2021 IEEE International Symposium on Information Theory (ISIT) | 978-1-5386-8209-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/ISIT45174.2021.9517890

Managing Noise and Interference Separately - Multiple Access Channel Decoding using Soft GRAND

Amit Solomon
RLE, MIT
Cambridge, MA 02139, USA
Email: amitsol@mit.edu

Ken R. Duffy
Hamilton Institute
Maynooth University, Ireland
Email: ken.duffy@mu.ie

Muriel Médard
RLE, MIT
Cambridge, MA 02139, USA
Email: medard@mit.edu

***Abstract*—Two main problems arise in the Multiple Access Channel (MAC): interference from different users, and additive noise channel noise. Maximum A-Posteriori (MAP) joint decoding or successive interference cancellation are known to be capacity-achieving for the MAC when paired with appropriate codes. We extend the recently proposed Soft Guessing Random Additive Noise Decoder (SGRAND) to guess, using soft information, the effect of noise on the sum of users' transmitted codewords. Next, we manage interference by applying ZigZag decoding over the resulting putative noiseless MAC to obtain candidate codewords. Guessing continues until the candidate codewords thus obtained pertain to the corresponding users' codebooks. This MAC SGRAND decoder is a MAP decoder that requires no coordination between users, who can use arbitrary moderate redundancy short length codes of different types and rates.**

***Index Terms*—MAC, MAP decoding, GRAND, ZigZag decoding.**

I. Introduction

The non-orthogonal Multiple Access Channel (MAC) is a commonly used communications model [1]. In MAC, users transmit simultaneously over a shared channel. The users' modulated codewords are added to each other and to channel noise. The receiver must thus resolve two effects: interference and noise. The approach for doing so that was first proposed by Ahlswede [2] and by Liao [3], which attempts to solve the problem of interference and the noise concurrently, is the following: when decoding one user's message, the receiver treats undecoded modulated codewords as noise, with attendant effects on that user's rate [1]. It can achieve capacity, i.e. reach the boundary of the Cover-Wyner region [1], with some level of coordination in users' transmission. Types of coordination include separating different users into orthogonal channel in time through Time Division Multiplexing Access (TDMA), in frequency through Frequencey Division Multiplexing Access (FDMA), or through code selection Code Division Multiple Access (CDMA) [4], possibly with rate splitting Rate Splitting Multiple Access (RSMA) [5]).

The approach we propose requires no coordination whatsoever from different users. Different users can transmit using different codes, lengths, rates, etc. In effect, the method lets users operate as if they use orthogonal channels. Its core is to deal separately with additive interference and with noise. Let us consider the former first. If we have interference in the absence of noise, the resulting noiseless MAC, sometimes termed additive MAC, has a capacity region, for any given input discrete signal alphabets, that matches that of a TDMA channel [6]–[10]. When noise is absent or negligible, one may instantiate the noiseless MAC by the use of ZigZag decoding [11]. In this case, users send packets, possibly uncoded. Each packet corresponds to a codeword. We can expect distinct packet transmissions to experience slightly different delays. Such differences will tend to happen naturally in MAC without further coordination, or we may introduce them. When packets are transmitted multiple times with different delays over a noiseless channel, the receiver observes different collision events, leading to different summations of packets in the channel, with different time offsets. Using ZigZag decoding, the receiver can then perform interference cancellation by considering the offsets as providing different equations. An example of ZigZag decoding is shown in Figure 1. This decoding method has been shown to, in effect, resemble orthogonal channels for different users in the noiseless regime [12], [13].

Let us now envisage the issue of noise in the channel. A natural candidate for dealing with noise is the recently proposed Guessing Random Additive Noise Decoder (GRAND) decoder [14]. GRAND based decoders attempt to identify the additive noise that corrupted the channel input, rather than identifying the channel input directly. From most likely to least likely based on a channel model and soft information, if it is available, GRAND schemes sequentially invert putative noise effect sequences from the received signal and query whether the remainder is a member of the codebook. The first time a member of the codebook is identified, GRAND terminates and returns that word as its output. GRAND based decoders are code-agnostic and noise-centric. If noise is additive and GRAND's noise effect query order is consistent with the channel, it provably identified a Maximum Likelihood (ML) decoding.

When paired with Random Linear Code (RLC)s, GRAND has been shown to be capacity achieving in the hard Single-Input Single-Output (SISO) setting [14]. Several GRAND variants for SISO channels have been proposed that use soft information: 1) Symbol Reliability GRAND (SRGRAND), which uses one bit of soft information per channel output [15], [16]; 2) Soft GRAND (SGRAND), a soft detection variant of GRAND that uses (potentially) real-valued soft information per channel symbol, and is a ML decoder in the soft detection SISO scenario [17]; and 3) Ordered Reliability Bits GRAND (ORBGRAND), which serves as a middle ground between the two, as it uses $\lceil \log_2 n \rceil$ soft information bits per channel symbol [18]. Abbas et al. demonstrated hardware architectures for both GRAND [19] and ORBGRAND [20].

In the spirit of GRAND decoders, we wish to guess the effect of the noise added by the channel, and remove that effect to obtain an equivalent noise-free MAC channel. GRAND based decoders exploit the fact that the entropy of the noise is substantially lower than the entropy of the inputs, hence guessing the noise effect is easier than guessing the codeword. Treating signals from the modulated codewords of undecoded users as noise, as is done in the Ahlswede and the Liao frameworks, is undesirable for GRAND, since those signals would generally have high entropy. We instead tailor SGRAND to fit MAC in order to remove the effects of the noise, followed by ZigZag decoding, described earlier. We prove that the suggested algorithm is a Maximum A-Posteriori (MAP) decoder, which is known to be optimal.

II. Model and Background

A. Definitions and Notation

Let $x, \vec{x}, X, \vec{X}$ denote a scalar, vector, matrix, and a random vector respectively. All vectors are row vectors unless stated otherwise, and x_i denotes the i-th element of $\vec{x}$. Composition of functions is denoted by $\circ$, i.e. $f \circ g(x) = f(g(x))$. Non-subscript/superscript i denotes $\sqrt{-1}$.

B. System Model

We consider a standard MAC model where s transmitters attempt to send modulated codewords of length n_i, i.e. the i-th transmitter transmits $\vec{x}_i \in \mathbb{C}^{n_i}$, to be sent over an analog channel. To focus on the novel aspects of the work, we assume that $n_i = n$ for all i, but this assumption can be dropped in practice. A codebook is the set of all possible codewords, and is denoted by $\mathcal{X}_i \subseteq \mathbb{C}^n$. The j-th element of $\vec{x}_i$ may take values from $\mathcal{X}_i^j$, where $\mathcal{X}_i \subseteq \mathcal{X}_i^1 \times \ldots \times \mathcal{X}_i^n$. All messages are added together in the channel into $\vec{x} = \sum_{i=1}^s \vec{x}_i$. The codebook of the sum is denoted by $\mathcal{X} \subseteq \mathbb{C}^n$. Similarly, the j-th element of $\mathcal{X}$ is an element of $\mathcal{X}^j$, where $\mathcal{X} \subseteq \mathcal{X}^1 \times \ldots \mathcal{X}^n$. Note that $\mathcal{X}^j$ is different from the individual codebooks $\mathcal{X}_i^j$, and specifically, of a higher constellation. For example, when $s = 2$ transmitters use a Binary Phase Shift Keying (BPSK) modulation, with a phase shift of $\pi/2$ radians, $\vec{x}$ has 4 constellation points, which constitute a Quadrature Phase Shift Keying (QPSK) modulation [21], as shown in Figure 1. In addition, the memoryless channel adds random additive noise $\vec{Z}$, to the channel input, resulting in channel output $\vec{Y} = \vec{x} + \vec{Z}$. The noise is independent of the transmitted information.

A soft decoder is a mapping $\mathbb{C}^n \rightarrow \mathcal{X}_1 \times \ldots \times \mathcal{X}_s$. A soft detection MAP decoder outputs $\underset{\vec{x}_1,\ldots,\vec{x}_s \in \mathcal{X}_1 \times \ldots \times \mathcal{X}_s}{\arg\max} p\left(\vec{x}_1, \ldots, \vec{x}_s \mid \vec{y}^{(1)}, \ldots \vec{y}^{(s)}\right)$, where p denotes the likelihood function. A demodulator is a function $\theta : \mathbb{C}^n \rightarrow \mathcal{X}^1 \times \ldots \times \mathcal{X}^n$ that returns the MAP modulated version of the channel input, namely $\underset{x_1 \in \mathcal{X}^1,\ldots,x_n \in \mathcal{X}^n}{\arg\max} \prod_{i=1}^n p(x_i \mid y_i)$, where the maximum is taken on each element individually, and not on $\mathcal{X}$. Note that a ML demodulation does not suffice, as $\vec{x}$ may not to be uniformly distributed, even if $\{\vec{x}_i\}_i$ are, depending on the modulation.

C. Transmission delays

In ZigZag decoding, transmitters send the same codewords, or, equivalently, packets s times. As different users are not coordinated and transmit codewords at uncoordinated times, there is a delay between transmission of different messages. We denote the delay of the i-th message in the j-th transmission, relative to the first transmission, by $d_i^{(j)}$. A negative delay means that the first codeword is delayed with respect to the j-th message. A *delay* operator, denoted by $D(\vec{x}_i, d_i)$, is an operator that delays $\vec{x}_i$ by $d_i^{(j)}$ units, and pads it with zeros so all messages in the same transmission are of the same length. By definition, $d_1^{(j)} = 0$ for all j. The first $\vec{x}_i$ input of $D\left(\vec{x}_i, d_i^{(j)}\right)$ may be omitted, if it is clear from context which codeword is delayed. Elements of the j-th transmission are denoted by a (j) superscript. The channel input of the j-th transmission is denoted by $\vec{x}^{(j)} = \sum_{i=1}^s D\left(\vec{x}_i, d_i^{(j)}\right)$. The channel output in the j-th transmission is $\vec{Y}^{(j)} = \vec{x}^{(j)} + \vec{Z}^{(j)}$, where $\left\{\vec{Z}^{(j)}\right\}_j$ are mutually independent, as the channel is memoryless. Note that different elements of $\vec{x}^{(j)}$ may have different modulations, as a result of edge-effects. This phenomenon is illustrated in Figure 1, where both transmitters use BPSK modulation with a phase shift of $\pi/2$ radians. The second message is delayed by one/two time units in the first/second transmission, respectively. Observing $\vec{x}^{(1)}$, we notice that $x_{1,1} \in \{i, -i\}$, while $x_{1,2} + x_{2,1} \in \{1+i, 1-i, -1+i, -1-i\}$. As common in MAC literature, we assume that the channel responses, including delays, are perfectly known to the receiver. Kazemi et al. [22] have studied the effect on ZigZag decoding of imperfectly known channel responses.

D. SGRAND

In the soft detection setting, SGRAND provides a means to generate error effect sequences in order, on the fly, from most likely to least likely for any arbitrary memoryless channel [17]. In Section III-B, we modify SGRAND to rank order error sequences on the fly for MAC. To check for codebook membership, we use ZigZag decoding to cancel interference, as we explain in Section III-A. We then check whether each of the individual putative codewords are a members of their codebook. An efficient way of doing so for linear codes would be to test whether the syndrome of said remainder is zero [23, Chapter 2].

III. MAC SGRAND

A. Noise free transmission

In order to better understand the decoding strategy proposed in this paper, we first discuss the case where no noise is added to the channel input, i.e. $\vec{Z}^{(j)} = \vec{0}$, so $\vec{Y}^{(j)} = \vec{x}^{(j)}$. The goal is to recover the transmitted channel inputs $\vec{x}_1, \ldots, \vec{x}_s$. Decodability is not guaranteed in this case: we need to ensure that the mapping from $\vec{x}_1, \ldots, \vec{x}_s$ to $\vec{x}^{(1)}, \ldots, \vec{x}^{(s)}$ is invertible.

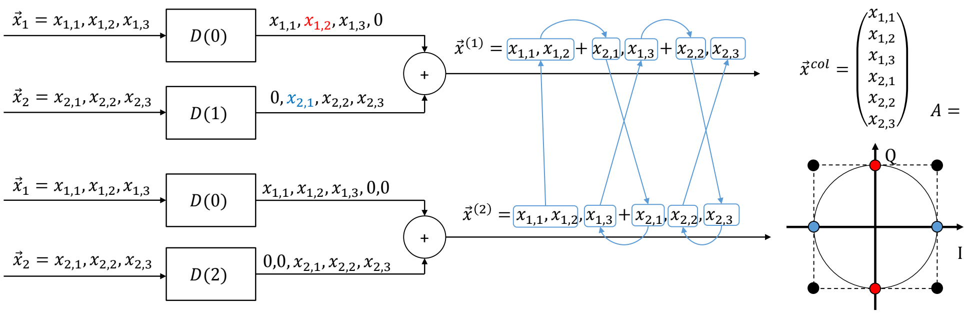

Fig. 1: Multiple Access Channel example. There are $s = 2$ transmitters, that send their messages twice. The delays of the second transmitter in the first/second transmissions are 1 and 2, respectively. The modulated messages are $\vec{x}_1 = x_{1,1}, x_{1,2}, x_{1,3}$ and $\vec{x}_2 = x_{2,1}, x_{2,2}, x_{2,3}$. Both messages are delayed and summed up into $\vec{x}^{(1)} = x_{1,1}, x_{1,2} + x_{2,1}, x_{1,3} + x_{2,2}, x_{2,3}$ and $\vec{x}^{(2)} = x_{1,1}, x_{1,2}, x_{1,3} + x_{2,1}, x_{2,2}, x_{2,3}$. Upon receiving a noiseless version of $\vec{x}^{(1)}, \vec{x}^{(2)}$, the decoder, which knows the delays, can recover $\vec{x}_1, \vec{x}_2$, as suggested by the blue arrows. In the example, the first/second transmitter use BPSK modulation with a phase shift of $\pi/2$ radians, which are shown as the red/blue points, respectively. The effective constellation of the sum is a QPSK modulation, which is illustrated by the black points. The column stack vectors and the participation matrix A are also shown in the figure.

$$\vec{x}^{col} = \begin{pmatrix} x_{1,1} \\ x_{1,2} \\ x_{1,3} \\ x_{2,1} \\ x_{2,2} \\ x_{2,3} \end{pmatrix} \quad A = \begin{pmatrix} 1&0&0&0&0&0 \\ 0&1&0&1&0&0 \\ 0&0&1&0&1&0 \\ 0&0&0&0&0&1 \\ 1&0&0&0&0&0 \\ 0&1&0&0&0&0 \\ 0&0&1&1&0&0 \\ 0&0&0&0&1&0 \\ 0&0&0&0&0&1 \end{pmatrix} \quad \vec{y}^{col} = \begin{pmatrix} x_{1,1} \\ x_{1,2}+x_{2,1} \\ x_{1,3}+x_{2,2} \\ x_{2,3} \\ x_{1,1} \\ x_{1,2} \\ x_{1,3}+x_{2,1} \\ x_{2,2} \\ x_{2,3} \end{pmatrix}$$

For example, consider a case with $s = 2$ transmitters and two transmissions. Suppose in both transmissions, the delay between the two transmitters is the same. In this case, the receiver gets in the second transmission a message that is linearly dependent on the first message, hence it cannot recover $\vec{x}_1, \vec{x}_2$. To define the problem formally, we define $\vec{x}^{col}, \vec{y}^{col}$ a column stack of $\vec{x}_1, \ldots, \vec{x}_s$, and all the transmissions of $\vec{y}$, respectively, i.e. $\vec{x}^{col} = (\vec{x}_1, \ldots, \vec{x}_s)^T, \vec{y}^{col} = \left(\vec{y}^{(1)}, \ldots, \vec{y}^{(s)}\right)^T$. The *col* superscript indicates a column stack vector. The length of $\vec{y}^{col}$, denoted by n', satisfies

$$n' = sn + \sum_{j=1}^{s} \left(\max_{i:d_i^{(j)} \geq 0} d_i^{(j)} - \min_{i:d_i^{(j)} < 0} d_i^{(j)} \right)$$

The codebook spanned by the elements of $\vec{x}^{col}$ is denoted by $\mathcal{X}^{col} = \mathcal{X}_1 \times \ldots \times \mathcal{X}_s$. We define as $\mathcal{Y}^{col}$ as the noiseless codebook of $\vec{y}^{col}$, and for each element of $\vec{y}^{col}$, we define $\mathcal{Y}^{col,i}, i \in \{1, \ldots, n'\}$ as a set of all possible options for noiseless version of y_i^{col}. Similarly, $\mathcal{Y}^{col} \subseteq \mathcal{Y}^{col,1} \times \ldots \times \mathcal{Y}^{col,n'}$. Note that $\mathcal{Y}^{col,i} \neq \mathcal{Y}^{col,j}$ may occur, as a result of edge effects. Let A be the *participation matrix*, defined as follows: $A_{i,j} = 1$ if element j of $\vec{x}^{col}$ participates in the sum of the i-th element of $\vec{y}^{col}$, and 0 otherwise. Figure 1 demonstrates these definitions. When applying the definitions of a demodulator from Section II-B, we get $\theta\left(\vec{y}^{col}\right) = \arg\max_{(A\cdot\vec{x}^{col})_1 \in \mathcal{Y}^{col,1}, \ldots, (A\cdot\vec{x}^{col})_{n'} \in \mathcal{Y}^{col,n'}} \prod_{i=1}^{n'} p\left(\left(A \cdot \vec{x}^{col}\right)_i \mid y_i^{col}\right)$

The ZigZag decoder attempts to recover $\vec{x}^{col}$ from $A, \vec{y}^{col}$. In the noiseless case, this is equivalent to solving the following system of linear equations $\vec{y}^{col} = A \cdot \vec{x}^{col}$, which can be solved efficiently e.g. with Gaussian Elimination [24]. In the noise-free case We say that the code is *decodable* if such solution exists. By construction of $\vec{x}^{col}$, recovering $\vec{x}_1, \ldots, \vec{x}_s$ is equivalent to recovering $\vec{x}^{col}$. In MAC SGRAND, decodability serves as a part of the codebook membership check.

B. MAC SGRAND - noisy channel

We now modify SGRAND to fit an additive MAC. The channel output in this case is given by $\vec{Y}^{col} = A \cdot \vec{x}^{col} + \vec{Z}^{col}$, where as explained in Section II, channel inputs are not necessarily distributed uniformly. We first prove that determining the MAP noise sequence $\vec{z}^{col}$ is equivalent to finding the MAP channel input $\vec{x}^{col}$, then we present MAC SGRAND that finds the MAP error sequence, hence it is an optimal decoder.

Theorem III.1. *Let $\vec{x}^{col} \in \mathcal{X}^{col}$ be a transmitted message on an additive memoryless MAC, such that $\vec{Y} = A \cdot \vec{x}^{col} + \vec{Z}^{col}$, and $A, \vec{x}^{col}$ are decodable (as defined in Section III-A). Then finding $\arg\max_{\vec{x}^{col} \in \mathcal{X}^{col}} p\left(\vec{x}^{col} \mid \vec{y}^{col}\right)$ is equivalent to finding $\arg\max_{\vec{z}^{col}:\vec{y}^{col}-\vec{z}^{col} \in \mathcal{Y}^{col}} p\left(\vec{z}^{col} \mid \vec{y}^{col}\right)$.*

Proof. Note that

$$\begin{aligned} &\arg\max_{\vec{x}^{col} \in \mathcal{X}^{col}} p\left(\vec{x}^{col} \mid \vec{y}^{col}\right) \\ &= \arg\max_{\vec{x}^{col} \in \mathcal{X}^{col}} p\left(\vec{y}^{col} \mid \vec{x}^{col}\right) p\left(\vec{x}^{col}\right) && (1) \\ &= \arg\max_{\vec{x}^{col} \in \mathcal{X}^{col}} p\left(\vec{z}^{col}\right) p\left(\vec{x}^{col}\right) && (2) \\ &= \arg\max_{\vec{x}^{col} \in \mathcal{X}^{col}} p\left(\vec{z}^{col}\right) p\left(A \cdot \vec{x}^{col}\right) && (3) \\ &= \arg\max_{\vec{x}^{col} \in \mathcal{X}^{col}} p\left(\vec{z}^{col}\right) p\left(A \cdot \vec{x}^{col} + \vec{z}^{col} \mid \vec{z}^{col}\right) && (4) \\ &= \arg\max_{\vec{z}^{col}:\vec{y}^{col}-\vec{z}^{col} \in \mathcal{Y}^{col}} p\left(\vec{z}^{col} \mid \vec{y}^{col}\right) && (5) \end{aligned}$$

where (1) and (5) follow from Bayes rule, (2) and (4) follow since $\vec{X}^{col}$ and $\vec{Z}^{col}$ are independent, and (3) follows since the code is decodable. In (5), we also use the fact that $\vec{x}^{col} \in \mathcal{X}^{col}$ iff $\vec{y}^{col} - \vec{z}^{col} \in \mathcal{Y}^{col}$, as the code is decodable. □

To implement any GRAND algorithm, we need to check for codebook membership and rank order all noise sequences from most likely to least likely. For MAC SGRAND, we use ZigZag decoding, as explained in Section III-A, to cancel interference, and check whether the interference-free vector $\vec{x}^{col}$ is in the code $\mathcal{X}^{col}$, for example by checking the syndromes for linear codes [23, Chapter 2]. We now present a method that rank orders all noise sequences from most likely to least likely.

We first define for each symbol i of $\vec{y}^{col}$ the Ordered Symbol Indices (OSI) as a vector that ranks order, from most likely to least likely, the error element that affected the i-th demodulated symbol. Formally speaking, $\vec{s}^i \in \mathbb{C}^{\left|\mathcal{Y}^{col,i}\right|}$ is an OSI vector, where the j-th element satisfies $\theta\left(y_i^{col}\right) - s_j^i \in \mathcal{Y}^{col,i}$. A vector $\vec{s}^i$ is an OSI of the i-th symbol if it satisfies the following: 1) $s_1^i = 0$; 2) $p\left(\theta\left(y_i^{col}\right) - s_j^i \mid y_i^{col}\right) \geq p\left(\theta\left(y_i^{col}\right) - s_l^i \mid y_i^{col}\right) \forall j < l$; and 3) $\vec{s}^i$ does not contain duplicate elements. The OSI of each symbol is a function of y_i^{col}, the modulation and the channel statistics, but this dependency is left implicit for notational simplicity. An OSI is unique almost surely for continuous memoryless channels. To demonstrate the definition of OSI, let us consider the example of Figure 1, assuming the transmission power is 1 and the channel is an Additive White Gaussian Noise (AWGN) channel. Suppose that $y_i^{col} = \sqrt{3/2} + \sqrt{1/2}i$, and y_i^{col} does not suffer from an edge effect, i.e. $\mathcal{Y}^{col,i} = \{1+i, 1-i, -1-i, -1+i\}$. Then $\theta\left(y_i^{col}\right) = 1+i$, and $\vec{s}^i = (0, 2i, 2, 2+2i)$, so $\theta\left(y_i^{col}\right) - \vec{s}^i = (1+i, 1-i, -1+i, -1-i)$, which lists all possible channel inputs, from most likely to least likely. We define the *previous* function, denoted by $\phi(\cdot)$, as a function that returns the previous element in the OSI vector, namely:

$$\phi\left(s_j^i\right) = \begin{cases} s_{j-1}^i, & \text{if } j > 1 \\ \perp, & j = 1 \end{cases}$$

and $\phi^{-1}(\cdot)$ as the inverse of $\phi(\cdot)$, where $\phi^{-1}\left(s^i_{\left|\mathcal{Y}_i^{col}\right|}\right) = \perp$. For example, $\phi\left(\theta\left(y_i^{col}\right)\right) = \perp$, as $\theta\left(y_i^{col}\right)$ is the MAP demodulated symbol.

Using the definitions of OSI, $\phi(\cdot)$ and $\phi^{-1}(\cdot)$, pseudocode for MAC SGRAND with ABandonment (SGRANDAB) is given in Algorithm 1. We prove that Algorithm 1 is a MAP decoder for a memoryless MAC.

C. Main Theorem

Theorem III.2. *Algorithm 1 satisfies the following, for an additive memoryless MAC:*

1) ***Correctness:*** *Error vectors are queried in non increasing order of likelihood, hence a returned codeword is a MAP codeword.*
2) ***Progress:*** *Each error vector is queried at most once.*

Proof. The proof carries similar flavor of the SISO SGRAND proof [17]. Let the *parent* of $\vec{z}^{col} \neq \vec{0}^T$, denoted by $\pi\left(\vec{z}^{col}\right)$, be defined as the follows:

$$\left(\pi\left(\vec{z}^{col}\right)\right)_j = \begin{cases} \phi\left(z_j^{col}\right), & \text{if } j = j_*, \\ z_j^{col}, & \textit{otherwise} \end{cases}$$

where j_* is defined as in Algorithm 1. A vector $\vec{z}^{col}$ is called a *child* of $\pi\left(\vec{z}^{col}\right)$. Note that a *child* is not unique, while a parent is. Moreover, while every non-zero error vector has a unique parent, not all vectors have children: an error vector where $j_* = n'$ and $\phi^{-1}\left(z_{j_*}\right) = \perp$ has no children. Any error vector has up to n' children. Observe that Algorithm 1 adds to $\mathcal{S}$ all the children of $\vec{z}$, after removing it from $\mathcal{S}$. To prove the theorem, we first prove the following lemma:

Algorithm 1: MAC SGRANDAB

Input: $\vec{y}^{col}, A, b, \mathcal{X}^{col}$ ▷ max #queries b, SGRAND: $b = \infty$
Output: $\vec{x}^{col}$
$g \leftarrow 0$ ▷ g counts queries performed
$\mathcal{S} \leftarrow \left\{\vec{0}^T\right\}$ ▷ $\mathcal{S}$ contains candidate error vectors
while $g \leq b$ **do**
  $\vec{z}^{col} \leftarrow \arg\max_{\vec{v}^{col} \in \mathcal{S}} p\left(\theta\left(\vec{y}^{col}\right) - \vec{v}^{col} \mid \vec{y}^{col}\right)$
  $\hat{\vec{y}}^{col} \leftarrow \theta\left(\vec{y}^{col}\right) - \vec{z}^{col}$
  $\mathcal{S} = \mathcal{S} \setminus \left\{\vec{z}^{col}\right\}$
  $g \leftarrow g + 1$
  $\vec{x}^{col} \leftarrow \text{ZigZagDecoder}\left(\hat{\vec{y}}^{col}, A, \mathcal{X}^{col}\right)$
  if $\vec{x}^{col} \neq \perp$ **then**
    return ▷ MAP codeword found
  else ▷ Update $\mathcal{S}$ for the next query
    if $\vec{z}^{col} = \vec{0}^T$ **then**
      $j_* \leftarrow 0$
    else
      $j_* \leftarrow \max\{j : z_j \neq 0\}$ ▷ $j_* > 0$
    end if
    $j \leftarrow j_*$
    while $j \leq n'$ **do**
      if $j > 0$ **then**
        if $\phi^{-1}\left(z_j^{col}\right) \neq \perp$ **then**
          $z_j^{col} = \phi^{-1}\left(z_j^{col}\right)$
          $\mathcal{S} = \mathcal{S} \cup \left\{\vec{z}^{col}\right\}$
          $z_j^{col} = \phi\left(z_j^{col}\right)$
        end if
      end if
      $j \leftarrow j + 1$
    end while
  end if
end while
$\vec{x}^{col} \leftarrow \perp$ ▷ Codeword not found in b queries; failure
return
procedure ZIGZAGDECODER($\hat{\vec{y}}^{col}, A, \mathcal{X}^{col}$)
  if $\exists \vec{x}^{col}$ s.t. $A \cdot \vec{x}^{col} = \hat{\vec{y}}^{col}$ & $\vec{x}^{col} \in \mathcal{X}^{col}$ **then**
    return $\vec{x}^{col}$
  else
    return $\perp$
  end if
end procedure

Lemma III.3. *Let $\vec{z}^{col} \neq \vec{0}^T$ be an error vector. Define $m \in \{1, \ldots, n'\}$ to be the codebook with most elements, i.e. $m = \arg\max_{i \in \{1,\ldots,n'\}} \left|\mathcal{Y}_i^{col}\right|$. The following hold for a memoryless MAC:*

1) $\pi\left(\vec{z}^{col}\right) \neq \vec{z}^{col}$
2) $p\left(\theta\left(\vec{y}^{col}\right) - \vec{z}^{col} \mid \vec{y}^{col}\right) \leq p\left(\theta\left(\vec{y}^{col}\right) - \pi\left(\vec{z}^{col}\right) \mid \vec{y}^{col}\right)$
3) $\pi \circ \ldots \circ \pi\left(\vec{z}^{col}\right) = \vec{0}^T$ *after at most* $\left|\mathcal{Y}_m^{col}\right| \cdot j_*$ *compositions.*

Proof. Let $\vec{z}^{col} \neq \vec{0}^T$, and j_* (with respect to $\vec{z}^{col}$) be defined as in Algorithm 1. Note that $j_* > 0$. We prove each of the

properties:
1) Let $\vec{z}^{col} \neq \vec{0}^T$. By definition $\pi\left(\vec{z}^{col}\right)_{j_*} = \phi\left(z^{col}_{j_*}\right)$, and $z^{col}_{j_*} \neq 0$. By property 3 of OSI, $\phi\left(z^{col}_{j_*}\right) \neq z^{col}_{j_*}$.
2) Let $q_j(w) = p\left(\theta\left(\vec{y}^{col}\right)_j - w \mid y^{col}_j\right)$ and $q(\vec{w}) = \prod_{j=1}^n q_j(w_j)$. The claim is established if exists j such that $q\left(\vec{z}^{col}\right) = 0$. Otherwise, assume $q\left(\vec{z}^{col}\right) > 0$. Then $q\left(\vec{z}^{col}\right) = q_{j_*}\left(z^{col}_{j_*}\right)\prod_{j\neq j_*} q_j\left(z^{col}_j\right)$, and $q\left(\pi\left(\vec{z}^{col}\right)\right) = q_{j_*}\left(\phi\left(z^{col}_{j_*}\right)\right)\prod_{j\neq j_*} q_j\left(z^{col}_j\right)$. Observe that $q\left(\pi\left(\vec{z}^{col}\right)\right)/q\left(\vec{z}^{col}\right) = q_{j_*}\left(\phi\left(z^{col}_{j_*}\right)\right)/q_{j_*}\left(z^{col}_{j_*}\right) \geq 1$, by the definition of OSI, which completes the proof.
3) Let $\vec{z}^{col} \neq \vec{0}^T$, and j_* be defined as in Algorithm 1. Let r_{j_*} denote the position of $z^{col}_{j_*}$ at the OSI. After applying π, $r_{j_*} \leq \left|\mathcal{Y}^{col}_m\right|$ times, j_* decreases by at least 1. By iterating this argument at most j_* times, we get the zero vector. □

We now prove Theorem III.2. Note that in order to prove Property 2, it is enough to show that an error vector can be added to $\mathcal{S}$ at most once. We prove the theorem by induction on the number of queries performed g, which is evaluated at the end of the while loop. To prove the base case of $g = 1$, we notice that $\mathcal{S}$ is initialized to contain only $\vec{0}^T$, which is the most likely error vector, so Property 1 is satisfied. We also notice that in the while loop, $\vec{0}^T$ is removed from $\mathcal{S}$, and all of its children, which are different than $\vec{0}^T$ (by Lemma III.3.1) and one from another, so Property 2 is satisfied. Assume now that the theorem holds after g queries, and we establish that it holds after $g+1$ queries. Suppose the algorithm queries the error vector $\vec{z}^{col}$ in query number g. To prove Property 2, suppose by contradiction that $\vec{v}^{col}$, an error vector that was previously added to $\mathcal{S}$, is added to $\mathcal{S}$ for the second time. An error vector is added to $\mathcal{S}$ only when its unique parent (by Lemma III.3.1) is queried, so $\vec{z}^{col} = \pi\left(\vec{v}^{col}\right)$. As a result, we conclude that $\vec{z}^{col}$ has been queried more than once within g queries, which contradicts the induction assumption. To prove Property 1, suppose by contradiction that it is not satisfied, and there exists another error vector $\vec{v}^{col}$ that has not been queried before and is more likely than $\vec{z}^{col}$, namely $p\left(\theta\left(\vec{y}^{col}\right) - \vec{z}^{col} \mid \vec{y}^{col}\right) < p\left(\theta\left(\vec{y}^{col}\right) - \vec{v}^{col} \mid \vec{y}^{col}\right)$. If there exists more than one such vector, let us pick the most likely one. Notice that $\vec{v}^{col} \notin \mathcal{S}$ must hold for each of the first $g+1$ queries. Otherwise, Algorithm 1 would pick $\vec{v}^{col}$ as its queried error vector. If $\pi\left(\vec{v}^{col}\right) \in \mathcal{S}$ ever held, then $\vec{v}^{col}$ would be contained at $\mathcal{S}$ at some point, since Algorithm 1 adds the children of the queried noise vector to $\mathcal{S}$, which contradicts our assumption. Therefore, assume that $\pi\left(\vec{v}^{col}\right) \notin \mathcal{S}$ at any point. By Lemma III.3.2, we know that $p\left(\theta\left(\vec{y}^{col}\right) - \pi\left(\vec{v}^{col}\right) \mid \vec{y}^{col}\right) \geq p\left(\theta\left(\vec{y}^{col}\right) - \vec{v}^{col} \mid \vec{y}^{col}\right)$. If $p\left(\theta\left(\vec{y}^{col}\right) - \pi\left(\vec{v}^{col}\right) \mid \vec{y}^{col}\right) > p\left(\theta\left(\vec{y}^{col}\right) - \vec{v}^{col} \mid \vec{y}^{col}\right)$, we contradict the assumption that $\vec{v}^{col}$ is the most likely noise sequence that should have been queried in query number $g+1$. Otherwise, assume that $p\left(\theta\left(\vec{y}^{col}\right) - \pi\left(\vec{v}^{col}\right) \mid \vec{y}^{col}\right) = p\left(\theta\left(\vec{y}^{col}\right) - \vec{v}^{col} \mid \vec{y}^{col}\right)$. By repeating this argument at most $\left|\mathcal{X}^{col}_m\right| \cdot j_*$ times, we conclude that the zero vector was never added $\mathcal{S}$ (by Lemma III.3.3), which is false, since $\mathcal{S}$ is initialized to contain only the zero vector. □

D. Complexity

At each iteration of the algorithm, we remove the most likely vector from $\mathcal{S}$, perform one membership test, and add at most n' new vectors to $\mathcal{S}$, hence $|\mathcal{S}| = \mathcal{O}(n'g)$ after g queries. An efficient way of implementing $\mathcal{S}$ for these operations is via a Max-Heap [25]. The complexity of the membership test, denoted by $f\left(n', s, \mathcal{X}^{col}\right)$, is twofold. At first, Gaussian elimination [24] has to be performed, and also a codebook membership test. For linear codes, syndrome check is a simple mechanism of checking for codebook membership [23], which boils down to matrix multiplication, whose complexity depends on implementation, e.g. [26], [27]. When $\mathcal{S}$ is implemented as a Max-Heap, the complexity after g queries is $\mathcal{O}\left(n'gf\left(n', s, \mathcal{X}^{col}\right)\log(n'g)\right)$. Hence, the worst-case complexity is $\mathcal{O}\left(n'bf\left(n', s, \mathcal{X}^{col}\right)\log(n'b)\right)$, where b is the querying thershold. It is worth noting that in practice, we expect the average case to be significantly lower, as we can anticipate that the total number of queries until a codeword is found is significantly lower than b. Also worth mentioning is the fact that as Signal to Noise Ratio (SNR) improves, the expected number of queries until a codeword is found reduces, as is the case with all GRAND-based decoders. MAC SGRAND is therefore suitable for high-rate short-length codes in the high SNR regime.

IV. Conclusion and Discussion

In this paper, we presented MAC SGRAND, a MAP algorithm for noisy MAC channels, which is known to be optimally accurate. It requires no coordination between users. This algorithm opens a new route for handling MAC: one that addresses interference cancellation and noise separately, where a dedicated algorithm that specializes in each of the problems is applied (SGRAND for the noise, ZigZag decoding for interference cancellation), without user coordination.

For future work, we plan to show a simple capacity achieving scheme which involves MAC SGRAND. In particular, non capacity-achieving approaches for transmitting over MAC are based on ALOHA and related protocols [28], where transmitters send their packets sporadically, without any coordination Future directions would consider an ALOHA based protocol, where users transmit packets whenever they are available. When one user is idle, a single user may transmit without interference. Unlike traditional ALOHA relying on backoff, a MAC SGRAND based scheme may request that colliding users collide anew.

References

[1] T. M. Cover, *Elements of information theory*. John Wiley & Sons, 1999.
[2] R. Ahlswede, "Multi-way communication channels," in *Second International Symposium on Information Theory: Tsahkadsor, Armenia, USSR, Sept. 2-8, 1971*, 1973.
[3] H. H.-J. Liao, "Multiple access channels." HAWAII UNIV HONOLULU, Tech. Rep., 1972.
[4] R. Gallager, "A perspective on multiaccess channels," *IEEE Transactions on Information Theory*, vol. 31, no. 2, pp. 124–142, 1985.
[5] A. J. Grant, B. Rimoldi, R. L. Urbanke, and P. A. Whiting, "Rate-splitting multiple access for discrete memoryless channels," *IEEE Transactions on Information Theory*, vol. 47, no. 3, pp. 873–890, 2001.

[6] S. Ray, M. Medard, and J. Abounadi, “Random coding in noise-free multiple access networks over finite fields,” in *IEEE Global Telecommunications Conference (IEEE Cat. No.03CH37489)*, vol. 4, 2003, pp. 1898–1902 vol.4.
[7] 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.
[8] B. L. Hughes and A. B. Cooper, “Nearly optimal multiuser codes for the binary adder channel,” *IEEE Transactions on Information Theory*, vol. 42, no. 2, pp. 387–398, 1996.
[9] G. H. Khachatrian and S. S. Martirossian, “Code construction for the t-user noiseless adder channel,” *IEEE Transactions on Information Theory*, vol. 44, no. 5, pp. 1953–1957, 1998.
[10] Shih-Chun Chang and E. Weldon, “Coding for t-user multiple-access channels,” *IEEE Transactions on Information Theory*, vol. 25, no. 6, pp. 684–691, 1979.
[11] S. Gollakota and D. Katabi, “Zigzag decoding: combating hidden terminals in wireless networks,” in *Proceedings of the ACM SIGCOMM 2008 conference on Data communication*, 2008, pp. 159–170.
[12] A. ParandehGheibi, J. K. Sundararajan, and M. Médard, “Collision helps-algebraic collision recovery for wireless erasure networks,” in *2010 Third IEEE International Workshop on Wireless Network Coding*. IEEE, 2010, pp. 1–6.
[13] ——, “Acknowledgement design for collision-recovery-enabled wireless erasure networks,” in *48th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2010, pp. 435–442.
[14] K. R. Duffy, J. Li, and M. Médard, “Capacity-achieving Guessing Random Additive Noise Decoding,” *IEEE Transactions on Information Theory*, vol. 65, no. 7, pp. 4023–4040, 2019.
[15] K. R. Duffy and M. Médard, “Guessing random additive noise decoding with soft detection symbol reliability information,” in *International Symposium on Information Theory*, 2019, pp. 480–484.
[16] K. R. Duffy, A. Solomon, K. M. Konwar, and M. Médard, “5G NR CA-Polar maximum likelihood decoding by GRAND,” in *2020 54th Annual Conference on Information Sciences and Systems (CISS)*. IEEE, 2020, pp. 1–5.
[17] A. Solomon, K. R. Duffy, and M. Médard, “Soft Maximum Likelihood Decoding using GRAND,” in *IEEE International Conference on Communications*, 2020.
[18] K. R. Duffy, “Ordered reliability bits guessing random additive noise decoding,” in *IEEE ICASSP*, 2021.
[19] S. M. Abbas, T. Tonnellier, F. Ercan, and W. J. Gross, “High-Throughput VLSI Architecture for GRAND,” in *IEEE International Workshop on Signal Processing Systems (SiPS)*. IEEE, 2020, pp. 1–6.
[20] S. M. Abbas, T. Tonnellier, F. Ercan, M. Jalaleddine, and W. J. Gross, “High-throughput VLSI architecture for soft-decision decoding with ORBGRAND,” 2021, accepted to International Conference on Acoustics, Speech and Signal Processingng Systems (ICASSP).
[21] J. Proakis, “Digital communications,” *Me Graw-Hill*, pp. 777–778, 2001.
[22] M. Kazemi, T. M. Duman, and M. Médard, “Double-zipper: Multiple access with zigzag decoding,” in *IEEE International Conference on Communications*. IEEE, 2020, pp. 1–6.
[23] R. Roth, *Introduction to coding theory*. CUP, 2006.
[24] K. E. Atkinson, *An introduction to numerical analysis*. John wiley & sons, 2008.
[25] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithms*. MIT press, 2009.
[26] P. Bürgisser, M. Clausen, and M. A. Shokrollahi, *Algebraic complexity theory*. Springer Science & Business Media, 2013, vol. 315.
[27] M. Frigo, C. E. Leiserson, H. Prokop, and S. Ramachandran, “Cache-oblivious algorithms,” in *40th Annual Symposium on Foundations of Computer Science*. IEEE, 1999, pp. 285–297.
[28] N. Abramson, “The ALOHA system: another alternative for computer communications,” in *Proceedings of the Fall Joint Computer Conference*, 1970, pp. 281–285.