords where the left nodes correspond to users and the right nodes corresponds to sub-blocks. The degree of the left nodes is determined by ℓ_{w_i} and choosing w_i uniformly at random induces a distribution on ℓ_{w_i} through the function g. Let the left degree distribution (d.d) from node perspective be $L(x) = \sum_{i=1}^{l_{\max}} L_i x^i$, where L_i denotes the fraction of user (left) nodes that are connected to i slot(right) nodes. Similarly let the left d.d from edge perspective be denoted by $\lambda(x) = \sum_{i=1}^{l_{\max}} \lambda_i x^{i-1}$, where λ_i denotes the fraction of edges in G that are connected to left nodes connected to $i-1$ other edges. The two distributions $L(x)$ and $\lambda(x)$ are related as $L(x) = \frac{L'(x)}{L'(1)}$. We choose the mapping g such that a desired left d.d. $L(x)$ (or equivalently $\lambda(x)$) is obtained.

During the j-th sub-block, let $\mathcal{N}_j$ denote the set of users who transmit. During the n-th use of the channel within the j-th sub-block, the i-th user transmits symbols of positive power if $i \in \mathcal{N}_j$. Otherwise, the i-th user remains silent. The received signal during the j-th sub-block is given by

$$\vec{\tilde{y}}_j = \sum_{i \in \mathcal{N}_j} \vec{\tilde{x}}_{w_i} + \vec{\tilde{z}}_j. \tag{3}$$

B. Transmission policy within each sub block - same codebook schemes for the T-user real output channel

There are two components in the proposed transmission scheme within each sub-block: a good sensing matrix for a T-sparse robust compressed sensing (CS) problem and a good channel code for the T-user binary-input real-adder channel that is decodable with low computational complexity. The B bits to be transmitted are split into two groups of size $B_{\rm p}$ and $B_{\rm c} = B - B_{\rm p}$ bits, respectively. For convenience, we define $M_{\rm p} := 2^{B_{\rm p}}$ and $M_{\rm c} := 2^{B_{\rm c}}$. The main idea is to use a linear code $\mathcal{C}$ to encode $B_{\rm c}$ bits, which we refer to as channel coding message index/bits; and to use the remaining $B_{\rm p}$ bits, which we refer to as preamble message index/bits, to pick a permutation of the codeword encoded through the channel coding message bits. Typically, we want $B_{\rm p} \ll B_{\rm c}$.

The codebook $\tilde{\mathcal{C}}$ is constructed as follows. We begin with a good linear block code such as a low density parity check (LDPC) code or a spatially-coupled low density parity check (SCLDPC) code $\mathcal{C}$ of rate $\tilde{R} = \frac{B_{\rm c}}{\tilde{N}-J}$ and length $N' := \tilde{N} - J$. As an example, we will consider the case when $\mathcal{C}$ is chosen uniformly at random from the $(\mathsf{l}, \mathsf{r}, \mathsf{w}, \mathsf{N}')$ SCLDPC ensemble[7]. Let the modulated codewords of $\mathcal{C}$ be denoted by $\{\vec{c}_1, \vec{c}_2, \ldots, \vec{c}_{M_{\rm c}}\}$, where $\vec{c}_w = [c_w(1), c_w(2), \ldots, c_w(N')]$, $c_w(i) \in \{\pm\sqrt{P_1}\}\ \forall i$ satisfying the power constraint

$$||\vec{c}_w||_2^2 \leq N' P_1 \tag{4}$$

denotes the modulated SCLDPC codeword corresponding to message index w. For the second part of the encoder let $\mathbf{A} \in \{-\sqrt{P_2}, +\sqrt{P_2}\}^{J \times M_2}$ denote a sensing matrix that can recover the sum of any T columns of $\mathbf{A}$ with low error probability. Let $f : [1 : M_{\rm p}] \to [1 : N'!]$ denote a hash function which maps B_2 preamble message bits into an integer $\tau_w = f(w)$ such that τ_w is uniformly distributed over possible permutations of length N', $[1 : N'!]$. Note that here the integer τ_w chooses the permutation $\pi_{\tau_w} \in S_{N'}$ of the encoded codeword from $\mathcal{C}$ before transmission where $S_{N'}$ is the symmetric group.

Let $\vec{\tilde{c}}_w$ be the codeword corresponding to message index $w = (w_{\rm p}, w_{\rm c})$, where the indices $w_{\rm p}, w_{\rm c}$ correspond to the preamble and coding message indices respectively. $\vec{\tilde{c}}_w$ is obtained by first permuting the codeword of the SCLDPC code $\vec{c}_{w_{\rm c}}$ according to permutation $\pi_{\tau_{w_{\rm p}}} = [\pi^1_{\tau_{w_{\rm p}}}, \pi^2_{\tau_{w_{\rm p}}}, \ldots, \pi^{N'}_{\tau_{w_{\rm p}}}]$ and then inserting the $w_{\rm p}$th column from the compressed sensing matrix $\mathbf{A}$ at the beginning of the permuted codeword i.e.,

$$\vec{\tilde{c}}_w = [\vec{a}_{w_{\rm p}}, c_{w_{\rm c}}(\pi_{\tau^1_{w_{\rm p}}}), c_{w_{\rm c}}(\pi_{\tau^1_{w_{\rm p}}}), \ldots, c_{w_{\rm c}}(\pi_{\tau_{w_{\rm p}}})]. \tag{5}$$

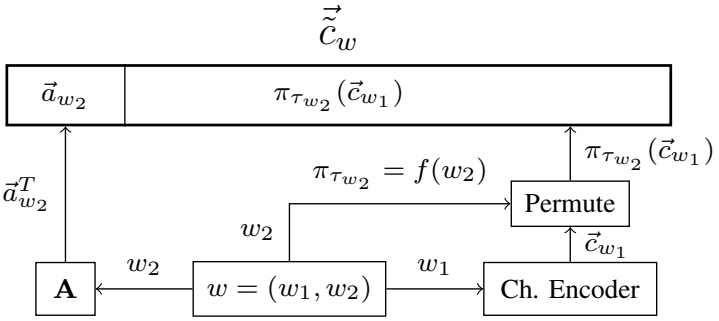

Fig. 2: At any user, for a message bit sequence $w = (w_{\rm p}, w_{\rm c})$ the schematic shows the encoder and the final transmitted codeword $\vec{\tilde{c}}_w$.

The overall encoding process is shown in the Fig. 2. The main idea here is that permuting the codeword $\vec{c}_{w_{\rm c}}$ decorrelates the multiple access interference from users even though they use identical linear codes and results in a performance that is similar to that obtained by using different codes for the different users. This is similar to interleave-division multiple access scheme that was originally proposed in [4]. The overall code is non-linear because of the random permutations for different codewords and $\vec{a}_{w_{\rm p}}$ being appended at the beginning; however, if $\vec{a}_{w_{\rm p}}$ is identified (and consequently also $w_{\rm p}$) and removed at the receiver, then the permutations can be determined and decoding the users can be accomplished using a belief propagation decoder that works on the joint graph of the two users.

C. Decoder

The overall decoder has two components - a decoder for the T-user Gaussian multiple access(GMAC) channel and a successive interference canceler that works across sub-blocks. Let $R_j = |\mathcal{N}_j|$ denote the number of codewords transmitted during the j-th sub-block. The decoder observes the received signal during each sub-block and first estimates $R_j, \forall j \in [1 : V]$. A simple estimate can be obtained by observing the energy of the received signal during each sub-block, but more sophisticated estimates based on decoding can be obtained, if necessary.

1) Compressed Sensing Decoder: The first J received symbols in the jth sub-block can be written in matrix vector form as

$$\vec{y}_j[1:J] = \mathbf{A}\vec{b} + z_j[1:J] \tag{6}$$

where $\mathbf{A} \in \mathbb{R}^{J \times M_{\rm p}}$ is the sensing matrix and $\vec{b} \in \{0,1\}^{M_{\rm p}}$ is a T-sparse vector that indicates the set of transmitted messages during the jth slot. Our proposed receiver to recover $\vec{b}$ from $\vec{y}_j[1:J]$ exploits the sparsity of $\vec{b}$ as well as the fact that the non-zero entries of $\vec{b}$ are positive integers (particularly, 1's). The latter aspect makes our proposed receiver different from many standard compressed sensing reconstruction algorithms.

We first run a non-negative least squares (or non-negative ℓ_1-regularized LASSO) algorithm that gives us an estimate $\hat{\vec{b}}$ of $\vec{b}$. But these algorithms do not guarantee an output signal either of the required sparsity or with elements strictly from the set $\{0,1\}$ (as we know apriori from the problem). To address this, we perform a hard thresholding operation on each element of $\hat{\vec{b}}$ and form a list of non-negative indices $\mathcal{N}_{\rm list} = \{i : \hat{b}_i > \eta_{Th}\}$. The value of parameter η_{Th} is chosen such that the list size is larger than T. We then implement a maximum likelihood decoder within the above list of indices to find the set of T indices that best explain the received vector $\vec{\rm y}$ i.e.,

$$\widehat{\mathcal{N}} = \underset{S \subseteq \mathcal{N}_{\rm list}, |S|=T}{\arg\min} ||\vec{y} - \sum_{i \in S} \vec{a}_i||_2^2. \tag{7}$$

As one can observe as we decrease the value of the threshold η_{Th} the list size increases which increases the complexity of the MMSE estimator in Eq. (7) whereas if we increase the value of the threshold the

list size decreases and the performance worsens. Clearly for a given SNR the value of the threshold η_{Th} needs to be optimized. The CS decoder outputs the set of message indices $\widehat{\mathcal{N}}$, where $|\widehat{\mathcal{N}}| = T$, to the message passing decoder.

2) Message Passing for Gaussian MAC: We describe the decoding process for $T = 2$ which can be generalized to larger values of T in a straight forward manner. Consider a sub-block j with only two users transmitting and let us assume their message indices are $w^1 = (w_{\mathrm{p}}^1, w_{\mathrm{c}}^1)$ and $w^2 = (w_{\mathrm{p}}^2, w_{\mathrm{c}}^2)$ respectively. Note that the estimates of preamble message indices $\{w_{\mathrm{p}}^1, w_{\mathrm{p}}^2\}$ are available at the message passing decoder, output from the CS decoder. Assuming appropriate demodulation is performed before the decoding step the received signal in terms of the SC-LDPC codewords can be given by:

$$\begin{aligned}\tilde{\vec{y}}_j =&[\vec{a}_{w_{\mathrm{p}}^1}, c_{w_{\mathrm{c}}^1}(\pi_{\tau^1_{w_{\mathrm{p}}^1}}), c_{w_{\mathrm{c}}^1}(\pi_{\tau^2_{w_{\mathrm{p}}^1}}), \ldots, c_{w_{\mathrm{c}}^1}(\pi_{\tau^{N'}_{w_{\mathrm{p}}^1}})]+\\ &[\vec{a}_{w_{\mathrm{p}}^2}, c_{w_{\mathrm{c}}^2}(\pi_{\tau^1_{w_{\mathrm{p}}^2}}), c_{w_{\mathrm{c}}^2}(\pi_{\tau^2_{w_{\mathrm{p}}^2}}), \ldots, c_{w_{\mathrm{c}}^2}(\pi_{\tau^{N'}_{w_{\mathrm{p}}^2}})] + z_j\end{aligned}$$

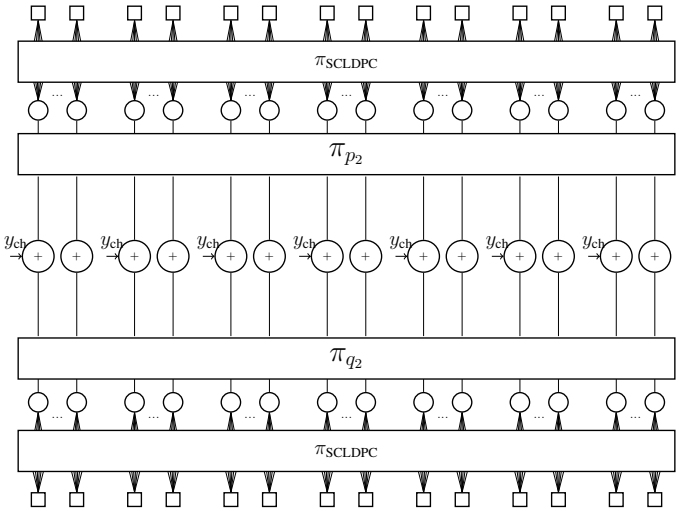

Fig. 3: Schematic showing the joint Tanner graph

Recall that $N' = \tilde{N} - J$ where $\tilde{N}$ is the number of channel uses in a sub-block. First we separate the channel coding part of the received vector by considering only the last N' values of the $\tilde{N}$ sized received vector i.e.,

$$\begin{aligned}\vec{\mathrm{y}}' =&[\vec{a}_{w_{\mathrm{p}}^1}, c_{w_{\mathrm{c}}^1}(\pi_{\tau^1_{w_{\mathrm{p}}^1}}), c_{w_{\mathrm{c}}^1}(\pi_{\tau^2_{w_{\mathrm{p}}^1}}), \ldots, c_{w_{\mathrm{c}}^1}(\pi_{\tau^{N'}_{w_{\mathrm{p}}^1}})]+\\ &[\vec{a}_{w_{\mathrm{p}}^2}, c_{w_{\mathrm{c}}^2}(\pi_{\tau^1_{w_{\mathrm{p}}^2}}), c_{w_{\mathrm{c}}^2}(\pi_{\tau^2_{w_{\mathrm{p}}^2}}), \ldots, c_{w_{\mathrm{c}}^2}(\pi_{\tau^{N'}_{w_{\mathrm{p}}^2}})] + z_j\end{aligned}$$

which is input to the joint belief propagation(BP) decoder. As we can observe, the code bits/symbols in each codeword are permuted according to a random permutation (chosen according to the second part of the message index) before being transmitted across the GMAC channel. Therefore in the joint BP decoder we need to apply these permutations and their inverses on the messages whenever they are being sent to and from the MAC nodes respectively. The schematic of the joint Tanner Graph of the two users is shown in Fig. 3.

Given the received signal $\vec{\mathrm{y}}'$ the joint BP decoder proceeds iteratively in a similar manner to that of a single user AWGN channel coding problem apart from an extra step of messages being sent to and received from the MAC node in each iteration. Throughout this sub-section we use superscript to distinguish between users 1 and 2.

- $u^1_{i,\text{MAC}}, u^1_{i,j}$: Messages passed from the bit node corresponding to i^{th} code bit of user 1 to the corresponding MAC node and SCLDPC check node j respectively.
- $v^1_{j,i}$: Message passed from SCLDPC check node j of user 1 to the bit node i of user 1
- $v^1_{\text{MAC},i}$: Message passed from i^{th} MAC node to the connected bit node of user 1.

The messages for user 2 are defined similarly. Refer to Fig. 4 for a graphical representation of the messages.

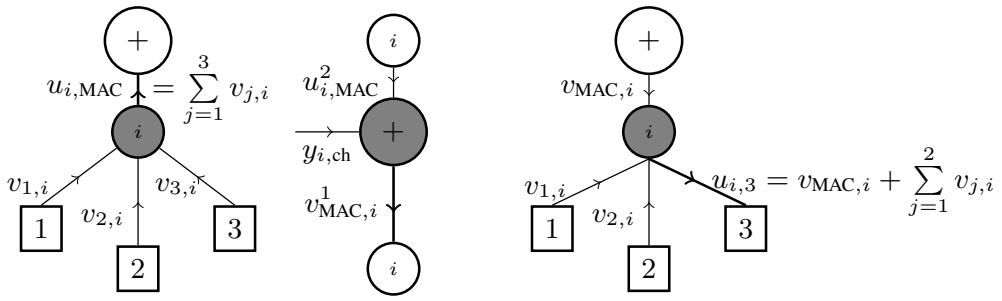

Fig. 4: Message passing rules at individual nodes on the joint Tanner graph of two users. The message passing rules at the check nodes of the SCLDPC code are identical to the single user channel coding case.

Bit Node:

$$u^1_{i,j} = v^1_{\text{MAC},i} + \sum_{j' \neq j, j' \in \mathcal{N}(i)} v^1_{j',i}$$

$$u^1_{i,\text{MAC}} = \sum_{j \in \mathcal{N}(i)} v^1_{j,i}$$

LDPC Check Node:

$$v^1_{j,i} = 2 \ \tanh^{-1}\left(\prod_{i' \neq i} \tanh\left(\frac{u^1_{i',j}}{2}\right)\right)$$

MAC Node:

$$\begin{aligned} v^1_{\text{MAC},i} &= h(u^2_{i,\text{MAC}}, y_{i,\text{ch}}) \\ v^2_{i,\text{MAC}} &= h(u^1_{i,\text{MAC}}, y_{i,\text{ch}}) \quad \text{where} \\ h(l, y) &= \log \frac{1 + e^l e^{2(y-1)/\sigma^2}}{e^l + e^{-2(y+1)/\sigma^2}}. \end{aligned} \tag{8}$$

The function $h(l, y|\sigma^2)$ can be seen as the log-likelihood of variable x_2 when $y = x_1 + x_2 + z$, $x_1, x_2 \in \{-1, +1\}$ when the log-likelihood ratio of variable x_1 is known to be l and $z \sim \mathcal{N}(0, \sigma^2)$.

IV. Choice of Parameters and Analysis

In this section we will look at the parameters of the different components of the proposed and how each of them affect the overall performance. As described in Sec. II let ε be the maximum per user error probability i.e. we are interested in decoding on an average $(1 - \varepsilon)$ fraction of the K_a active users successfully. Let us define the following error events before we analyze the per user error probability P_e. At any fixed slot j the events $\mathcal{E}_{1j}$, $\mathcal{E}_{2j}, \mathcal{E}_{3j}$ are defined as:

- $\mathcal{E}_{1j}$: Event that at least two users pick the same preamble message index
- $\mathcal{E}_{2j}$: Event that the CS decoder does not decode the preamble message index $w_{i,2}$ correctly for at least one user i.e., $\exists\ i \in \mathcal{N}_j : w_{i,2} \notin \widehat{\mathcal{N}}_j$

- $\mathcal{E}_{3j}$: Event that the MAC channel decoder fails to recover the remaining users, if less than or equal to T, in the slot
- $\mathcal{E}_{\text{SIC}}$: Event that a random active user is not recovered by the symmetric interference cancellation process

The per user error probability $P_e = \Pr(\mathcal{E}_{\text{SIC}})$ can be upper bounded as following:

$$\begin{aligned} P_e &= \Pr(\cap_{i,j}\mathcal{E}_{ij}^c)\Pr\left(\mathcal{E}_{\text{SIC}}|\bigcap_{i,j}\mathcal{E}_{ij}^c\right) + \Pr\left(\bigcup_{i,j}\mathcal{E}_{ij}\right)\Pr(\mathcal{E}_{\text{SIC}}|\cup_{i,j}\mathcal{E}_{ij}) \\ &\leq \Pr\left(\mathcal{E}_{\text{SIC}}|\bigcap_{i,j}\mathcal{E}_{ij}^c\right) + \Pr\left(\bigcup_{i,j}\mathcal{E}_{ij}\right) \\ &= \Pr\left(\mathcal{E}_{T\text{-peeling}}\right) + \Pr\left(\bigcup_{i,j}\mathcal{E}_{ij}\right) \\ &\leq \Pr\left(\mathcal{E}_{T\text{-peeling}}\right) + \sum_{j}(\Pr(\mathcal{E}_{1j}) + \Pr(\mathcal{E}_{2j}) + \Pr(\mathcal{E}_{3j})) \\ &= \Pr\left(\mathcal{E}_{T\text{-peeling}}\right) + V\left(\Pr(\mathcal{E}_1) + \Pr(\mathcal{E}_2) + \Pr(\mathcal{E}_3)\right) \end{aligned} \tag{9}$$

where we refer to the SIC process under the assumption that the preamble collision, CS decoder error and the MAC channel decoder error do not occur in any of the slots as T-peeling or ideal SIC process. In other words T-peeling is an iterative process in which at any slot if there are less than or equal to T-users connected we assume those T-users are decoded and their contributions are subtracted off (peeled off from the slot nodes) from the other slots in which these users have transmitted. Thus the event $\mathcal{E}_{\text{SIC}}$ conditioned on $\cap_{i,j}\mathcal{E}_{ij}^c$ is equivalent to the event a random user is not recovered under the T-peeling process which is denoted $\mathcal{E}_{T\text{-peeling}}$. The multiplicative factor V in the upper bound of P_e in Eqn. (9) union bounds the total number of instances(slots) a preamble collision ($\mathcal{E}_1$) or a compressed sensing decoder error ($\mathcal{E}_2$) or a MAC channel decoder error ($\mathcal{E}_3$) can occur.

A. Hash Collision

Lemma 1. $\Pr\left(\mathcal{E}_1\right) \approx \frac{T(T-1)}{2M_2}$

Proof. Let us consider the event $\mathcal{E}_{1j}^c$ where the R_j users in slot j picked a unique preamble message index (or equivalently a unique permutation). Note that in total there are M_2 possible preamble indices for each user.

$$\begin{aligned} \Pr(\mathcal{E}_{1j}^c) &= \frac{M_2(M_2-1)\dots(M_2-(R_j-1))}{M_2^{R_j}} \\ \implies \Pr(\mathcal{E}_{1j}) &= 1 - \prod_{i=0}^{R_j-1}(1-\frac{i}{M_2}) \\ &\approx \frac{R_j(R_j-1)}{2M_2}. \end{aligned} \tag{10}$$

According to the encoding scheme each user chooses the slot randomly according to the message index and hence the degree distribution for the slots is non-uniform. But we observe that the CS and the channel decoders are run on a slot only when the remaining number of connections are estimated to be less than T and hence we can approximate the number of connections to T to get a uniform upper bound across the slots. Substituting T for R_j in Eqn. (10) completes the proof. □

B. Compressed sensing problem and design choices

Now we discuss our choice of parameters for T, B_1, B_2 and the sensing matrix $\mathbf{A}$. First let us start with the sensing matrix and the recovery of the sparse signal that aids in recovering the set of preamble message indices. In this paper the typical values considered for T are $\{2, 4\}$. A low probability of error for event $\mathcal{E}_2$, for such low values of T, translates to designing a sensing matrix $\mathbf{A}$ where we require:

1) A large minimum distance in the Euclidean space between distinct T-sums of columns and
2) a minimal number of T-sets of columns whose sum is identical.

Before we formalize the above mentioned notions, we would like to note that, for the choice of $\mathbf{A}$, we considered the superimposed codes proposed by authors Fan, Darnell and Honary for the multiaccess binary adder channel [8]. In this work the authors consider binary codes and show that every constant weight code with weight w and maximum correlation c corresponds to a subclass of disjunctive code of order $T < \frac{w}{c}$. In other words, for any $T < \frac{w}{c}$ sum of any T codewords from this code results in a distinct output. Although the superimposed codes solve the second requirement we mentioned above they do not consider the first requirement i.e., the larger minimum distance of the resulting signal space of T-sums of codewords which is also critical to obtain a low probability of decoding error values. We defer the discussion of these results and our result relaxing the constraint of *constant weight* to appendix.

In the following subsection we introduce lattice and derive upper bounds on $\Pr(\mathcal{E}_2)$ based on maximum-likelihood decoder for lattices.

Definition 2. *A lattice Λ in n-dimensional Euclidean space $\Lambda \subset \mathbb{R}^n$ can be defined as:*

$$\Lambda = \{\lambda \in \mathbb{R}^n : \lambda = \mathbf{G}\mathbf{u}, \mathbf{u} \in \mathbb{Z}^m\} \tag{11}$$

where $\mathbf{M} \in \mathbb{R}^{n \times m}$ is called the generator matrix of the lattice. We define the minimum distance $d_{\min}(\Lambda)$ of the lattice Λ as

$$d_{\min}(\Lambda) \triangleq \min_{\lambda_1, \lambda_2 \in \Lambda} ||\lambda_1 - \lambda_2||_2.$$

Let the set of codewords/columns of $\mathbf{A}$ be denoted by $\mathcal{C}$ and $\mathcal{C} \subseteq \mathcal{C}_{\text{lin}}$ where $\mathcal{C}_{\text{lin}}$ is a binary linear code. We can then observe that the set of T-sums of codewords is a subset of lattice formed from $\mathcal{C}_{\text{lin}}$ ie..,

$$\sum_{j=1}^{T} \vec{a}_{i_j} \in \Lambda \quad i_j \in [1 : M_2]$$

where $\Lambda = \{\mathbf{G}\mathbf{u}, \mathbf{u} \in \mathbb{Z}^m\}, \mathbf{G}$ is the generator matrix of the binary code $\mathcal{C}_{\text{lin}}$. Now that the connection between the T-sums of the binary code and the lattice in which they are contained in is established we formalize the two requirements on $\mathbf{A}$ mentioned above.

Definition 3. *For a given binary code $\mathcal{C}$ and fixed T, for a set subset S of size T, we define the indicator parameter*

$$\beta_T(S) \triangleq \mathbf{1}[\exists S' \text{ s.t. } \; u(S) = u(S'), |S'| = T, S' \neq S],$$

where $u(S) := \sum_{i \in S} \vec{c}_i$ and $\beta_T(S)$ indicates if the T sum of codewords for the index set S is unique in the set of T-sums of codewords from $\mathcal{C}$. The second requirement mentioned above translates to minimizing $\beta_T(\mathcal{C})$. Similarly we define $\beta_T(\mathcal{C}) \triangleq \sum_{S \subset [1:|\mathcal{C}|]} \beta_T(S)$ counts the total number of subsets whose sum is not unique in the set of T-sums of codewords from $\mathcal{C}$.

Definition 4. *For a given binary code $\mathcal{C}$, fixed T, we define the minimum Euclidean distance of a set S in the space of T-sums of codewords as*

$$d_{\min}(S; \mathcal{C}) \triangleq \min_{S \neq S', |S| = |S'| = T} ||u(S) - u(S')||_2 .$$

Also the following relation combining the three quantities above can be observed:

$$d_{\min}(S;\mathcal{C}) \begin{cases} \geq d_{\min}(\Lambda) & \text{if } \beta_T(S) = 0 \\ = 0 & \text{otherwise.} \end{cases} \tag{12}$$

We will upper bound the probability of decoding error for the CS problem in terms of the parameters defined in Def. 3 and 4.

Lemma 5. *Let $\mathcal{C} \subseteq \mathcal{C}_{lin}$, where $\mathcal{C}_{lin}$ is a linear code containing $\mathcal{C}$, be a binary code with parameters $(n, M, d_{\min})$. The probability of error of the bounded distance decoder in decoding $\vec{z} = \sum_{i\in S,|S|=T} \vec{c}_i + \vec{n}$ where $\vec{n} \sim \mathcal{N}(0, \sigma^2\mathbf{I})$ can be upper bounded by*

$$Pr(\mathcal{E}_2) \leq \frac{\beta_T(\mathcal{C})}{\binom{|\mathcal{C}|}{T}} + \left(\frac{e d_{\min}^2(\Lambda)}{4\sigma^2 N} e^{\frac{-d_{\min}(\Lambda)^2}{4\sigma^2 N}} \right)^{N/2}$$

where N is the blocklength of the code $\mathcal{C}$.

Proof. We recall that the error event $\mathcal{E}_2$ is defined as the event in which the CS decoder fails to decode the set S exactly from

$$\vec{\mathrm{y}} = \sum_{i\in S} \vec{a}_i + \vec{z}$$

where $|S| = T$. When we condition event $\mathcal{E}_2$ on the T-sum of vectors from S not being unique, which happens with probability $\frac{\beta_T(\mathcal{C})}{\binom{|\mathcal{C}|}{T}}$ the first part of the bound is obtained. If we assume that the T-sum of vectors from S is unique, then the probability of error in decoding the set S under bounded distance decoding can be upper bounded by $\Pr\left[||\vec{z}|| \geq \frac{d_{\min}(\Lambda)}{2}\right]$ which is equivalent to

$$\Pr\left[\sum_{i=1}^{N} z_i^2 \geq \frac{d_{\min}^2(\Lambda)}{4\sigma^2}\right]$$

where $z_i \sim \mathcal{N}(0,1)$. The result is obtained by using the right tail bounds of Chi-squared distribution. □

We should note that it is not easy to compute the values of $\beta_T(\mathcal{C})$ especially for higher values of T or M_2. Following the constant weight code argument in [8] we provide upper bounds, in terms of the minimum and maximum Hamming weights of the binary code $\mathcal{C}$, on the values of T for which $\beta_T(\mathcal{C}) = 0$.

Lemma 6. *A binary code $\mathcal{C}$ with parameters $(n, M, d_{\min}, w_{max})$ is also a disjunctive code of order (n, M, T) for all T satisfying*

$$T < \frac{w_{min}}{w_{max} - d_{\min}/2}. \tag{13}$$

where w_{min} and w_{max} respectively are the minimum and maximum Hamming weights of all codewords in the code. Note that the values of $d_{\min}$ and w_{min} are not necessarily equal for non-linear codes.

The proof is provided in Appendix. We look at an example of sensing matrix $\mathbf{A}$ based on BCH code to consider how the results so far impact the probability of decoding error.

Example 7. *Consider a binary BCH code $\mathcal{C}$ with parameters $(n, k, d_{\min}) = (63, 10, 27)$. Let the subset $\mathcal{C}_0 \subset \mathcal{C}$ be obtained via the following decomposition:*

$$\mathcal{C} = \mathcal{C}_0 \cup \mathcal{C}_1 \quad \textit{such that } c \in \mathcal{C}_0 \iff \bar{c} \in \mathcal{C}_1,$$

where $\bar{c} = \mathbf{1} \oplus c$ is the one's complement of c. For the code $\mathcal{C}_0 \backslash \mathbf{0}$, $\mathbf{0}$ being the all-zero codeword, the weight and distance parameters are computed to be $(w_{min}, w_{max}, d_{\min}) = (27, 36, 27)$ for which the bound in Eqn. (13) is $T \leq 1$. But numerically we observe that this code produces unique outputs from the

MAC adder channel atleast upto values of $T = 3$*. The parameters* $\beta_T(\mathcal{C}_0)$ *and* $d_{\min}(\mathcal{C}_0, T)$ *are computed numerically for* $T \leq 1$ *and are given by :*

T	$d_{\min}$	$\beta_T(\mathcal{C})$
1	$\sqrt{27}$	*0*
2	$\sqrt{27}$	*0*
3	$\sqrt{27}$	*0* .

C. Channel coding problem

In the following subsection we will look at the analysis of the T-GMAC channel coding problem and the bounds on performance. Although the information theoretic limits for the multiple access problem especially the symmetric rate region are well known these do not prove to be very useful for our purposes. It is because even though the block lengths we are interested in are considerably large the information length (or equivalently the code size for each user) is small. Therefore we will be considering the finite length performance especially we will use the finite length random coding bounds for the Gaussian multiple access channel derived by Polyanskiy [1]. The following lemma is identical to Thm. 1 in [1] except for the difference that we are interested in the case where error is declared if atleast one of the users messages is not in the decoded set (see event $\mathcal{E}_{3j}$) in contrast to [1] where the error probability is defined similar to Eqn. (2).

Lemma 8. *There exists an* (N', M_1) *random-access code for* T*-user satisfying the power constraint* P *(see Eqn.* (4)*) with the probability of error under maximum-likelihood decoder bounded by*

$$P(\mathcal{E}_3) \leq \mathsf{h}_{FBL}(N', M_1, T, P) := \sum_{t=1}^{T} \min(p_t, q_t) + p_0, \tag{14}$$

where

$$\begin{aligned}
p_0 &= \frac{\binom{T}{2}}{M_1} + T\Pr\left(\sum_{j=1}^{N'} Z_j^2 > \frac{N'P}{P'}\right) \\
p_t &= e^{-N'E(t)} \\
E(t) &= \max_{0\leq\rho,\rho_1\leq 1} -\rho\rho_1 t R_1 - \rho_1 R_2 + E_0(\rho, \rho_1) \\
E_0 &= \rho_1 a + \frac{1}{2}\log(1 - 2b\rho_1) \\
a &= \frac{\rho}{2}\log(1 + 2P't\lambda) + \frac{1}{2}\log(1 + 2P't\mu) \\
b &= \rho\lambda - \frac{\mu}{1 + 2P't\mu}, \mu = \frac{\rho\lambda}{1 + 2P't\lambda} \\
\lambda &= \frac{P't - 1 + \sqrt{D}}{4(1 + \rho_1\rho)P't} \\
D &= (P't - 1)^2 + 4P't\frac{1 + \rho\rho_1}{1 + \rho} \\
R_1 &= \frac{1}{N'}\log M_1 - \frac{1}{N'}\log(t!) \\
R_2 &= \frac{1}{N'}\log\binom{T}{t} \\
q_t &= \inf_{\gamma} \Pr[I_t \leq \gamma] + \exp\{N'(R_1 + R_2) - \gamma\}.
\end{aligned}$$

and

$$I_t = \min_{|S_0|=t, S_0 \subseteq [T]} N' C_t + \frac{\log e}{2} \left(\frac{|| \sum_{i \in S_0} \vec{c}_i + \vec{z}||_2^2}{1 + P't} - ||\vec{z}||_2^2 \right)$$
$$C_t = \frac{1}{2} \log(1 + P't)$$
$$\vec{z} \sim \mathcal{N}(0, \mathbf{I}_{N'}).$$

Proof. In [1], author Y. Polyanskiy considers the T-user GMAC problem with power constraint P according to Eqn. (4). Let W be the set of messages of size T, chosen by the users uniformly without replacement and $\hat{W}$ be the set of messages of size T output by the decoder. The author considers a random Gaussian codebook generated from Gaussian process $\mathcal{N}(0, P'\mathbf{I}_n)$, $(P' < P)$, and maximum-likelihood decoder and shows that

$$\Pr(|W \backslash \hat{W}| = t) = \min(p_t, q_t). \tag{15}$$

It was also shown that p_0 is the total variation distance of a random variable of maximum value 1 when the measure under which a) the messages are sampled *independently* rather than *without replacement* and b) the codeword is set to zero-vector if the total power of the random codeword is larger than nP is replaced by the measure considered in showing Eqn. (15) i.e., messages sampled independently and disregarding the strict power constraint on each codeword. These results along with the observation that

$$\Pr(\mathcal{E}_3) = 1 - \prod_{i=1}^{t} \Big(1 - \Pr(\hat{w}_i \notin \hat{W})\Big) + p_0$$
$$\leq \sum_{t=1}^{T} \Pr(|S \backslash \hat{S}| = t)$$

completes the proof. □

D. Symmetric Interference Cancellation

In the channel coding literature for LDPC codes on binary erasure channel and sparse signals via Tanner graphs literature the symmetric interference cancellation is traditionally studied under the name of peeling decoder which is an iterative process in which if a right node (slot in our case) is connected to only one left node (user) the corresponding left node and all its connections are peeled off from the bipartite graph. This is essentially the symmetric interference cancellation process described in Sec. II except that we peel off the connections from a right node if the number of variable nodes connected is less than or equal to T instead of 1. Although density evolution methods are well studied to predict the performance of such decoding processes all the existing density evolution methods are for values of $T = 1$. Before we address this issue let us define the considered peeling process precisely.

Definition 9 (T-peeling)**.** *We define an ideal SIC decoder as the decoder in which at each slot, if the number of users transmitted and are still undecoded is less than or equal to T, then the remaining undecoded users in that slot are decoded with zero error. In other words in the ideal SIC process there are no hash collisions in any slot and the channel and sparse signal decoders are assumed to be zero-error. This process proceeds iteratively until all the users are decoded or there are no slots with undecoded users less than or equal to T. We also refer to this as T-peeling process.*

Lemma 10 (Density Evolution (DE))**.** *Let the left and right degree distributions (d.d.) of the bipartite graph from the edge perspective be $\lambda(x)$ and $\rho(x)$. Then let x_t be the probability that an edge in the*

graph, in iteration t of the T-peeling process, is connected to a left node that is undecoded yet. Then the recurrence relation for x_t corresponding to the T-peeling process is given by

$$y_t = \left[\sum_{r=1}^{T} \rho_r + \sum_{r>T} \rho_r \left(\sum_{t=0}^{T-1} \binom{r-1}{t} (1-x_t)^{r-1-t} x_t^t\right)\right], \tag{16}$$

$$x_{t+1} = \lambda(1-y_t). \tag{17}$$

Proof. In the context of low density parity check(LDPC) codes the bipartite graph corresponds to the parity check matrix where the left and right nodes represent the bits of the codeword and the parity check equations respectively. If we consider an LDPC code under binary erasure channel where each bit is erased with probability ε, under the assumption that the bipartite graph is a tree, the probability that a random edge in the graph is an erasure in iteration t of the peeling process is given by [9]

$$y_t = \sum_{r=1}^{r_{\max}} \rho_r (1-x_t)^{r-1}, \tag{18}$$

$$x_{t+1} = \varepsilon\lambda(1-y_t). \tag{19}$$

Eqn. (18) is due to the observation that all the incoming messages at a check node are independent, due to the tree assumption, and the outgoing message on an edge from a check node of degree r is a non-erasure if and only if all the incoming messages are non-erasures. For degree distributions with finite maximum degree on the left and right it is shown that a graph chosen randomly from the ensemble (N, λ, ρ) is a tree with probability approaching 1 asymptotically in blocklength of the code.

Now if we consider an edge e connected to check node of degree r in the T-peeling process, the outgoing message is a non-erasure if and only if there are at most $T-1$ erasures in the remaining $r-1$ incoming edges. Thus the probability that the outgoing message from a check node of degree r is non-erasure, denoted by $y_{t,r}$, if the incoming message on the remaining $r-1$ edges is an erasure with probability x_t is equal to

$$\begin{aligned} y_{t,r} &= \sum_{t=0}^{T-1} \binom{r-1}{t} (1-x_t)^{r-1-t} x_t^t && \text{if } r > T \\ &= 1 && \text{else if } r \le T. \end{aligned}$$

Averaging over all edges where an edge is connected to a check node of degree r with probability ρ_r gives us Eqn. (16). □

Let $L(x) = \sum_{i=1}^{l_{\max}} L_i x^i$ be the left d.d according to which the the users choose their repetition parameters as described in Sec. III-A i.e., $\Pr(L_w = i) = L_i$. Also let the average left degree of this distribution be $l_{\text{avg}} = \sum_i i L_i$. Then according to our transmission policy the right d.d. $R(x)$ is Binomial distributed with parameters $(K_{\text{a}} l_{\text{avg}}, 1/V)$ and in the limit $K_{\text{a}} \to \infty$ $R(x)$ can be approximated as Poisson distribution with parameter $r_{\text{avg}} = \frac{K_{\text{a}} l_{\text{avg}}}{V}$. Thus, asymptotically in K_{a}, it can be seen that $R(x) = e^{-r_{\text{avg}}(1-x)}$ and $\rho(x) = R'(x)/R'(1) = e^{-r_{\text{avg}}(1-x)}$. For more details refer to [10].

Lemma 11. *For $V = \alpha K_a$ where α is fixed the asymptotic performance of our transmission scheme under the ideal SIC decoding process can be characterized by*

$$\lim_{K_a \to \infty} \Pr(\mathcal{E}_{SIC}(K_a, T)) = L(1-y_\infty)$$

$$\textit{where } y_\infty = \lim_{t \to \infty} y_t,$$

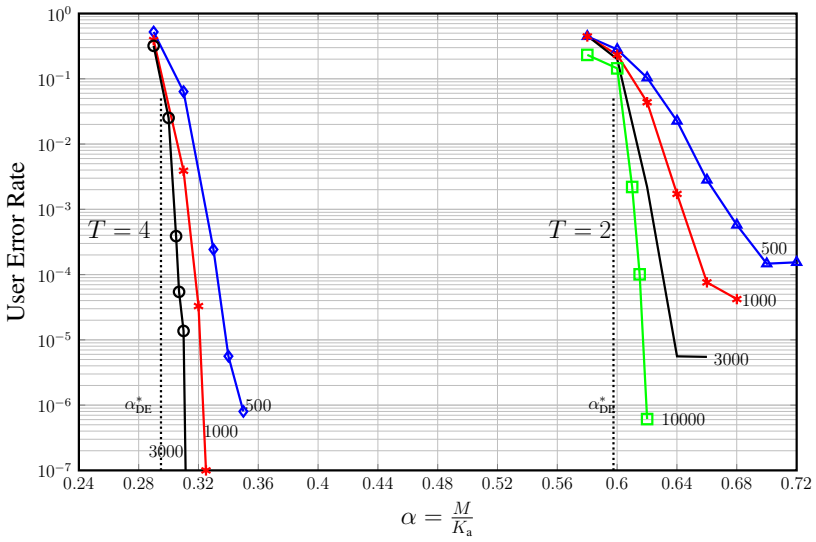

Fig. 5: α^*_{DE} is the density evolution threshold computed for $L(x) = x^2$ and $T = \{2, 4\}$ from Lemmma. (10). We validate the threshold behavior by evaluating the T-peeling performance via Monte Carlo simulations for increasing blocklengths. We observe that the simulations indeed confirm the threshold behavior for values of α above the DE threshold.

and $\Pr(\mathcal{E}_{SIC}(K_a, T))$ *is the probability that the ideal SIC process does not recover a user given there are* K_a *users. Here the initial condition is* $x_0 = 1$ *and the evolution of* x_t, y_t *is given by the DE relationship in Lem. 10.*

As we can see from Eqns. (17) and (16) that $x_t = 0$ is a fixed point if and only if $\lambda_0 = 0$. This leads us to the following result characterizing he threshold behavior of the system.

Definition 12 (Density Evolution Threshold)**.** *If* $L_1 = 0$ *we define the density evolution threshold* α^*_{DE} *to be*

$$\alpha^*_{DE} \triangleq \inf\{\alpha : \lim_{K_a \to \infty} \Pr(\mathcal{E}_{SIC}(K_a, T)) = 0\}.$$

We validate the threshold behavior via simulations. For a fixed left d.d $L(x) = x^2$ we first compute the density evolution thresholds according to Def. 12 to be 0.5975 and 0.2949 for $T = 2, 4$ respectively. We then perform Monte Carlo simulations where each time a random graph is chosen as described in Sec. III-A for increasing values of K_{a} and plot the performance as we incrase the number of slots. The results are presented in Fig. 5. In both the cases the threshold behavior can be clearly seen that as K_{a} increases the probability of a user not being decoded decreases sharply for values of $\alpha > \alpha^*_{\text{DE}}$ and remains fairly constant for values of $\alpha \leq \alpha^*_{\text{DE}}$.

V. Numerical Results

In this section we evaluate the overall performance of the proposed scheme and compare with the other schemes known in the literature. In [2] apart from proposing a low complexity coding scheme for the uncoordinated Gaussian multiple access the authors Ordentlich and Polyanskiy also compare the performance of their proposed scheme with ALOHA, treating inference as noise (TIN), and the random coding achievability bounds. To make the comparison fair and convenient we pick the same parameters. We fix the number of bits each user intend to transmit $B = 100$, total number of channel uses $n = 30,000$, total number of active users $K_{\text{a}} \in [25 : 300]$ and the maximum per user error probability $P_e \leq \varepsilon = 0.05$.

Remark 13. *Note that in Lemma. 1 the probability of preamble message collision scales only as* $\Pr(\mathcal{E}_1) \propto 1/M_2$ *and if the desired error probability is less than* 0.05 *which requires* $V\Pr(\mathcal{E}_1) \ll 0.05$ *which in turn*

requires M_2 to be a large value. Therefore we try to circumvent this issue by not counting a preamble collision in a slot as an error instead we modify the scheme where sparse signal decoder outputs a collision if the output set of messages of size T do not sufficiently explain the received vector i.e., if

$$||\vec{y} - \sum_{i \in \widehat{\mathcal{N}}} \vec{a}_i||_2^2 > J/2$$

then we do not decode this slot in this iteration. In our numerical simulations we update the definitions of $\mathcal{E}_{T\text{-peeling}}$ and $\mathcal{E}_2$ accordingly where $\mathcal{E}_{T\text{-peeling}}$ corresponds to the peeling process where a slot is not peeled off if it has atleast two users connected with identical preamble messages and $\mathcal{E}_2$ corresponds to the CS decoder error event where two or more users may have colliding preambles. With that the upper bound for error probability is modified to

$$P_e \leq \Pr(\mathcal{E}_{T\text{-peeling}}) + V\left(\Pr(\mathcal{E}_2) + \Pr(\mathcal{E}_3)\right). \tag{20}$$

With the parameters $B, n, K_{\mathrm{a}}, P_e$ fixed our proposed scheme has the following design parameters:

1) The maximum number of users to be jointly decoded at a slot $T \in \{2, 4\}$.
2) From Eqn. 20 we want $\Pr(\mathcal{E}_{T\text{-peeling}}) + V\left(\Pr(\mathcal{E}_2) + \Pr(\mathcal{E}_3)\right) \leq 0.05$. Therefore we set the target error probabilities for the individual events as $\Pr(\mathcal{E}_{T\text{-peeling}}) \leq \varepsilon_0 = 0.04$ and $\Pr(\mathcal{E}_i) \leq 0.01/2/V$, $i = 1, 2$.
3) The left d.d is chosen to be $L(x) = \lambda x + (1-\lambda)x^2$. Although in Sec. IV-Dwe remarked that if the minimum left degree is one then zero is not a fixed point for the DE equations or in other words, in the asymptotic regime, we will have error floors rather than threshold behavior. But the effects of a minimum left degree of one in the finite number of users regime are not very clear. The free parameter is chosen from the set $\lambda \in \{0, 0.1, \ldots, 0.9, 1\}$.
4) We choose $B_2 = 9$ and $B_1 = B - B_2 = 91$. Note that $M_2 = 2^{B_1} = 512$ is the size of the sensing matrix.
5) Let $\mathcal{C}_0$ be the binary code of size 2^9 described in Ex. 7. We choose the sensing matrix of size $J \times M_2 = 63 \times 512$ as

$$\mathbf{A} = [\vec{a}_1, \vec{a}_2, \ldots, \vec{a}_{M_2}], \tag{21}$$

where $\vec{a}_i = \sqrt{P_2}(1 - 2\vec{c}_i), \vec{c}_i \in \mathcal{C}_1$, i.e., $a_{ij} \in \{-\sqrt{P_2}, \sqrt{P_2}\}$.

For a fixed T the performance of the overall scheme i.e., the minimum E_b/N_0 required for achieving $P_e \leq \varepsilon$ is computed as following:

$$\frac{E_b}{N_0} = \min_{\lambda} \frac{(2-\lambda)(JP_2 + N'P_1)}{2B} \tag{22}$$

where

$$P_2 := \arg\min_{P_2} \ \Pr(\mathcal{E}_3) \leq \frac{\varepsilon - \varepsilon_0}{2V} \tag{23}$$

$$P_1 := \arg\min_{P_1} \ \mathsf{h}_{\mathrm{FBL}}(N', B_1, T, P_1)) \leq \frac{\varepsilon - \varepsilon_0}{2V} \qquad \text{(see Eqn. (14))} \tag{24}$$

$$N' := \tilde{N} - J$$

$$\tilde{N} := \left\lfloor \frac{n}{V} \right\rfloor$$

$$V := \arg\min_{V} \ \Pr(\mathcal{E}_{T\text{-peeling}}(K_{\mathrm{a}}, V, T)) \leq \varepsilon_0. \tag{25}$$

A remark on how we compute $\Pr(\mathcal{E}_3)$ is in order. With the sensing matrix fixed, as described in Eqn. (21), the bound given in Lem. 5 is firstly difficult to compute especially the parameters $K(\Lambda), \beta_T(\Lambda)$ and secondly it is not very tight in the non-asymptotic regime when compared with the results obtained through numerical simulations. Specifically, for simulations, we choose T random preamble message indices independently from the available M_2 indices and form the measurement vector. Then we use

a two-stage decoder as already described in Sec. III-C1 where for the first stage we employ a non-negative least squares algorithm and compute the probability of error $\Pr(\mathcal{E}_3)$ in Eqn. (23) from atleast 10^5 simulations. Similarly for $\Pr(\mathcal{E}_{T\text{-peeling}})$ in Eqn. (25) we rely on numerical simulations where we implement the T-peeling process described earlier in Def. 9. The results are presented in Fig. 6.

In Fig. 6 the curves labelled $T = 2$, $T = 4$ correspond to the performance of our proposed scheme assuming the existence of a coding scheme for the T-user multiple access channel that achieves the finite block length(FBL) bounds in [1]. The curve labelled 4-fold ALOHA is the performance of the 4-fold ALOHA scheme from [2] which also assumes exactly the same code. It can be seen that for large values of K_{a}, our proposed scheme with $T = 4$ substantially outperforms the 4-fold ALOHA and this gain is due to the SIC. The curve labelled OP-Exact is a reproduction of the results from [2] of the practical scheme introduced there. The x mark represents our proposed scheme where for the channel coding part instead of the FBL bounds we use the actual simulation results. We use a rate-1/4 $(364, 91)$ LDPC code obtained from repeating every coded bit of (3,6) LDPC code twice and a message passing decoder for $T = 2$. It can be seen that the simulation results with the $(3, 6)$ LDPC code are only 0.5 dB away from the curve corresponding to $T = 2$ showing that the pragmatic coding scheme can perform close to the finite length bounds. It can also be seen that our proposed scheme provides substantial gain over the results in [2]. The performance of OP-Exact is already substantially better than many other multiple access schemes and our proposed scheme can potentially provide more than 14 dB improvement over OP-Exact for large values of K_{a} and is only about 7 dB away from Polyanskiy's FBL bound [1]. The slope of the $T = 4$ curve also is much closer to the finite block length bound which is encouraging for larger values of K_a.

In the proposed encoding scheme, for $L(x) = \lambda x + (1 - \lambda)x^2$ each user may transmit once or twice depending on the message index chosen. We need to point out that the power constraint employed is an average over all the message indices i.e

$$\mathbb{E}_w \left[||\tilde{\vec{c}}_w||_2^2 \right] = (2 - \lambda)P.$$

REFERENCES

[1] Y. Polyanskiy, "A perspective on massive random access." Submitted to IEEE International Symposium on Information Theory, 2017.

[2] O. Ordentlich and Y. Polyanskiy, "Low complexity schemes for the random access Gaussian channel." Submitted to IEEE International Symposium on Information Theory, 2017.

[3] B. Rimoldi and R. Urbanke, "A rate-splitting approach to the Gaussian multiple-access channel," *IEEE Trans. Inform. Theory*, vol. 42, no. 2, pp. 364–375, 1996.

[4] L. Ping, L. Liu, K. Wu, and W. K. Leung, "Interleave division multiple-access," *IEEE Transactions on Wireless Communications*, vol. 5, no. 4, pp. 938–947, 2006.

[5] J. Zhan, B. Nazer, U. Erez, and M. Gastpar, "Integer-forcing linear receivers," *IEEE Transactions on Information Theory*, vol. 60, no. 12, pp. 7661–7685, 2014.

[6] P. Mathys, "A class of codes for a t active users out of n multiple-access communication system," *IEEE Transactions on Information Theory*, vol. 36, no. 6, pp. 1206–1219, 1990.

[7] S. Kudekar, T. Richardson, and R. Urbanke, "Threshold saturation via spatial soupling: Why convolutional ldpc ensembles perform so well over the bec," *IEEE Trans. Inform. Theory*, vol. 57, no. 2, pp. 803–834, 2011.

[8] P. Z. Fan, M. Darnell, and B. Honary, "Superimposed codes for the multiaccess binary adder channel," *IEEE transactions on information theory*, vol. 41, no. 4, pp. 1178–1182, 1995.

[9] T. Richardson and R. Urbanke, *Modern coding theory*. Cambridge University Press, 2008.

[10] K. R. Narayanan and H. D. Pfister, "Iterative collision resolution for slotted ALOHA: An optimal uncoordinated transmission policy," in *Proc. Intl. Symp. Turbo Codes and Rel. Topics.*, pp. 136–139, IEEE, 2012.

APPENDIX

In the following subsection we first present the main results from [8] that enabled the authors to show that constant weight codes are a subclass of disjunctive code. Then we follow it up with our result where we relax the constant weight constraint on the code to *nearly* constant weight.

Definition 14. *The maximum correlation* c *of a binary code* $\mathcal{C}$ *is defined as*

$$c = \max_{\vec{c}_i, \vec{c}_j \in \mathcal{C},\ i \neq j} < \vec{c}_i, \vec{c}_j > .$$

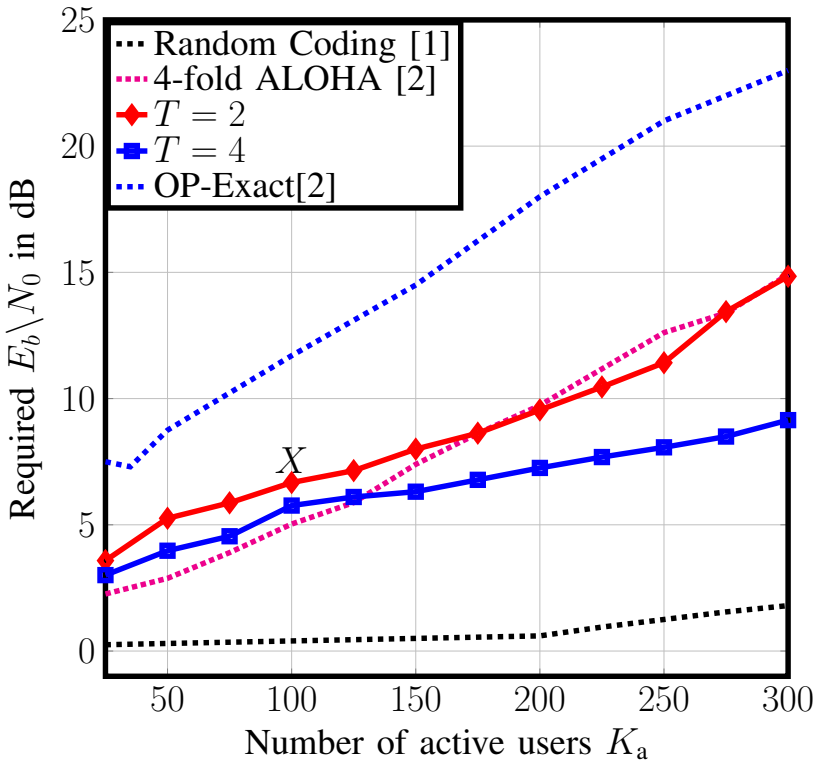

Fig. 6: Minimum E_b/N_0 required as a function of number of users. For the cases $T = 2, 4$ and the 4-fold ALOHA, the parameters chosen are $\varepsilon_0 = 0.04$ and $\varepsilon = 0.05$. This results in a total $\Pr_e \leq 0.05 (= \varepsilon)$.

Definition 15. *A binary vector $\vec{c} = [c(1), c(2), \ldots, c(n)]$ is said to be included in a vector $\vec{z} = [z(1), z(2), \ldots, z(n)]$ if and only if $z(i) \geq c(i)$ $\forall i$.*

Definition 16. *A binary code $\mathcal{C}$ with length n, size M is said to be a disjunctive code of order T if each subset $S \subset \mathcal{C}$ with size $|S| \leq T$ has the property that the vector $\vec{z}$ includes only those codewords in $\mathcal{C}$ that belong to S where*

$$\vec{z} = \sum_{\vec{c}_i \in S} \vec{c}_i \tag{26}$$

is the output of the multiple access real adder channel. We denote a disjunctive code by $D(n, M, T)$.

Definition 17. *A constant weight(CW) binary code is one in which all the codewords have equal weight w. For a CW code, the minimum distance d_{min} and the maximum correlation c are related as*

$$2c = 2w - d_{\min}.$$

We denote a constant code by parameters CW(n, M, w, c) where n, M are blocklength and size of the code respectively.

Lemma 18 ([8] Theorem 1)**.** *A constant weight binary code $\mathcal{C}$ with parameters (n, M, w, c) is also a disjunctive code of order (n, M, T) for all T satisfying*

$$T < \frac{w}{c}.$$

Example 19. *Consider a Reed-Solomon code $RS(n, k, d_{\min}) = RS(7, 3, 5)$. As described in [8] we construct a constant weight code by mapping each symbol in a codeword from GF(2^3) to a length 8*

binary vector of weight one

$$0 \to 10000000$$
$$1 \to 01000000$$
$$\dots$$
$$7 \to 00000001.$$

Note that this code has parameters $n = 56, M = 2^9, w = 7, d_{\min} = 10$ *which implies* $c = w - d_{\min}/2 = 2$. *Thus any* T*-sum of the codewords from this CW code is unique for all* $T \leq 3 < \frac{w}{c}$.

Now we relax the constant weight constraint in Lemma. 18 and give the corresponding bounds on the disjunctive code parameters.

Lemma 20. *For a binary code* $\mathcal{C}$ *with parameters* $(n, M, d_{\min}, w_{max})$*, where* w_{max} *is the maximum Hamming weight of all the codewords in the code, the maximum correlation between any two codewords can be given by*

$$c \leq w_{max} - d_{\min}/2.$$

Proof. For any two codewords $\vec{c}_i, \vec{c}_j \in \mathcal{C}$ the relationship between correlation, Hamming distance and sum of Hamming weights can be given by

$$d_H(\vec{c}_i, \vec{c}_j) + 2c(\vec{c}_i, \vec{c}_j) = w_H(\vec{c}_i) + w_H(\vec{c}_j)$$

where d_H and w_H are the Hamming distance and weights respectively. By substituting the lower and upper bounds $d_{\min}$ and $w_{\max}$ for the two parameters gives us the required upper bound on maximum correlation of any two codewords of the binary code. □

Proof of Lem. 6. Without loss of generality consider a set $S = \{\vec{c}_1, \vec{c}_2, \dots, \vec{c}_T\}$ of codewords of size T and let the output of the real adder multiple access channel, given by Eqn. (26), be $\vec{z}$. Let us consider codeword $\vec{c}_e \in \mathcal{C} \backslash S$ and look at the event in which $\vec{z}$ does not include $\vec{c}_e$. Let $s_{ie} := \{k : c_i(k) = c_e(k) = 1\}\ \forall i \leq T$ and $s_e = \{k : c_e(k) = 1\}$. Since $\vec{z} = \sum_{i \leq T} \vec{c}_i \implies z(k) \geq 1\ \forall k \in \cup s_{ie}$. Hence the condition that needs to be satisfied for $\vec{z}$ to not include $\vec{c}_e$ is that $\exists k : k \in s_e \backslash \cup\, s_{ie}$ which translates to

$$|\cup\, s_{ie}| < |s_e|. \tag{27}$$

The inequality in Eq. (27) is satisfied when $\sum_i c(\vec{c}_i, \vec{c}_e) < w_H(c_e)$ which is implied by the condition $Tc_{\max} < w_{\min}$ and from Lemma. 20 the required result follows. □