Enhanced GCD through ORBGRAND-AI: Exploiting Partial and Total Correlation in Noise

Jiewei Feng, Ken R. Duffy and Muriel Médard

***Abstract*—There have been significant advances in recent years in the development of forward error correction decoders that can decode codes of any structure, including practical realizations in synthesized circuits and taped out chips. While essentially all soft-decision decoders assume that bits have been impacted independently on the channel, for one of these new approaches it has been established that channel dependencies can be exploited to achieve superior decoding accuracy, resulting in Ordered Reliability Bits Guessing Random Additive Noise Decoding Approximate Independence (ORBGRAND-AI). Building on that capability, here we consider the integration of ORBGRAND-AI as a pattern generator for Guessing Codeword Decoding (GCD). We first establish that a direct approach delivers mildly degraded block error rate (BLER) but with reduced number of queried patterns when compared to ORBGRAND-AI. We then show that with a more nuanced approach it is possible to leverage total correlation to deliver an additional BLER improvement of $\sim$0.75 dB while retaining reduced query numbers.**

***Index Terms*—Soft input, correlation, GRAND, GCD**

I. Introduction

CODE-AGNOSTIC decoders have been the subject of considerable research attention, including Guessing Random Additive Noise Decoding (GRAND) [1]–[4], Guessing Codeword Decoding (GCD) [5]–[8], and Ordered Reliability Direct Error Pattern Testing (ORDEPT) [9], [10]. These decoders offer the flexibility of decoding different codes with a single algorithmic architecture. The practical promise of these approaches has been established primarily for GRAND variants, where its inherent parallelizability and the simplicity of its essential operations have resulted in synthesised circuits, e.g. [11]–[16], and taped out chips, e.g. [17]–[19], for both hard and soft detection systems. Moreover, it has been shown that soft-input versions of both GRAND and GCD can readily produce accurate soft output, enabling iterative decoding of long, powerful concatenated codes [20], [21].

One line of GRAND's development focuses on correlated channels, where variants have been introduced that exploit statistical knowledge of correlation to improve decoding, including GRAND Markov Order (GRAND-MO) [22] for hard input and Ordered Reliability Bits GRAND Approximate Independence (ORBGRAND-AI) [23]–[26] for soft input. The latter development leverages ideas from thermodynamic probability theory, e.g. [27], to achieve multiple dB decoding improvement in channels with memory. ORBGRAND-AI's practicality has been demonstrated in a taped out chip [28].

This work was supported by the Defense Advanced Research Projects Agency (DARPA) under Grant HR00112120008

J. Feng and K. R. Duffy, Northeastern University, (e-mails: {feng.ji,k.duffy}@northeastern.edu).

M. Médard, Massachusetts Institute of Technology, (e-mail: medard@mit.edu).

A natural question arises as to whether the principles underlying ORBGRAND-AI can be adapted for use with GCD. Here we first demonstrate that a direct combination of ORBGRAND-AI with GCD results in mildly degraded BLER in return for much fewer queries. With a more nuanced understanding of these ideas, we establish that this combination can be modified to leverage total channel state information (CSI), resulting in up to an additional $\sim$0.75 dB improvement in BLER over ORBGRAND-AI while retaining some query reduction.

The remainder of this paper is organized as follows. Section II introduces preliminaries, including summaries of ORBGRAND-AI and GCD. Section III describes the two proposed decoding algorithms whose performance is shown in Section IV. Conclusions can be found in Section V.

II. Preliminaries

While GRAND decoders can decode non-linear codes, e.g. [29], here we restrict our attention to binary linear codes as usually considered with GCD. A k-bit binary message $U^k = (U_1, \ldots, U_k) \in \mathbb{F}_2^k$ is encoded with the $k \times n$ binary generator matrix G resulting in an n-bit binary sequence $X^n = U^k G$. The collection of all 2^k possible codewords, i.e. the codebook, is denoted by the set C. Let H denote the codebook's parity check matrix. A sequence X^n is a codeword if and only if $X^n H^T = 0^{n-k}$, the zero vector of length $n-k$. For a set $\mathcal{B} \subseteq \{1, \ldots, n\}$, we use $H_{:,\mathcal{B}}$ to denote the sub-matrix of H containing the columns with indices in $\mathcal{B}$ and $\mathtt{x}_{\mathcal{B}}$ to denote the subsequence of $\mathtt{x}^n$ containing the entries with indices in $\mathcal{B}$. The codeword X^n is modulated to $\mathcal{X}^{n_s} = \mathcal{M}(X^n)$, where n_s is the number of symbols needed to represent n bits and $\mathcal{M}$ represents the modulation function. $\mathcal{X}^{n_s}$ is transmitted through a noisy channel and, after equalization, the channel output is $\mathcal{Y}^{n_s} = \mathcal{X}^{n_s} + \mathcal{N}^{n_s}$ where $\mathcal{N}^{n_s}$ represents the continuous noise effects impacting the codeword. An maxmimum likelihood (ML) decoder returns the codeword

$$\arg\max_{\mathtt{x}^n \in C} f_{\mathcal{Y}^{n_s}|X^n}(y^{n_s}|\mathtt{x}^n)$$

where f is the probability density function that characterizes the channel and can be calculated using CSI. In this letter, we develop enhanced decoders that integrate ORBGRAND-AI with GCD to effectively leverage CSI and exploit correlated channel effects, achieving reduced query numbers and superior decoding performance.

A. ORBGRAND-AI

ORBGRAND-AI [23], [26] originates from symbol-level ORBGRAND [30], which performs soft decoding at the

symbol level without bit demapping. ORBGRAND-AI further leverages correlated channel effects to achieve enhanced decoding performance. The key idea underlying ORBGRAND-AI can be understood as follows. The sequence $\mathcal{Y}^{n_s}$ is split into n_b blocks with each block containing b symbols, where we assume b and $n_b = n_s/b$ are integers for notational simplicity:

$$\begin{aligned}\underline{\mathcal{Y}}^{n_b} &= (\underline{\mathcal{Y}}_1, \ldots, \underline{\mathcal{Y}}_{n_b}) \\ &= \left(\mathcal{Y}_1, \ldots, \mathcal{Y}_b, |\mathcal{Y}_{b+1}, \ldots, \mathcal{Y}_{2b}, | \ldots, |\mathcal{Y}_{n_s-b+1}, \ldots, \mathcal{Y}_{n_s}\right).\end{aligned}$$

Similarly, every b symbols in $\mathcal{X}^{n_s}$ are combined into one block, resulting in $\underline{\mathcal{X}}^{n_b} = (\underline{\mathcal{X}}_1, \ldots, \underline{\mathcal{X}}_{n_b})$. Each block can also be seen as a higher-level symbol that contains multiple symbols. We denote $\underline{x}^{n_b} = \mathtt{x}^n$ if the two sequences represent the same n-bit binary sequence. Each symbol is treated separately, and CSI is used to evaluate the PDF $f_{\underline{\mathcal{Y}}_i|\underline{\mathcal{X}}_i}(\underline{y}_i|\underline{x}_i)$, which is denoted as $f(\underline{y}_i|\underline{x}_i)$.

The block-level hard demodulation for each block is given by $\underline{x}_i^* = \arg\max_{\underline{x}_i} f(\underline{y}_i|\underline{x}_i)$. As established in [23], considering the correlation within relatively small blocks suffices to take advantage of the majority of the information in correlations across the whole codeword. Therefore, by ignoring the correlation between blocks, the likelihood of a decision $\underline{\hat{x}}^{n_b}$ can be approximated as

$$f(\underline{y}^{n_b}|\underline{\hat{x}}^{n_b}) \approx \prod_{i=1}^{n_b} f(\underline{y}_i|\underline{\hat{x}}_i) \tag{1}$$

$$\propto \exp\left(\sum_{i=1}^{n_b} \log f(\underline{y}_i|\underline{\hat{x}}_i)\right) \propto \exp\left(\sum_{i=1}^{n_b} \log \frac{f(\underline{y}_i|\underline{\hat{x}}_i)}{f(\underline{y}_i|\underline{x}_i^*)}\right), \tag{2}$$

where we call

$$\delta(\underline{\hat{x}}_i, \underline{x}_i^*) = -\log \frac{f(\underline{y}_i|\underline{\hat{x}}_i)}{f(\underline{y}_i|\underline{x}_i^*)} \geq 0$$

the *relative reliability* of the demodulated block $\underline{x}_i^*$ to the alternative block $\underline{\hat{x}}_i$. The relative reliability reduces to bit-level reliability, i.e. the absolute value of a log-likelihood ratio, in the absence of any correlation. To rank order the likelihood of decisions from most likely to least likely, it suffices to rank order the sum of relative reliabilities $\sum_{i=1}^{n_b} \delta(\underline{\hat{x}}_i, \underline{x}_i^*)$ from least to highest. Each term $\delta(\underline{\hat{x}}_i, \underline{x}_i^*)$ quantifies the degradation of likelihood resulting from replacing the hard demodulated block $\underline{x}_i^*$ with $\underline{\hat{x}}_i$.

For the i-th block, there are $|\mathcal{X}|^b$ possible decisions of $\underline{\mathcal{X}}_i$ with $\underline{\mathcal{X}}_i^*$ having the least relative reliability, of zero. This results in $n_b(|\mathcal{X}|^b - 1)$ positive relative reliabilities for the alternative decisions except for the hard demodulated blocks. If the rank-ordered positive relative reliabilities are approximately a line, the generator from ORBGRAND [3] can be adapted to efficiently generate noise effects in approximately decreasing order of likelihood, where each positive relative reliability is treated as a binary reliability. Each generated noise effect corresponds to a unique decision for the blocks, $\underline{\hat{x}}^{n_b}$, provided that each hard demodulated block is swapped at most once. If a hard demodulated block is swapped more than once, the corresponding noise effect will be discarded. The first decision $\underline{\hat{x}}^{n_b}$ that passes the parity check is a near ML decoding. As demonstrated in [23], for channels subject to correlation ORBGRAND-AI offers superior decoding accuracy by leveraging CSI. Moreover, its taped out implementation [28] demonstrates the ability to do so efficiently in practice.

B. GCD

GCD [6], [7] assumes a binary-discrete memoryless channel, in which case the channel output can instead be represented by Y^n. As a result, the likelihood of $\mathtt{x}^n$, denoted as $f_{Y^n|X^n}(\mathtt{y}^n|\mathtt{x}^n)$, can be calculated as $\prod_{i=1}^n f_{Y_i|X_i}(\mathtt{y}_i|\mathtt{x}_i)$. The bit-level hard modulation is calculated as $\mathtt{x}_i^* = \arg\max_{\mathtt{x}_i} f_{Y_i|X_i}(\mathtt{y}_i|\mathtt{x}_i)$. Without loss of generality, GCD assumes that the parity check matrix is written in a systematic format, i.e.,

$$H = [I \quad B], \tag{3}$$

where B is an $n-k$ by k binary matrix and I is an $n-k$ by $n-k$ identity matrix. That form can be achieved for any binary linear code by using Gauss-Jordan elimination and column permutation, which only needs to be performed once offline. For GCD, let $\mathcal{I} = \{1, \ldots, n-k\}$ and $\mathcal{B} = \{n-k+1, \ldots, n\}$. The assumption (3) enables GCD to extend any binary subsequence $\hat{\mathtt{x}}_{\mathcal{B}}$ to a complete codeword $\hat{\mathtt{x}}^n$ via $\hat{\mathtt{x}}_{\mathcal{I}} = \hat{\mathtt{x}}_{\mathcal{B}} B^T$. We will call the bits in $\hat{\mathtt{x}}_{\mathcal{B}}$ the *base bits* as they are extrapolated to identify a whole codeword.

In GCD, a pattern generator is used to sequentially provide decisions of the base bits. For a decision of the base bits denoted as $\hat{\mathtt{x}}_{\mathcal{B}}^n$, its likelihood is calculated as $\prod_{i\in\mathcal{B}} f_{Y_i|X_i}(\mathtt{y}_i|\hat{\mathtt{x}}_i)$, where the extended codeword has likelihood $f_{Y^n|X^n}(\mathtt{y}^n|\hat{\mathtt{x}}^n)$. GCD keeps a record of the running maximum of $f_{Y^n|X^n}(\mathtt{y}^n|\hat{\mathtt{x}}^n)$, which is denoted as p^*, and the corresponding codeword. Assuming the pattern generator generates decisions for the base bits in decreasing order of $\prod_{i\in\mathcal{B}} f_{Y_i|X_i}(\mathtt{y}_i|\hat{\mathtt{x}}_i)$, GCD terminates when

$$\prod_{i\in\mathcal{B}} f_{Y_i|X_i}(\mathtt{y}_i|\hat{\mathtt{x}}_i) \prod_{i\in\mathcal{I}} f_{Y_i|X_i}(\mathtt{y}_i|\mathtt{x}_i^*) < p^*, \tag{4}$$

as no future pattern could lead to a codeword that has higher likelihood than the running maximum p^*. The codeword whose likelihood is the running maximum is the ML decoding. If the decisions of base bits are generated in approximately decreasing order of likelihood, the resulting decoding is a near-ML decoding.

III. GCD driven by ORBGRAND-AI

We investigate the use of ORBGRAND-AI as the pattern generator of GCD to incorporate correlations. When leveraging correlations instead of assuming independent noise effects, column permutation to place H in systematic format would require modifying the order of transmission of bits at the transmitter side. To integrate ORBGRAND-AI with GCD for all linear binary codes, including all systematic and non-systematic codes, we assume Gauss-Jordan elimination without column permutation results in

$$H_{:,\mathcal{B}} = B, H_{:,\mathcal{I}} = I, \tag{5}$$

for some $\mathcal{B} \subseteq \{1, \ldots, n\}$ with $|\mathcal{B}| = k$ and $\mathcal{I} = \{1, \ldots, n\} \setminus \mathcal{B}$. Note that, in this format any subsequence $\hat{\mathrm{x}}_{\mathcal{B}}$ can be extended to a complete codeword via $\hat{\mathrm{x}}_{\mathcal{I}} = \hat{\mathrm{x}}_{\mathcal{B}} B^T$. Equation (5) differs from (3) in the sense that $\mathcal{B}$ and $\mathcal{I}$ can be non-convex integer sets. The non-convexity can occur for some non-systematic codes, as illustrated in Section IV.

As in ORBGRAND-AI, consecutive symbols are grouped into n_b blocks, with each block containing b symbols or fewer if needed. In addition, we assume that each block only contains either bits in $\mathcal{B}$ or bits in $\mathcal{I}$. This ensures that the n_b blocks are separated into blocks containing only the base bits and blocks containing the bits in $\mathcal{I}$. The separation allows us to use ORBGRAND-AI to generate decisions for the base bits and extend the decisions using B. An exception occurs when a symbol contains both base bits and bits in $\mathcal{I}$. If that a case, we adopt the framework of Locally Constrained-GCD [5] as follows: we use ORBGRAND-AI to generate decisions for the symbols that contain at least one base bit. Even though the generated partial decisions will be overspecified, each of them will be examined by the parity check matrix to see if it can lead to a valid codeword. A partial decision that passes the parity check will be extended using the decision of the base bits, as in GCD.

We denote $\{\underline{\mathcal{X}}_a\}_{a\in\mathcal{B}'}$ as the *base blocks* containing only the base bits, and similarly we denote $\{\underline{\mathcal{X}}_a\}_{a\in\mathcal{I}'}$ as the rest of the blocks. Every specification of the base blocks $\{\underline{\mathcal{X}}_{a,i}\}_{a\in\mathcal{B}'}$ can be extended to a full codeword using H similarly to GCD. Note that we only group consecutive symbols into blocks, as noise effects on consecutive symbols are more correlated compared to the noise effects on separated symbols. For example, in a Gauss-Markov channel with BPSK where consecutive noise has correlation ρ, the noise N_i and N_j have correlation $\rho^{|i-j|}$ which decreases exponentially as $|i-j|$ increases. Therefore, when $\mathcal{B}$ or $\mathcal{I}$ are not convex, not every group of b consecutive symbols is combined into one block. Further illustrations will be given in Section IV using a concrete example.

A. ORBGRAND-AI driven GCD (ORBGRAND-AI-GP)

With ORBGRAND-AI, we can sequentially generate decisions of the base blocks and use (1) to calculate the likelihoods of decisions. Denote $\{\underline{\mathcal{X}}_{a,i}\}_{a\in\mathcal{B}'}$ as the base blocks generated by ORBGRAND-AI at the i-th query. For a running maximum likelihood p^*, it is updated to be the likelihood of the i-th query if

$$\prod_{a=1}^{n_b} f(\underline{\mathcal{Y}}_a|\underline{\mathcal{X}}_{a,i}) > p^*, \tag{GP}$$

which is referred to as the maximum update condition. Adapting the derivation from (4) yields the following stopping criterion:

$$\left(\prod_{a\in\mathcal{B}'} f(\underline{\mathcal{Y}}_a|\underline{\mathcal{X}}_{a,i})\right)\left(\prod_{a\in\mathcal{I}'} f(\underline{\mathcal{Y}}_a|\underline{\mathcal{X}}^*_a)\right) < p^*, \tag{6}$$

which serves as the stopping criterion for ORBGRAND-AI generated GCD. This stopping criterion is consistent with the pattern generator in that it incorporates the correlation within each block but assumes independence between blocks. We will refer to the ORBGRAND-AI generated GCD with maximum update condition (GP) and stopping criterion (6), which leverages partial correlation, as the ORBGRAND-AI-GP. In addition, the decoding process will stop early if the first query leads to the same codeword as the hard demodulation, in which case, it can be shown that the second query will satisfy the stopping criterion.

B. ORBGRAND-AI integrated GCD (ORBGRAND-AI-GT)

Although (1) uses approximate independence across blocks to generate patterns, which is also used in (GP) and (6), we can incorporate the omitted correlation into the maximum update rule. For the i-th generated decision $\underline{x}^{n_b,i} = (\underline{x}_{1,i}, \ldots, \underline{x}_{n_b,i})$, the running maximum p^* is updated if

$$f(\underline{y}^{n_b}|\underline{x}^{n_b,i}) > p^*, \tag{GT}$$

where $f(\underline{y}^{n_b}|\underline{x}^{n_b,i})$ is evaluated using the full CSI.

For the stopping criterion, note that the second product in (6), which serves as an upper bound for the likelihood multiplier corresponding to the blocks in $\mathcal{I}'$, relies on block-level hard demodulation and the assumption of independence between blocks. The principle here is to use the approximate independence where it is needed, to facilitate efficient pattern generation, but to keep full CSI to inform the maximum update condition.

Essentially, the principle is to retain the efficiency of pattern generation using idea of approximate independence, but have the update criterion reflect full CSI. We will refer to ORBGRAND-AI-GP with the maximum update condition replaced by (GT), which leverages total correlation, as ORBGRAND-AI-GT. As will be shown in Section IV, replacing the maximum update condition suffices to leverage correlation between blocks for lower BLER. In a circuits implementation, the replacement of (GP) by (GT) can be seen as changing the block size b to n_s in the calculation of block-level likelihood.

IV. Performance Evaluation

The standard channel for soft decision decoding benchmarking is BPSK subject to additive White Gaussian noise (AWGN). We consider the natural correlated generalization where AWGN is replaced with a first order Gauss-Markov process [26]. That arises, for example, in an equalized inter-symbol-interference (ISI) channel subject to AWGN. Let $Z_1, \ldots, Z_n$ be i.i.d. Gaussian random variables with mean zero and variance σ^2. A Gauss-Markov process, $\{N^n\}$, with correlation ρ is defined by $N_1 = Z_1$ and $N_i = \rho N_{i-1} + \sqrt{1-\rho^2} Z_i$ for $i = 2, \ldots, n$. The received signal is $Y^n = (1 - 2X^n) + N^n \in \mathbb{R}^n$. For any $m \leq n$, we have that

$$f_{Z^m}(z^m) = f_N(z_1) \prod_{i=2}^{m} f_N\left(\frac{z_i - \rho z_{i-1}}{\sqrt{1-\rho^2}}\right),$$

where f_N is the PDF of the mean 0 Gaussian distribution with variance σ. The likelihood of a decision $\hat{x}^m$ needed in the decoding process is calculated by

$$f_{Y^m|X^m}(\mathrm{y}^m|\hat{\mathrm{x}}^m) = f_{Z^m}(\mathrm{y}^m - (1 - 2\hat{\mathrm{x}}^m)).$$

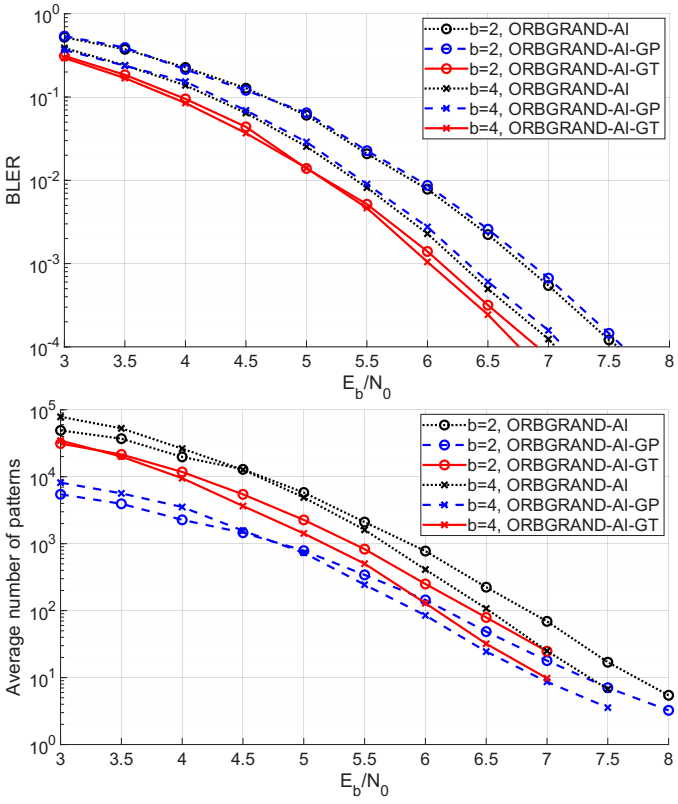

Fig. 1. BLER (upper panel) and pattern count (lower panel) vs Eb/N0 for a CRC $[64, 48]$. BPSK channel subject to Gauss-Markov noise with $\rho = 0.5$.

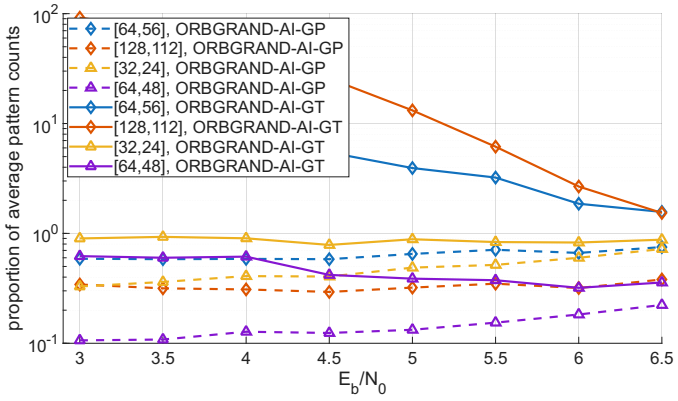

Fig. 2. Ratio of average pattern counts to ORBGRAND-AI for various code dimensions.

Fig. 1 considers the two combinations using ORBGRAND-AI and GCD when the code is systematic. CRC codes are widely used for error detection, but GRAND and GCD can upgrade them to error correction. Here we use CRC $[64, 48]$, where the 16-bit CRC is specified in the Distributed Network Protocol (DNP), and set $\rho = 0.5$, representing reasonable levels of ISI. For systematic codes where the first k bits are information bits, the set $\mathcal{B}$ equals $\{1, \ldots, k\}$ which contains consecutive integers. The black lines represent ORBGRAND-AI while the blue lines represent ORBGRAND-AI-GP. The red lines represent ORBGRAND-AI-GT. The lines with circles represent a block size $b = 2$ and the lines with crosses represent $b = 4$. The top plot records the BLER performance. The bottom plot records the average number of generated patterns, including the discarded patterns where a symbol is flipped more than once.

ORBGRAND-AI has mildly superior performance to ORBGRAND-AI-GP in terms of BLER, but ORBGRAND-AI-GP reduces the average number of tested patterns by up to 90%. In contrast, ORBGRAND-AI-GT, which uses the full CSI in its update rule, experiences a ∼0.75 dB gain when all decoders use $b = 2$ and around ∼0.4 dB when all decoders have $b = 4$. As could be expected, this gain in BLER performance comes at the cost of an increased number of tested patterns, but still fewer than ORBGRAND-AI.

Fig. 2 further investigates the reduction of pattern counts for different code dimensions where all 3 decoders use $b = 2$. The dashed lines represent the ratio of average pattern counts between ORBGRAND-AI-GP and ORBGRAND-AI, while the solid line is ORBGRAND-AI-GT and ORBGRAND-AI. ORBGRAND-AI-GP demonstrates consistent query reduction ability for all code rates and code lengths, where the reduction is more significant for lower-rate codes. As a counterpoint, to obtain better BLER, ORBGRAND-AI-GT has less query reduction gain for codes with a lower rate and can require more queries compared to ORBGRAND-AI for codes with a higher rate.

For non-systematic codes, the parity check matrix can be rewritten in either of the formats (3) or (5) without column permutation. In the case of (3), the blocks are constructed as if the code is systematic and therefore ORBGRAND-AI-GP performs as previously illustrated. In the case of (5), however, the set of base bits $\mathcal{B}$ and its complement $\mathcal{I}$ no longer contain only consecutive numbers, which makes one or more blocks contain fewer than b bits.

We illustrate the resolution of this issue using a $[128, 110]$ 5G NR CA-Polar code with an 11-bit CRC. The base bits of this code are $\mathcal{B} = \{18, \ldots, 32, 34, \ldots, 128\}$ and the complement is $\mathcal{I} = \{1, \ldots, 17, 33\}$. For $b = 2$, every two bits in $\mathcal{I}$ can be grouped into one block except for the 17-th bit and the 33-th bit as they are non-consecutive. When bits 17 and 33 are transmitted consecutively, their experience of the channel is highly correlated, enabling this correlation to be leveraged for decoding. However, implementing such consecutive transmission would necessitate column permutation of H and G, which requires modifications at the transmitter. By default, these two bits are transmitted with a time gap, which reduces their noise correlation to a near-negligible level, and hence can be ignored in the pattern generator as described in Section III. In this default configuration, each of the two bits becomes an individual block when using ORBGRAND-AI-GP, resulting in a major reduction in average query number at the cost of a minor BLER degradation, as depicted in Fig. 3.

V. Conclusions

We leverage ORBGRAND-AI and GCD in two ways. ORBGRAND-AI-GP provides a reduction in average pattern counts compared to ORBGRAND-AI at the cost of a minor

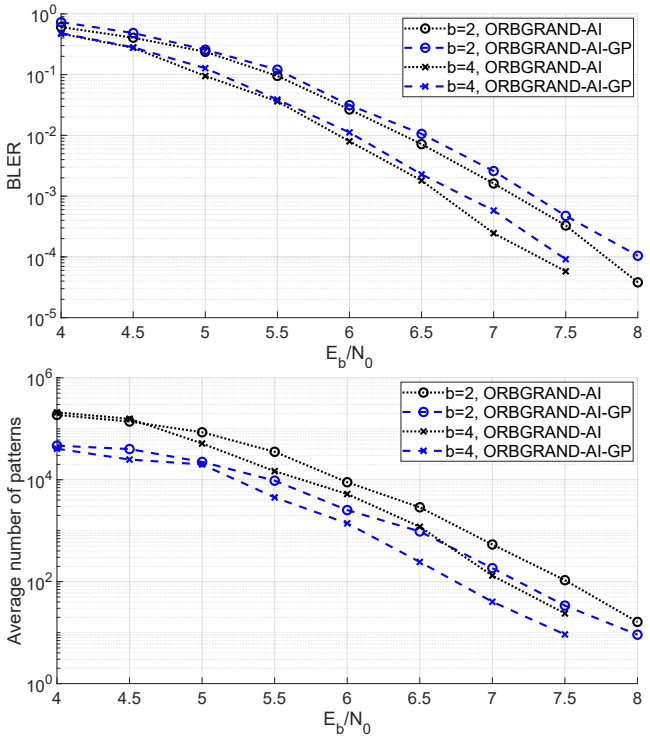

Fig. 3. BLER and average query numbers for a CA-Polar $[128, 110]$ code.

degradation in BLER performance. By modifying the maximum update condition to leverage more CSI, ORBGRAND-AI-GT provides better BLER performance at the cost of a higher average number of generated patterns. It still provides query reduction compared to ORBGRAND-AI only for codes with lower rates and higher values of $n-k$. We note that techniques have been successfully developed to leverage binary linear code structure to reduce query numbers with GRAND decoders [4], [31]–[35]. However, it remains unclear how those skipping techniques can be directly adapted to ORBGRAND-AI owing to its non-binary sensibilities.

References

[1] K. R. Duffy, J. Li, and M. Médard, "Capacity-achieving guessing random additive noise decoding," *IEEE Trans. Inf. Theory*, vol. 65, no. 7, pp. 4023–4040, 2019.

[2] A. Solomon, K. R. Duffy, and M. Médard, "Soft maximum likelihood decoding using GRAND," in *IEEE ICC*, 2020.

[3] K. R. Duffy, W. An, and M. Medard, "Ordered reliability bits guessing random additive noise decoding," *IEEE Trans. Signal Proc.*, vol. 70, pp. 4528–4542, 2022.

[4] L. Rapp, M. Médard, and K. R. Duffy, "SOGRAND assisted guesswork reduction," *IEEE Commun. Lett.*, 2025.

[5] X. Zheng and X. Ma, "Locally constrained guessing codeword decoding of short block codes," in *IEEE ITW*, 2024, pp. 442–447.

[6] ——, "A universal list decoding algorithm with application to decoding of polar codes," *IEEE Trans. Inf. Theory*, vol. 71, no. 2, pp. 975–995, 2025.

[7] Q. Wang, Y. Wang, X. Zheng, and X. Ma, "Ordered reliability bits guessing codeword decoding of short codes," *IEEE Wirel. Commun. Lett.*, vol. 14, no. 9, pp. 2823–2827, 2025.

[8] J. Griffin, P. Yuan, K. R. Duffy, and M. Medard, "Using a single-parity-check to reduce the guesswork of guessing codeword decoding," in *CISS*, 2025.

[9] R. Hadavian, D. Truhachev, K. El-Sankary, H. Ebrahimzad, and H. Najafi, "Ordered reliability direct error pattern testing (ORDEPT) algorithm," in *GLOBECOM*, 2023, pp. 6983–6988.

[10] R. Hadavian, X. Huang, D. Truhachev, K. El-Sankary, H. Ebrahimzad, H. Najafi, Y. Ge, and A. Zokaei, "Ordered reliability direct error pattern testing decoding algorithm," *IEEE Trans. Commun.*, 2025.

[11] S. M. Abbas, T. Tonnellier, F. Ercan, and W. J. Gross, "High-throughput VLSI architecture for GRAND," in *IEEE Workshop Signal Process. Syst.*, 2020, pp. 1–6.

[12] S. M. Abbas, T. Tonnellier, F. Ercan, M. Jalaleddine, and W. J. Gross, "High-throughput VLSI architecture for soft-decision decoding with ORBGRAND," in *IEEE Int. Conf. Acoust. Speech and Signal Process.*, 2021, pp. 8288–8292.

[13] C. Condo, V. Bioglio, and I. Land, "High-performance low-complexity error pattern generation for ORBGRAND decoding," in *IEEE GLOBECOM Workshops*, 2021, pp. 1–6.

[14] S. M. Abbas, M. Jalaleddine, and W. J. Gross, "High-throughput VLSI architecture for GRAND Markov order," in *IEEE Workshop Signal Process. Syst.*, 2021, pp. 158–163.

[15] C. Ji, X. You, C. Zhang, and C. Studer, "Efficient ORBGRAND implementation with parallel noise sequence generation," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 33, no. 2, pp. 435–448, 2025.

[16] S. M. Abbas, M. Jalaleddine, C.-Y. Tsui, and W. J. Gross, "Improved Step-GRAND: low-latency soft-input guessing random additive noise decoding," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2025.

[17] A. Riaz, V. Bansal, A. Solomon, W. An, Q. Liu, K. Galligan, K. R. Duffy, M. Medard, and R. T. Yazicigil, "Multi-code multi-rate universal maximum likelihood decoder using GRAND," in *IEEE Eur. Solid State Circuits Conf.*, 2021, pp. 239–246.

[18] L. D. Blanc, V. Herrmann, Y. Ren, C. Müller, A. T. Kristensen, A. Levisse, Y. Shen, and A. Burg, "A GRANDAB decoder with 8.48 Gbps worst-case throughput in 65nm CMOS," in *IEEE ESSERC*, 2024, pp. 685–688.

[19] A. Riaz, A. Yasar, F. Ercan, W. An, J. Ngo, K. Galligan, M. Médard, K. R. Duffy, and R. T. Yazicigil, "A sub-0.8-pJ/bit universal soft-detection decoder using ORBGRAND," *IEEE J. Solid-State Circuits*, 2025.

[20] P. Yuan, M. Medard, K. Galligan, and K. R. Duffy, "Soft-output (SO) GRAND and iterative decoding to outperform LDPC codes," *IEEE Trans. Wireless Commun.*, vol. 24, no. 4, pp. 3386–3399, 2025.

[21] K. R. Duffy, P. Yuan, J. Griffin, and M. Médard, "Soft-output guessing codeword decoding," *IEEE Commun. Lett.*, vol. 29, no. 2, pp. 328–332, 2024.

[22] W. An, M. Médard, and K. R. Duffy, "Keep the bursts and ditch the interleavers," *IEEE Trans. Commun.*, vol. 70, no. 6, pp. 3655–3667, 2022.

[23] K. R. Duffy, M. Grundei, and M. Médard, "Using channel correlation to improve decoding – ORBGRAND-AI," in *IEEE GLOBECOM*, 2023.

[24] J. A. Millward, K. R. Duffy, M. Rangaswamy, and M. Médard, "Using equalization-induced noise coloring to improve error-correcting decoding," in *Asilomar Conf. Signals, Sys., and Comput.*, 2024, pp. 1400–1403.

[25] ——, "Enhancing guessing random additive noise decoding using channel estimation," in *SPAWC*, 2024, pp. 676–680.

[26] K. R. Duffy, M. Grundei, J. A. Millward, M. Rangaswamy, and M. Médard, "Decoding in the presence of ISI without interleaving – ORBGRAND-AI," *arXiv:2510.14939*, 2025.

[27] J. T. Lewis, C.-E. Pfister, and W. G. Sullivan, "Entropy, concentration of probability and conditional limit theorems," *Markov Process. Relat. Fields*, vol. 1, no. 3, pp. 319–386, 1995.

[28] Z. E. Kizilates, A. Riaz, A. Bali, M. Grundei, M. Médard, K. R. Duffy, and R. T. Yazicigil, "Low-latency modulation- and correlation-adaptive ORBGRAND-AI decoder," in *IEEE ESSERC*, 2025.

[29] A. Cohen, R. G. D'Oliveira, K. R. Duffy, J. Woo, and M. Médard, "AES as error correction: Cryptosystems for reliable communication," *IEEE Commun. Lett*, vol. 27, no. 8, pp. 1964–1968, 2023.

[30] W. An, M. Médard, and K. R. Duffy, "Soft decoding without soft demapping with ORBGRAND," in *IEEE ISIT*, 2023, pp. 1080–1084.

[31] M. Rowshan and J. Yuan, "Constrained error pattern generation for GRAND," in *IEEE ISIT*, 2022, pp. 1767–1772.

[32] ——, "Low-complexity GRAND by segmentation," in *IEEE GLOBECOM*, 2023, pp. 6145–6151.

[33] ——, "Segmented GRAND: Complexity reduction through sub-pattern combination," *IEEE Trans. Commun.*, pp. 1–1, 2025.

[34] Y. Wang, J. Liang, and X. Ma, "Partially constrained GRAND of linear block codes," in *Int. Conf. Comput. Commun. Sys.*, 2024, pp. 617–622.

[35] L. Rapp, J. Feng, M. Médard, and K. R. Duffy, "A balanced tree transformation to reduce GRAND queries," in *IEEE ISIT*, 2025.