



INSTITUTO
UNIVERSITÁRIO
DE LISBOA

**Signal Processing and Coding Techniques for 6G Ultra-reliable Low-latency Wireless
Machine-type Communications**

Sahar Allahkaram

PhD in Information Science and Technology

Supervisor:

Doctor Francisco António Taveira Branco Nunes Monteiro, Associate Professor, Iscte – Instituto
Universitário de Lisboa

March, 2026



TECNOLOGIAS
E ARQUITETURA

Department of Information Science and Technology

**Signal Processing and Coding Techniques for 6G Ultra-reliable Low-latency Wireless
Machine-type Communications**

Sahar Allahkaram

PhD in Information Science and Technology

Jury:

Professor Américo Manuel Carapeto Correia, Full Professor, Iscte – Instituto Universitário de Lisboa

Professor Muriel Médard, Full Professor, Massachusetts Institute of Technology (MIT)

Doctor Marco Alexandre Cravo Gomes, Assistant Professor, Universidade de Coimbra

Doctor Nuno Manuel Branco Souto, Associate Professor, Iscte – Instituto Universitário de Lisboa

Doctor Francisco António Taveira Branco Nunes Monteiro, Associate Professor, Iscte – Instituto Universitário de Lisboa

March, 2026

Direitos de cópia ou Copyright ©Copyright: Sahar Allahkaram.

O Iscte - Instituto Universitário de Lisboa tem o direito, perpétuo e sem limites geográficos, de arquivar e publicitar este trabalho através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, de o divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objetivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

Acknowledgements

Today is the day after four years of intense work; I am putting the final touches on my dissertation by sending this letter of appreciation. It has been a moment of intense learning for me, personally and in science. This dissertation's writing has significantly impacted me. I want to reflect on the people who have been supportive and helpful to me throughout this time.

Foremost, I would like to express my deep gratitude to my advisor, Prof. Francisco Monteiro, for his ongoing support of my Ph.D. research and his patience, inspiration, and vast knowledge. His advice was helpful to me throughout the entire research and thesis-writing process.

Most importantly, I am grateful for my family's unconditional, unequivocal, and loving support. Although they were not physically beside me, their love and support were with me the whole time.

I gratefully acknowledge ISCTE – Instituto Universitário de Lisboa for the merit-based scholarship awarded to me. This support was fundamental to the successful completion of this PhD thesis; without it, the development and realization of this research would not have been possible. I am deeply thankful for the institutional trust, financial support, and academic environment provided by ISCTE, which were essential to carrying out this work.

This work has also been supported by Instituto de Telecomunicações, via national funds through FCT – Fundação para a Ciência e a Tecnologia, I.P., and, when eligible, co-funded by EU funds under project/support UID/50008/2025 – Instituto de Telecomunicações, with DOI identifier <https://doi.org/10.54499/UID/50008/2025>.

Resumo

As redes sem fios de sexta geração (6G) impõem requisitos rigorosos ao nível da camada física, em particular no que respeita a comunicações ultra-fiáveis de baixa latência (URLLC). Esta deve ser inferior a 1 ms, e com uma fiabilidade superior a 99,999%. Os esquemas convencionais de codificação com códigos longos são incompatíveis com estas restrições devido aos atrasos inerentes à decodificação e ao uso de entrelaçadores. Esta dissertação desenvolve técnicas de processamento de sinal e de codificação de baixa complexidade, direcionadas para comunicações máquina-a-máquina (MTC) com códigos curtos.

A primeira contribuição demonstra a viabilidade de sistemas massive multiple-input multiple-output (mMIMO) codificados, integrando códigos lineares aleatórios (RLC), multiplexagem espacial baseada em *zero-forcing* e decodificação baseada em *guessing random additive noise* (GRAND). Os resultados evidenciam diminuições significativas da latência face aos códigos polares do 5G, alcançando também ganhos de codificação assinaláveis, especialmente para valores elevados de relação sinal-ruído (SNR).

A segunda contribuição aplica o GRAND ao nível dos símbolos da modulação em sistemas mMIMO com constelações arbitrárias, através de ordenação das antenas e restrição dos padrões de erro, reduzindo a complexidade em mais de 90% sem perda de desempenho face à decodificação de máxima verosimilhança (ML).

A terceira contribuição introduz variantes do GRAND com restrições para códigos LDPC de bloco curto, permitindo reduzir mais de 99% dos padrões de erro candidatos mantendo a optimalidade ML.

Por fim, são desenvolvidas sequências de assinatura com soma única para acesso aleatório em canais aditivos ruidosos, permitindo identificação dos utilizadores.

Palavras-chave: Comunicações ultrafiáveis de baixa latência (URLLC), Massive MIMO (mMIMO), Decodificação GRAND, Codificação de bloco curto, Acesso aleatório sem concessão

Abstract

The emergence of sixth-generation (6G) wireless networks imposes physical-layer requirements, particularly ultra-reliable low-latency communication (URLLC), demanding sub-millisecond latency and reliability exceeding 99.999%. Conventional long-blocklength coding schemes are incompatible with these constraints due to inherent decoding delays and interleaving latency. This dissertation develops low-complexity, high-reliability signal processing and coding techniques tailored for short-blocklength wireless machine-type communications (MTC).

The first contribution demonstrates the feasibility of coded massive multiple-input multiple-output (mMIMO) systems by integrating random linear codes (RLCs), zero-forcing spatial multiplexing, and guessing random additive noise decoding (GRAND). Results show significant reliability and latency improvements over comparable 5G polar codes, achieving significant block error rate (BLER) gains and reduced decoding time, particularly at high signal-to-noise ratios (SNRs).

The second contribution extends symbol-level GRAND to mMIMO with arbitrary constellations through antenna-reliability ordering and constrained error enumeration, reducing decoding complexity by over 90% while preserving maximum-likelihood (ML) performance.

The third contribution introduces constraint-based GRAND variants for short low-density parity-check (LDPC) codes, including set-partitioning, Sieving-GRAND, and largely constrained GRAND, enabling over 99% reduction in candidate error patterns while maintaining ML optimality.

Finally, uniquely decodable Bose–Chowla–Lindström signature sequences are developed for grant-free random access over noisy adder channels, enabling reliable user identification and collision resolution.

Keywords: Ultra-reliable low-latency communication (URLLC), Massive MIMO (mMIMO), Guessing random additive noise decoding (GRAND), Short-blocklength coding, Grant-free random access

Index

Acknowledgements	i
Resumo	ii
Abstract	iii
Index	iv
Tables Index	vi
Figures Index	vii
List of Acronyms	viii
1 Introduction	1
1.1 Context and Motivation	1
1.1.1 6G Wireless and Machine-Type Communications	1
1.1.2 URLLC and Finite Blocklength Communication	4
1.1.3 Limitations of Short-Block Coding and of Traditional Decoders	7
1.1.4 Massive MIMO for Spatial Multiplexing	12
1.1.5 The Limitations of Code-Centric Decoders for Short Blocklengths	14
1.1.6 Universal Decoding	18
1.1.7 Noise-Centric Decoding Paradigm	20
1.1.8 Random Access and Grant-Free Communication in 6G	21
1.1.9 Research Objectives and Contributions	23
1.2 Technical Foundation	24

1.2.1	Random Linear Codes	24
1.2.2	Syndrome-based Error Detection and Correction	26
1.2.3	Low-density Parity Check Codes	26
1.2.4	Quasi-regular LDPC codes	27
1.2.5	Massive Multiple-Input Multiple-Output (mMIMO)	29
1.2.6	Generalized Quadrature amplitude modulation (QAM)	30
1.2.7	Guessing Random Additive Noise Decoding (GRAND)	33
1.2.8	Bose-Chowla-Lindström Code	36
1.3	Thesis Structure	38
2	URLLC with Coded Massive MIMO via Random Linear Codes and GRAND	40
3	Constrained Symbol-Level Noise-Guessing Decoding with Antenna Sorting for Massive MIMO	46
4	Largely Constrained Noise-guessing Decoding of Short LDPC codes	65
5	Uniquely Decodable Signature Sequences Over the Noisy Adder Channel	86
6	Conclusions and future research	100
6.1	Main conclusions	100
6.2	Future research proposals	102
	References	105

Tables Index

1.1	Key performance requirements for next-generation URLLC systems	8
-----	--	---

Figures Index

1.1	Overview of emerging 6G use cases such as robotic surgery, autonomous vehicles, drones, underwater communication, reconfigurable factories, satellite links, and IoT that demand unprecedented reliability and very low latency. The figure was created by using figures available on the internet.	2
1.2	Encoding process of a random linear code (RLC), mapping an information message $\mathbf{a}$ into a binary codeword $\mathbf{x}_b$	24
1.3	Structure of a codeword in a systematic linear block code.	25
1.4	Example of a LDPC parity-check matrix ($n = 20, k = 5, w_c = 3, w_r = 4$). . . .	27
1.5	Example of a quasi-regular LDPC code matrix ($n = 30, k = 10, w_c = 4$). . . .	28
1.6	General massive multiple-input multiple-output (mMIMO) model, where $N_T \gg N_R$ or $N_R \gg N_T$	30
1.7	Type I, Type II, and Type III QAM constellations.	31
1.8	Gray-coded 16-QAM.	32
1.9	Visualization of GRAND error pattern enumeration for a block length $n = 6$. Each subplot corresponds to error patterns of a given Hamming weight. The x-axis indexes error patterns in order of generation, and the y-axis shows the bit positions affected by the error. This illustrates the principle of GRAND: systematically guessing possible noise patterns in increasing order of likelihood (here, increasing Hamming weight) until a valid codeword is found.	35

List of abbreviations

3GPP	Third Generation Partnership Project
2D	Two-Dimensional
AED	Approximate Exhaustive Decoding
AoI	Age of Information
AoL	Age of Loop
AWGN	Additive White Gaussian Noise
BCH	Bose–Chaudhuri–Hocquenghem
BCL	Bose–Chowla–Lindström
BLER	Block Error Rate
BS	Base Station
CC	Convolutional Code
CDF	Cumulative Distribution Function
CRC	Cyclic Redundancy Check
CSI	Channel State Information
cMTC	Critical Machine-Type Communications
eMBB	Enhanced Mobile Broadband
GCD	Guessing Codeword Decoding
GF	Galois Field
GRAND	Guessing Random Additive Noise Decoding
GRANDAB	Guessing Random Additive Noise Decoding with ABandonment

HARQ	Hybrid Automatic Repeat Request
HQAM	Hexagonal Quadrature Amplitude Modulation
IDER	Identification Error Rate
IOT	Internet of Things
ISAC	Integrated Sensing and Communication
LC-GRAND	Largely Constrained Guessing Random Additive Noise Decoding
LDPC	Low-Density Parity-Check
LTE	Long-Term Evolution
MAAC	Multi-Access Adder Channel
ML	Maximum Likelihood
MMSE	Minimum Mean-Squared Error
mMIMO	Massive Multiple-Input Multiple-Output
MIMO	Multiple-Input Multiple-Output
mMTC	Massive Machine-Type Communications
MRC	Maximum-Ratio Combining
MTC	Machine-Type Communications
MS	Min-Sum
NOMA	Non-Orthogonal Multiple Access
NMS	Normalized Min-Sum
ORBGRAND	Ordered Reliability Bits Guessing Random Additive Noise Decoding
OFDM	Orthogonal Frequency Division Multiplexing
OSD	Ordered-Statistics Decoding
OMS	Offset Min-Sum
PDF	Probability Density Function
PLNC	Physical Layer Network Coding

PMF	Probability Mass Function
QAM	Quadrature Amplitude Modulation
QoS	Quality of Service
RLC	Random Linear Code
RQAM	Rectangular Quadrature Amplitude Modulation
SER	Symbol Error Rate
SCL	Successive Cancellation List
S-GRAND	Sieving Guessing Random Additive Noise Decoding
SIC	Successive Interference Cancellation
SINR	Signal-to-Interference-plus-Noise Ratio
SNR	Signal-to-Noise Ratio
SPA	Sum-Product Algorithm
SP-GRAND	Set Partitioning Guessing Random Additive Noise Decoding
SQAM	Square Quadrature Amplitude Modulation
SRGRAND	Symbol Reliability Guessing Random Additive Noise Decoding
TB	Tail Biting
TEP	Test Error Pattern
UD	Uniquely Decodable
UAV	Unmanned Aerial Vehicle
URLLC	Ultra-Reliable Low-Latency Communications
XQAM	Cross Quadrature Amplitude Modulation
ZF	Zero-Forcing

Chapter 1

Introduction

1.1 Context and Motivation

1.1.1 6G Wireless and Machine-Type Communications

The transfer from fifth-generation (5G) wireless networks to sixth-generation (6G) systems signifies not only an increase in data rates and spectral efficiency but also a significant qualitative transformation in wireless communication. While earlier generations of mobile networks were fundamentally designed around human-centric traffic patterns, future 6G systems are being architected for a world in which machines, rather than humans, dominate both the volume and the criticality of communication flows [1]. This transformation is driven by the rapid convergence of autonomous systems, distributed intelligence, immersive environments, cyber–physical control loops, and large-scale industrial automation. In such environments, wireless communication ceases to be an auxiliary data pipe and instead becomes a deterministic control substrate for physical processes.

As discussed in [1], emerging 6G applications, including autonomous vehicles, collaborative robotics, industrial digital twins, mobile augmented reality, distributed sensor fabrics, and tactile internet systems, impose communication requirements that extend far beyond high throughput. These systems demand deterministic latency, extreme reliability, and massive device density, and often simultaneously. In many control-centric scenarios, the value of information is not measured by its bit rate but by its timeliness and its impact on system stability. Update intervals below one millisecond, packet-loss probabilities below 10^{-7} , and jitter on the order of tens of microseconds are not exceptional requirements, but normative design constraints.

The inadequacy of incremental 5G evolution to meet these requirements has been widely acknowledged. Despite major advances in 5G, including flexible orthogonal frequency division multiplexing (OFDM) numerologies, massive multi-input multi-output (mMIMO), and low-latency scheduling, its architectural foundations remain rooted in human-centric broadband

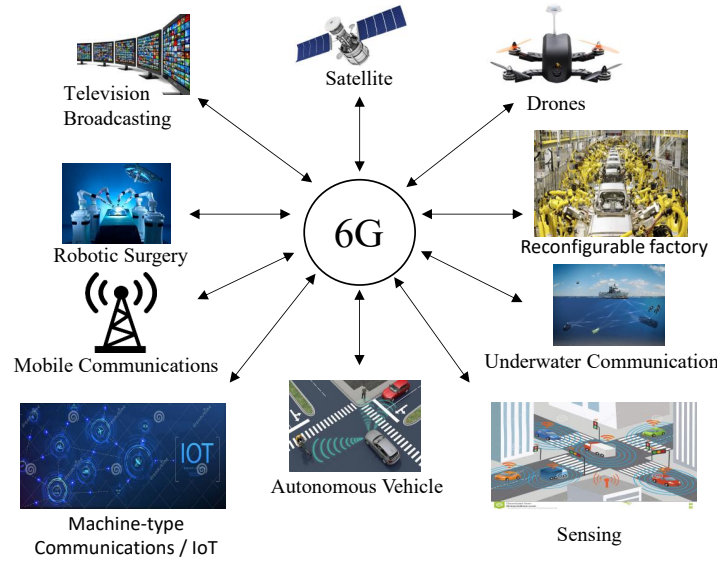


Figure 1.1: Overview of emerging 6G use cases such as robotic surgery, autonomous vehicles, drones, underwater communication, reconfigurable factories, satellite links, and IoT that demand unprecedented reliability and very low latency. The figure was created by using figures available on the internet.

communication, characterized by bursty but predictable traffic, latency tolerance in the tens of milliseconds, and dominant downlink flows [1],[2]. These assumptions fundamentally collapse in machine-type environments, where short periodic packets, random access, and ultra-high reliability define the traffic fabric.

This insight motivates a broader re-evaluation of the foundational principles of communication theory itself. In [3], the authors argue that 6G cannot be realized through incremental extensions of Shannon’s classical framework. Shannon theory assumes ergodicity, infinite blocklengths, and independence between communication and control. However, 6G systems operate in precisely the opposite regime: finite blocklengths, non-ergodic channels, strict delay budgets, and deep integration with sensing, actuation, and learning. The notion of “communication efficiency” is therefore undergoing a redefinition from maximizing bits per second per Hertz to maximizing the relevance, freshness, and control utility, that is, the task-level usefulness of information in supporting control and decision making of delivered information.

This paradigm shift introduces new communication concepts such as semantic communication [3], goal-oriented communication, and task-aware networking, in which packets are prioritized not only by quality of service (QoS) class but by their marginal utility to a control or inference task [4], [3]. Metrics such as age of information (AoI) and age of loop (AoL) are replacing classical throughput as the dominant performance indicators, revealing that the physical layer must now be optimized not for average rates, but for information timeliness and the stability of the closed-loop control system [1]. These ideas fundamentally reshape the role of coding and decoding in future networks: coding is no longer just a mechanism for approaching asymptotic capacity, but a tool for ensuring deterministic reliability under strict latency con-

straints [5]. With the emergence of 6G, there is increasing interest in rethinking the fragmented landscape of channel coding schemes used in different communication scenarios, motivating research into unified and adaptable coding frameworks [6].

Another defining feature of the 6G principle is the deep integration of communication and sensing, commonly referred to as integrated sensing and communication (ISAC) [7]. Unlike previous generations, where sensing and communication were treated as disjoint systems, 6G systems must support joint communication, sensing control loops sharing spectrum, hardware, and signal processing infrastructures. This integration introduces new theoretical challenges, including joint waveform optimization, tradeoffs between sensing precision and decoding reliability, and fundamental limits on parameter estimation under simultaneous data transmission [3]. These challenges directly affect the design of modulation, coding, and detection algorithms, particularly in URLLC and critical machine-type systems.

Machine-type communications (MTC) emerged in 5G as a new service category distinct from enhanced mobile broadband (eMBB). However, 6G MTC transcends the conceptual boundaries of 5G. In 5G, MTC was divided into massive MTC (mMTC) serving a large number of low-power, low-rate devices and ultra-reliable low-latency communication (URLLC) serving a small number of devices with extreme reliability and latency constraints. While this classification was adequate for early internet of things (IoT) and industrial automation, it does not generalize well to the heterogeneity and scale of 6G applications [2].

As shown in [2] future 6G machine-type systems must simultaneously support dense of autonomous robots and unmanned aerial vehicles (UAVs), ultra-low-power zero-energy sensors, immersive haptic interfaces, safety-critical industrial controllers, and autonomous mobility infrastructures. These systems do not fall cleanly into either mMTC or URLLC. Instead, they require hybrid service classes that combine: extreme reliability of the order of $1 - 10^{-7}$, sub-millisecond deterministic latency, massive device density, three-dimensional coverage, energy-neutral operation. To address this gap, [2] proposes a new service taxonomy for 6G MTC, including dependable critical MTC (cMTC), broadband cMTC, scalable cMTC, globally scalable mMTC, and zero-energy mMTC. Unlike traditional QoS classifications, these categories are defined not only by latency and reliability, but also by energy neutrality, synchronization accuracy, predictability, security, and information freshness. In dependable cMTC, for example, deterministic latency with bounded jitter and application-level reliability approaching $1 - 10^{-7}$ implies per link physical-layer reliabilities approaching $1 - 10^{-9}$, well beyond the targets of classical URLLC.

The architectural consequences of these requirements are profound. In [1] demonstrates that future systems must concurrently support sparse bursts from millions of devices and tightly synchronized communication among small subsets of critical nodes. This convergence mitigates the classical separation between massive access and critical communication. Random access must be grant-free and collision-resilient; metadata signaling, that is, the control information

required for device identification, synchronization, and decoding, must be ultra-compressed, and decoding must operate reliably in the finite-blocklength regime.

Random access emerges as one of the most severe bottlenecks. In current long-term evolution (LTE/4G) and 5G NR systems, grant-based access requires multi-step handshake procedures that alone exceed the entire URLLC latency budget. Grant-free random access removes this overhead but introduces severe challenges of preamble collision, inter-user interference, and unreliable channel estimation under short packets. New paradigms such as unsourced random access, compressed-sensing-based activity detection, and multi-antenna covariance estimation are therefore required to scale access under extreme density [1]. These techniques intrinsically rely on sparse signal processing and code-detection co-design, reinforcing the need for new short-packet coding paradigms.

Energy efficiency constitutes another core constraint in large-scale MTC. In UAV-assisted and aerial IoT deployments, 6G must support both URLLC-grade reliability and stringent battery constraints. UAVs operate as airborne relays, mobile base stations (BS), and sensing platforms, but their energy consumption is dominated not only by propulsion but also by communication, sensing, and processing. Finite-blocklength theory becomes critical in this setting, as achievable throughput and decoding reliability are fundamentally constrained under short-packet transmissions, which directly impacts the ability to meet latency requirements [8]. The integration of non-orthogonal multiple access (NOMA), cognitive radio, and joint power allocation demonstrates that signal processing, multiple access, and coding must be co-designed to meet URLLC and mMTC requirements simultaneously [8], [9].

These observations lead to a central conclusion: 6G MTC represents a fundamentally new communication regime, in which classical asymptotic coding theory, long-packet assumptions, and code-specific decoders can no longer serve as reliable design foundations. Instead, 6G demands short-blocklength-optimized, universal decoding, low-latency, and energy-aware signal processing and coding techniques. This dissertation proposes an ensemble of techniques that address such problems.

1.1.2 URLLC and Finite Blocklength Communication

URLLC systems operate under hard real-time constraints where even a single packet error or millisecond-scale delay can trigger system-level failure. Applications such as factory automation, tele-surgery, intelligent transportation systems, smart grids impose end-to-end delays below one millisecond and reliabilities exceeding 99.999% or 99.99999% [10], [2].

The third generation partnership project (3GPP) URLLC specification targets the successful transmission of a 32-byte packet within a 1 ms user-plane latency with reliability above $1 - 10^{-5}$. However, as demonstrated in [10], many industrial and medical use cases demand even stricter conditions, with end-to-end delays well below 1 ms and reliability exceeding $1 - 10^{-9}$. These

constraints place URLLC far beyond the feasible operating region of conventional LTE and 5G-NR physical layers, which rely on long blocklengths, retransmission-based reliability, and iterative decoding processes with unpredictable latency.

At the heart of URLLC physical-layer design lies a conflict between latency and reliability. On the one hand, low latency mandates the use of very short packets, often consisting of only a few tens or hundreds of bits. On the other hand, high reliability traditionally requires strong channel coding consistent with Shannon’s channel coding theorem, which guarantees arbitrarily small error probabilities at a finite rate ($R = k/n < \mathcal{C}$), where k information bits are encoded into n bits that constitute a codeword, and $\mathcal{C}$ denotes the Shannon capacity of the underlying channel, only in the limit ($n \rightarrow \infty$), long blocklengths, and sometimes multiple retransmissions. These mechanisms are inherently incompatible with hard real-time constraints. Even a single hybrid automatic repeat request (HARQ) round trip in LTE-based systems consumes several milliseconds, already exceeding URLLC latency budgets [10]. As a consequence, URLLC must rely on one-shot decoding, where the packet must be decoded correctly in a single transmission opportunity with extremely low block error probability. This requirement eliminates the safety net of time diversity and retransmissions that underpins classical broadband reliability, and it places pressure on the joint design of modulation, coding, detection, and decoding.

Paper [10] provides a systematic comparison of the most promising short-block coding families for URLLC, including BCH, tail-biting convolutional codes (TB-CC), turbo codes, LDPC codes, and polar codes. Under optimal ML decoding, extended BCH (eBCH) and high-memory TB-CCs approach the finite-blocklength normal approximation limits and exhibit exceptional error performance due to their large minimum distances. However, their decoding algorithms, such as ordered-statistics decoding (OSD) for BCH or high-memory Viterbi decoding for TB-CC, are computationally prohibitive for real-time URLLC operation. In particular, OSD is a near-ML soft-decision decoding algorithm that exploits the reliability information of the received symbols by ordering them according to their estimated reliabilities and generating multiple candidate codewords through a structured reprocessing of the least reliable positions, selecting the codeword with the minimum Euclidean distance to the received signal. More practical decoders, such as cyclic redundancy check (CRC)-aided polar successive cancellation list (CA-SCL) decoders and iterative LDPC belief-propagation (BP) decoders [11], offer a more favorable complexity-performance trade-off. Nevertheless, even these state-of-the-art solutions fall short of the theoretical finite-blocklength performance limits at ultra-low block error rate (BLER) targets (e.g., 10^{-6} , 10^{-7}), revealing a persistent performance complexity gap in the URLLC regime. For instance, decoding short LDPC codes of blocklength $n = 128$ using BP results in a performance loss of approximately 1.5 dB relative to ML decoding. As reliability requirements become increasingly stringent, this gap emerges as a critical limiting factor in mission-critical communication systems [10].

The limitations of classical block coding are further amplified by the architectural constraints

of 6G. In dense machine-type environments, URLLC services must coexist with massive connectivity. As shown in [2], reliable mMTC must guarantee deterministic latency and bounded jitter even under extreme spatial density and high interference. These conditions invalidate many assumptions underlying 6G URLLC, which was primarily designed for a small number of high-priority users. Beyond coding, multiple access and random access constitute additional physical-layer bottlenecks for URLLC. Grant-based access introduces handshake delays incompatible with sub-millisecond operation, while grant-free access creates collision, interference, and detection challenges under short-packet operation [1]. These access-layer constraints directly impact decoding reliability because short packets carry insufficient redundancy to tolerate unresolved collisions or channel-estimation errors.

Energy efficiency further complicates URLLC design in UAV-assisted and IoT-dense networks. In cognitive UAV-aided URLLC and mMTC systems, finite blocklength effects significantly degrade the achievable throughput predicted by Shannon capacity, and imperfect successive interference cancellation (SIC) significantly increases the error probability due to error propagation [8]. Even when sophisticated power-allocation strategies yield large improvements in average performance, achieving strict URLLC deadlines under interference remains fundamentally difficult [12].

Collectively, these results lead to a critical conclusion: URLLC is not merely a demanding service class; it is a fundamentally new physical-layer operating regime in which asymptotic coding theory, iterative decoding, and retransmission-based reliability mechanisms lose their validity. Instead, URLLC requires decoding strategies that can deliver near-ML reliability at extremely short blocklengths with deterministic latency and bounded complexity. This conclusion forms the direct motivation for the noise-centric and constrained decoding frameworks developed in this dissertation.

In the short-blocklength regime relevant to URLLC, finite-blocklength effects become fundamental to reliability analysis. In particular, channel dispersion captures the variability of the information conveyed over finite-length transmissions and introduces a non-negligible penalty relative to the asymptotic capacity limit [13]. Consequently, achievable rates depend not only on channel capacity but also on blocklength, reliability requirements, and latency constraints.

From a broader theoretical perspective, [3] argues that non-asymptotic analysis must become a central core of 6G communication theory. Almost all machine-type applications operate under finite blocklength, time-varying channels, and non-ergodic conditions that invalidate long-term averaging assumptions. Consequently, performance must be evaluated using: 1) outage-limited metrics, 2) dispersion-aware bounds, 3) short-horizon reliability guarantees, 4) task-oriented information measures.

Codes must now be optimized not for asymptotic distance growth but for finite-length distance spectra (i.e., the distribution of the codeword weights), error-floor behavior, and decoder determinism. Decoders must deliver predictable latency and bounded complexity rather than

average-case efficiency. The practical impact of finite blocklength theory is further illustrated in UAV-assisted URLLC systems. Finite blocklength-based throughput expressions more accurately capture the degradation induced by short packets, imperfect SIC, and fast-varying aerial channels. Even when sophisticated joint power-allocation strategies drastically improve energy efficiency, meeting strict URLLC deadlines under interference remains a strong challenge [12].

The lesson from these works is that finite-blocklength information theory is not merely a refinement of Shannon theory for URLLC, it is the only valid theoretical framework for the design of 6G mission-critical communication systems. This realization directly motivates the development of near-ML decoding strategies with bounded complexity, short-code-optimized decoding frameworks, and noise-centric universal decoders, all of which constitute the methodological foundation of this dissertation.

1.1.3 Limitations of Short-Block Coding and of Traditional Decoders

In this regime, the performance of coding schemes is no longer governed by asymptotic distance spectra but by finite-length effects, structural weaknesses of code graphs, and the computational limits of practical decoders.

Paper [14] provides a system-level diagnoses of these difficulties. It highlights that while contemporary systems such as 5G New Radio and DVB-S2/S2X [15] rely on powerful coding families including LDPC and polar codes, their near capacity behavior is achieved only under sufficiently long blocklengths. When packet sizes shrink due to stringent latency, signaling, and scheduling constraints, several detrimental effects emerge simultaneously: 1) reduction in coding gain due to fewer parity constraints, 2) a larger number of short cycles in LDPC Tanner graphs, which destroy the independence assumptions underlying belief-propagation, 3) error propagation in successive-cancellation-based polar decoding, which becomes increasingly destructive as blocklength decreases, and 4) amplification of burst errors in mmWave and satellite channels, where short packets encounter severe fading outages. As a result, the gap between theoretical capacity and practical performance significantly widens in the finite-length regime. Paper [14] explicitly demonstrates that, for short blocklengths, neither LDPC nor polar codes operate close to their Shannon limits, even under optimistic decoding assumptions.

From a decoder complexity standpoint, [14] also provides a detailed breakdown of the computational burden associated with short-block LDPC and polar decoding. For LDPC codes, belief-propagation decoding, whether via the sum-product algorithm (SPA) or its approximations (min-sum (MS), normalized min-sum (NMS), and offset min-sum (OMS)), incurs a complexity that scales with: 1) the density of the parity-check matrix, 2) the number of iterations required for convergence, and 3) the degree distribution and girth of the code graph. In short-block scenarios, reduced girth (i.e., the presence of short cycles in the Tanner graph) and dense connectivity (i.e., higher node degrees and increased edge density in the parity-check matrix) yield

highly non-linear convergence behavior, forcing either a large number of iterations (violating latency constraints) or premature termination with elevated error floors (violating reliability requirements).

For polar codes, deterministic channel polarization enables low complexity encoding and successive cancellation (SC) decoding. However, SC decoding suffers from severe sequential error propagation, particularly at short blocklengths. Enhanced schemes such as successive-cancellation list (SCL) and CRC-aided SCL (CA-SCL) decoding dramatically improve error performance but at the cost of exponentially growing list-induced complexity and decoding latency [14]. Simulation results reported in [14] indicate that polar codes can achieve a performance gain of approximately 0.3–0.8 dB over LDPC codes for blocklengths between 256 and 1024 bits. This gain is obtained only by increasing the list size, which directly undermines deterministic latency guarantees.

These observations are fully consistent with the large-scale URLLC benchmarking results reported in short blocklength codes for ultra-reliable low-latency communications [10]. Under optimal maximum-likelihood decoding, extended BCH codes and tail-biting convolutional codes exhibit the best finite-length performance, often approaching the normal-approximation benchmark. However, these gains are obtained at the cost of prohibitively high decoding complexity, rendering such decoders infeasible for real-time URLLC operation.

More practical decoding schemes, such as CA-SCL for polar codes and layered OMS decoding for LDPC codes, achieve acceptable performance only at moderate BLER targets (10^{-4}). At the extreme URLLC regime of $10^{-7} - 10^{-8}$, a persistent and irreducible gap remains between theoretical ML performance and feasible decoder realizations [10].

This performance complexity gap represents one of the most severe unresolved challenges in 6G physical-layer design. The strategic outlook of channel coding toward 6G reinforces this conclusion. In [16], next-generation systems are expected to satisfy several stringent performance requirements, summarized in Table 1.1.

Table 1.1: Key performance requirements for next-generation URLLC systems

Requirement	Target Value
End-to-end latency	$< 100 \mu s$
Reliability	$\geq 99.99999\%$
Throughput	Multi-Gbps to Tbps
Energy efficiency	Strict constraints across heterogeneous devices

Under these conditions, existing 5G coding solutions, LDPC for data channels and polar codes for control channels, are not intrinsically designed for the joint optimization of ultra-reliability, ultra-low latency, and finite decoding complexity. Although they offer excellent

throughput and asymptotic efficiency, they do not scale favorably into the short-packet, ultra-reliable regime with bounded and predictable decoding latency and complexity required by 6G mission-critical services [16].

Taken together, the results of [10], [14], [16] reveal a consistent and deeply structural problem. Existing channel coding paradigms, even in their most advanced 5G forms, are not fundamentally suited for the ultra-short, ultra-reliable, and ultra-low-latency regime of 6G machine-type communication.

This realization forms the intellectual bridge from traditional short-block coding theory to the universal and noise-driven decoding frameworks that underpin the guessing random additive noise decoding (GRAND) family of algorithms and the constrained ML decoding techniques developed in this dissertation.

While short blocklength channel coding theory reveals the fundamental limits of reliability under latency constraints, it is the computational complexity and determinism of decoding algorithms that often decide whether these limits can be approached in practice. In classical broadband systems, decoding latency and complexity are secondary concerns compared to throughput and spectral efficiency. In URLLC, however, decoding becomes a real-time control operation, where variability of execution time directly translates into violations of latency guarantees. As a result, the decoding process itself emerges as one of the most critical components of the 6G physical layer [10], [16], [17]. The central difficulty lies in the fact that reliability and complexity are intrinsically coupled. To achieve ultra-low block error rates at short blocklengths, decoders must operate near-ML performance. However, ML decoding is, in general, an NP-hard problem whose worst-case complexity grows exponentially with blocklength. Every practical decoding algorithm, therefore, represents a compromise between: 1) proximity to ML performance, 2) predictability of execution time, 3) energy efficiency, and 4) hardware implementability. In URLLC systems, this trade-off becomes extreme because:

- blocklengths are short (often < 200 bits),
- target BLERs are extremely low ($10^{-5} - 10^{-7}$),
- latency budgets are deterministic and tight (< 1 ms),
- energy constraints apply across heterogeneous devices [10], [2].

Under these constraints, even modest increases in average decoding iterations or list sizes can exceed the system's latency budget.

1- Limitations of Iterative Decoding at Short Blocklengths

Iterative message-passing decoders such as belief-propagation for LDPC codes and Turbo decoding have been extraordinarily successful in broadband systems. Their performance relies

on the statistical independence of messages exchanged along sparse Tanner graphs over many iterations. However, as shown in [14] and reinforced in [16], short-block LDPC codes violate the core assumptions that underpin iterative decoding. At short blocklengths, Tanner graphs necessarily contain a large number of short cycles, message passing independence collapses, convergence becomes highly irregular, and error floors emerge at moderate to high SNR. To counteract these effects, the decoder must: 1) increase the number of iterations, 2) perform layered or flooding schedules, 3) use normalized or offset min-sum refinements. Each of these remedies directly increases decoding latency and power consumption [14]. More critically, the required number of iterations becomes highly variable across channel realizations, making deterministic latency guarantees impossible. Moreover, the computational burden of check-node and variable-node updates scales with the density of the parity-check matrix and the number of edges in the Tanner graph. Even for moderate blocklengths (e.g., 128–256 bits), iterative decoders can require thousands of arithmetic operations per iteration. This creates a severe mismatch with the energy and hardware constraints of battery-limited IoT and embedded URLLC devices [16].

2- List Decoding of Polar Codes and Latency Explosion

Polar codes represent another basis of modern cellular coding. Their theoretical appeal lies in their capacity-achieving property under successive-cancellation decoding in the limit of infinite blocklength. In practice, however, URLLC performance is achieved only through SCL decoding, often aided by CRC concatenation [10], [14]. SCL decoding improves reliability by maintaining multiple candidate decoding paths and selecting the most likely at the end of the decoding process. More importantly, SCL decoding latency is inherently variable:

- the number of surviving paths fluctuates with noise realization.
- path metric updates involve frequent memory sorting and pruning.
- hardware pipelines stall under worst-case path proliferation.

As a result, while CRC-aided SCL (CA-SCL) decoders deliver excellent error performance at short blocklengths, they cannot guarantee deterministic execution time under all conditions, a property that is absolutely mandatory for URLLC control loops [17].

3- Performance-Complexity Tradeoffs in URLLC

The benchmarking results in [10] reveal a striking phenomenon, as BLER targets move from 10^{-3} to 10^{-5} and below, the decoding complexity required to approach ML performance increases sharply. For example:

- LDPC decoders require disproportionately more iterations to suppress rare error events.
- Polar CA-SCL decoders require large list sizes to avoid catastrophic early path decisions.

- BCH and TB-CC decoders require near exhaustive search to reach ML-like performance.

This phenomenon represents a performance complexity wall: beyond a certain reliability threshold, incremental increases in decoding effort yield diminishing reliability gains but explosive increases in latency and energy consumption. This wall is not an implementation artifact; it is a direct consequence of operating in the finite-blocklength extreme reliability regime [10], [14], [16].

4- Decoder Universality Versus Code-Specific Architectures

A final and deeply structural limitation of classical decoding frameworks is their tight coupling to specific code families. LDPC decoders are tailored to specific parity-check matrices; Turbo decoders rely on particular convolutional structures; polar decoders require recursively defined generator matrices. This code-decoder dependence becomes a critical drawback in 6G, where:

- multiple short-block codes may coexist across services,
- grant-free access introduces dynamically varying code ensembles,
- non-terrestrial networks, UAVs, and industrial subsystems impose heterogeneous waveform and coding requirements [17].

In such environments, the need to deploy multiple, code-specific decoders increases: hardware redundancy, configuration overhead, reconfiguration latency, system fragility.

As argued forcefully in [17], decoder universality is no longer optional; it is foundational. The paper demonstrates that classical decoders fail not merely due to insufficient coding gain, but due to their structural rigidity and dependence on code algebra. The comparative benchmarking of approximate exhaustive decoding (AED), GRAND, ordered statistics decoding (OSD), bit-flip decoders, and belief-propagation variants in [17] confirms that universal, noise-centric, or automorphism-based decoders consistently outperform code-specific iterative decoders in the short-packet, ultra-reliable regime. More importantly, they offer: greater adaptability across code families, more predictable execution profiles, superior suitability for heterogeneous URLLC environments.

5- Fundamental Implication for 6G Physical-Layer Design

The main message of [10], [14], [16], and [17] is that in URLLC the dominant obstacle to achieving theoretical reliability is not the lack of good codes, but the lack of decoders that can approach ML performance with deterministic latency and bounded complexity. This insight shifts the core of 6G physical-layer design from coding theory to decoding architecture and algorithmic structure. It implies that: i) further incremental improvements to LDPC and polar code constructions will yield diminishing returns, and ii) breakthroughs must instead emerge

from new decoding paradigms that are: universal, noise-driven, near-ML in performance, and energy-efficient in hardware.

This realization provides the precise conceptual entry point for the noise-centric universal decoding philosophy embodied by the GRAND family of algorithms, which forms the central methodological backbone of this dissertation.

1.1.4 Massive MIMO for Spatial Multiplexing

In classical broadband wireless communication, reliability is typically achieved through a combination of time diversity, frequency diversity, and coding across large blocklengths. However, URLLC explicitly prohibits the use of time-domain diversity via retransmissions and long interleaving due to hard latency constraints. As a result, spatial diversity provided by multi-antenna systems becomes the dominant reliability resource for 6G mission-critical machine-type communication [13]. Massive MIMO systems can simultaneously support high spatial multiplexing gains and significant reliability improvements through array gain and spatial combining. In URLLC scenarios, the system design prioritizes the use of spatial degrees of freedom to enhance reliability rather than to maximize throughput.

The fundamental principles of massive MIMO are established in [18]. As the number of base-station antennas grows large, two key asymptotic phenomena may arise under favorable propagation conditions:

- 1) **Channel hardening:** the effective channel gains concentrate sharply around their mean values due to the law of large numbers, transforming a highly random fading channel into a quasi-deterministic communication link.
- 2) **Favourable propagation:** user channel vectors become asymptotically orthogonal, enabling strong inter-user interference suppression using simple linear processing.

While both effects are enabled by large antenna arrays, they are distinct phenomena and need not occur simultaneously. Together, these effects allow mMIMO systems to achieve: highly predictable signal-to-interference-plus-noise-ratio (SINR), reduced dependence on power control, strong robustness to small-scale fading, and simultaneous support of many users with high reliability [18]. These properties are particularly attractive for URLLC, where deterministic latency and reliability must replace stochastic performance guarantees. Channel hardening directly aligns with the URLLC requirement that system behaviour be dominated by worst-case bounds rather than average-case statistics.

Another critical advantage of mMIMO is its extreme power-scaling capability. By exploiting array gain, user transmit power can theoretically scale inversely with the number of base-station

antennas ($1/N$) under perfect channel state information (CSI), and inversely with the square root of the number of antennas ($1/\sqrt{N}$) under imperfect CSI [18]. This makes mMIMO a physical-layer enabler of energy-neutral mMTC and battery-limited URLLC devices, including dense IoT deployments and UAV-assisted communication.

However, the classical benefits of mMIMO were derived primarily under asymptotic capacity analysis, assuming infinite blocklengths, perfect channel state information, ergodic fading, and average-rate optimization.

These assumptions are not present in the URLLC regime. This limitation is addressed in [13], which provides one of the most definitive treatments of what mMIMO actually delivers under the operational constraints of URLLC.

1- Massive MIMO Under Finite Blocklength Constraints

Paper [13] shows that, in URLLC, massive MIMO primarily serves as a source of spatial reliability rather than throughput enhancement. Since time- and frequency-domain diversity are severely constrained by sub-millisecond latency budgets, spatial processing becomes the dominant mechanism to achieve ultra-high reliability. Using a finite-blocklength information-theoretic framework, the authors characterize the achievable block error probability in the uplink as a function of the number of base-station antennas, the quality of channel state information, pilot contamination, and the adopted linear combining scheme. In particular, maximum-ratio combining (MRC) and minimum mean-squared error (MMSE) are analyzed, and it is shown that only MMSE processing enables the error probability to vanish asymptotically with the number of antennas, whereas MRC does not suffice. Zero-forcing (ZF) is also a linear detection technique that exploits the spatial degrees of freedom in mMIMO systems to cancel interference and decouple the multi-antenna channel into approximately independent parallel links.

Critically, the study reveals that pilot contamination becomes more destructive under finite-blocklength operation than under asymptotic models. In classical mMIMO theory, pilot contamination limits the achievable SINR but does not fundamentally destroy capacity growth. Under finite blocklength, however, pilot contamination inflates the dispersion of the effective channel and drives the error probability into regimes that violate URLLC reliability targets by orders of magnitude [13].

As it is well-known in wireless communication:

- MRC amplifies both desired signal and interference, causing residual interference and pilot contamination to dominate the error probability floor.
- ZF cancels multiuser interference by projecting the received signal onto the null space of interferers; however, under imperfect CSI and finite blocklength, noise enhancement prevents the error probability from vanishing asymptotically.

- MMSE detection actively suppresses both noise and interference, restoring the property that the block error probability vanishes as $N \rightarrow \infty$, even under imperfect CSI [13], [19].

This result has implications for 6G URLLC design: MMSE processing is not an optional performance enhancement for URLLC; it is a necessity.

The collective insights of [13], [18], [19] lead to several decisive conclusions:

- 1) Spatial diversity is the dominant reliability resource for URLLC, replacing time and frequency diversity.
- 2) Finite-blocklength effects fundamentally change mMIMO behaviour, invalidating classical capacity-based insight.
- 3) Pilot contamination and imperfect CSI become first-order URLLC impairments, as channel estimation errors directly increase the dispersion of the effective channel and dominate the block error probability under finite blocklength constraints.
- 4) MMSE detection is mandatory for URLLC-grade reliability.

Most importantly, these conclusions reveal that mMIMO alone cannot guarantee URLLC reliability. Instead, it must be tightly integrated with:

- finite-blocklength coding,
- near-maximum-likelihood (near-ML) detection and decoding strategies that are robust to rare noise realizations,
- deterministic low-latency digital processing chains.

This observation completes the theoretical justification for shifting part of the URLLC reliability burden from channel coding and spatial processing to universal, noise-driven decoding architectures, which is precisely the role played by the GRAND-based decoding frameworks developed in this dissertation.

1.1.5 The Limitations of Code-Centric Decoders for Short Blocklengths

Digital communication theory has historically approached error correction through a code-centric decoding paradigm: one first designs an algebraic code with desirable distance properties, and then constructs a decoding algorithm specialized to exploit the structure of that particular code. This approach has proven extraordinarily successful in long-blocklength broadband systems, where iterative message passing, trellis-based decoding, and list decoding can operate over thousands of symbols with manageable complexity and near-capacity performance. However,

as the preceding sections have demonstrated, the operational regime of 6G URLLC violates the fundamental assumptions that enable the effectiveness of code-centric decoding. Short blocklengths, extreme reliability targets, deterministic latency constraints, massive connectivity, and stringent energy limitations collectively create a setting in which code-centric decoding becomes structurally ill-suited, not merely suboptimal [10], [14], [16], [17]. The failure of classical decoders in this regime is not a consequence of poor engineering, but a fundamental manifestation of the mismatch between asymptotic coding principles and finite, ultra-reliable operation.

1- Dependence on Long-Block Asymptotics

Most modern decoding algorithms, belief-propagation for LDPC codes, iterative decoding of Turbo codes, and successive cancellation for polar codes derive their theoretical guarantees from asymptotic behaviour as blocklength tends to infinity. In this limit:

- Tanner graphs become locally tree-like,
- message independence assumptions hold approximately,
- law of large numbers effects suppress statistical fluctuations,
- error probabilities decay exponentially with blocklength as predicted by Shannon’s theory.

As a result, the theoretical justification for message-passing, list-based, and trellis-based decoders evaporates. These algorithms continue to operate, but their convergence behaviour becomes unpredictable, their error floors rise sharply, and their worst-case latencies become unbounded [10], [14].

2- Problem of Traditional Coding of Short Codes

At moderate target BLERs (e.g., 10^{-3}), many classical decoders perform adequately at short blocklengths. However, URLLC pushes systems into error-probability regimes as low as 10^{-5} – 10^{-7} , where decoding failures are dominated by rare noise events, and this creates a fundamental difficulty, including for our methods. In this extreme reliability regime:

- iterative decoders require disproportionately more iterations to suppress rare errors,
- list decoders require exponentially larger candidate sets,
- trellis-based decoders require large constraint lengths to achieve extremely low error probabilities at short blocklengths, leading to exponential growth in state-space complexity.

Paper [10] explicitly demonstrates this phenomenon in its comprehensive benchmarking study: for all practical LDPC and polar decoding configurations, the complexity required to push BLER below 10^{-5} rises superlinearly, while the performance gap to ML decoding remains

stubbornly non-zero. This behaviour exposes a fundamental structural fragility: code-centric decoders are optimized for typical noise realizations, not for the tails of the noise distribution that dominate ultra-reliable operation. In URLLC, decoding failure is governed by the extreme value behaviour of the noise rather than its average statistics. Classical decoders are not inherently designed to handle this regime efficiently.

3- Latency Variability

URLLC systems require deterministic worst-case latency guarantees, not just favourable average latency. However, even when classical decoders exhibit acceptable average complexity, their worst-case execution time is unbounded or prohibitively large:

- LDPC belief-propagation decoders exhibit iteration counts that fluctuate widely with channel realization,
- polar CRC-aided successive cancellation list (CA-SCL) decoders show variable path growth and pruning behaviour,
- turbo decoders require unpredictable numbers of iterations to converge.

As shown in [17], this variability is not an incidental implementation artifact; it is intrinsic to the stochastic nature of iterative and list-based decoding. In a real-time control system, a single outlier decoding event that exceeds the latency budget is sufficient to destabilize the entire control loop, regardless of how rare such events may be. Thus, from the perspective of URLLC system safety, average-case decoding complexity is irrelevant; only worst-case complexity matters. Classical code-centric decoders are fundamentally incompatible with this requirement.

4- Inherent Rigidity of Code-Specific Architectures

Perhaps the most profound limitation of code-centric decoding lies in its structural rigidity. Classical decoders are designed for a specific algebraic structure, optimized for a single code family, and fine-tuned for specific blocklengths, rates, and parity-check matrices.

In 6G machine-type networks, multiple short-block codes may coexist dynamically, grant-free access may induce unknown or rapidly changing code ensembles, heterogeneous devices may employ different coding strategies depending on energy, latency, and reliability requirements, and non-terrestrial and aerial networks introduce further diversity in waveform and coding formats [2], [17].

Deploying a separate, code-specific decoder for each of these possibilities is not only infeasible from a hardware perspective but also incompatible with the reconfigurability and scalability demands of 6G. The paper [17] demonstrates that this rigidity alone disqualifies classical decoding architectures from serving as the universal physical-layer backbone of future machine-type networks.

5- The Mismatch Between Code-Centric Decoding and URLLC

The weaknesses of classical code-centric decoders in URLLC can be summarized as follows:

- 1) Dependence on long-block asymptotics, which collapse in the finite-blocklength regime.
- 2) Reliance on average-case performance metrics and stochastic latency, which is inadequate for ultra-reliable communication, where rare noise realizations can dominate the block error probability, and latency must satisfy deterministic URLLC deadlines.
- 3) Severe energy inefficiency and unpredictability, incompatible with battery-limited platforms.
- 4) Rigid dependence on specific code sizes, incompatible with heterogeneous and reconfigurable 6G environments.

These limitations are not marginal engineering issues; they are fundamental mismatches between the design philosophy of classical decoding and the operational realities of 6G URLLC.

6- The Noise-Centric Decoding Paradigm

This noise-centric decoding philosophy reverses the traditional logic: instead of searching for the codeword that best fits the received signal, one searches for the most likely noise realization that could have produced the received word from some valid codeword. Such a transformation aligns the decoding process directly with the maximum-likelihood principle, because ML decoding is fundamentally equivalent to finding the most probable noise vector consistent with a valid codeword.

This noise-centric viewpoint possesses several properties that classical code-centric decoding lacks:

- universality across all linear block codes, because it only requires a code membership test,
- natural compatibility with short blocklengths, where rare noise patterns dominate.

As formally demonstrated in [20], [21], [17], decoders based on this paradigm, especially GRAND, can achieve ML decoding performance across a wide range of codes with substantially lower and more predictable complexity than classical code-centric decoders in the short-block URLLC regime.

7- A New Decoding Paradigm

The analysis above shows that the traditional, code-centric decoding paradigm is misaligned with the finite-blocklength, ultra-reliable, and deterministic-latency requirements of 6G URLLC. This observation is reinforced by recent work on noise-centric decoding, in particular the GRAND framework introduced in [20]. In contrast to conventional iterative or list-based

decoders, GRAND formulates maximum-likelihood decoding as an ordered search over channel error patterns rather than over codeword structure. This simple but fundamental change of viewpoint leads to a universal decoding strategy applicable to arbitrary linear block codes, including short LDPC, polar, BCH, and RLCs [20], [17].

The performance and complexity results reported in [20] and extended in [17] demonstrate that, in the short-blocklength regime of interest for URLLC, GRAND and its variants can achieve frame error rates extremely close to maximum-likelihood decoding while maintaining an average complexity that decreases rapidly with signal-to-noise ratio. Moreover, soft-input variants such as symbol-reliability GRAND (SRGRAND) [22] and ordered reliability bits GRAND (ORBGRAND) [23] exploit soft channel information to further reduce the number of required noise guesses and thus the decoding effort. These properties directly address several of the key shortcomings identified for classical decoders.

A first book on GRAND appeared in 2023 [21], providing an overview of the first noise-guessing algorithms and their variants, including guessing random additive noise decoding with abandonment (GRANDAB), ORBGRAND, and SRGRAND. The book also provides an account of the combination of GRAND with MIMO, as well as symbol-level GRAND, both of which are techniques explored in the present dissertation. That work describes hardware architectures and shows that GRAND-based decoders rely on simple and repetitive processing steps that can take advantage of parallelization for efficient test-error-pattern (TEP) generation and testing. It is shown that such decoding techniques can have bounded worst-case decoding latency, making them well-suited for URLLC deployment.

A more recent book on GRAND appeared in 2026 [24], offering a more comprehensive and updated overview of the algorithms underpinning noise-guessing decoding, and also focusing on hardware implementation aspects. The implementation aspects are, in fact, very relevant, and some noise-guessing techniques have been developed in tandem with their hardware implementation [25].

Taken together, [20], [17], [22], [23], and [21] clearly indicate that the most promising path forward for 6G URLLC decoding is to abandon purely code-centric designs and adopt noise-centric, universal decoding strategies with carefully controlled complexity. This dissertation is positioned precisely along this direction: it adopts GRAND-style noise-centric decoding and develops constrained, implementation-oriented variants tailored to RLCs and short LDPC codes.

1.1.6 Universal Decoding

1- Universality in the GRAND Framework

Decoder universality is a central property of the GRAND family of algorithms. In [20] it is shown that GRAND can achieve channel capacity when used with random codebooks, and

that its operation depends only on the ability to test whether a candidate vector belongs to the codebook. This test can be performed by a simple syndrome check for linear block codes. As a result, GRAND is inherently agnostic to the internal algebraic structure of the code and can be applied without modification to BCH, LDPC, polar, CRC-aided, and RLCs [20], [17]. The survey in [17] extends this observation and evaluates multiple universal decoding techniques, including GRAND variants and ordered-statistics-based decoders, across a wide range of short blocklength codes and operating conditions relevant to URLLC. The results confirm that noise-centric universal decoders, and in particular GRAND and its soft-input variants, consistently provide near-maximum-likelihood performance across different code families without code-specific algorithm redesign.

2- Universality Under Finite Blocklength and URLLC Constraints

In the short blocklength regime, universality must be achieved without sacrificing reliability or violating strict latency. Numerical results in [20], [17] show that:

- GRAND variants achieve frame error rates close to ML decoding for short BCH, LDPC, and polar codes;
- soft-input variants such as SRGRAND and ORBGRAND approach or match OSD with significantly reduced complexity [22], [23];
- the average number of codebook membership queries decreases sharply as SNR increases, which is precisely the regime in which URLLC typically operates [20], [21].

Because the noise-guessing order in GRAND-type decoders is deterministic and parametrizable (e.g., through the abandonment threshold in GRANDAB or ORBGRAND), system designers can impose explicit bounds on worst-case complexity and latency. This property aligns directly with URLLC requirements and is difficult to achieve with classical iterative decoders.

3- Universal Decoders in 6G

From a 6G system perspective, decoder universality provides several critical advantages:

- 1) A single decoding algorithm can be reused across multiple services and code families, simplifying base-station and edge-hardware design.
- 2) New codes or short-block configurations can be adopted without redesigning the decoder core, which is essential in evolving 6G deployments.

In this dissertation, decoder universality is adopted as a core design principle. GRAND-type algorithms, particularly ORBGRAND and SRGRAND-inspired constrained variants, are used as universal decoding engines for both structured short LDPC codes and for RLCs. This positions universal, noise-centric decoding as a cornerstone of the proposed 6G URLLC physical-layer architecture.

1.1.7 Noise-Centric Decoding Paradigm

1- Maximum-Likelihood Decoding via Noise Guessing

For memoryless channels and linear block codes, ML decoding can be interpreted as the search for the most probable noise vector that transforms a valid codeword into the received word. GRAND formalizes this interpretation by explicitly ordering and testing candidate noise patterns instead of navigating the structure of the code [20]. Given a received vector $\mathbf{y}$, GRAND generates a sequence of test error patterns $\mathbf{e}$ in decreasing probability order, forms $\mathbf{y} \oplus \mathbf{e}$, and checks whether the result belongs to the code. The first pattern that passes the membership test corresponds to the ML solution. This formulation is conceptually simple but powerful: it separates the noise-search process from the code structure, which is used only to perform a membership test. GRAND is therefore naturally universal across linear block codes [20], [17].

2- Performance of GRAND in the Short-Blocklength Regime

The ML optimality of GRAND for random codebooks is proved in [20], where it is shown that GRAND is capacity-achieving when used with random linear codes in the limit of large blocklengths. More importantly for this dissertation, simulation results in [20], [17] demonstrate that, at short blocklengths relevant for URLLC, GRAND and its variants closely match the performance of exhaustive ML decoding and high-order OSD, i.e., OSD implementations that test high-order combinations of the least reliable bits and therefore approach ML performance at the cost of rapidly increasing complexity, for Bose-Chaudhuri-Hocquenghem (BCH), polar, and CRC-aided codes. In particular:

- In [20] hard-input GRANDAB already outperforms classical algebraic decoders for BCH codes at moderate SNR;
- ORBGRAND and SRGRAND reach near-ML frame error rates comparable to OSD of order 2, but with markedly lower average complexity [22], [23];
- the performance advantage is especially pronounced in the ultra-reliable regime, where iterative decoders exhibit error floors and increasing iteration counts [17].

3- Complexity and Latency Behaviour of GRAND

A key metric for GRAND-type decoders is the number of codebook membership queries per decoded block. Each query consists of applying a test error pattern to the received word and performing a syndrome check for membership. In [21] provides a detailed analysis of both worst-case and average complexity for several GRAND variants. It is shown that: 1) worst-case complexity can be bounded by parameters such as the maximum Hamming weight in GRANDAB or in ORBGRAND; 2) average complexity decreases rapidly with SNR because typical noise realizations are corrected using only a few of the lowest-weight patterns; 3) soft-input variants

such as ORBGRAND and SRGRAND exhibit substantially lower average complexity than hard-input GRANDAB at the same error rate [22], [23], [21].

4- GRAND in the Context of Universal Decoding

The comparative study in [17] shows that GRAND-based decoders provide a compelling balance between universality, near-ML performance, and controlled complexity when compared with other universal or semi-universal decoders such as OSD. While OSD can reach ML performance, its combinatorial growth in complexity with reprocessing order makes it less suitable for real-time short-packet URLLC. GRAND variants, especially ORBGRAND [23] and SRGRAND [22], achieve similar reliability with more favourable and better structured complexity scaling.

Moreover, the modular architecture described in [21] confirms that GRAND can serve as a practical universal decoding core in 6G systems, able to support both structured and unstructured codes through simple membership tests.

1.1.8 Random Access and Grant-Free Communication in 6G

While much of classical communication theory focuses on the reliable transmission of payload data, 6G machine-type communication is increasingly dominated by an often-overlooked but fundamentally limiting factor: metadata and access signalling. In dense IoT and mission-critical control systems, the information required to coordinate communication itself, including access requests, scheduling grants, pilot assignments, acknowledgments, and synchronization, can exceed the size of the payload data by orders of magnitude [1]. As a result, random access is no longer a higher-layer concern alone but directly impacts physical-layer efficiency and latency in 6G systems. In conventional LTE and 5G NR systems, access is predominantly grant-based, relying on multi-step handshake procedures that include: random access preamble transmission, random access response, scheduled uplink transmission, and contention resolution.

Even under ideal conditions, this control-plane exchange introduces latencies on the order of several milliseconds, rendering it fundamentally incompatible with the sub-millisecond deadlines of URLLC [1]. Grant-based access also scales poorly in dense mMTC environments because the probability of preamble collision grows rapidly with the number of simultaneously active devices.

To resolve these limitations, 6G must adopt grant-free random access as a default operating mode for both URLLC and mMTC [1], [2]. In grant-free random access, devices transmit their data packets immediately without waiting for scheduling grants. This eliminates handshake latency but introduces a new and far more severe physical-layer challenge: collision resolution and multiuser detection under extreme uncertainty, short packets, and massive concurrency.

1- Unsourced and Massive Random Access

The unsourced random access paradigm, highlighted in [1], fundamentally redefines user identification and data association. Instead of decoding both message and user identity, the receiver decodes only a set of transmitted messages, with user identity resolved at higher protocol layers. This abstraction is particularly well-suited for IoT systems in which a very large population of low-power devices sporadically transmit short updates. Unsourced random access systems are typically modeled as: a sparse activity pattern, a very large number of potential users, extremely short transmitted messages, and strict latency constraints.

Compressed-sensing-based activity detection, covariance-based detection in mMIMO, and joint detection–decoding algorithms have been proposed as candidate solutions [1]. However, these methods critically depend on accurate noise modeling, efficient sparse recovery, and robust decoding in the presence of strong multi-user interference.

When short packets and finite blocklength constraints are imposed, these challenges intensify dramatically. The results in [1] demonstrate that joint activity detection and data decoding must be performed under severe dispersion and overhead constraints, where even a few extra control bits can drastically reduce access efficiency.

2- Multi-user Detection Under Finite Blocklength and URLLC Constraints

In grant-free access, multiple users transmit simultaneously without coordination. The base station must therefore solve a joint multiuser detection and decoding problem under unknown user activity, short packets, severe interference, finite blocklength effects, and ultra-low-latency requirements.

This problem is more difficult than traditional single-user decoding. The receiver must determine: i) which users are active, ii) which signals belong to which users, and iii) what data each user transmitted, all within microsecond-scale deadlines. Paper [1] shows that classical matched filtering and interference cancellation are insufficient in ultra-dense access scenarios. Even small levels of channel estimation error or residual interference can greatly impact short packets. Similarly, iterative multi-user detection algorithms such as expectation maximization or belief-propagation-based detectors suffer from unpredictable convergence times, making them unsuitable for deterministic URLLC operation. Finite blocklength theory further exacerbates this difficulty. As shown in [5] and reinforced by [13], short packets dramatically reduce the reliability margin available for collision resolution and interference suppression. With little redundancy in each packet, even a single unresolved collision or detection error can cause catastrophic packet loss.

Once again, the limitations of classical iterative, correlation-based, and interference cancellation–based random-access schemes become evident in short-packet scenarios. In the grant-free random-access setting considered in this work, the fundamental resolution to the collision problem is to redesign the signalling itself so that collisions are resolvable at the physical layer. This requirement leads naturally to the use of uniquely decodable signature sequences. By assigning each terminal a Bose–Chowla–Lindström (BCL) signature sequence with the unique-sum

property, the superposition of a bounded number of simultaneously active users produces a sum that uniquely identifies the corresponding active set. In other words, the collision pattern is encoded combinatorially in the algebraic structure of the sum itself. This signature-based formulation fundamentally shifts grant-free random access from a probabilistic collision-avoidance problem to a structurally resolvable multi-user identification problem, thereby enabling reliable short-packet random access under the stringent latency and reliability constraints of 6G.

1.1.9 Research Objectives and Contributions

The previous sections highlighted the stringent requirements of sixth-generation (6G) URLLC and the limitations of existing channel coding and decoding techniques when operating under short-blocklength constraints. Although modern coding schemes can achieve near-capacity performance, their decoding complexity and latency remain significant obstacles to meeting the demanding requirements of future wireless systems. These challenges become even more pronounced in mMIMO systems and grant-free machine-type communications, where reliability, scalability, and computational efficiency must be achieved simultaneously. The main objective of this dissertation is to develop signal processing and channel decoding techniques that enable reliable and low-latency communications for future 6G wireless systems while maintaining practical computational complexity. In particular, this work investigates how noise-centric decoding strategies can be adapted and accelerated to support short-blocklength communications, massive MIMO scenarios, and grant-free random access.

These objectives are addressed through four main technical contributions. First, this dissertation demonstrates the feasibility of integrating GRAND with coded massive MIMO systems employing random linear codes to achieve low-latency decoding. Second, it proposes symbol-level constrained GRAND together with antenna sorting techniques that significantly reduce decoding complexity while maintaining ML-equivalent performance. Third, it develops constrained GRAND algorithms for short LDPC codes, eliminating the vast majority of infeasible candidate error patterns and enabling practical ML-level decoding. Finally, it investigates BCL-based uniquely decodable signature sequences for reliable user identification in grant-free random access systems over the noisy adder channel.

The following section provides the technical background required to understand these contributions by reviewing the coding schemes, decoding algorithms, communication models, and mathematical concepts used throughout this dissertation.

1.2 Technical Foundation

This section provides the theoretical background needed for understanding the articles that describe the proposals made in this dissertation. The section covers RLCs, LDPC codes, mMIMO models, QAM modulation, noise-guessing decoding (GRAND), and uniquely decodable signature constructions such as Lindström and BCL sequences. This chapter establishes the mathematical tools used throughout the thesis.

1.2.1 Random Linear Codes

A *random linear codes* (RLC) is a fundamental error-correcting code used in digital communications and data storage to enhance reliability by detecting and correcting errors that occur during transmission or storage. These codes operate on fixed-length sequences known as *codewords*, where each element is randomly selected from an alphabet of size q . The set of elements from which the codewords are formed corresponds to a *finite field*, also known as a *Galois Field*, denoted as $\text{GF}(q)$. In the special case where $q = 2$, the code is referred to as a *binary random linear code*, and its codewords consist of *binary digits (bits)*. The field $\text{GF}(2)$ contains only two elements, 0 and 1, and all arithmetic operations within this field adhere to the rules of binary arithmetic [26].

A *binary block code* consists of codewords of length n , meaning each contains n elements from $\text{GF}(2)$. Consequently, there are a total of 2^n possible words. However, not all codewords are valid within a specific encoding scheme; only a carefully selected subset is utilized for transmission, based on predefined encoding rules. The encoding process involves mapping a length message k consisting of $\mathbf{a} = (a_1, a_2, \dots, a_k)$ into a *codeword* of length n constituting $\mathbf{x}_b = [b_1, b_2, \dots, b_n]$.



Figure 1.2: Encoding process of a random linear code (RLC), mapping an information message $\mathbf{a}$ into a binary codeword $\mathbf{x}_b$.

Since each information bit is an element of $\text{GF}(2)$, the number of possible input message sequences is given by 2^k , leading to 2^k codewords. The resulting block code is denoted as a (n, k) block code, where [27]:

- n represents the length of the codeword after encoding,
- k is the number of information bits in the original message,

- The remaining $(n - k)$ bits are *parity check bits*, added to introduce redundancy for error detection and correction.

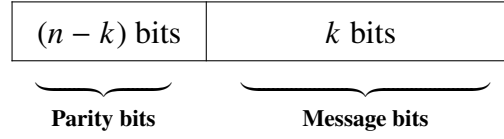


Figure 1.3: Structure of a codeword in a systematic linear block code.

The *code rate* is defined as:

$$R = \frac{k}{n}. \quad (1.1)$$

Since $n > k$, the code rate is always less than 1. A lower code rate indicates a higher level of redundancy, which improves error correction performance at the cost of reduced transmission efficiency. A block code is classified as *systematic* if the original k information bits remain unchanged within the final codeword, with the parity check bits appended to them. In contrast, a *nonsystematic* code does not retain the original bits in their unmodified form; instead, the codeword transforms encoding.[28]

A systematic linear (n, k) block code is specified by a $k \times n$ generator matrix, (**G**) which is composed of two submatrices: a $k \times (n - k)$ submatrix **P** that is called the parity matrix and a $k \times k$ identity matrix **I_k** [29]:

$$\begin{aligned} \mathbf{G} &= \left[\mathbf{P} \mid \mathbf{I}_k \right] \\ &= \left[\begin{array}{cccc|cccc} p_{0,0} & p_{0,1} & \cdots & p_{0,n-k-1} & 1 & 0 & \cdots & 0 \\ p_{1,0} & p_{1,1} & \cdots & p_{1,n-k-1} & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ p_{k-1,0} & p_{k-1,1} & \cdots & p_{k-1,n-k-1} & 0 & 0 & \cdots & 1 \end{array} \right] \end{aligned} \quad (1.2)$$

$\underbrace{\hspace{15em}}$
Parity matrix
 $\underbrace{\hspace{10em}}$
Identity matrix

If a generator matrix (**G**) for a (n, k) random linear codes are provided in systematic form, and the associated parity-check matrix (**H**) is shown below:

$$\begin{aligned} \mathbf{H} &= \left[\mathbf{I}_{n-k} \mid \mathbf{P}^T \right] \\ &= \left[\begin{array}{cccc|cccc} 1 & 0 & \cdots & 0 & p_{0,0} & p_{1,0} & \cdots & h_{k-1,0} \\ 0 & 1 & \cdots & 0 & p_{0,1} & p_{1,1} & \cdots & h_{k-1,1} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & p_{0,n-k-1} & p_{1,n-k-1} & \cdots & h_{k-1,n-k-1} \end{array} \right]. \end{aligned} \quad (1.3)$$

It is straightforward to verify that $\mathbf{G} \cdot \mathbf{H}^T = 0$.

1.2.2 Syndrome-based Error Detection and Correction

One method for detecting potential transmission errors in linear block coding is syndrome error detection. It provides the foundation for several error-correcting codes, such as BCH and Hamming codes, and is essential to digital communications and storage systems. The codeword $\mathbf{x}_b = [b_1, b_2, \dots, b_n]$, taken from the $\text{GF}(2)$, is transmitted through a noisy channel; therefore, the received codeword includes some possible errors, which are different from the transmitted codewords. The received vector and an error vector, also called *error pattern*, that affects the received vector are denoted by $\mathbf{y}_b \in \text{GF}(2)$ and, $\mathbf{e} \in \text{GF}(2)$ respectively. Therefore, the received vector can be expressed by [29]:

$$\mathbf{y}_b = \mathbf{x}_b + \mathbf{e}. \quad (1.4)$$

A syndrome vector is the result of multiplying the received vector by the transpose of the parity-check matrix. The result shows whether an error has occurred and, in some cases, where the error is located.

$$\mathbf{s} = \mathbf{y}_b \cdot \mathbf{H}^T \in \mathbb{F}_2^{n-k}. \quad (1.5)$$

Remark: In this section, vectors are written in row form for notational convenience; this is equivalent to the column-vector convention used elsewhere in the thesis, since $\mathbf{y}_b \mathbf{H}^T = (\mathbf{H} \mathbf{y}_b^T)^T$.

The received vector is subjected to a detection operation. If the result is an all-zero vector, the received vector is regarded as a valid codeword; if not, the decoder has discovered errors.

$$\begin{aligned} \mathbf{s} &= \mathbf{y}_b \cdot \mathbf{H}^T = \\ &= (\mathbf{x}_b + \mathbf{e}) \cdot \mathbf{H}^T = \mathbf{e} \cdot \mathbf{H}^T. \end{aligned} \quad (1.6)$$

If the error vector is the all-zero vector, the syndrome vector will also be all-zero, indicating that the received vector is a valid codeword. When the syndrome vector has at least one nonzero component, it can identify errors in the receiving vector [28].

1.2.3 Low-density Parity Check Codes

Robert G. Gallager introduced the low-density parity-check (LDPC) codes in his 1960 PhD dissertation. Gallager's pioneering work proved that these codes can approach the Shannon limit. LDPC codes were not extensively utilized because of their computational complexity, which was beyond the capabilities of current technology in the 1960s. Interest in LDPC codes resurfaced in the late 1990s and early 2000s with the work of MacKay, Luby, and others [30], [31], [32], [33], who noticed, apparently independently of Gallager's work, the advantages of linear block codes with sparse (low-density) parity-check matrices.

LDPC codes are linear block codes that are built by creating a sparse parity-check matrix $\mathbf{H}$, which means that most of its elements are zeros, with only a few ones scattered throughout. A *regular* Gallager code is a low-density parity-check code that has a length of n and each column of $\mathbf{H}$ has the same weight w_c and each row has the same weight w_r . If the number of ones per row and column is allowed to vary, the result is known as an *irregular* LDPC code. These codes are generated at random based on these constraints [34], [29], [11].

To create an ensemble of (n, w_c, w_r) low-density codes, the matrix is split into w_c submatrices, with a single 1 in each column. The first of these submatrices has all of its 1s in descending order, i.e., the i 'th row has 1s in columns $(i - 1)w_r + 1$ through $i \times w_r$. The other submatrices are just column permutations of the first. An ensemble of (n, w_c, w_r) codes is the outcome of randomly permuting the columns of a matrix's bottom $w_c - 1$ submatrices with equal probability assigned to each permutation. This ensemble yields two significant results: the minimum distance between member codes and the probability of decoding error [34]. Here is an example of an LDPC parity-check matrix based on this structure for $n = 20$, $w_c = 3$, $w_r = 4$ [34]:

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Figure 1.4: Example of a LDPC parity-check matrix ($n = 20$, $k = 5$, $w_c = 3$, $w_r = 4$).

1.2.4 Quasi-regular LDPC codes

A quasi-regular LDPC code is defined by its parity-check matrix and follows two basic principles [30], [31]:

- Generate sparse columns for $\mathbf{H}$ randomly, ensuring that each has the same weight w_c ;
- Any newly generated column, $\mathbf{h}_{\text{new}}$, must have an overlap of at most 1 with each previously accepted column.

The overlap between two columns of the parity-check matrix is defined as the number of row positions in which both columns contain a value of one. The design constraint requires that any

newly added column share at most one common nonzero entry with each previously accepted column. If a new column shares two or more ones in the same row positions with an existing column, the overlap constraint is violated.

In their standard construction, LDPC codes are non-systematic, meaning that the information bits do not appear explicitly in the transmitted codeword, but are instead distributed across all codeword positions through the parity-check constraints. However, when creating short LDPC codes, the second restriction is eased, allowing up to two overlaps. These overlaps are efficiently checked using the inner product between columns in the extended matrix $\mathbf{H}_{\text{ext}} = [\mathbf{H}, \mathbf{I}]$. A new column is rejected if it violates the described constraints. A simple way of testing whether a new column is admissible is by checking whether $\mathbf{H}_{\text{ext}}^T \mathbf{H}_{\text{ext}} - w_c \mathbf{I} > 2$.

0	1	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0	0	0	1	1	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0
0	0	0	1	0	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1
0	1	1	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	1	0	0	1	1	0	0	0	0	0	1
0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1
1	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1	0	0	0	0	0	1	0	0	1	0	0	0	0
0	1	1	0	0	1	0	0	1	0	0	1	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	1	0
0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1	1	0	0	1	0
1	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	1	0	1	0	1	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1	0	0	0	0	1	0	0	0	0	1	1	0
0	0	0	1	0	1	1	0	1	0	1	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	1	0	1	0	1	0	0	0	1	0	1	0	0	1	0	0	0	0	0	0	1	0
1	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0
0	1	0	0	1	0	1	1	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0	0	0	0	0	1	0	1
0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
1	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0

Figure 1.5: Example of a quasi-regular LDPC code matrix ($n = 30, k = 10, w_c = 4$).

1.2.5 Massive Multiple-Input Multiple-Output (mMIMO)

Multiple-input multiple-output (MIMO) technology has been extensively explored over the last two decades and implemented in numerous wireless standards because it has the potential to dramatically increase the capacity and reliability of wireless networks. While early research on the problem focused on point-to-point MIMO links, in which two devices with multiple antennas communicate with each other, the focus has shifted in recent years to more practical multi-User MIMO (MU-MIMO) systems, in which a base station with multiple antennas simultaneously serves a set of single-antenna users and the multiplexing gain is shared by all users [35]. Massive MIMO systems, also known as large-scale antenna systems, have been suggested in [36] and [37] to serve many users simultaneously on the same frequency and time resource, obtain significant gains, and simplify signal processing.

The mMIMO technique is based on the phase-coherent but computationally simple processing of signals from all antennas at the base station. The mMIMO may boost capacity tenfold or more while improving radiated energy efficiency by a factor of 100. The mMIMO's aggressive spatial multiplexing leads to a large increase in capacity. The key idea that enables the enormous gain in energy efficiency is that with a large number of antennas, energy can be directed with extraordinary precision into small regions of space [38].

Figure 1.6 illustrates a general massive MIMO communication model. The transmitted signal vector $\mathbf{x} = [x_1, x_2, \dots, x_{N_T}]^T \in \mathbb{C}^{N_T \times 1}$ passes through the channel matrix $\mathbf{H} \in \mathbb{C}^{N_R \times N_T}$ and is received as $\mathbf{y} = [y_1, y_2, \dots, y_{N_R}]^T \in \mathbb{C}^{N_R \times 1}$. In mMIMO systems, either $N_T \gg N_R$ or $N_R \gg N_T$. In this dissertation, an uplink scenario is considered, and therefore the base station is equipped with a much larger number of receive antennas than the number of simultaneously transmitting user antennas, i.e., $N_R \gg N_T$. The complex MIMO model can be represented as [19]

$$\mathbf{y}^{(c)} = \mathbf{H}^{(c)} \mathbf{x}^{(c)} + \mathbf{n}^{(c)}, \quad (1.7)$$

where $\mathbf{n}^{(c)} = [n_1, \dots, n_{N_R}]^T$ denotes the additional noise at the receiver. The items in $\mathbf{H}^{(c)}$ and $\mathbf{n}$ are independent random variables drawn from the complex normal distribution $\mathcal{CN}(0, 1)$.



Figure 1.6: General massive multiple-input multiple-output (mMIMO) model, where $N_T \gg N_R$ or $N_R \gg N_T$.

Since the decoding algorithms considered in this dissertation operate on real-valued vectors and matrices, here we show how the complex-valued MIMO system can be converted into its equivalent real-valued representation [19]. This transformation splits the real and imaginary components of the transmitted symbols, the received signal, the noise vector, and of the channel matrix itself, allowing the detection problem to be formulated with real-valued linear algebra, while preserving the information contained in the original complex model. The equivalent real-valued representation is given by

$$\begin{bmatrix} \Re\{\mathbf{y}\} \\ \Im\{\mathbf{y}\} \end{bmatrix} = \begin{bmatrix} \Re\{\mathbf{H}^{(c)}\} & -\Im\{\mathbf{H}^{(c)}\} \\ \Im\{\mathbf{H}^{(c)}\} & \Re\{\mathbf{H}^{(c)}\} \end{bmatrix} \begin{bmatrix} \Re\{\mathbf{x}\} \\ \Im\{\mathbf{x}\} \end{bmatrix} + \begin{bmatrix} \Re\{\mathbf{n}\} \\ \Im\{\mathbf{n}\} \end{bmatrix}, \quad (1.8)$$

where $\Re\{\cdot\}$ and $\Im\{\cdot\}$ denote the real and imaginary components, respectively.

This may be simply stated as the real-valued MIMO model:

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}. \quad (1.9)$$

For the real channel matrix $\mathbf{H} \in \mathbb{R}^{2N_R \times 2N_T}$, $\mathbf{x} \in \mathbb{R}^{2N_T \times 1}$, and $\mathbf{y}$, and $\mathbf{n} \in \mathbb{R}^{2N_R \times 1}$.

1.2.6 Generalized Quadrature amplitude modulation (QAM)

Quadrature amplitude modulation (QAM) is a widely used digital modulation scheme that conveys information through the joint variation of the in-phase and quadrature components of a transmitted signal. Because the techniques developed in Chapter 3 can be applied to general QAM constellations, a brief overview of the main QAM constellation structures is provided here. The evolution of QAM constellations has led to different geometric arrangements of constellation points, each offering different trade-offs between spectral efficiency, error performance, and implementation complexity. QAM was developed as an improvement over classical amplitude

modulation in the development of digital modulation schemes throughout the late 1950s and early 1960s. In 1960, C. R. Cahn presented the first QAM design, which is today known as the Type I constellation. This design arranged constellation points along a circular geometry and established the foundation for subsequent QAM developments. Hancock and Lucy developed an enhanced version of Cahn's concept, known as the Type II constellation, in which points were dispersed throughout many concentric rings. When compared to Type I, this structure performed better (in terms of error rates) since it permitted more points on outer circles and fewer on interior ones. Later, in 1962, Campopiano and Glazer introduced the Type III constellation, which included QAM constellations in the shape of ordered squares. Despite having a little worse performance than Type II, the Type III design was much easier to decode and apply, which made it useful for practical applications. Many further advancements in QAM constellations utilized in contemporary digital communication systems were made possible by these early designs [39], [40], [41], [42], [43], [44].

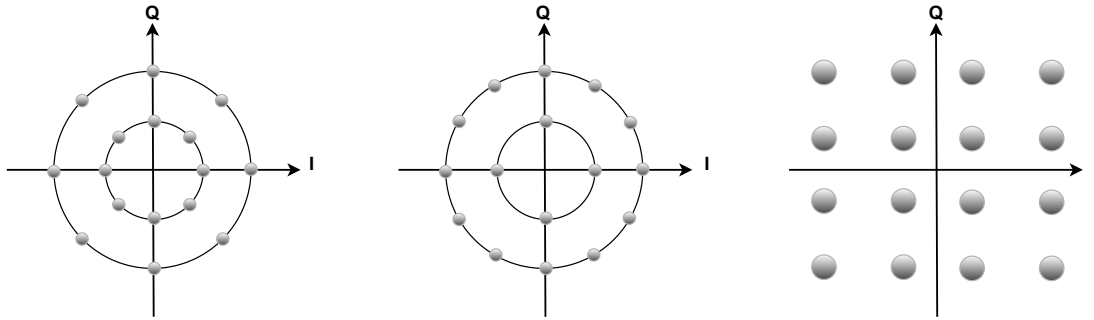


Figure 1.7: Type I, Type II, and Type III QAM constellations.

The QAM family has garnered a lot of interest lately in wireless networks because of its potential to deliver high data rates with enhanced bandwidth and power efficiency. The spectral efficiency associated with an M -QAM modulation scheme can be expressed as

$$\eta = \frac{R_b}{\log_2(M)} \quad [\text{bits/s/Hz}], \quad (1.10)$$

where R_b denotes the bit rate and M is the constellation size. This expression highlights the fundamental trade-off in QAM systems: increasing the constellation order improves spectral efficiency, but at the cost of reduced noise robustness and higher implementation complexity.

Square QAM (SQAM), rectangular QAM (RQAM), cross QAM (XQAM), star QAM, and hexagonal QAM (HQAM) are the primary members of the QAM family. SQAM is usually preferred for even powers of 2 QAM constellations. Odd power-of-2 constellations like RQAM are favored for improved rate and channel adaptation. However, because of its greater average and peak outputs, RQAM is not a desirable option. This is addressed by proposing a cross-shaped constellation called cross QAM (XQAM), which is created by arranging the outer corner points

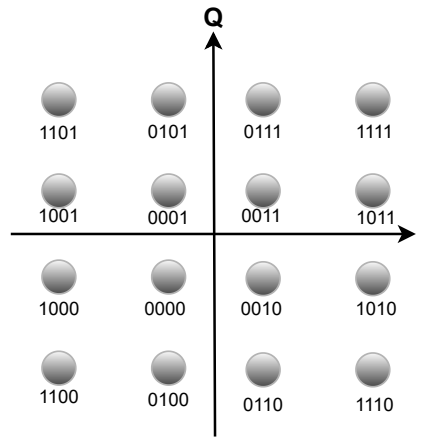


Figure 1.8: Gray-coded 16-QAM.

of RQAM in a cross shape and deleting them, lowering the average energy of the constellations. High data rates at low energy consumption further steered research toward HQAM, a more power-efficient two-dimensional (2D) hexagonal constellation. [45].

SQAM usually takes a perfect square shape for the even power of 2 signaling points 4, 16, 64, 256, 1024, 4096, etc. This arrangement maximizes the minimum Euclidean distance between points for a given average symbol power, enabling efficient transmission with a straightforward ML detection process. Compared to M -ary PSK, M -ary QAM needs a lower carrier-to-noise ratio, making it a preferred modulation scheme in many wireless communication systems and IEEE standards. Lower-order QAM schemes imply reduced spectral efficiency but are more resilient to noise and distortion, aiding in better cell overlap control and improved bit error rate (BER) / symbol error rate (SER) performance. On the other hand, higher-order QAM constellations provide higher data rates but demand tighter SER / BER constraints, introduce greater intercell interference, reduce coverage range, and increase hardware complexity [44].

The information bits of M -QAM modulations are encoded as variations in the signal's phase and amplitude. A sequence of information symbols, each containing k bits of information ($k = \log_2(M)$), is sent by the M -QAM modulator. These symbols are selected from the set $m \in \{1, 2, \dots, M\}$. The information symbols are Gray-coded to limit incorrect receiver decisions to single-bit errors. In any M -QAM constellation, the transmitter constellation's adjacent symbols should not differ by more than one bit to limit incorrect symbol choices to single-bit errors. To do this, the input symbols are typically converted to Gray-coded symbols and then mapped to the intended QAM constellation. A Gray-coded example of a 16-QAM constellation is shown in Figure 1.8.

1.2.7 Guessing Random Additive Noise Decoding (GRAND)

Recent coding techniques such as LDPC codes, Turbo codes, and polar codes have shown remarkable performance in terms of approaching the Shannon limit for long code lengths [34], [46], [47], [48], [49]; nevertheless, these capacity-approaching and capacity-attaining strategies perform poorly in the short-to-medium block length regime. [50]. This problem is particularly important for contemporary communication systems like 5G URLLC, IoT, and augmented reality, which require extremely ultra-reliable and low-latency connections. Guessing random additive noise decoding (GRAND) presents a compelling decoding technique.

GRAND is a class of decoding algorithms that can decode any block code, provided a membership test is available to determine whether a candidate word is a valid codeword. It was initially suggested in 2018 for hard-detection channels [51], [20]. GRAND is a maximum likelihood decoding technique particularly fit for short-length random linear codes with high code rates. Unlike traditional decoding methods that rely on the structure of the codeword itself, GRAND focuses on the noise affecting the transmitted signal [20]. Notably, the only prerequisite for GRAND is the existence of a *membership test* to determine whether a given word is a legitimate codeword. Since it is a universal decoder, it may be used for binary or multi-level codes, whether they are random or have some mathematical structure (such as polar codes, BCH codes, Hamming codes, etc.) [52] [23]. A brute force comparison of the received codeword with every codeword in the codebook is far slower than guessing the noise. This significant reduction in search time is made possible by the noise's low entropy, which results in a manageable list size for potential error patterns affecting a codeword. It may also be the result of some prior knowledge about the noise's statistics, which can speed up the search [53].

GRAND has two primary variants: hard input and soft input. Hard-input GRAND uses binary decisions (e.g., 0 or 1) based on the received signal to estimate the noise, but soft-input GRAND takes into account soft information, such as probabilities or likelihoods, derived from the Euclidean distance of the received signal to the constellation points or decision regions, making it more flexible and potentially more accurate in noisy situations. The primary distinction between the two lies in the sequence used to infer the channel-induced noise [54].

In particular, this technique uses all types of codes and has been shown to attain channel capacity across a wide range of finite-state and memoryless channels. It represents a major theoretical breakthrough because of its universality and independence from the code structure [20]. Expanding on the original idea, the symbol reliability GRAND (SRGRAND) algorithm enhances GRAND by incorporating soft information about symbol reliability, such as log-likelihood ratios obtained from the channel. SRGRAND uses these confidence levels to prioritize which noise patterns to guess, giving more attention to bits considered less reliable. This approach significantly improves both decoding speed and error-correction performance, specifically in noisy settings like AWGN channels. While it introduces extra complexity due to the use of soft data, SRGRAND retains the capacity-achieving of GRAND and suggests practical

improvements by lowering the number of required guesses [22]. ORBGRAND addresses this challenge by assigning each candidate error pattern a logistic weight derived from the reliability information of the received bits. The error patterns are then searched in increasing order of this weight, providing an efficient approximation to the optimal ordering while prioritizing the most likely noise patterns. This strategy can substantially reduce the number of guesses required to identify a valid codeword while retaining the maximum-likelihood decoding principle of GRAND [23].

GRAND is a universal decoding method for high-rate and short-length channel codes such as random linear codes (n, k) with a well-specified encoding framework that exemplifies structured codes. GRAND is a noise-centric decoding method [20] that estimates channel-induced noise in transmitted codewords by producing test error patterns $(\hat{\mathbf{e}})$, which are subsequently applied to the received vector $(\mathbf{y}_b)$ from the channel. The resultant vector $(\mathbf{y}_b \oplus \hat{\mathbf{e}})$ is then assessed for codebook membership [54]. GRAND achieves capacity when utilized with random codebooks, as seen in [20].

The purpose of the decoder is to estimate the transmitted codeword $\mathbf{x}_b$ given the received word $\mathbf{y}_b$. The core idea of GRAND is to identify the error pattern $\hat{\mathbf{e}}$ that distorted the transmitted codeword during transmission, since $\mathbf{y}_b = \mathbf{x}_b \oplus \mathbf{e}$. Instead of decoding the codeword directly, GRAND generates and tests possible error candidate error patterns $\hat{\mathbf{e}}$ in decreasing order of likelihood. Each candidate is applied to the received word to obtain a codeword candidate, and this is checked for codebook membership.

For linear block codes, this membership is efficiently verified using the *parity check matrix* $\mathbf{H} \in \mathbb{F}_2^{(n-k) \times n}$, which defines the codebook as: $C = \{\mathbf{x}_b \in \mathbb{F}_2^n : \mathbf{H}\mathbf{x}_b^T = \mathbf{0}\}$. Given a received word $\mathbf{y}_b$, GRAND tests error patterns $\hat{\mathbf{e}}$ by checking: $\mathbf{s}(\mathbf{y}_b) = \mathbf{H}\mathbf{y}_b^T = \mathbf{H}(\mathbf{x}_b^T \oplus \mathbf{e}^T) = \mathbf{H}\mathbf{e}^T$, which is the $\mathbf{0}$ vector only if $\mathbf{e}$ is $\mathbf{0}$ or $\mathbf{e}$ is a valid codeword; in both cases $\mathbf{y}_b \in C$. If this condition is satisfied, the error pattern $\hat{\mathbf{e}}$ is estimated to be the correct noise pattern, and the estimated codeword is:

$$\hat{\mathbf{x}}_b = \mathbf{y}_b \oplus \hat{\mathbf{e}}.$$

Every error pattern has a syndrome that is not exclusive to it. The number of unique syndromes is only $2^{(n-k)}$ and the number of possible error patterns with up to t errors is

$$\sum_{i=0}^t \binom{n}{i}, \quad (1.11)$$

where t is the maximum number of errors considered. Therefore, inferring the correct error pattern using $\mathbf{s}(\mathbf{x}_b)$ results in ML decoding. As explained in Algorithm 1, for a given fixed value of t , GRAND tries to decode $\mathbf{s}(\mathbf{x}_b)$ by iteratively examining error patterns in decreasing probability order.

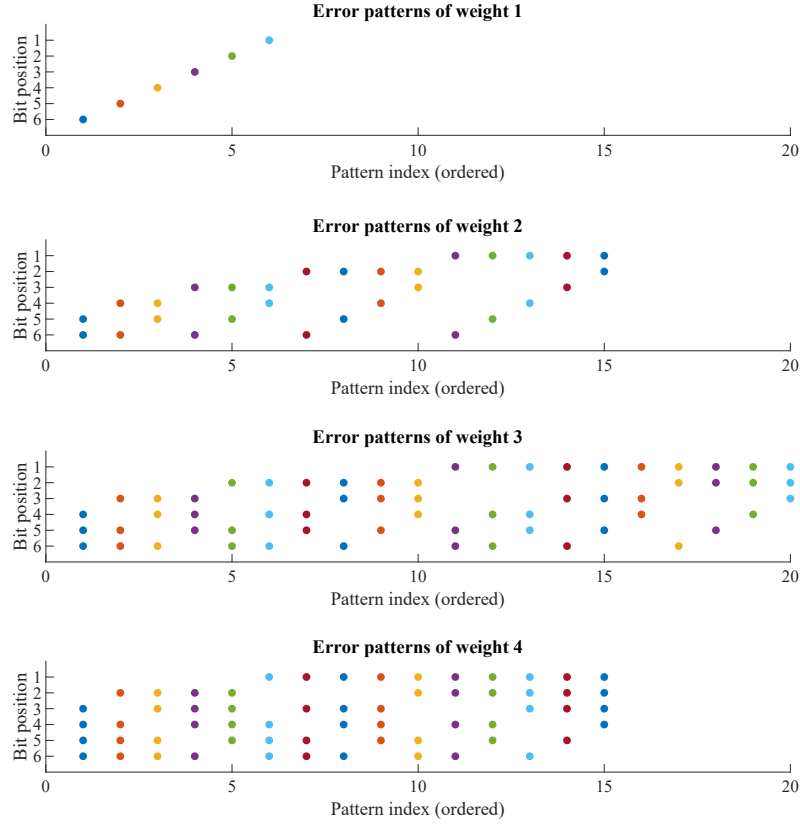


Figure 1.9: Visualization of GRAND error pattern enumeration for a block length $n = 6$. Each subplot corresponds to error patterns of a given Hamming weight. The x-axis indexes error patterns in order of generation, and the y-axis shows the bit positions affected by the error. This illustrates the principle of GRAND: systematically guessing possible noise patterns in increasing order of likelihood (here, increasing Hamming weight) until a valid codeword is found.

Algorithm 1 GRAND for Random Linear Codes

- 1: **Input:** $\mathbf{y}_b = \mathbf{x}_b \oplus \mathbf{e}$
 - 2: **Output:** $\hat{\mathbf{x}}_b = \mathbf{y}_b \oplus \hat{\mathbf{e}}$
 - 3: $\hat{\mathbf{e}} \leftarrow$ Generate Error Patterns
 - 4: **if** $\mathbf{s}(\mathbf{y}_b) = \mathbf{H}(\mathbf{x}_b^T \oplus \mathbf{e}^T) = \mathbf{0}$ **then** ▷ Syndrome check
 - 5: **Output:** $\hat{\mathbf{x}}_b = \mathbf{y}_b \oplus \hat{\mathbf{e}}$ ▷ Valid codeword found
 - 6: **else**
 - 7: Go back to Step 3
 - 8: **Output:** $\hat{\mathbf{x}}_b = \mathbf{y}_b \oplus \hat{\mathbf{e}}$
 - 9: **end if**
-

Figure 1.9 illustrates the principle of the GRAND algorithm. For a block length of $n = 6$, all possible error patterns are generated and ordered by Hamming weight. The algorithm proceeds by applying these candidate noise patterns to the received word, starting from the most likely (low-weight) errors. The figure shows the structure of these error patterns: each row (subplot) corresponds to patterns of a fixed Hamming weight, and within each row, the dots indicate the bit positions that are flipped in a given pattern. This systematic exploration of noise patterns underlies the decoding process of GRAND, where the first candidate that produces a valid

codeword is declared the decoder output.

1.2.8 Bose-Chowla-Lindström Code

In current wireless systems, particularly sensor networks and machine-type communications, numerous devices (terminals) are linked but broadcast only intermittently. Traditional grant-based random access systems, which need devices to wait for permission before sending, are unable to fulfill the ultra-reliable low-latency communication (URLLC) requirements of 5G and beyond. To address this, grant-free random access enables devices to broadcast without delay [55], [56], [57], [58]. However, this results in collisions, which occur when many devices broadcast concurrently. Instead of deleting collision packets, physical layer network coding (PLNC) can use the total of these signals to extract useful information—as long as it can uniquely identify which devices were active [59], [60], [61]. This necessitates the employment of uniquely decodable (UD) signature sequences, which allow the identities of the sending users to be uniquely identified by adding a subset (m out of r) of user sequences. While orthogonal sequences (such as Hadamard or Zadoff-Chu) and pseudo-orthogonal sequences (such as Gold codes) are commonly utilized, they are not ideal for this application. In short, from a sum, one can uniquely determine the terms involved in it, and this is known as the signature coding problem [62], [63], [64], [65]. There are two types of UD sequences: those acquired by the Chang-Weldon construction [64], which is connected to the Cantor and Mills construction [66], and those obtained by the Lindström construction [63], [64], [67].

Lindström sequences are completely dependent on an algebraic construction based on Galois fields, which was initially presented by Bose and Chowla in [68], hence in this work, we will refer to such sequences as Bose-Chowla-Lindström (BCL) sequences.

The length of the sequence is $n(r, m) \sim m \log r$ bits for a system with r users, of which m are active [63], [67], [68], assuming that:

$$\lim_{m \rightarrow \infty} \lim_{r \rightarrow \infty} \frac{n(r, m)}{m \log r} \leq 1. \quad (1.12)$$

BCL sequences are not necessarily binary. Due to their construction from integers that can be transferred onto various Galois fields [63].

The purpose of the BCL construction is to produce a set of $r \geq t$ integers that have UD sums for each group of m out of these r integers. This configuration is called $\text{BCL}(r, m)$. The construction requires that r be a prime power, which means that it has the form p^i , where i is an integer and p is a prime number. Building a code for t users begins with building a code for r users, where $r \geq t$ and is selected as a prime power. After that, the $r - t$ integers are eliminated, which is equivalent to muting the $r - t$ users, resulting in a final set of t users. The prime number theorem demonstrates that r fulfills the upper bound $r \leq 2^{\lceil \log_2 t \rceil}$.

In the construction, let α be a primitive element of the Galois field $\text{GF}(r^m)$, meaning that all elements of the field can be expressed as powers of α : $\text{GF}(r^m) = 0, \alpha^0, \alpha^1, \dots, \alpha^{r^m-1}$. Primitive elements always satisfy $\alpha > 1$. Let $\beta_1, \beta_2, \dots, \beta_s$ be elements of $\text{GF}(r)$. Define integers $d_1, d_2, \dots, d_s$ such that:

$$\alpha^{d_i} = \alpha + \beta_i \quad \text{for } i = 1, 2, \dots, s. \quad (1.13)$$

Python's Galois field library can compute both α and d_i values. Because $\alpha + \beta_i$ is always a nonzero element in $\text{GF}(r^m)$, each i has a corresponding d_i value. Given $\alpha > 1$, we have $d_i \neq 0$, and given primitive elements' characteristics, $0 < d_i \leq r^m - 1$. The integers $d_1, d_2, \dots, d_s$ have a critical property: their m -term sums are unique. This means that no two alternative combinations of m elements give the same sum. This can be formally expressed as:

$$\sum_{i=1}^m d_{a_i} = \sum_{i=1}^m d_{b_i} \iff \{a_1, a_2, \dots, a_m\} = \{b_1, b_2, \dots, b_m\} \quad (1.14)$$

This equation implies that if two subsets of m components provide the same total sum when combined, they must be equal (up to reordering). In other words, no two unique m -sized selections of the d_i values can provide the same result.

Each integer d_i is converted to binary and added with an extra 1 to create a binary codeword $\mathbf{x}_i$. This additional bit allows the base station (BS) to detect how many users are participating in a transmission (i.e., the collision level).

Since each $d_i < r^m - 1$, it needs at most $\lceil \log_2(r^m - 1) \rceil$ bits to define d_i in binary:

$$n(r, m) = \lceil \log_2(r^m - 1) \rceil + 1. \quad (1.15)$$

Given that $r \leq 2 \log_2 t$, we can derive an upper bound for $n(t, m)$ as follows:

$$\begin{aligned} n(r, m) &\leq m \log_2 r + 2 \\ &\leq m \lceil \log_2 r \rceil + 2 \\ &\leq m(1 + \log_2 r) + 2. \end{aligned} \quad (1.16)$$

This codeword design provides that for any m , the binary sums of m different codewords are unique. That is, no two distinct groups of m users yield the same sum, which is essential for reliably identifying active users in multi-user communication systems when using non-orthogonal identification sequences.

1.3 Thesis Structure

This dissertation is organized as follows:

Chapter 2 – URLLC with Coded Massive MIMO via Random Linear Codes and GRAND

This chapter investigates the integration of RLCs with GRAND decoding in mMIMO systems. Using zero-forcing spatial multiplexing and different QAM mapping schemes, it demonstrates that using RLCs with GRAND significantly outperforms 5G polar codes in both block-error rate (BLER) and decoding latency. The results show substantial decoding-complexity reductions.

Chapter 3 – Constrained Symbol-Level Noise-Guessing Decoding with Antenna Sorting for Massive MIMO

Chapter 3 expands GRAND into a symbol-level constrained framework for mMIMO systems. An antenna-reliability sorting method and a novel lightweight membership test are introduced to fast prune infeasible noise patterns early. This yields over 90% complexity reduction while maintaining ML-equivalent performance. Importantly, the proposed techniques constitute general decoding mechanisms that apply to a wide class of linear block codes and are not inherently tied to a particular coding structure.

Chapter 4 – Largely Constrained ML-Level Decoding of Short LDPC Codes

Building on the constraint-driven decoding ideas, this chapter adapts GRAND to short LDPC codes. It proposes three different techniques to remove the number of error patterns that need to be tested: 1) set partitioning, a general constraint mechanism applicable to any linear block codes, 2) sieving, and 3) large-constraint methods that eliminate more than 99% of infeasible patterns before enabling practical ML-level decoding for short LDPC codes.

Chapter 5 – Uniquely Decodable Signature Sequences Over the Noisy Adder Channel

This chapter proposes using BCL-based UD signature sequences that are constructed through an algebraic method combining related Galois fields. These sequences guarantee m -sum uniqueness, enabling reliable user identification under noisy collisions in grant-free random access systems.

Chapter 6 – Conclusions and Future Work

The final chapter summarizes the dissertation's contributions and outlines avenues in coding theory, URLLC, and multiuser random access.

Related Publications

This dissertation is based on the following peer-reviewed and submitted publications. Each technical chapter directly builds upon the corresponding published or submitted work.

- **Chapter 2:** S. Allahkaram, F. A. Monteiro, and I. Chatzigeorgiou, “URLLC with Coded Massive MIMO via Random Linear Codes and GRAND,” in *Proceedings of the IEEE 96th Vehicular Technology Conference (VTC-Fall)*, London, UK, Sep, 2022. **Status: Published.**
- **Chapter 3:** S. Allahkaram, F. A. Monteiro, and I. Chatzigeorgiou, “Constrained Symbol-Level Noise-Guessing Decoding With Antenna Sorting for Massive MIMO,” *IEEE Open Journal of the Communications Society*, vol. 6, pp. 9486–9503, Nov, 2025. **Status: Published.**
- **Chapter 4:** S. Allahkaram, F. A. Monteiro, J. Ribeiro, and C. Salema, “Largely Constrained Noise-Guessing Decoding of Short LDPC Codes,” *IEEE Open Journal of the Communications Society*, vol. 7, Aug, 2026. **Status: Accepted. Available as Early Access article.**
- **Chapter 5:** S. Allahkaram and F. A. Monteiro, “Uniquely Decodable Signature Sequences Over the Noisy Adder Channel,” in *Proceedings of the International Symposium on Networks, Computers and Communications (ISNCC)*, Paris, France, Oct, 2025. **Status: Published.**
- **Chapter 5:** S. Allahkaram and F. A. Monteiro, “Uniquely Decodable Bose–Chowla–Lindström Signature Sequences Over the Multiple Access Adder Channel,” submitted to *IEEE Access*, Nov, 2025. **Status: Under review.** *This manuscript is an extended journal version of the conference paper listed above.*

Chapter 2

URLLC with Coded Massive MIMO via Random Linear Codes and GRAND

Conference: 2022 IEEE 96th Vehicular Technology Conference (VTC2022-Fall)

Date: 18 January 2023

Location: London, United Kingdom

Pages: 1-5

DOI: 10.1109/VTC2022-Fall57202.2022.10012803

URL: <https://ieeexplore.ieee.org/document/10012803>

URLLC with Coded Massive MIMO via Random Linear Codes and GRAND

Sahar Allahkaram*, Francisco A. Monteiro*, and Ioannis Chatzigeorgiou†

*Instituto de Telecomunicações, and ISCTE - Instituto Universitário de Lisboa, Lisbon, Portugal

†School of Computing and Communications, Lancaster University, Lancaster, United Kingdom

emails: {sahar.allahkaram, francisco.monteiro}@lx.it.pt, i.chatzigeorgiou@lancaster.ac.uk

Abstract—A present challenge in wireless communications is the assurance of ultra-reliable and low-latency communication (URLLC). While the reliability aspect is well known to be improved by channel coding with long codewords, this usually implies using interleavers, which introduce undesirable delay. Using short codewords is a needed change to minimizing the decoding delay. This work proposes the combination of a coding and decoding scheme to be used along with spatial signal processing as a means to provide URLLC over a fading channel. The paper advocates the use of random linear codes (RLCs) over a massive MIMO (mMIMO) channel with standard zero-forcing detection and guessing random additive noise decoding (GRAND). The performance of several schemes is assessed over a mMIMO flat fading channel. The proposed scheme greatly outperforms the equivalent scheme using 5G's polar encoding and decoding for signal-to-noise ratios (SNR) of interest. While the complexity of the polar code is constant at all SNRs, using RLCs with GRAND achieves much faster decoding times for most of the SNR range, further reducing latency.

Index Terms—Ultra-reliable and low-latency communications (URLLC), massive MIMO, Random linear codes (RLCs), Guessing random additive noise decoding (GRAND)

I. INTRODUCTION

The physical layer has much to contribute to the goal of wireless ultra-reliable and low-latency communication (URLLC), whose major goals are to reduce latency to 1 ms and simultaneously ensure at least 99.999 % reliability [1].

Noise-guessing decoding was very recently proposed as a universal decoder for codes having codewords of moderate length or sufficiently high rate, which are ideal for applications in wireless URLLC, where the codewords are desirably short. The technique, dubbed guessing random additive noise decoding (GRAND), allows ditching interleavers and eliminating the decoding delay bottleneck they imposed [2]. Given that the entropy of the noise is much smaller than the entropy of the codewords, decoding of the noise pattern that affected a codeword greatly diminishes the complexity of maximum likelihood (ML) decoding. Moreover, because it is a universal decoder, it opened doors to using random linear codes (RLCs), which are able to attain the maximum rate of the finite-blocklength regime, precisely the one of interest in URLLC. The combination of RLC encoding with GRAND is a perfect fit for URLLC, where the flexibility of choice of the codewords' lengths and rates of a RLC greatly exceeds any limitation regarding their length.

As it is well known, massive multiple-input multiple-output (mMIMO) allowed the very high spectral efficiencies in 5G due to spatial multiplexing. In order to cater for both objectives in URLLC, this paper proposes coded mMIMO links using a RLC and GRAND. GRAND has been recently proposed for single-input single-output (SISO) flat Rayleigh channels and shown to outperform Bose–Chaudhuri–Hocquenghem (BCH) codes using the Berlekamp–Massey (B-M) decoder [3]. To the best of our knowledge, the use of RLCs and GRAND is for the first time proposed and assessed in the context of a MIMO fading channel.

II. RANDOM LINEAR CODES

Shannon's random code-construction [4] provably reaches capacity when the length of the codewords tends to infinity and the decoder performs ML decoding. However, selecting the codebook members randomly leads to two different problems: i) a storage problem, given that all codewords would have to be stored both at the encoding side and at the decoding side, and ii) a decoding complexity problem given that, when applying the ML principle, a corrupted codeword needs to be compared with all codewords in the codebook in order to find the most likely one.

The storage problem posed by Shannon's construction has been overcome by RLCs, because, as in the case of any linear block code, the generator matrix $\mathbf{G}$ encapsulates a very short description of a code, and is the only piece of information needed to be stored.

RLCs are known to be capacity-achieving in the binary symmetric channel (BSC), [4], [5], in the asymptotic regime. Notably, they also attain the maximum rate possible in the *finite-blocklength regime* [6], [7], the one of interest for practical URLLC. Furthermore, RLCs can be constructed with *any size and any rate*, and thus having those degrees of freedom is a major practical advantage for most engineering applications.

III. GUESSING RANDOM ADDITIVE NOISE DECODING

Albeit classical RLCs are known to be capacity-achieving, they do not offer a solution to the remaining problem of having a practical decoder. A practical solution for the RLC's decoding conundrum only recently emerged with the advent of GRAND-based algorithms [2], [8], [9]. GRAND focuses on guessing the noise that corrupted the transmitted codeword, rather than exhaustively going through all the possible

codewords, and is proven to still lead to ML decoding [8]. GRAND is a *universal* decoder, enabling the decoding of *any* block code of moderate length and with a sufficiently high code rate, and can be used whether the code is random or has some mathematical structure (e.g. polar codes [10], [11], BCH codes [3], [11], Hamming codes, etc), being a binary or multi-level code. Notably, the sole requirement for GRAND is that a membership test exists to detect if a received word is a valid codeword. In the case of RLCs, the test relies on the codeword's syndrome.

Because GRAND is a universal decoder, it opened doors to using the capacity-achieving RLCs for which no practical decoder was yet available. Recent research has shown that RLCs supersede the performance of polar codes of the same length, even with short codewords [2], [10]. However, while off-the-shelf polar codes do not exist for any wanted length or desired rate, RLCs can be constructed with any desired *blocklength* and *rate*.

For uniform-at-random RLCs, encoding k bits onto n , with rate $R = k/n$, where the codewords are chosen uniformly at random from the set (or space) where 2^n possible words exist, out of which only $2^k = 2^{nR}$ are valid codewords. The distance between any two codewords is also uniformly distributed. As $n \rightarrow \infty$, GRAND permits a successful decoding in $2^n/2^{nR} = 2^{n(1-R)}$ trials, on average. Consequently, higher code rates lead to a faster decoding when using GRAND because the words' space becomes denser.

IV. SYSTEM MODEL FOR CODED MASSIVE MIMO

A coded mMIMO system is considered, with a RLC encoder at the transmitter and GRAND at the receiver, as depicted in Fig. 1. The description of the transmission chain is made considering only one packet of information comprising k bits, which is encoded onto a codeword, sent via spatial multiplexing. For longer streams the process can be repeated by slicing the string of bits in packets of size k .

A. RLC encoding scheme

A block $\mathbf{a}$ of k i.i.d. information bits is linearly encoded onto a codeword of bits $\mathbf{x}_b$ of length n using a systematic RLC with rate $R = \frac{k}{n}$ over the binary Galois $\mathbb{F}_2$. A (n, k) RLC defines a codebook $\mathcal{C}$ with $2^k = 2^{nR}$ codewords of length n , constituting a linear subspace of the discrete vector space $\mathbb{F}_2^n$. The minimal Hamming distance between two codewords in $\mathcal{C}$ is d , however, its role in the context of RLC is not as relevant to determine the performance of some code [12, Ch.13]. The code is defined by a binary *random* generator matrix $\mathbf{G} \in \mathbb{F}_2^{k \times n}$, which acts as the basis matrix for the code subspace, such that $\mathcal{C} = \{\mathbf{x}_b = \mathbf{a}\mathbf{G} : \mathbf{a} \in \mathbb{F}_2^k\}$. The generator matrix is of the form $\mathbf{G} = [\mathbf{P} \mid \mathbf{I}_k]$, where $\mathbf{P} \in \mathbb{F}_2^{k \times (n-k)}$ is a binary random matrix, and $\mathbf{I}_k$ is the identity matrix of size $k \times k$, responsible for the systematic part of the encoding. As a result of this construction, all codewords $\mathbf{x}_b$ are equally probable.

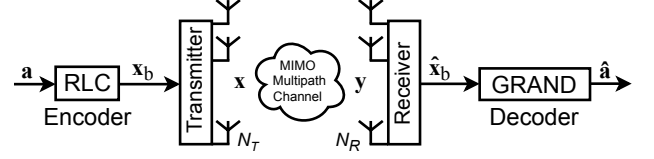


Fig. 1. System model for coded mMIMO URLLC.

B. Spatial multiplexing with mMIMO

The baseband bit encoder outputs the n -bit codeword $\mathbf{x}_b$, which is fed to a transmitter equipped with N_T transmit antennas which maps the n bits onto N_T complex symbols belonging to a M -QAM constellation defined by the symbol alphabet $\mathcal{A} \in \mathbb{C}$, each of which carrying $\log_2(M)$ bits. These symbols form the complex transmit vector is $\mathbf{x}^{(c)} = [x_1, \dots, x_{N_T}]^T$. In this work one uses a “one-shot” approach to the transmission of each codeword. This means that the number of transmit antennas fits the number of complex symbols needed to transmit one codeword when using a given cardinality M for the modulation, and is given by

$$N_T = \frac{n}{\log_2(M)}, \quad (1)$$

nevertheless, this constraint can be easily alleviated.

The codeword of n bits is divided in small strings with $\log_2(M)$ bits, which are mapped onto N_T constellation symbols using some mapping scheme. Two mapping schemes will be considered: *natural mapping* and *Gray mapping*. In both cases the complex constellation alphabet $\mathcal{A}$ is constructed as the Cartesian product of two pulse amplitude modulation (PAM) alphabets, where each of the I and Q components carry $\frac{\log_2(M)}{2}$ PAM bits. The system is designed such that the first $n/2$ bits of a codeword control the PAM symbols in the I component, and the remaining $n/2$ bits control the PAM symbols in the Q component. Square QAM constellations with $M = 2^{2\theta}$ are considered, with $\theta = 2, 3$.

The symbols transmitted from each antenna are assumed to have unit power, that is $\mathbb{E}\{(\mathbf{x}^{(c)})^H \mathbf{x}^{(c)}\} = M$, where $(\cdot)^H$ is the Hermitian transpose operator. The coded transmitted signal $\mathbf{x}$ goes through a channel (perfectly known at the receiver), characterized by the matrix $\mathbf{H} \in \mathbb{C}^{N_R \times N_T}$ and is received at the receiver equipped with $N_R \gg N_T$ antenna elements. Using the complex MIMO model for flat Rayleigh fading, the received signal $\mathbf{y} = [y_1, \dots, y_{N_R}]^T$ can be expressed as:

$$\mathbf{y}^{(c)} = \sqrt{\frac{\text{SNR}}{M}} \mathbf{H}^{(c)} \mathbf{x}^{(c)} + \mathbf{n}^{(c)}, \quad (2)$$

where $\mathbf{n}^{(c)} = [n_1, \dots, n_{N_R}]^T$ represents the additive noise at the receiver. Both the entries in $\mathbf{H}^{(c)}$ and in $\mathbf{n}$ are i.i.d. random variables taken from a complex normal distribution $\mathcal{CN}(0, 1)$. For implementation purposes, the complex-valued MIMO model was converted to a real-valued one [13].

While it is well-known that the performance of ZF is very poor in symmetric MIMO ($N_R = N_T$), ZF attains quasi-optimal performance in highly asymmetric MIMO with

$N_R \gg N_T$. Matrix inversion may become expensive but very good approximate solutions for the inverse matrix can be obtained via Neumann series [14]. Considering $N_R \gg N_T$, ZF incurs no performance loss and boosts the receiver array gain. ZF detection amounts to applying the Moore-Penrose pseudo-inverse [13]

$$\mathbf{H}^+ = (\mathbf{H}^H \mathbf{H})^{-1} \mathbf{H}^H, \quad (3)$$

resulting at the receiver:

$$\mathbf{H}^+ \mathbf{y} = \mathbf{I}_{N_T} \mathbf{x} + \underbrace{\mathbf{H}^+ \mathbf{n}}_{\mathbf{u}}, \quad (4)$$

where $\mathbf{I}_{N_T}$ is the identity matrix of size N_T and $\mathbf{u} \in \mathbb{C}^{N_T}$ denotes the new noise vector after the ZF filter.

Symbol detection is made via a slicer defined by the shape of the constellation, and finally the bits are demapped from the detected symbols, reconstructing a binary word $\mathbf{y}_b = \mathbf{x}_b \oplus \mathbf{e}$, with $\oplus$ denoting the modulo-2 addition. This word is fed to the decoder, which applies GRAND to infer the transmitted word $\hat{\mathbf{x}}_b$, which was corrupted by the error pattern $\mathbf{e}$ (where $\mathbf{y}_b$, $\mathbf{x}_b$, and $\mathbf{e}$, are all strings of n -bits).

C. Decoder applying GRAND

The task of the decoder is to estimate $\mathbf{x}_b$ given $\mathbf{y}_b$. The central idea of GRAND is that this task is equivalent to the one of decoding the error pattern $\hat{\mathbf{e}}$ affecting $\mathbf{x}_b$.

Instead of using the generator matrix $\mathbf{G}$, any linear block code can also be defined by its parity check matrix $\mathbf{H} \in \mathbb{F}_2^{(n-k) \times n}$ whose kernel is: $\mathcal{C} = \{\mathbf{x}_b \in \mathbb{F}_2^n : \mathbf{H}\mathbf{x}_b^T = \mathbf{0}\}$, reminding that $\mathbf{x}_b \in \mathcal{C}$. *Note:* the notation of the parity matrix coincides with the one of the MIMO channel matrix; given the very different contexts in which they appear, we have opted to keep the traditional notations.

The *syndrome* of a detected word $\mathbf{y}$ is $\mathbf{s}(\mathbf{x}_b) = \mathbf{H}\mathbf{x}_b^T = \mathbf{H}(\mathbf{x}_b^T \oplus \mathbf{e}^T) = \mathbf{H}\mathbf{e}^T$, which is the $\mathbf{0}$ vector only if $\mathbf{e}$ is $\mathbf{0}$ or $\mathbf{e}$ is a valid codeword; in both cases $\mathbf{y}_b \in \mathcal{C}$. The syndrome associated with each error pattern is not unique. The number of distinct syndromes is only $2^{(n-k)}$ and the number of possible error patterns is $\sum_t^n \binom{n}{t}$. Therefore, identifying the true error pattern based on $\mathbf{s}(\mathbf{x}_b)$ leads to highly sub-optimal decoding.

Unlike other deployed decoders, GRAND concentrates on identifying the error pattern rather than the codeword itself; it attempts to decode $\mathbf{x}_b$ by successively testing error patterns in decreasing probability order, as described in Fig. 2.

Because of the combined effect of using a M -QAM modulation and having a MIMO flat fading channel, the noise patterns $\mathbf{e}$ will not have the same probability distribution of the additive white Gaussian noise (AWGN) model that has mostly been considered when analysing GRAND. Nevertheless, in this work one considers that the probability of the error patterns gets smaller with an increasing weight, denoted as $|\mathbf{e}|$. Thus, error patterns of weight $|\mathbf{e}| = i$ are tested, with $i = 1, \dots, n_b$, meaning that a maximum number of n_b bit-flips is considered in a n -bit codeword.

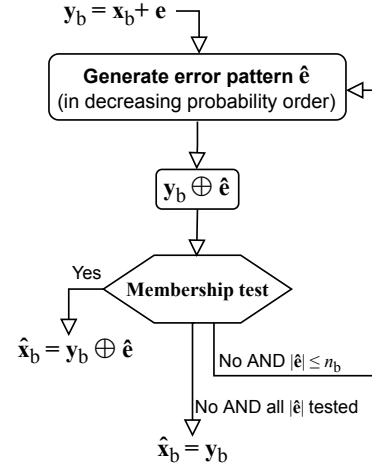


Fig. 2. Description of the GRAND algorithm.

V. PERFECT CHANNEL HARDENING LOWER-BOUND

The error patterns $\mathbf{e}$ that corrupt the transmitted codeword are not simply due to the thermal AWGN noise at the receive antennas. As seen in Section IV-B, in (4), when the receiver makes ZF detection, the noise affecting the decision is an amplified version of the thermal noise. The effect the ZF filter has on the original noise $\mathbf{n}$ can be tracked by considering the autocorrelation matrix of the new noise $\mathbf{u} = \mathbf{H}^+ \mathbf{n}$, which is:

$$\begin{aligned} \mathbf{R}_u &= \mathbb{E} \{ \mathbf{u} \mathbf{u}^H \} = \mathbb{E} \{ (\mathbf{H}^+ \mathbf{n}) (\mathbf{H}^+ \mathbf{n})^H \} \\ &= \mathbb{E} \{ (\mathbf{H}^+ \mathbf{n}) (\mathbf{n}^H (\mathbf{H}^+)^H) \} \\ &= \mathbf{H}^+ \mathbb{E} \{ \mathbf{n} \mathbf{n}^H \} (\mathbf{H}^+)^H = \sigma_n^2 \mathbf{H}^+ (\mathbf{H}^+)^H, \end{aligned} \quad (5)$$

where the autocorrelation of additive noise, $\mathbb{E} \{ \mathbf{n} \mathbf{n}^H \} = \mathbf{R}_n = \sigma_n^2 \mathbf{I}_{N_R}$, was used. Replacing (3) in (5), after some matrix algebra, it is possible to obtain

$$\mathbf{R}_u = \sigma_n^2 (\mathbf{H}^H \mathbf{H})^{-1} = \sigma_n^2 \mathbf{T}^{-1}, \quad (6)$$

where $\mathbf{T} = \mathbf{H}^H \mathbf{H} \in \mathbb{C}^{N_T \times N_T}$ is the Gram matrix [15].

As seen in (4), the correct detection of $\hat{\mathbf{x}}$ is perturbed by the modified noise vector $\mathbf{u}$. It is possible to show that the output SNR after ZF detection of the N_T incoming signals is always lower than the input SNR [16] [13, sec. 2.5.2]:

$$\text{snr}_i^{(ZF)} = \frac{\text{snr}_i}{\left[(\mathbf{H}^H \mathbf{H})^{-1} \right]_{ii}}, \quad 1 \leq i \leq N_T, \quad (7)$$

meaning that, in real-world channels, ZF detection always leads to noise enhancement in the detection of $\hat{\mathbf{x}}$.

There is only one particular (and ideal) situation that does not lead to noise enhancement: when all the column vectors in $\mathbf{H}$ are mutually orthogonal. This is precisely what tends to happen when N_T is fixed and $N_R \rightarrow \infty$, as in the case of mMIMO, leading to the so-called *channel hardening effect*. The geometric interpretation is the following: one has N_T random Gaussian vectors living in a N_R -dimensional space; with high probability any pair of the N_T vectors will be

orthogonal to each other. In that case, the Gram matrix, which comprises all the inner products $\mathbf{h}_i^H \mathbf{h}_j$, $i, j = 1, \dots, N_T$, becomes a diagonal matrix of the form:

$$\mathbf{T} = \text{diag}(\|\mathbf{h}_{11}\|^2, \dots, \|\mathbf{h}_{N_T N_T}\|^2) = N_R \mathbf{I}_{N_T}, \quad (8)$$

given that $\|\mathbf{h}_{ii}\|^2 = \sum_{i=1}^{N_R} |h_i|^2 = N_R$, for all the N_T vectors.

Replacing (8) in (5), in the case of perfect channel hardening, the autocorrelation of the noise after ZF is $\mathbf{R}_{\mathbf{u}} = \frac{\sigma_n^2}{N_R} \mathbf{I}_{N_T}$. Finally, the power of $\mathbf{u}$ is $\|\mathbf{u}\|^2 = \text{Tr}(\mathbf{R}_{\mathbf{u}}) = \frac{\sigma_n^2 N_T}{N_R}$.

It is now possible to establish the equivalent channel model if the $N_T \times N_R$ mMIMO configurations were to attain perfect channel hardening at those (finite) dimensions: $\mathbf{y} = \mathbf{I}_{N_T} \mathbf{x} + \mathbf{u}$. One has N_T independent parallel channels with each of the N_T component of $\mathbf{u}$ having power $|u_i|^2 = \frac{\sigma_n^2}{N_R}$, shedding light on the tremendous benefit of having a larger receiver array.

VI. RESULTS

The proposed system was assessed by numerical simulation in terms of performance and decoding complexity when transmitting a short codeword in “one shot”, i.e., only using one MIMO burst, aiming at URLLC. The results are also compared with the (128,103) benchmark polar code used in the control channel of the 5G air interface, thus having $R = 0.8$. For an increasing spectral efficiency of the modulation, the number of transmit antennas is reduced to accommodate the same payload of n bits, according to expression (1). In the case of $M = 64$, expression (1) would lead to a non-integer $N_T = 21.3$. To make it an integer number of antennas, the RLC code is slightly changed to (132,106) in the case of 64-QAM, keeping the same code rate $R = \frac{106}{132} = 0.8$.

The assessment of the system’s performance is made via the block error rate (BLER) versus $\frac{E_b}{N_0}$, as it is most common in recent papers assessing GRAND. E_b denotes the energy per information bit and N_0 is the bilateral spectral density of the noise at each one of the receive antennas. The decoding complexity is measured by the expected number of membership tests needed at each $\frac{E_b}{N_0}$. All the results are ergodic in sense that each Monte Carlo iteration uses a different channel matrix $\mathbf{H}$ and a different generator matrix $\mathbf{G}$ for the RLC.

Figures 3 and 4 show the performance and complexity results respectively for $M = 16, 64$. One should highlight that n remains constant in the different cases, while N_T and M change, with opposite effects and thus the joint effect does not lend itself to an easy interpretation. As one would expect, with all other parameters unchanged, Gray mapping always outperforms the equivalent scheme using natural mapping.

By increasing the value of n_b , threshold error patterns of larger weight are tested and therefore the performance dramatically increases, at the expense of a much larger member of membership tests. The upper bound for the number of membership tests is given by

$$UB = \sum_{t=1}^{n_b} \binom{n}{t} = \binom{n}{1} + \binom{n}{2} + \dots + \binom{n}{n_b}, \quad (9)$$

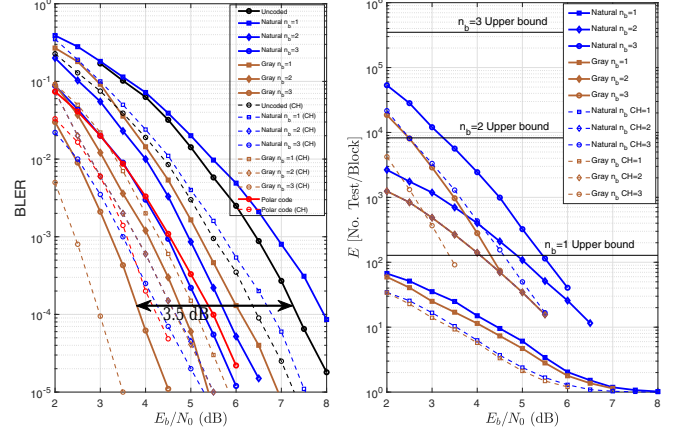


Fig. 3. Performance (left) and decoding complexity (right) for different n_b thresholds in GRAND, using RLC (128,103), with $N_T = 32$, and 16-QAM. The corresponding perfect channel hardening lower-bounds are also plotted. The performance of the polar-code (128,103) is also included.

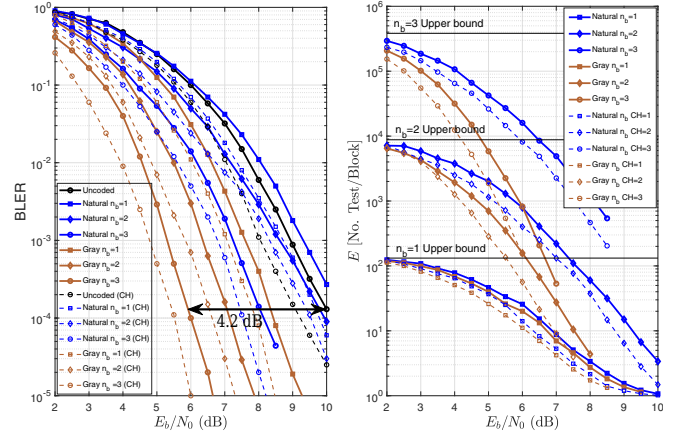


Fig. 4. Performance (left) and decoding complexity (right) for different n_b thresholds in GRAND, using RLC (133,106), with $N_T = 22$, and 64-QAM. The corresponding perfect channel hardening lower-bounds are also plotted.

however, the results show that the average number of membership tests is much lower than (9) in the case $M = 16$ (and also with $M = 4$, not shown due to space limitations). With 64-QAM the complexity can approach the upper bound when the noise is too large due to the sheer number of modulation symbols in error. It is interesting to observe how the average number of membership tests always tends to one, which is a consequence of the decreasing number of bits in error, such that eventually almost all received words are valid codewords.

The comparison of the performance of the proposed schemes using RLC and GRAND with the performance attained using the polar code (128,103) decoded using the list decoding technique used in 5G [17], for a list length of 8, shows that the RLC with GRAND remarkably outperforms the polar code. More surprisingly, this is true even when GRAND runs with only $n_b = 2$ (seen in Fig. 3 and also with 4-QAM, not shown due to space limitations). Note that for $M = 64$

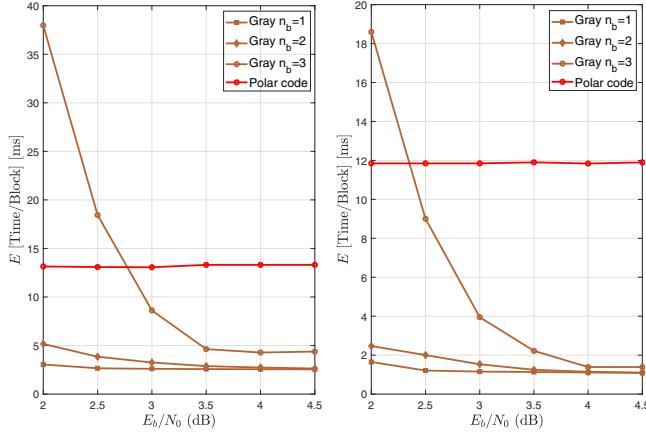


Fig. 5. Comparison of the average decoding time per codeword: RLC (128,103) using GRAND with different n_b thresholds vs the equivalent (128,103) polar code using list decoding. The results are for $N_T = 64$ with 4-QAM (left), and $N_T = 32$ with 16-QAM (right).

one could easily construct a RLC with a rate that fits the setup, however, a polar code counterpart does not exist and could not be included for comparison in Fig. 4).

Figures 3 and 4 include both the performance and the complexity results for the situation with perfect channel hardening (see Sec. V). This gauges the loss coming from the non-ideal mMIMO. As expected, the more asymmetrical the MIMO configuration becomes, the closer one is from that ideal case. One has a fixed $N_R = 200$ and $N_T = 32, 22$. In fact, in the ($M = 64, N_T = 22$) setup the gap to the ideal channel is only ≈ 0.5 dB for all n_b and for both mapping schemes.

The complexity involved in decoding the polar code is assessed by means of the average decoding time of codewords, as depicted in Fig. 5. The decoding time of polar codes remains constant for all $\frac{E_b}{N_0}$. However, as a consequence of the accentuated reduction of the number of membership tests performed (as observed in figures 3, and 4), GRAND almost always spends much less time decoding a codeword than the list decoder. Depending on $\frac{E_b}{N_0}$, when GRAND is set with $n_b = 1, 2$ it is ≈ 2.5 to ≈ 5 times faster in the ($M = 4, N_T = 64$) setup and ≈ 5 to ≈ 12 times faster in the ($M = 16, N_T = 32$) setup. Even with the best performant $n_b = 3$, applying GRAND to the RLC is faster than decoding the polar code for $\frac{E_b}{N_0} > 2.75$ dB in the first setup and > 2.3 dB in the second. Using RLC with GRAND is up to over ≈ 3 times faster in the first setup and up to ≈ 8 times faster in the second setup. The average decoding time was measured using MATLAB on PC equipped with a CPU Intel Core i7-12700 and 32GB of RAM with maximum clock speed of 4.9 GHz.

VII. CONCLUSIONS

This paper proposed transmission schemes for high-throughput, high-reliability, and very-low latency, adequate for URLLC. The proposal puts together RLCs using GRAND transmitted over mMIMO. The coded schemes attain a large gain (3.5 – 4.7 dB) in respect to the uncoded transmission,

significantly outperform the polar-coded counterparts with state-of-the-art list decoding, and for most of the SNR range GRAND delivers a much faster decoding time, thus reducing decoding latency, in addition to the delay savings for not requiring the use of interleavers. Performance can be traded-off with decoding time, and one can get decoding speeds up to 12 times faster than the schemes with polar codes, while still beating the performance of the polar code.

ACKNOWLEDGMENTS

This work has been funded by Instituto de Telecomunicações and FCT/MCTES (Portugal) through national funds and when applicable co-funded EU funds under the project UIDB/50008/2020. Sahar Allahkaram is funded by ISCTE-IUL with a Merit Scholarship awarded by the ISCTE School of Technology and Architecture (ISTA).

REFERENCES

- [1] P. Nouri, H. Alves, M. A. Uusitalo, O. Alcaraz López, and M. Latva-aho, "Machine-type wireless communications enablers for beyond 5G: Enabling URLLC via diversity under hard deadlines," *Computer Networks*, vol. 174, no. 3, p. 107227, 2020.
- [2] W. An, M. Médard, and K. R. Duffy, "Keep the bursts and ditch the interleavers," *IEEE Transactions on Communications*, vol. 70, no. 6, pp. 3655–3667, 2022.
- [3] S. M. Abbas, M. Jaleleddine, and W. J. Gross, "GRAND for Rayleigh fading channels." [Online]. Available: <https://arxiv.org/abs/2205.00030>
- [4] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, pp. 379–423, 1948.
- [5] R. Gallager, "The random coding bound is tight for the average code (corresp.)," *IEEE Transactions on Information Theory*, vol. 19, no. 2, p. 244–246, Mar 1973.
- [6] Y. Polyanskiy, H. V. Poor, and S. Verdú, "Channel coding rate in the finite blocklength regime," *IEEE Transactions on Information Theory*, vol. 56, no. 5, pp. 2307–2359, 2010.
- [7] W. Yang, G. Durisi, T. Koch, and Y. Polyanskiy, "Quasi-static multiple-antenna fading channels at finite blocklength," *IEEE Transactions on Information Theory*, vol. 60, no. 7, pp. 4232–4265, 2014.
- [8] 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, Jul. 2019.
- [9] K. R. Duffy, M. Médard, and W. An, "Guessing random additive noise decoding with symbol reliability information (SRGRAND)," *IEEE Transactions on Communications*, vol. 70, no. 1, pp. 3–18, 2022.
- [10] K. R. Duffy, A. Solomon, K. M. Konwar, and M. Médard, "5G NR CA-polar maximum likelihood decoding by GRAND," *54th Annual Conf. on Info. Sciences and Systems (CISS)*, 2020.
- [11] K. R. Duffy, "Ordered reliability bits guessing random additive noise decoding," in *Proc. of IEEE Inter. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Toronto, Canada, Jun 2021, p. 8268–8272.
- [12] D. J. C. MacKay, *Information Theory, Inference and Learning Algorithms*. Cambridge, UK: Cambridge University Press, 2003.
- [13] F. Monteiro, "Lattices in MIMO spatial multiplexing: Detection and geometry," Ph.D. dissertation, University of Cambridge, United Kingdom, 2012.
- [14] F. Rosário, F. A. Monteiro, and A. Rodrigues, "Fast matrix inversion updates for massive MIMO detection and precoding," *IEEE Signal Processing Letters*, vol. 23, no. 1, pp. 75–79, 2016.
- [15] F. A. Monteiro and I. J. Wassell, "Recovery of a lattice generator matrix from its Gram matrix for feedback and precoding in MIMO," in *2010 4th Inter. Symp. on Comm., Control and Sig. Proc. (ISCCSP)*, 2010.
- [16] Y. Jiang, M. K. Varanasi, and J. Li, "Performance analysis of ZF and MMSE equalizers for MIMO systems: An in-depth study of the high SNR regime," *IEEE Transactions on Information Theory*, vol. 57, no. 4, pp. 2008–2026, 2011.
- [17] I. Tal and A. Vardy, "List decoding of polar codes," *IEEE Transactions on Information Theory*, vol. 61, no. 5, pp. 2213–2226, 2015.

Chapter 3

Constrained Symbol-Level Noise-Guessing Decoding with Antenna Sorting for Massive MIMO

Journal: IEEE Open Journal of the Communications Society

Publication Date: November 2025

Volume: 6

Pages: 1-18

DOI: 10.1109/OJCOMS.2025.3630551 URL:

<https://ieeexplore.ieee.org/document/11235519>

Received 23 October 2025; accepted 3 November 2025. Date of publication 7 November 2025; date of current version 14 November 2025.

Digital Object Identifier 10.1109/OJCOMS.2025.3630551

Constrained Symbol-Level Noise-Guessing Decoding With Antenna Sorting for Massive MIMO

SAHAR ALLAHKARAM^{1,2} (Student Member, IEEE), FRANCISCO A. MONTEIRO^{1,2} (Member, IEEE), AND IOANNIS CHATZIGEORGIOU³ (Senior Member, IEEE)

¹Instituto de Telecomunicações, 1049-001 Lisbon, Portugal

²ISCTE—Instituto Universitário de Lisboa, 1649-026 Lisboa, Portugal

³School of Computing and Communications, Lancaster University, LA1 4YW Lancaster, U.K.

CORRESPONDING AUTHOR: F. A. MONTEIRO (e-mail: francisco.monteiro@lx.it.pt)

This work was funded by national funds through FCT — Fundação para a Ciência e a Tecnologia, I.P., and, when eligible, co-funded by EU funds under project/support ID/50008/2025 – Instituto de Telecomunicações, with DOI identifier <https://doi.org/10.54499/UID/50008/2025>. Sahar Allakhkaram is funded by a merit scholarship from ISCTE – Instituto Universitário de Lisboa.

ABSTRACT Supporting ultra-reliable and low-latency communication (URLLC) is a challenge in current wireless systems. Channel codes that generate large codewords improve reliability but necessitate the use of interleavers, which introduce undesirable latency. Only short codewords can eliminate the requirement for interleaving and reduce decoding latency. This paper suggests a coding and decoding method which, when combined with the high spectral efficiency of spatial multiplexing, can provide URLLC over a fading channel. Random linear coding and modulation are used to transmit information over a massive multiple-input multiple-output (mMIMO) uplink channel, followed by zero-forcing detection and guessing random additive noise decoding (GRAND) at a receiver. This work considers symbol-level GRAND, which is a variant of GRAND that was originally proposed for single-antenna systems employing square M -ary quadrature amplitude modulation, and generalizes it to schemes that combine spatial multiplexing with any M -ary modulation method. The paper studies the impact of the orthogonality defect of the underlying mMIMO lattice on symbol-level GRAND, and proposes to leverage side-information that comes from the mMIMO channel-state information and relates to the reliability of each receive antenna. Additionally, a lightweight membership test is introduced to reduce the number of error patterns that undergo full membership tests, by making use of a row in the parity-check matrix that eliminates candidate error patterns. All proposals reduce the decoding speed without compromising the decoding performance. The proposed decoder operating at the symbol level, when combined with antenna sorting and syndrome-constrained decoding, has the potential to reduce complexity by 90% when compared to bit-level GRAND in some of the tested configurations.

INDEX TERMS Antenna sorting, constrained decoding, guessing random additive noise decoding (GRAND), massive multiple-input multiple-output (mMIMO), random linear codes (RLCs), symbol-level decoding, ultra-reliable and low-latency communications (URLLC).

I. INTRODUCTION

IN ADDITION to other crucial requirements for the sixth generation (6G) of wireless networks, such as low energy consumption, high scalability, stability, security, and ubiquitous connectivity, the physical layer of wireless communications will have to significantly contribute to the goal of ultra-reliable and low-latency communications (URLLC). To meet the important requirements of applications like the industrial Internet of Things (IIoT), virtual reality, or

self-driving cars, URLLC's main objectives are to reduce latency to 1 ms while concurrently guaranteeing at least 99.999% dependability [1]. Using error-correcting codes with short codewords is one way of achieving the sought low-latency objective, because that allows to discard the interleavers that are typically employed in wireless links to make the errors look independent and identically distributed (i.i.d.) [2]. However, developing codes with large codewords was prioritized in pre-5G systems to reach Shannon's

capacity [3], [4]. An interest in codes from the 1960s, such as Reed-Solomon and BCH codes, was rekindled, aiming at URLLC applications [5]. While these codes can have short codewords, they only exist for a limited number of code rates. Contrary to that, random linear codes (RLCs) can be constructed with any code rate, even though decoding long RLCs is impractical [6].

It was previously known that short RLCs could be decoded using trellis decoding [7], [8], [9], [10], [11], [12], [13], [14] or information set decoders [15], however, given the historical emphasis on capacity-achieving codes (with long codewords), that path of research seems to have been abandoned by the coding community. Recently, noise-guessing decoding has been proposed as a universal decoding technique for codes with moderate length or sufficiently high rate, which are particularly suited for wireless URLLC [16]. The method, known as guessing random additive noise decoding (GRAND) allows maximum likelihood (ML) decoding with a considerably reduced complexity, chiefly because it focuses on “decoding the noise” rather than the codewords, by taking advantage of the entropy of the noise being much lower than the entropy of the codewords. The sole requirement is that a code membership test exists to decide whether some word is a valid codeword. Consequently, GRAND can perform ML decoding of binary or nonbinary linear block codes (such as polar codes [17], [18], BCH codes [18], [19], or Hamming codes) without the need to compute a trellis or store a large table.

Also, GRAND opened doors to using RLCs, known to be capacity-achieving in the asymptotic regime (i.e., with infinite length codewords) in the binary symmetric channel (BSC) [3], [4], and they also reach capacity in the finite-blocklength regime [16], [20], [21], which is the regime of interest for URLLC applications. Several recent research works have shown that RLCs supersede the performance of polar codes of the same length and rate in the classical case [22], [23]. Most importantly, while off-the-shelf nonrandom codes, such as polar codes, do not exist for any desired pair of code length and code rate, one has great flexibility of choice regarding the length and code rate when employing RLCs with GRAND [2], [16], [24]. For higher spectral efficiency, GRAND has been proposed in combination with massive multiple-input multiple-output (mMIMO) in [23]. The ideas behind GRAND have also been adapted to allow the decoding of quantum random linear codes in a practical manner [25], [26], and also to decode quantum stabilizer codes with a given structure (i.e., nonrandom known codes) [27].

Several works have recently proposed enhancements to reduce the complexity of decoders based on guessing techniques. Guessing codeword decoding (GCD) is an ML decoding algorithm [28] that orders error patterns based on a newly defined metric that serves as a “soft Hamming weight” for each candidate error pattern, and that allows them to be sorted in decreasing probability; moreover, the decoder only

tests error patterns that have a given structure associated with the systematic form of the code (which is always available via Gaussian elimination). The ordered reliability direct error pattern testing (ORDEPT) decoder [29] introduced a technique to test error patterns that are at a Hamming distance-one of a currently tested error pattern. This early tests will change the optimal order of testing error patterns and may lead to finding a codewords that is not the ML solution. To compensate, the authors proposed populating a list of error candidates that lead to valid codewords and only make a decision in the end. In doing that, ORDEPT attains ML performance, and outperforms both the (sub-optimal) ordered reliability bits GRAND (ORBGRAND) [18] and the even more complex first soft GRAND decoder [30]. Soft-output GRAND (SO-GRAND) [31] employs a list decoder, which also keeps track of the probability of the correct ML codeword being in the accumulated list of candidate codewords at any given moment, allowing to dynamically adjust the list size by defining a threshold for that probability. Symbol-level GRAND has been recently proposed in [32] for single-input single-output (SISO) block fading channels, of which the additive Gaussian noise (AWGN) channel is a special case. Symbol-level GRAND attains significantly faster decoding than the original bit-level GRAND. In [2], the authors have suggested modifying GRAND to use knowledge about the adopted modulation scheme for channels with memory. Symbol-level GRAND takes a different approach: it relies on a closed-form expression for the probability that the input stream of bits contains a specific combination of bit strings representing various constellation symbols. These constellation symbols have different numbers of nearest and next-nearest neighbors. When the transmission is done over a block fading channel, the expression allows to order the error patterns according to their likelihood.

With the aim of attaining the URLLC objectives, this work proposes a transmission scheme that integrates RLC encoding and symbol-level GRAND, aided by antenna sorting, into a mMIMO spatial multiplexing system. Both zero-forcing (ZF) detection and minimum mean square error (MMSE) detection are considered, however, as it would be expected in mMIMO setups, MMSE provides no benefit regarding performance or complexity reduction in respect to the ZF detection filter. While RLCs cater for the sought-after high reliability and GRAND offers reduced decoding complexity, mMIMO techniques enable high spectral efficiency through spatial multiplexing. We explain that symbol-level GRAND can be directly extended to mMIMO, if strong channel hardening (CH) conditions are assumed. Furthermore, we show that the considered mMIMO system can cope with adverse CH conditions, if the symbols at the output of the linear detector are ordered according to their reliability, which can be derived from channel state information (CSI).

The optimized re-ordering of symbols can be seen as an antenna sorting problem. Antenna sorting has been known to greatly impact the detection performance of MIMO systems

that use a small number of spatial streams. Optimal antenna sorting strategies that rely on the notion of the *effective* signal-to-noise ratio (SNR) of a stream at the output of the MIMO detector [33], [34] have been devised for different MIMO detection methods. For example, antenna sorting was used to increase the performance of V-BLAST detectors [34], or to simultaneously improve the performance and reduce the complexity of sphere decoders [35]. Any extra information regarding the a priori probability of the error patterns should be used, and that concept is at the core of the proposals in this paper. In this paper, we use the effective SNR after ZF or MMSE detection filters as a sorting metric akin to the reliability of the QAM symbols, which carry the bit strings that build up the codewords. The proposed antenna sorting method further reduces the complexity of symbol-level GRAND. Moreover, we introduce a constraint based on a carefully selected row of the parity-check matrix to discard incompatible error patterns. While related ideas appear in [36], [37], our method offers a simpler and more easily implementable alternative with minimal modification to the decoder, which, combined with antenna sorting, can reduce the complexity by more than 90% in some configurations when compared to bit-level GRAND.

In summary, the contributions of the paper are as follows:

- 1) Symbol-level GRAND, introduced in [32] for RLCs combined with Gray-coded square M -QAM for transmission over SISO block Rayleigh fading channels, has been abstracted into a generalized framework that can be applied to any M -ary modulation scheme for which a constellation can be defined.
- 2) Antenna sorting has been proposed as a simple technique that can be easily integrated into symbol-level GRAND to make it fit for MIMO channels. The sorting that results from MMSE detection is compared with the one arising from ZF detection, and it is shown that ZF is capable of offering equal performance and complexity reduction at any SNR in mMIMO scenarios.
- 3) A general bit-level lightweight membership test has been developed that can be applied to any GRAND decoder. A study of this technique is presented that leads to finding the rules that maximize the number of eliminated error patterns to be tested. This proposal is combined with symbol-level sorted GRAND to filter out a large number of considered error patterns that would have otherwise been subjected to the standard – more computationally intensive – membership test, thus reducing the overall computational requirements of the noise-guessing decoder.
- 4) The model of a system, which encompasses a transmitter employing RLC encoding, M -ary modulation and spatial multiplexing, a mMIMO channel and a receiver that uses ZF or MMSE detection and GRAND, has been created. It is shown that block error rate (BLER) measurements have been obtained for the

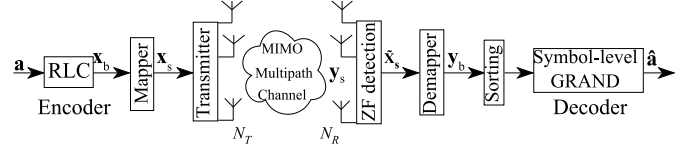


FIGURE 1. System model for coded mMIMO URLLC.

case when the GRAND algorithm is either bit-level GRAND [16] or symbol-level GRAND [32], and is optionally combined with antenna sorting. The ideal case of perfect channel hardening (PCH) has been used as a lower-bound benchmark. The impact of the chosen GRAND algorithm on complexity has also been investigated.

The paper starts by describing the system model in detail in Section II. Section III shows how symbol reliability can be obtained from CSI and how antenna sorting should be implemented. Section IV generalizes symbol-level GRAND for any M -ary modulation scheme and integrates it with antenna sorting. Section V introduces a lightweight membership test that constrains the number of error patterns queried by symbol-level GRAND and, consequently, reduces computational cost. Section VI describes how symbol-level GRAND can be adapted to Gray-coded square M -QAM in the case of PCH, and discusses memory requirements. Section VII shows performance and complexity results of the proposed scheme, and Section VIII summarizes key conclusions.

II. SYSTEM MODEL FOR CODED MASSIVE MIMO

A coded uplink massive MIMO system is considered, making use of an RLC encoder at the transmitter and symbol-level GRAND at the receiver. A block of k information bits is mapped onto a n -bit codeword and sent “over the air” via spatial multiplexing, as illustrated in Fig. 1. This process may be repeated when transmitting longer information streams by dividing the bit stream into blocks of size k .

A. RLC ENCODING AND SPATIAL MULTIPLEXING WITH MMIMO

A block a of k i.i.d. information bits is linearly encoded into a codeword $\mathbf{x}_b$ of length n using a systematic binary RLC with rate $R = k/n$, denoted by $\text{RLC}(n, k)$. The $\text{RLC}(n, k)$ defines a codebook $\mathcal{C}$ with $2^k = 2^{nR}$ codewords of length n , which constitutes a linear subspace of the discrete vector space $\mathbb{F}_2^n$. Although the minimum Hamming distance between two codewords determines the error correction capability of (random or nonrandom) linear block codes (LBCs) at high-SNR, in very noisy conditions the minimum distance is not as relevant in determining the performance of *long* RLCs. In fact, when the objective is to approach the Shannon limit, the code needs to decode beyond the minimum distance [38, Ch.13]. This work focuses on the finite blocklength, using short RLCs. The $\text{RLC}(n, k)$ is described by a *random* binary

generator matrix $\mathbf{M} \in \mathbb{F}_2^{k \times n}$, which acts as the basis matrix for the code subspace, such that $\mathcal{C} = \{\mathbf{x}_b = a\mathbf{M} : a \in \mathbb{F}_2^k\}$. The generator matrix of a systematic RLC is of the form $\mathbf{M} = [\mathbf{I}_k \mid \mathbf{P}]$, where $\mathbf{P} \in \mathbb{F}_2^{k \times (n-k)}$ is a random binary matrix, and $\mathbf{I}_k$ is the $k \times k$ identity matrix responsible for the systematic part of the encoding. The corresponding parity-check matrix will take the form $\mathbf{C} = [\mathbf{P}^T \mid \mathbf{I}_{n-k}] \in \mathbb{F}_2^{(n-k) \times n}$.

The n bits, $b_1, \dots, b_n$, of a codeword are input to a M -ary constellation. The mapper divides the sequence of n bits into L strings of $\log_2(M)$ bits, that is, $L = n/\log_2(M)$, and maps the L strings onto L complex-valued symbols, $s_1, \dots, s_L$, taken from the alphabet $\mathcal{A} \in \mathbb{C}$ of the M -ary constellation. The cardinality of $\mathcal{A}$ is $|\mathcal{A}| = M$. We denote the n -bit codeword and the sequence of L modulated symbols by $\mathbf{x}_b = [b_1, \dots, b_n]$ and $\mathbf{x}_s = [s_1, \dots, s_L]^T$, respectively. Furthermore, we denote by $\mathcal{S}(s_i)$ the string of $\log_2(M)$ bits that has been mapped onto symbol s_i . Thus, the codeword $\mathbf{x}_b$ can also be written as $\mathbf{x}_b = [\mathcal{S}(s_1), \dots, \mathcal{S}(s_L)]$. If E_b represents the energy per information bit, then $(k/n)E_b$ is the energy per codeword bit, and $\log_2(M)(k/n)E_b$ is the energy per string of $\log_2(M)$ bits, which also corresponds to the energy per symbol, $\mathbb{E}\{|s_i|^2\}$.

The system can be designed to allow the transmission of N_c codewords in each MIMO channel used. This implies that, when a specific cardinality M is employed for the modulation, the number of transmit antennas is $N_T = N_c L$. Without loss of generality, and to keep the notation simple, we will describe the system for $N_c = 1$, where one MIMO burst transmitted from the N_T antennas contains one codeword only (i.e., $N_T = L$). Later, in Section IV-B, the generalization for $N_c > 1$ will be commented on. A system with $N_c < 1$ can also be made operational by adding buffers both at the transmitter and at the receiver, hence creating a full separation between the mMIMO physical layer and channel coding and decoding such that symbol-level GRAND only starts decoding when the L symbols corresponding to a codeword have been received.

The coded signal $\mathbf{x}_s$ is transmitted over a MIMO Rayleigh fading channel, characterized by the matrix $\mathbf{H} \in \mathbb{C}^{N_T \times N_R}$, where $N_R \gg N_T$ is the number of antennas fitted at the receiver. The received signal $\mathbf{y}_s = [y_1, \dots, y_{N_R}]^T$ is given by

$$\mathbf{y}_s = \mathbf{H}\mathbf{x}_s + \mathbf{n}, \quad (1)$$

where $\mathbf{n} = [n_1, \dots, n_{N_R}]^T$ is the additive noise. The entries in both $\mathbf{H}$ and $\mathbf{n}$ are i.i.d. random variables taken from complex normal distributions: the ones in $\mathbf{H}$ are taken from $\mathcal{CN}(0, 1)$, and those in $\mathbf{n}$ are taken from $\mathcal{CN}(0, \sigma_n^2)$. The noise power at each receive antenna is $\sigma_n^2 = N_0$, where N_0 is the unilateral power spectral density of the noise. In this framework, the ergodic SNR per transmit antenna (i.e., per layer) at the receiver is

$$\text{snr} \triangleq \frac{\mathbb{E}\{|s_i|^2\}}{\sigma_n^2} = \log_2(M)(k/n)(E_b/N_0). \quad (2)$$

The symbols in $\mathcal{A}$ are normalized to unit average energy, so that $\mathbb{E}\{|s_i|^2\} = 1$, and therefore $\text{snr} = \frac{1}{\sigma_n^2}$. The $N_T \times N_R$ matrix $\mathbf{H}$ remains constant during the transmission of $\mathbf{x}_s$ but changes independently from channel use to channel use.

B. LINEAR DETECTION AND SYMBOL-LEVEL GRAND

The ZF and MMSE linear detection schemes are considered [34], [39]. The first amounts to applying the Moore-Penrose pseudo-inverse

$$\mathbf{W}_{\text{ZF}} = (\mathbf{H}^H \mathbf{H})^{-1} \mathbf{H}^H, \quad (3)$$

while the MMSE filter is given by

$$\mathbf{W}_{\text{MMSE}} = \left(\mathbf{H}^H \mathbf{H} + \frac{1}{\text{snr}} \mathbf{I}_{N_T} \right)^{-1} \mathbf{H}^H. \quad (4)$$

In the ZF case one obtains at the receiver

$$\mathbf{W}_{\text{ZF}} \mathbf{y}_s = \mathbf{I}_{N_T} \mathbf{x}_s + \underbrace{\mathbf{W}_{\text{ZF}} \mathbf{n}}_{\mathbf{u}}, \quad (5)$$

where $\mathbf{I}_{N_T}$ is the $N_T \times N_T$ identity matrix, and $\mathbf{u} \in \mathbb{C}^{N_T}$ denotes the new (now correlated) noise vector after ZF filtering. Although the performance of ZF detection is rather poor in symmetric MIMO, where $N_R = N_T$, it attains quasi-optimal performance in highly asymmetric MIMO, for example, when $N_R \gg N_T$ in an uplink scenario [39]. In this scenario, which is considered in our system model, the instantaneous SNR for each channel realization approaches its ergodic value at each received data stream after ZF detection. At the same time, the large value of N_R boosts the receiver array gain.

After applying $\mathbf{W}_{\text{ZF}}$ in (3) or $\mathbf{W}_{\text{MMSE}}$ in (4), which can both be abstracted to $\mathbf{W}$, a quantization operation $\mathcal{Q}(\cdot)$ is made to the M -ary constellation to obtain the sequence of detected symbols $\tilde{\mathbf{x}}_s = \mathcal{Q}(\mathbf{W}\mathbf{y}_s) = [\tilde{s}_1, \dots, \tilde{s}_L]^T$, which is an estimate of $\mathbf{x}_s$ corrupted by noise. The detected symbols $\tilde{s}_1, \dots, \tilde{s}_L$ are demapped to bit strings $\mathcal{S}(\tilde{s}_1), \dots, \mathcal{S}(\tilde{s}_L)$ and reconstruct a word of n bits, denoted by $\mathbf{y}_b$. The relationship between the reconstructed word $\mathbf{y}_b$ at the receiver and the codeword $\mathbf{x}_b$ at the transmitter is $\mathbf{y}_b = \mathbf{x}_b \oplus \mathbf{e}_b$, where $\mathbf{e}_b$ is the error pattern that has corrupted the transmitted codeword. The operation $\oplus$ denotes modulo-2 addition.

The most common method of decoding LBCs, including RLCs, is syndrome decoding, which makes use of the parity-check matrix $\mathbf{C} \in \mathbb{F}_2^{(n-k) \times n}$ to generate the syndrome $\mathbf{s} = \mathbf{C}\mathbf{y}_b \in \mathbb{F}_2^{(n-k)}$. Since all codewords verify $\mathbf{s} = \mathbf{0}$, this can be used as a simple *membership test*. Syndrome decoding achieves ML decoding, but a lookup table is required for the storage of possible syndromes and respective coset leaders. For example, suppose that we wish to correct received words that contain up to a threshold of w_{th} bit errors. The number of error patterns that need to be considered is given by $\sum_{t=0}^{w_{\text{th}}} \binom{n}{t}$. However, the number of all possible syndromes is 2^{n-k} . For large values of w_{th} and high code rates, that is $R \rightarrow 1$ and thus $n \rightarrow k$, the relationship $\sum_{t=0}^{w_{\text{th}}} \binom{n}{t} \gg 2^{(n-k)}$ holds, therefore the error correction capability of the code

is limited because a wide variety of error patterns result in the same syndrome. Choosing the coset leader associated with each particular syndrome depends on side information regarding the *a priori* probability of each error pattern. Over AWGN, the chosen coset leaders should be the error patterns with the lowest Hamming weight, which leads to ML decoding.

In the proposed transmission scheme, the word $\mathbf{y}_b$ is input to symbol-level GRAND, which attempts to estimate $\mathbf{e}_b$ and infer $\mathbf{x}_b$ using $\hat{\mathbf{x}}_b = \mathbf{y}_b \oplus \hat{\mathbf{e}}_b$, where $\hat{\mathbf{x}}_b$ and $\hat{\mathbf{e}}_b$ are estimates of $\mathbf{x}_b$ and $\mathbf{e}_b$, respectively. The first k of the n bits of the estimated codeword $\hat{\mathbf{x}}_b$ form the block of decoded information bits $\hat{\mathbf{a}}$, as shown in Fig. 1.

III. SYMBOL RELIABILITY AND ANTENNA SORTING

A. EFFECTIVE POST-PROCESSING SNR

After a ZF or MMSE filter, the decisions made by the quantizer $\mathcal{Q}(\cdot)$ to obtain $\hat{\mathbf{x}}_s$ are perturbed by the modified noise vector $\mathbf{u}$ that appears in (5). One can show that the output SNR after ZF or MMSE detection of the N_T incoming signals streams depends on the instantaneous channel realization $\mathbf{H}$ in the following manner [33] [34, Sec. 3.1.3–3.1.4]:

$$\begin{aligned} \text{snr}_i^{(\text{ZF})} &= \frac{\text{snr}}{[(\mathbf{H}^H \mathbf{H})^{-1}]_{ii}} \\ &= \frac{1}{[\mathbf{G}^{-1}]_{ii}} \text{snr} = g_i \text{snr}, \quad 1 \leq i \leq N_T \end{aligned} \quad (6)$$

$$\text{snr}_i^{(\text{MMSE})} = \frac{\text{snr}}{[(\mathbf{H}^H \mathbf{H} + \frac{1}{\text{snr}} \mathbf{I}_{N_T})^{-1}]_{ii}} - 1, \quad 1 \leq i \leq N_T. \quad (7)$$

In the ZF case in (6), the g_i are defined as the inverses of the elements in the diagonal of $\mathbf{G}^{-1}$, for $i = 1, \dots, N_T$. Note that $\mathbf{G} = \mathbf{H}^H \mathbf{H}$ is the Gram matrix of the lattice spanned by the columns of $\mathbf{H}$ (e.g., [40]). The value of each g_i in (6) should be as large as possible. In the case of a diagonal $\mathbf{G}$, that maximization happens for a $\mathbf{G}$ with large diagonal elements. If the energy spills over the diagonal, the elements in the diagonal get smaller due to energy conservation arguments. This corresponds to having the off-diagonal elements of $\mathbf{G}$ no longer close to zero due to non-orthogonality of the column vectors of $\mathbf{H}$. Note that a different definition of snr is used in [33], but that does not change the relation in (6).

Expressions (6) and (7) provide soft information about the reliability of each detected symbol, given the one-to-one relation between symbols and antenna streams. This information will be central to sorting the received symbols so that symbol-level GRAND can perform its guesswork of the transmitted symbols starting from the least reliable symbol to the most reliable one. While both expressions provide the absolute post-processing SNRs at each antenna, only the *relative* information about their magnitude ordering is required as a side information to be passed to the

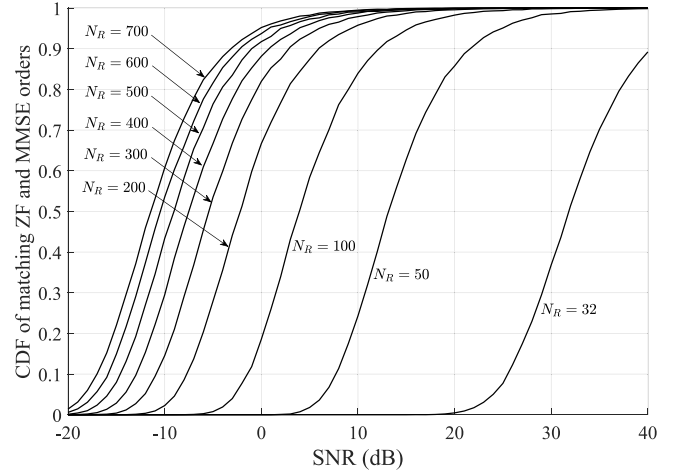


FIGURE 2. Cumulative distribution function of the number of exact matches between the ordered vectors defined by the sorting of the antennas when using ZF detection and when using MMSE detection filters. Different cases are represented for the configurations with $N_T = 32$ transmit antennas and a variable number of receive antennas, N_R . Simulation results obtained considering 5000 channel realizations at each SNR (computed at each integer value of the dB horizontal axis).

symbol-level GRAND decoder. It is known that in highly asymmetrical mMIMO configurations, MMSE delivers the same performance as ZF, while having a slightly more complex implementation. In our proposed receiver, it is important to determine whether the sorting provided by (6) and (7) has an impact on the symbol-level GRAND decoding complexity. At high snr, (6) tends to (7), and therefore the orders resulting from both should be equal. However, when snr decreases, the sorting orders could differ. The proportion of instances in which the ordered vectors produced by ZF matched those generated by MMSE was obtained and is shown in Fig. 2. The ordering vectors generated by sorting $\text{snr}_i^{(\text{ZF})}$ and $\text{snr}_i^{(\text{MMSE})}$ for the N_T antenna streams, by means of (6) and (7), were compared using 5000 channel realizations. The cumulative distribution function (CDF) shows that both resulting orders are the same at high SNR. As the SNR lowers, the ordered vectors start to differ. The cliff region of the CDF curves can be displaced to lower SNRs by applying an increasing number of receive antennas. The SNR range presented in Fig. 2, as well as the range of values for N_T and N_R , are covered in Section VII.

The sorting orders resulting from (6) and (7) are trivially the same as $\text{snr} \rightarrow \infty$ because (7) tends to (6). To investigate how sorting is impacted for $\text{snr} \rightarrow 0$, let us write the denominator in (7) as $(\mathbf{G} + \frac{1}{\text{snr}} \mathbf{I}_{N_T})^{-1} = (\frac{1}{\text{snr}} (\mathbf{I}_{N_T} + \text{snr} \mathbf{G}))^{-1} = \text{snr} (\mathbf{I}_{N_T} + \text{snr} \mathbf{G})^{-1}$. Therefore, the first term in (7), the one responsible for the sorting order, is

$$\frac{\text{snr}}{[(\mathbf{G} + \frac{1}{\text{snr}} \mathbf{I}_{N_T})^{-1}]_{ii}} = \frac{1}{[(\mathbf{I}_{N_T} + \text{snr} \mathbf{G})^{-1}]_{ii}}.$$

By using the Neumann series $(\mathbf{I}_{N_T} - \mathbf{G})^{-1} = \sum_{k=0}^{\infty} \mathbf{G}^k$, for low snr, one has $(\mathbf{I}_{N_T} + \text{snr} \mathbf{G})^{-1} = \mathbf{I}_{N_T} - \text{snr} \mathbf{G} + O(\text{snr}^2)$. Consequently, when $\text{snr} \rightarrow 0$, the first term in (7) tends to an identity matrix, turning equal all diagonal indexes i . Hence, at low SNR, the ZF sorting will be defined by the

g_i gains, while the MMSE sorting becomes irrelevant. In the mMIMO scenario, given that $\mathbf{G} \rightarrow \mathbf{I}_{N_T}$, the same effect will occur in ZF, eliminating the need for antenna sorting regardless of the linear detector used (see Section III-C). For a given snr, a larger N_R gives rise to a more orthogonal channel and therefore to a Gram matrix $\mathbf{G}$ that is closer to a scaled identity matrix. In that case, the denominators in (6) and (7) are more similar, leading to an increasing matching, as observed in the results in Fig. 2.

This section established that antenna sorting using ZF is equivalent to MMSE in mMIMO systems. For this reason, expression (6) will be used for the calculation of the SNR at the output of each receive antenna under the assumption of perfect CSI. A derivation of (6) is presented in Appendix-A. The impact of imperfect CSI on the SNR expression is briefly explored and discussed in Appendix-B.

B. LATTICE GEOMETRY WITH A FINITE NUMBER OF ANTENNAS

The geometry of ZF detection fully determines its detection performance. For ZF to approach ML detection using the Voronoi regions of the underlying real MIMO lattice, it is necessary that the so-called ZF detection region matches the Voronoi region with a low discrepancy (e.g., [34]). This is the fundamental cause for ZF detection becoming optimal as N_R increases. When $N_R \rightarrow \infty$ the lattice spanned by the columns of $\mathbf{H}$ would be a perfectly orthogonal lattice, and ZF would be optimal. Analytically, this effect can be captured by measuring the effect of the effective noise $\mathbf{u}$ in (5). The effect of the ZF filter on that noise power can be tracked by considering the autocorrelation matrix of the new noise $\mathbf{u} = \mathbf{W}_{ZF}\mathbf{n}$, calculated as:

$$\begin{aligned} \mathbf{R}_u &= \mathbb{E}\{\mathbf{u}\mathbf{u}^H\} = \mathbb{E}\{(\mathbf{W}_{ZF}\mathbf{n})(\mathbf{W}_{ZF}\mathbf{n})^H\} \\ &= \mathbb{E}\{(\mathbf{W}_{ZF}\mathbf{n})(\mathbf{n}^H(\mathbf{W}_{ZF})^H)\} \\ &= \mathbf{W}_{ZF}\mathbb{E}\{\mathbf{n}\mathbf{n}^H\}(\mathbf{W}_{ZF})^H = \sigma_n^2\mathbf{W}_{ZF}(\mathbf{W}_{ZF})^H, \end{aligned} \quad (8)$$

where the autocorrelation of the original Gaussian noise, $\mathbb{E}\{\mathbf{n}\mathbf{n}^H\} = \mathbf{R}_n = \sigma_n^2\mathbf{I}_{N_R}$, has been used. Replacing the Moore-Penrose pseudo-inverse from (3) in (8), and using the definition of the Gram matrix, it is possible to obtain

$$\mathbf{R}_u = \sigma_n^2(\mathbf{H}^H\mathbf{H})^{-1} = \sigma_n^2\mathbf{G}^{-1}. \quad (9)$$

From both (6) and (9), one can see that ZF detection always causes noise enhancement in the case of real-world channels (with a finite N_R). The noise amplification of ZF detection can be geometrically interpreted using lattices. Let $\mathcal{G} = \mathbb{Z} + i\mathbb{Z}$ denote the set of Gaussian integers. A complex lattice is defined as $\Lambda = \{\mathbf{H}\mathbf{z} : \mathbf{z} \in \mathcal{G}^{N_T \times 1}\}$. For a lattice basis $\mathbf{H} \in \mathbb{C}^{N_R \times N_T}$, the lattice has rank N_T , and lives in a N_R -dimensional space. The volume of the *fundamental region* of the lattice is $\text{vol}(\Lambda) = \sqrt{\det(\mathbf{H}^H\mathbf{H})} = \sqrt{\det(\mathbf{G})}$. In the case of square matrices, this simplifies to $\text{vol}(\Lambda) = \det(\mathbf{H})$. In MIMO detection it is preferable to use the real-valued

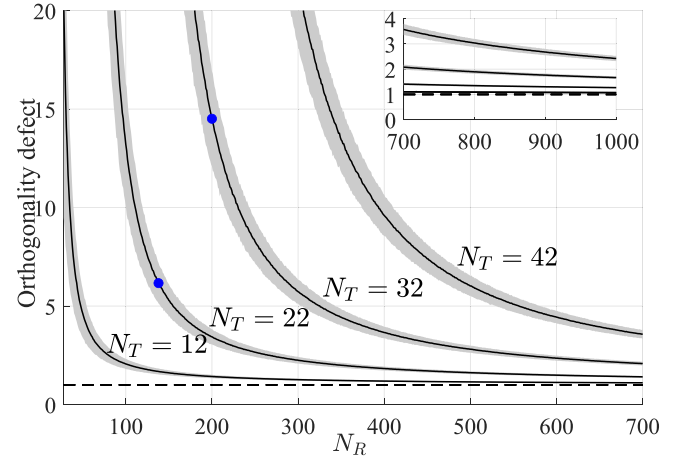


FIGURE 3. Evolution of $od(\mathcal{H})$ as a function of the number of receive antennas. The blue dots indicate the operating points of the two systems that will be assessed in Section VII that are closer to (but still far from) the PCH regime. The shaded region corresponds to two standard deviations of od .

equivalent lattice, defined as $\Lambda_{\mathbb{R}} = \{\mathcal{H}\mathbf{z} : \mathbf{z} \in \mathbb{Z}^{2N_T \times 1}\}$, having rank $2N_T$, and living in $2N_R$ dimensions. It uses the equivalent *real-valued* basis $\mathcal{H} \in \mathbb{R}^{2N_R \times 2N_T}$, constructed from the complex basis $\mathbf{H}$ [34]. Noise amplification is reduced when a lattice's fundamental region is closer to orthogonal. To measure how orthogonal a lattice $\Lambda_{\mathbb{R}}$ is, one can use the so-called *orthogonality defect* (OD) [41], which for a lattice spanned by a real basis $\mathcal{H}$ is defined as:

$$od(\mathcal{H}) = \frac{\prod_{i=1}^{2N_T} \|\mathcal{H}(:, i)\|}{\text{vol}(\Lambda_{\mathbb{R}})}. \quad (10)$$

The value of $od(\mathcal{H})$ is always greater than or equal to one, and can only attain the unit if the columns of $\mathbf{H}$ are orthogonal to one another. We now use this metric to investigate how N_R and N_T influence the geometry of the mMIMO lattice and, therefore, how far from optimal ZF detection is. Fig. 3 shows how the OD evolves with N_R , for different values of N_T . The figure shows the domain of N_R of more practical significance and an overlaid graph depicts the OD asymptotic convergence to the unit value when the number of receive antennas tends to infinity. The OD is assessed by generating random samples of $\mathcal{H}$, with its real entries drawn from $\mathcal{N}(0, \frac{1}{2})$. This corresponds to generating $2N_T$ random Gaussian vectors in a vector space of N_R (real) dimensions, with the dimension of the vector space being much larger than the number of random vectors drawn (i.e., $N_R \gg N_T$). When this happens, those vectors are mutually orthogonal with high probability. As expected, larger N_T necessitates having a larger N_R in order to maintain the same $od(\mathcal{H})$ value, and as N_R increases, the column vectors of $\mathcal{H}$ tend to be mutually orthogonal.

C. PERFECT CHANNEL HARDENING LOWER-BOUND

There is one specific (and ideal) circumstance in which noise amplification is prevented: when all the column vectors in $\mathbf{H}$ are mutually orthogonal. This occurs when N_T is fixed and $N_R \rightarrow \infty$, leading to the so-called *channel hardening*

effect [42]. For a geometric interpretation of this property, one could consider N_T random Gaussian vectors living in a finite N_R -dimensional space. With high probability any pair of the N_T vectors will be orthogonal to each other. With $N_R \rightarrow \infty$, this probability becomes 1. Let us consider a finite N_R and the special case of a channel matrix where an *ideal* MIMO channel is formed, i.e., a case where all columns of $\mathbf{H}$ are mutually orthogonal. In this case, the Gram matrix, which comprises all inner products $\mathbf{h}_i^H \mathbf{h}_j$, $i = 1, \dots, N_R$, $j = 1, \dots, N_T$, becomes a diagonal matrix of the form:

$$\mathbf{G} = \begin{bmatrix} \|\mathbf{h}_1\|^2 & & 0 \\ & \ddots & \\ 0 & & \|\mathbf{h}_{N_T}\|^2 \end{bmatrix} = N_R \mathbf{I}_{N_T}, \quad (11)$$

given that $\|\mathbf{h}_j\|^2 = \sum_{i=1}^{N_R} |\mathbf{h}_{ij}|^2 = N_R$, for all the N_T vectors. By replacing (11) in (8) one gets that the autocorrelation of the noise after ZF, in the case of PCH, is

$$\mathbf{R}_u = \frac{\sigma_n^2}{N_R} \mathbf{I}_{N_T}, \quad (12)$$

and the power of $\mathbf{u}$ comes as

$$\|\mathbf{u}\|^2 = \text{Tr}(\mathbf{R}_u) = \frac{\sigma_n^2 N_T}{N_R}. \quad (13)$$

It is now possible to establish the equivalent channel model if the $N_T \times N_R$ mMIMO configurations were to attain PCH at those (finite) dimensions:

$$\mathbf{W}_{ZF} \mathbf{y}_s = \mathbf{I}_{N_T} \mathbf{x}_s + \mathbf{u}. \quad (14)$$

In this scenario, one has N_T independent parallel channels, where the effective noise becomes again a vector of independent Gaussian entries. Each of these N_T components of $\mathbf{u}$ has power $|u_i|^2 = \frac{\sigma_n^2}{N_R}$, shedding light on the benefit of having a larger receiver array: with $N_R \rightarrow \infty$ there is a regression to the mean, and the effective noise power vanishes. Note that the snr in (14) is the *input* SNR, at each receive antenna before any baseband processing takes place. This asymptotic regime leads to a uniform *post-processing* $\text{snr}_i^{(ZF)}$ across the N_T spatially multiplexed layers. In that limit, the reliability of all symbols is equal, and therefore sorting would bring no benefit.

IV. GENERALIZED SYMBOL-LEVEL GRAND WITH ANTENNA SORTING

Symbol-level GRAND was originally proposed for a SISO block Rayleigh fading channel [32], thus it cannot be directly applied to a mMIMO system. Taking in consideration the analysis made in the previous section, this section outlines the principles of symbol-level GRAND, generalizes them for any modulation scheme and describes how this method can be integrated into the mMIMO setup by incorporating soft information emanating from the ZF detector.

A. SORTING ERROR PATTERNS GUIDED BY THE CONSTELLATION STRUCTURE

As described in Section II, the n -bit codeword $\mathbf{x}_b$ can be expressed as a sequence of L strings, i.e., $\mathbf{x}_b = [\mathcal{S}(s_i)]_{i=1}^L$, where s_i is a symbol of the M -ary modulation scheme and $\mathcal{S}(s_i)$ is the string of $\log_2(M)$ bits that has been mapped onto s_i . The demodulator outputs $\mathbf{y}_b = [\mathcal{S}(\tilde{s}_i)]_{i=1}^L$, where the hard-detected symbols $\tilde{s}_1, \dots, \tilde{s}_L$ are demapped onto bit strings $\mathcal{S}(\tilde{s}_1), \dots, \mathcal{S}(\tilde{s}_L)$, respectively. The relationship between $\mathcal{S}(\tilde{s}_i)$ at the receiver and $\mathcal{S}(s_i)$ at the transmitter is $\mathcal{S}(\tilde{s}_i) = \mathcal{S}(s_i) \oplus e_i$, for $i = 1, \dots, L$, where e_i is the error string of length $\log_2(M)$ that altered the i -th bit string $\mathcal{S}(s_i)$. The received word $\mathbf{y}_b$ assumes the form:

$$\begin{aligned} \mathbf{y}_b &= [\mathcal{S}(\tilde{s}_i)]_{i=1}^L \\ &= [\mathcal{S}(s_i) \oplus e_i]_{i=1}^L \\ &= [\mathcal{S}(s_i)]_{i=1}^L \oplus [e_i]_{i=1}^L \\ &= \mathbf{x}_b \oplus \mathbf{e}_b, \end{aligned} \quad (15)$$

where $\mathbf{e}_b = [e_i]_{i=1}^L$ is the error pattern that corrupted the transmitted codeword $\mathbf{x}_b$. The objective of algorithms based on GRAND is to obtain an estimate of $\mathbf{e}_b$, denoted by $\hat{\mathbf{e}}_b$, and then add it to $\mathbf{y}_b$ to estimate $\mathbf{x}_b$, that is, $\hat{\mathbf{x}}_b = \mathbf{y}_b \oplus \hat{\mathbf{e}}_b$ provided that $\hat{\mathbf{x}}_b \in \mathcal{C}$.

Bit-level GRAND [16] keeps generating error patterns $\hat{\mathbf{e}}_b$ in descending order of likelihood until an error pattern that satisfies the test $\mathbf{y}_b \oplus \hat{\mathbf{e}}_b \in \mathcal{C}$ is identified. The likelihood of an error pattern is assumed to be a monotonically decreasing function of its Hamming weight. In order to reduce complexity, bit-level GRAND introduced the notion of the abandonment threshold w_{th} . The generation of error patterns is abandoned when all error patterns of weight less than or equal to w_{th} have been tested and a valid codeword has not been found, i.e., $\mathbf{y}_b \oplus \hat{\mathbf{e}}_b \notin \mathcal{C}$ for every $\|\hat{\mathbf{e}}_b\|_1 \leq w_{th}$, where $\|\hat{\mathbf{e}}_b\|_1$ denotes the Hamming weight of $\hat{\mathbf{e}}_b$.

In symbol-level GRAND [32], the condition $\mathbf{y}_b \oplus \hat{\mathbf{e}}_b \in \mathcal{C}$ is replaced by the equivalent expression $[\mathcal{S}(\tilde{s}_i) \oplus \hat{e}_i]_{i=1}^L \in \mathcal{C}$, where $\hat{e}_i$ is the i -th error string of the estimated error pattern, i.e., $\hat{\mathbf{e}}_b = [\hat{e}_i]_{i=1}^L$. Differently from bit-level GRAND, symbol-level GRAND does not generate and verify every realization of $\hat{\mathbf{e}}_b$ for increasing Hamming weight; instead, it creates and tests only realizations of $\hat{\mathbf{e}}_b$, which are composed of error strings that are more likely to occur, as dictated by the properties of the constellation diagram of the modulation scheme and the method for mapping bit strings onto symbols.

The concept of symbol-level GRAND was introduced in [32] through the lens of Gray-coded square M -QAM, as the analysis and expressions therein were tailored to that particular modulation scheme. In this section, we derive a generalized expression for the probability of occurrence of an error pattern when symbol-level GRAND is applied to any M -ary modulation scheme. As in [32], we will initially study the case where the L received symbols have been affected by the same fading coefficient and, therefore, have the same reliability. We will then present a simple add-on

TABLE 1. Examples of error patterns with structure $[L_1 L_2]$ for $M = 16$ and $n = 16$.

Structure $[L_1 L_2]$	Examples of error patterns having structure $[L_1 L_2]$	Weight $(L_1 + 2L_2)$
$[2 0]$	Example 1: 0010 – 1000 – 0000 – 0000 Example 2: 0000 – 0001 – 0001 – 0000 Example 3: 0100 – 0000 – 0000 – 0010	2
$[0 1]$	Example 1: 0000 – 0011 – 0000 – 0000 Example 2: 0000 – 0000 – 1001 – 0000 Example 3: 1100 – 0000 – 0000 – 0000	2
$[1 1]$	Example 1: 0101 – 0001 – 0000 – 0000 Example 2: 0100 – 0000 – 0000 – 1100 Example 3: 0000 – 0000 – 0110 – 1000	3
$[2 1]$	Example 1: 0000 – 1000 – 0011 – 0100 Example 2: 0110 – 0001 – 0001 – 0000 Example 3: 0010 – 0000 – 0100 – 1010	4

to symbol-level GRAND to enable the decoding of symbols that have different reliabilities, as in the considered mMIMO setup.

Let $[L_1 L_2 \dots L_\vartheta]$ describe the structure of an error pattern, where L_w denotes the number of error strings in the error pattern that have Hamming weight w , for $w = 1, \dots, \vartheta$, $1 \leq \vartheta \leq \log_2(M)$ and $\sum_{w=1}^{\vartheta} L_w \leq L$. The Hamming weight of the error pattern is given by $\sum_{w=1}^{\vartheta} wL_w$. Table 1 shows examples of error patterns of length $n = 16$ composed of $L = 4$ error strings, each having length $\log_2(M) = 4$. For clarity, error strings in an error pattern have been separated by dashes. Error strings of weight up to $\vartheta = 2$ are considered in Table 1, thus the structure of an error pattern can be summarized by $[L_1 L_2]$. For example, the three error patterns listed at the bottom of Table 1 have structure $[L_1 L_2] = [2 1]$, i.e., each error pattern consists of $L_1 = 2$ error strings of weight 1 (displayed using blue typeface) and $L_2 = 1$ error string of weight 2 (rendered in red lettering). In all three cases, the weight of the error pattern is $L_1 + 2L_2 = 4$.

If the fading coefficient of the channel is constant over a block of L symbols, error patterns that have the same structure $[L_1 L_2 \dots L_\vartheta]$, have the same probability of occurrence denoted by $P(L_1, L_2, \dots, L_\vartheta)$. Symbol-level GRAND relies on an expression for $P(L_1, L_2, \dots, L_\vartheta)$ to calculate the likelihood of every possible structure – or every structure that satisfies $\sum_{w=1}^{\vartheta} wL_w \leq w_{th}$ if we wish to bound from above the Hamming weight of the considered error patterns. Error patterns are then generated and tested for each structure, from the most likely structure to the least likely structure.

The error strings $\hat{e}_1, \dots, \hat{e}_L$ that compose an error pattern do not take values uniformly at random, but depend on the transmitted symbols $s_1, \dots, s_L$ and the noise power that led to the detection of symbols $\tilde{s}_1, \dots, \tilde{s}_L$ at the receiver, given that $\hat{e}_i = \mathcal{S}(s_i) \oplus \mathcal{S}(\tilde{s}_i)$ for $i = 1, \dots, L$. To facilitate the derivation of an expression for $P(L_1, L_2, \dots, L_\vartheta)$, we:

- Define B neighborhoods for each symbol. Members of a neighborhood of a symbol are other symbols of the

constellation diagram that are adjacent to that symbol and have the same Euclidean distance from it.

- Define T types, whereby symbols of the same type share the same neighborhood characteristics in terms of size, Euclidean distance, and also Hamming distance.

Fig. 4(b) shows examples of neighborhoods and symbol types for the 16-ary regular triangular QAM (TQAM) [43], [44], also referred to as regular hexagonal QAM [45], and for Gray-coded square 16-QAM. In the case of 16-ary regular TQAM, the nearest neighbors of a symbol are equidistant from that symbol and surround it, thus forming a single neighborhood for that symbol ($B = 1$). Notice that if s_i is transmitted and the detected symbol $\tilde{s}_i$ is in the neighborhood of s_i , the Hamming distance between $\mathcal{S}(s_i)$ and $\mathcal{S}(\tilde{s}_i)$ corresponds to the Hamming weight of the error string $\hat{e}_i$. Depending on the size of a neighborhood and the Hamming distance between a symbol and each of its neighbors, the 16 symbols can be classified into $T = 6$ types. If each type is arbitrarily assigned a unique number $j \in \{1, \dots, T\}$, then t_j denotes the number of symbols of type j , where $\sum_{j=1}^T t_j = M$. For 16-ary regular TQAM, we have $t_1 = 2, t_2 = 2, t_3 = 4, t_4 = 2, t_5 = 2$ and $t_6 = 4$. In the case of Gray-coded square 16-QAM, the symbols that surround each symbol can be divided into $B = 2$ neighborhoods based on their Euclidean distance from that symbol. Each symbol can then be categorized into one of $T = 3$ types, where $t_1 = 4, t_2 = 8$ and $t_3 = 4$, as illustrated in Fig. 4(a).

Using the concepts of neighborhoods and types, the probability that an error pattern or, equivalently, a sequence of L error strings has structure $[L_1 L_2 \dots L_\vartheta]$ can be expressed as a function of the probability that L symbols classed into T types are selected uniformly at random for transmission, L' of them are received incorrectly, where $L' = \sum_{w=1}^{\vartheta} L_w$, and the remaining $L - L'$ symbols are received without errors. More specifically, we can write:

$$\begin{aligned}
 P(L_1, L_2, \dots, L_\vartheta) &= \\
 &= \frac{1}{M^L} \sum_{\sum_{j=1}^T \ell_j = L} \left\{ \binom{L}{\ell_1, \dots, \ell_T} \prod_{j=1}^T t_j^{\ell_j} \times \right. \\
 &\quad \times \sum_{\substack{\sum_{j=1}^T \ell'_{j,0} = L-L' \\ \sum_{j=1}^T \ell'_{j,w} = L_w \\ \forall w \in \{1, \dots, \vartheta-1\}}} \left[\prod_{j=1}^T \binom{\ell_j}{\ell'_{j,0}, \dots, \ell'_{j,\vartheta}} \prod_{w=0}^{\vartheta} p_{j,w}^{\ell'_{j,w}} \right] \}. \quad (16)
 \end{aligned}$$

The second line of (16) determines the fraction of sequences of L symbols, from a total of M^L possible sequences, that contain ℓ_j symbols of type j , for $j = 1, \dots, T$. Each of the ℓ_j positions in the sequence can be occupied by any of the t_j symbols of that type, resulting in $t_j^{\ell_j}$ possible outcomes. The third line of (16) enumerates all possible ways that $\ell'_{j,w}$ error strings of weight w can change any $\ell'_{j,w}$ of the ℓ_j transmitted symbols of type j into a neighboring symbol with probability $p_{j,w}^{\ell'_{j,w}}$, for $j = 1, \dots, T$ and $w = 0, \dots, \vartheta$. An error string of weight 0 leaves a transmitted symbol

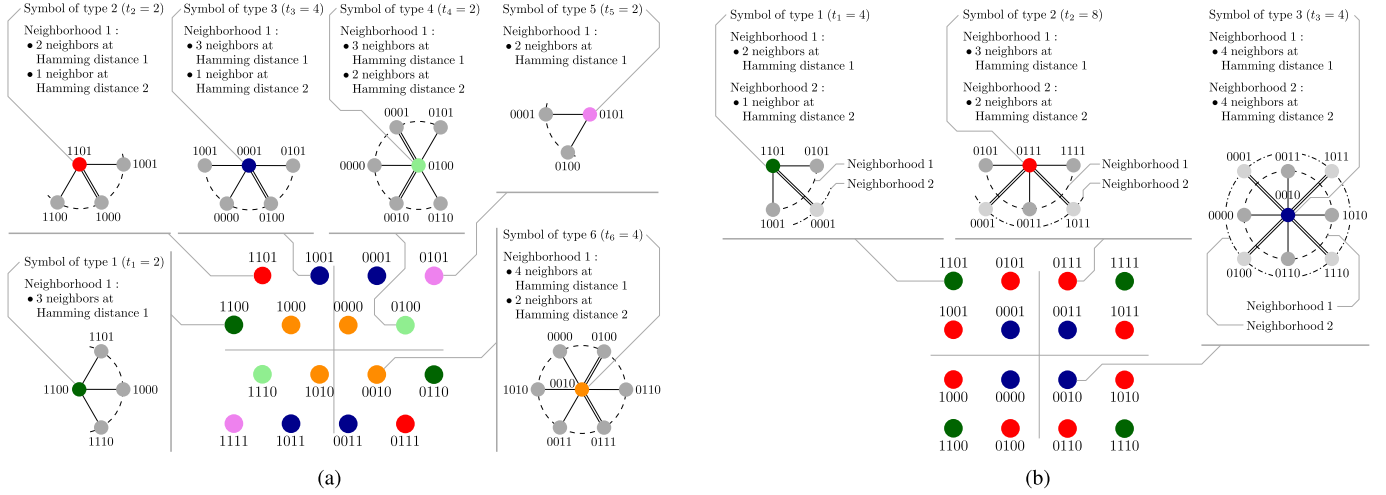


FIGURE 4. Constellation diagrams of (a) 16-ary regular TQAM and (b) Gray-coded square 16-QAM. In each case, every symbol has been classified into a type based on the number of adjacent symbols, their Euclidean distance and their Hamming distance from that symbol. Symbols located at the same Euclidean distance from a symbol compose a neighborhood for that symbol. Symbols connected by a single line are separated by a Hamming distance of 1. The Hamming distance between symbols linked by a double line is 2. If a symbol is transmitted and a neighboring symbol is received, the Hamming weight of the error string that altered the binary value of the transmitted symbol is equal to the Hamming distance of the two symbols.

unchanged, thus the first condition $\sum_{w=1}^T \ell'_{j,0} = L - L'$ ensures that the number of weight-0 error strings matches the number of correctly received symbols. The second condition requires that all error strings of weight w add up to L_w for $0 < w \leq \vartheta - 1$. Given the values of $\ell'_{j,0}, \dots, \ell'_{j,\vartheta-1}$, the value of $\ell'_{j,\vartheta}$ can be derived from $\ell'_{j,\vartheta} = \ell_j - \ell'_{j,0} - \dots - \ell'_{j,\vartheta-1}$. Expressions for $p_{j,w}$ depend on the M -ary modulation scheme. Section VI applies (16) to the case of Gray-coded square M -QAM and provides expressions for $p_{j,w}$.

Derivation of (16) assumed that received symbols have the same reliability, which is the case when symbols are transmitted over a SISO block Rayleigh fading channel. The following section explains how CSI could guide symbol-level GRAND in prioritizing error patterns of the same structure when received symbols have different reliabilities, as in the considered mMIMO scenario.

B. SORTING ERROR PATTERNS GUIDED BY CSI

In a non-ideal mMIMO scenario with $od(\mathcal{H}) > 1$, the reliability of spatial streams could vary significantly among them. At the receiver, a first processing block should implement the antenna sorting, as previously discussed in Section III.

Without loss of generality, we will discuss the case with $N_c = 1$. This can be accomplished by inserting a permutation matrix Π , which is a binary matrix whose columns are all columns of the identity $\mathbf{I}$ but placed in a different order. As it is well known, a permutation matrix is always an orthogonal matrix, and its inverse is its transpose: $\Pi^{-1} = \Pi^T$. These two matrices can be added before and after symbol-level GRAND, as presented in Fig. 5. The $g_1, \dots, g_{N_T}$ gains in (7) are sorted in ascending order of magnitude, and the corresponding permutation matrix Π is created. The permuted symbols $\tilde{\mathbf{x}}_s^{(\Pi)} = \tilde{\mathbf{x}}_s \Pi$ are fed to symbol-level GRAND, which will now not assume that

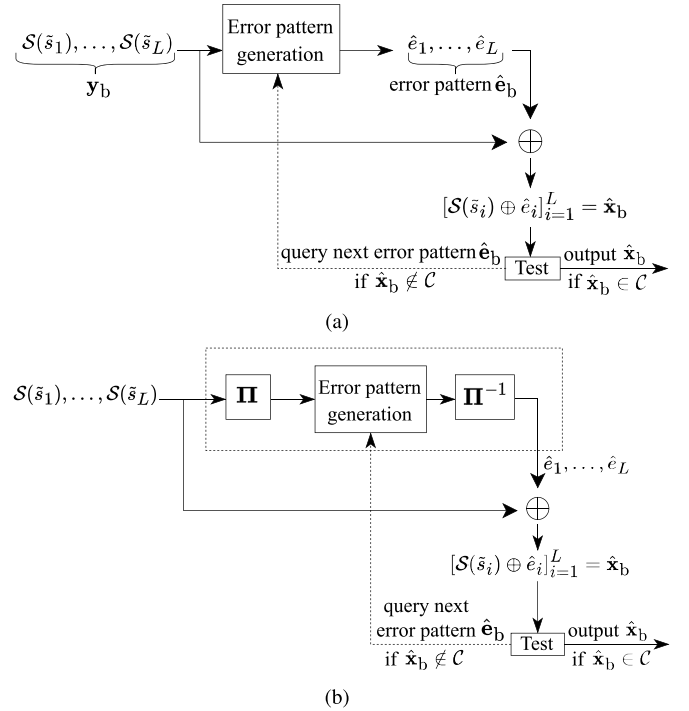


FIGURE 5. Types of symbol-level GRAND: (a) symbol-level GRAND proposed in [32], and (b) symbol-level GRAND with antenna sorting.

error patterns of the same structure are equiprobable and test them in an arbitrary order, as was the case in the previous section, but will test them in decreasing order of likelihood.

In the general case with $N_c > 1$ codewords per MIMO transmission, the set of the g_i , for $i = 1, \dots, N_T = N_c L$, is partitioned in N_c subsets and an independent sorting process is applied to each one of those subsets. Note that, for a faster overall decoding time, these sorting processes

satisfy $\mathbf{c}_u^T \hat{\mathbf{e}}_b = 1$ and, instead, results in $\mathbf{c}_u^T \hat{\mathbf{e}}_b = 0$, then it will fail the membership test $\mathbf{s} = \mathbf{C} \hat{\mathbf{e}}_b$. Therefore, the constraint $\mathbf{c}_u^T \hat{\mathbf{e}}_b = 1$ can be used as a lightweight test; only error patterns that meet the constraint and pass the lightweight test will be subjected to the membership test, whereas all other error patterns will be filtered out. In the latter case, the proposed constraint reduces the complexity associated with the computation of $\mathbf{C} \hat{\mathbf{e}}_b$ from $\mathcal{O}((n-k)n) = \mathcal{O}(n^2)$ to $\mathcal{O}(n)$, as multiplication is limited to a single row of $\mathbf{C}$.

The search space of all error patterns of weight $\|\hat{\mathbf{e}}_b\|_1$ that will be considered for the membership test has size $\binom{n}{\|\hat{\mathbf{e}}_b\|_1}$. Let α_0 be the number of error patterns that result in $\mathbf{c}_u^T \hat{\mathbf{e}}_b = 0$ and will thus be eliminated by the lightweight test. On the other hand, let α_1 be the number of error patterns that meet the condition $\mathbf{c}_u^T \hat{\mathbf{e}}_b = 1$ and, therefore, will pass the lightweight test and undergo the membership test. The relationship between α_0 , α_1 and the size of the unconstrained search space of all vectors of weight $\|\hat{\mathbf{e}}_b\|_1$ is

$$\alpha_0 + \alpha_1 = \binom{n}{\|\hat{\mathbf{e}}_b\|_1}. \quad (18)$$

To derive an expression for α_0 and α_1 , the conditions for $\mathbf{c}_u^T \hat{\mathbf{e}}_b$ being 0 or 1, respectively, need to be first determined.

If we shift our attention back to Fig. 7(b), we will notice that the product of the selected row $\mathbf{c}_u^T$ and the error pattern $\hat{\mathbf{e}}_b$ is 1 because only one of the two positions in $\hat{\mathbf{e}}_b$ that hold non-zero elements is in $S_1^{(u)} = \{2, 3, 9, 22\}$. If the two non-zero elements of $\hat{\mathbf{e}}_b$ occupied positions that were both members of $S_1^{(u)}$, or they both occupied any of the remaining $n - |S_1^{(u)}| = 26$ positions, the product would have been 0 and the lightweight test would have failed.

In general, an error pattern of weight $\|\hat{\mathbf{e}}_b\|_1$ will fail the lightweight test and will be eliminated from the search space of the membership test, if an *even* number of non-zero elements in the error pattern occupy any of the $|S_1^{(u)}|$ positions listed in $S_1^{(u)}$, while the rest of the non-zero elements occupy the remaining $n - |S_1^{(u)}| = |S_0^{(u)}|$ positions. Hence, the total number of error patterns that will be eliminated by the lightweight test is given by

$$\alpha_0 = \sum_{\rho=\rho_{lo}}^{\rho_{up}} \binom{|S_1^{(u)}|}{2\rho} \binom{n - |S_1^{(u)}|}{\|\hat{\mathbf{e}}_b\|_1 - 2\rho}. \quad (19)$$

Expressions for the lower and upper bounds of the summation, i.e., ρ_{lo} and ρ_{up} , can be deduced from the conditions $0 \leq 2\rho \leq |S_1^{(u)}|$ and $0 \leq (\|\hat{\mathbf{e}}_b\|_1 - 2\rho) \leq n - |S_1^{(u)}|$ that the two binomial coefficients in (19) are subject to. The upper bound is defined by having no more ones to distribute among the positions in $S_1^{(u)}$ or because there are no more empty positions in $S_1^{(u)}$ to fill in with ones. Conversely, the lower bound is related with the situation when very few ones are placed in $S_1^{(u)}$ and the remaining ones exceed the number of existing positions in the $S_0^{(u)}$ set. The interval where the

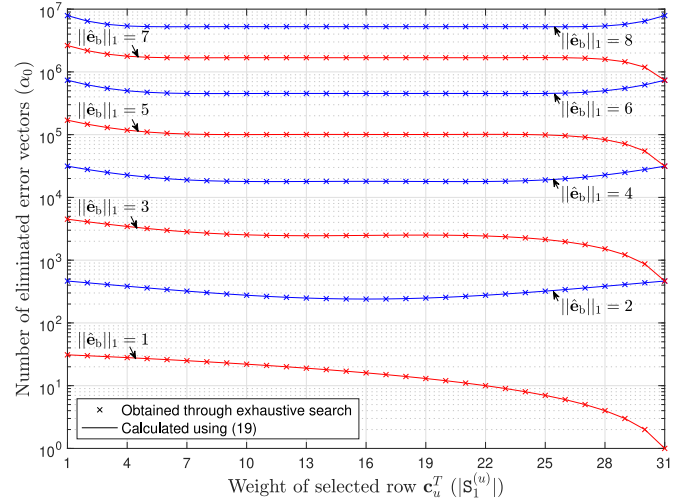


FIGURE 8. Number of error patterns that are eliminated by the lightweight test, as a function of the weight of the selected row $\mathbf{c}_u^T$. Error patterns of length $n = 32$ and weight $1 \leq \|\hat{\mathbf{e}}_b\|_1 \leq 8$ have been considered. Markers (x) correspond to values obtained through an exhaustive search, whereas solid lines (—) correspond to values obtained from (19).

two ranges overlap determines the domain $[\rho_{lo}, \rho_{up}]$ of ρ . We obtain

$$\rho_{lo} = \max \left\{ 0, \left\lceil \frac{|S_1^{(u)}| + \|\hat{\mathbf{e}}_b\|_1 - n}{2} \right\rceil \right\} \text{ and } \rho_{up} = \min \left\{ \left\lfloor \frac{|S_1^{(u)}|}{2} \right\rfloor, \left\lfloor \frac{\|\hat{\mathbf{e}}_b\|_1}{2} \right\rfloor \right\},$$

where $\lceil t \rceil$ and $\lfloor t \rfloor$ are operators that round t up or down, respectively, to the nearest integer.

Expression (19) has been plotted in Fig. 8 in terms of $|S_1^{(u)}|$, for $1 \leq \|\hat{\mathbf{e}}_b\|_1 \leq 8$ and $n = 32$. Similar trends emerge for different values of n . An exhaustive search has also been carried out to count all error patterns that do not satisfy $\mathbf{c}_u^T \hat{\mathbf{e}}_b = 1$, and compare the total with the value of α_0 obtained from (19) to visually confirm the validity of (19) in Fig. 8. We can infer from the plotted curves that selecting a row $\mathbf{c}_u^T$ out of the $\|\mathbf{s}\|_1$ rows of $\mathbf{C}_{red}$ – not arbitrarily but based on its weight – could maximize the value of α_0 . As shown in Fig. 8, for *even* values of $\|\hat{\mathbf{e}}_b\|_1$, choosing the row of $\mathbf{C}_{red}$ with either the lowest or the highest weight, i.e., with the greatest imbalance in the number of zero and non-zero elements, will maximize the number of error patterns that the lightweight test eliminates. The index u of this row can be determined from

$$\begin{aligned} u &= \arg \min_{1 \leq i \leq \|\mathbf{s}\|_1} \left\{ |S_1^{(i)}|, |S_0^{(i)}| \right\} \\ &= \arg \min_{1 \leq i \leq \|\mathbf{s}\|_1} \left\{ |S_1^{(i)}|, n - |S_1^{(i)}| \right\}, \text{ for } \|\hat{\mathbf{e}}_b\|_1 \text{ even.} \end{aligned} \quad (20)$$

If multiple rows have the same weight, a value for u is selected at random from the tied candidates. On the other hand, for *odd* values of $\|\hat{\mathbf{e}}_b\|_1$, choosing the row of $\mathbf{C}_{red}$

with the lowest weight has the potential to maximize the value of α_0 . In this case, the index u can be obtained from

$$u = \arg \min_{1 \leq i \leq \|\mathbf{s}\|_1} \left\{ |S_1^{(i)}| \right\}, \text{ for } \|\hat{\mathbf{e}}_b\|_1 \text{ odd.} \quad (21)$$

The size of the constrained and, thus, reduced search space of error patterns of weight $\|\hat{\mathbf{e}}_b\|_1$ for the *full* membership test can be derived from (18), if we solve for α_1 and substitute α_0 with the sum in (19). Note that in the case of systematic codes, due to the existence of $n - k - 1$ zeros in the $\mathbf{I}_{n-k}$ submatrix in $\mathbf{C}$, the most unbalanced row selected by (20) often ends up corresponding to the row with the fewest ones, selected by (21).

VI. ANALYSIS IN THE PERFECT CHANNEL HARDENING LIMIT

The system model considered in this paper and presented in Section II focuses on Gray-coded square M -QAM. Symbol-level GRAND with antenna sorting tests error patterns of structure $[L_1 \ L_2]$ and weight $L_1 + 2L_2 \leq w_{th}$. An expression for the likelihood of a structure $[L_1 \ L_2]$, which will be used to order all structures prior to the generation and testing of error patterns, can be obtained from (16) for $\vartheta = 2$, $T = 3$, $t_1 = 4$, $t_2 = 4(\sqrt{M} - 2)$, $t_3 = (\sqrt{M} - 2)^2$ and $L' = L_1 + L_2$, as follows:

$$\begin{aligned} P(L_1, L_2) &\approx \frac{1}{M^L} \sum_{\ell_1 + \ell_2 + \ell_3 = L} \left\{ \binom{L}{\ell_1, \ell_2, \ell_3} 4^{\ell_1 + \ell_2} (\sqrt{M} - 2)^{\ell_2 + 2\ell_3} \right. \\ &\times \sum_{\substack{\ell'_{1,0} + \ell'_{2,0} + \ell'_{3,0} = L - L' \\ \ell'_{1,1} + \ell'_{2,1} + \ell'_{3,1} = L_1}} \left[\prod_{j=1}^3 \binom{\ell_j}{\ell'_{j,0}, \ell'_{j,1}, \ell'_{j,2}} \right] \times \\ &\times \left. \prod_{w=0}^2 p_{j,w}^{\ell'_{j,w}} \right\}, \end{aligned} \quad (22)$$

which is in agreement with the expression derived in [32]. Exact and approximate expressions for $p_{j,w}$ in (22), for $j = 1, 2, 3$ and $w = 0, 1, 2$, have been included in [32] but are also listed in Table 2 for the sake of completeness. They are all functions of the halfway Euclidean distance d' between any two adjacent points along one dimension of the constellation diagram at the receiver.

As explained in Section III, PCH is achieved when N_T is fixed and $N_R \rightarrow \infty$, which essentially reduces the mMIMO channel into an equivalent non-fading AWGN channel. In this case, the ergodic SNR is given in (2) and the noise variance per dimension is $\sigma_n^2/2$. Therefore, d' assumes the form:

$$d' = d \frac{\sqrt{\text{snr}}}{\sqrt{0.5}} = d \sqrt{2 \text{snr}} = \sqrt{\frac{3 \text{snr}}{M-1}}, \quad (23)$$

where $d = \sqrt{3/(2(M-1))}$ is the halfway distance between adjacent points along one dimension of the M -QAM constellation diagram at the transmitter [46], [47].

TABLE 2. Expressions for the probability terms in (22). Function $Q(z) \triangleq (1/\sqrt{2\pi}) \int_z^\infty \exp(-t^2/2) dt$ is the tail distribution of the standard normal distribution. Variable d' is given by $d' = \sqrt{3 \text{snr}/(M-1)}$.

$p_{1,0} = (1 - Q(d'))^2$
$p_{2,0} = (1 - Q(d'))(1 - 2Q(d'))$
$p_{3,0} = (1 - 2Q(d'))^2$
$p_{1,1} = 2(1 - Q(d'))Q(d')$
$p_{2,1} \approx [2(1 - Q(d'))Q(d') + (1 - 2Q(d'))Q(d')]$
$p_{3,1} \approx 4(1 - 2Q(d'))Q(d')$
$p_{1,2} = Q^2(d')$
$p_{2,2} \approx 2Q^2(d')$
$p_{3,2} \approx 4Q^2(d')$

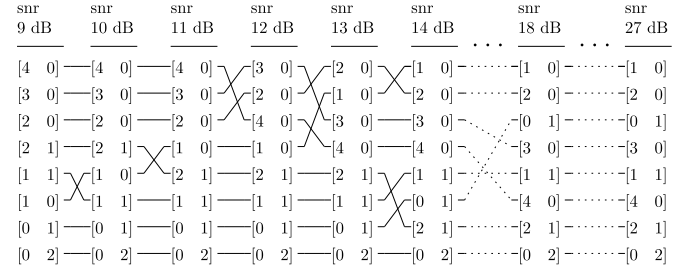


FIGURE 9. Evolution of the ranking of the error structures, based on (22), for $N_R \rightarrow \infty$, $w_{th} = 4$ and an increasing value of snr .

Fig. 9 provides an example that demonstrates the memory requirements of symbol-level GRAND. Lookup tables are presented side by side for snr values ranging from 9 dB to 27 dB in steps of 1 dB. Each lookup table contains all possible structures for $w_{th} = 4$, arranged in descending order of likelihood as determined by (22). One lookup table is required for each snr value in the range, since the ordering of the error structures depends on snr . In this example, the ordering of the error structures does not change for snr values greater than 18 dB, therefore lookup tables beyond 18 dB can be omitted. Observe in Fig. 9 that $0 \leq L_1 \leq 4$ and $0 \leq L_2 \leq 2$, hence L_1 and L_2 can be represented by 3 bits and 2 bits, respectively. A structure $[L_1 \ L_2]$ occupies $\lambda = 3 + 2 = 5$ bits, each lookup table contains $v = 8$ structures, and a total of $\tau = 10$ lookup tables are needed to cover the range between 9 dB and 18 dB in steps of 1 dB. Therefore, symbol-level GRAND will reserve $\lambda v \tau = 400$ bits of memory space in this example.

VII. RESULTS

The performance and the decoding complexity of the considered system were evaluated through numerical simulations. The number of codewords per MIMO channel use, previously defined in Section II, is here set to $N_c = 1$. In this setup, each codeword of n bits is transmitted in “one shot”, using Gray-coded M -QAM over a mMIMO channel with $N_T = n/\log_2(M)$ transmit antennas. Both ZF and MMSE detectors have been assessed, using (3) and (4) for filtering and (6) and (7) for antenna sorting. In all considered system configurations, the curves obtained using MMSE detection perfectly matched those obtained using ZF

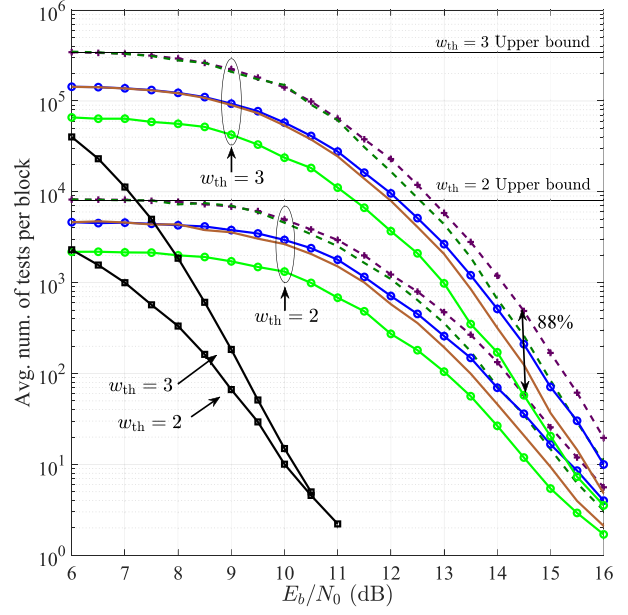
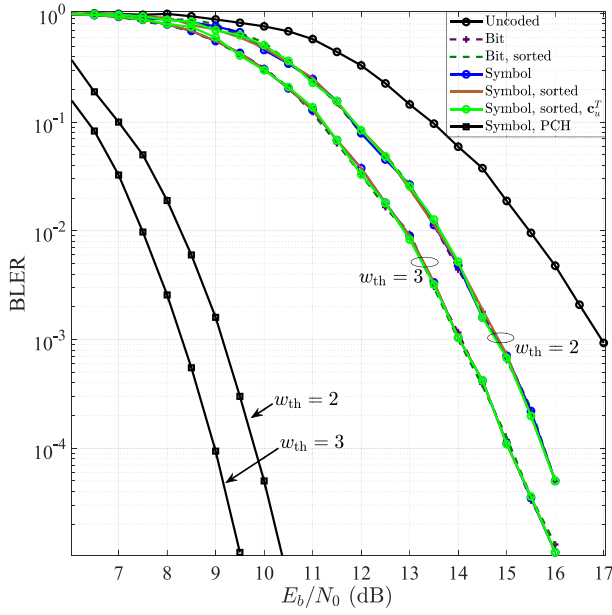


FIGURE 10. BLER performance (left) and decoding complexity (right) for different thresholds $w_{th} = 2, 3$, for different decoders, using RLC (128,103), with $N_T = 32$ and $N_R = 50$, and 16-QAM. The corresponding PCH lower bounds are also plotted.

detection, showing no advantage of MMSE over ZF. In the high-SNR regime, both (4) and (7) converge respectively to (3) and (6). In the low-SNR regime, as seen in Fig. 2, the different filters lead to different antenna sorting orders. However, for a large number of errors in a codeword, the error correction capability of the short codes does not suffice to provide any complexity reduction. Consequently, as MMSE does not offer an advantage over ZF in the considered system model, only results for ZF detection are presented in this section. Note that after the ZF detector, antenna sorting involves the arrangement of the set of g_i , for $i = 1, \dots, N_T$, in ascending order.

Two constellations were considered: 16-QAM and 64-QAM. As the M -arity of the modulation grows, the number of transmit antennas N_T is decreased to accommodate the same payload of n bits. The number of antennas was set to $N_T = 32$ for 16-QAM with a RLC (128, 103), and $N_T = 22$ for 64-QAM with a RLC (132, 106), such that the code rate $R = k/n = 0.8$ is kept constant. Three different values for N_R were tested for each M -arity: $N_R = 50, 100$ and 200 for 16-QAM, and $N_R = 38, 69$ and 138 for 64-QAM. To make the comparison between 16-QAM and 64-QAM fair, the number of receive antennas N_R in each configuration was chosen to yield similar load factors N_R/N_T , i.e., $50/32 \approx 38/22$, $100/32 \approx 69/22$ and $200/32 \approx 138/22$.

The antenna sorting preprocessing can be used to improve the decoding speed of symbol-level GRAND but also of bit-level GRAND. After arranging the bit-strings $\mathcal{S}(\tilde{s}_i)$ in ascending order of likelihood, one can also apply the original bit-level GRAND using its default flipping order for each bit. However, this leads to sub-optimal performance, given that the probability of the strings of $\log_2(M)$ bits is being used rather than the probability of the individual bits. We refer to this decoding method as *sorted-bit-level*

TABLE 3. Summary of the simulation curves presented in the figures.

Legend	Description
Uncoded	Baseline case without channel coding. Performance reflects only MIMO detection.
Bit	Conventional bit-level GRAND decoding combined with ZF detection. The test error patterns are generated at the bit level.
Bit, sorted	Bit-level GRAND decoding assisted by antenna sorting.
Symbol	Symbol-level GRAND decoding (no sorting). The test error patterns are generated at the symbol level (i.e. strings of length $\log_2(M)$).
Symbol, sorted	Symbol-level GRAND with test error patterns guided by the antenna sorting order.
Symbol, sorted, c_u^T	Constrained symbol-level GRAND (sorted) with the proposed lightweight membership test based on the optimized row c_u^T .
Symbol, PCH	Lower bound under perfect channel hardening (PCH) assumption, where all spatial streams are equally reliable and antenna sorting brings no benefit.

decoding. While sub-optimal, this ordering performs a step toward optimal bit ordering, and therefore reduces decoding complexity in comparison to standard unsorted bit-level GRAND. Afterward, decoding can be further accelerated by using the row of the parity-check matrix conveying most information about the errors' location as a filter on the candidate error patterns coming from the previous decoding steps as described in Section V.

For clarity and completeness, Table 3 summarizes all the labels that appear in the legends of the simulation figures. Fig. 10, Fig. 11 and Fig. 12 illustrate the performance and decoding complexity results for 16-QAM, and Fig. 13, Fig. 14 and Fig. 15 show the performance and decoding

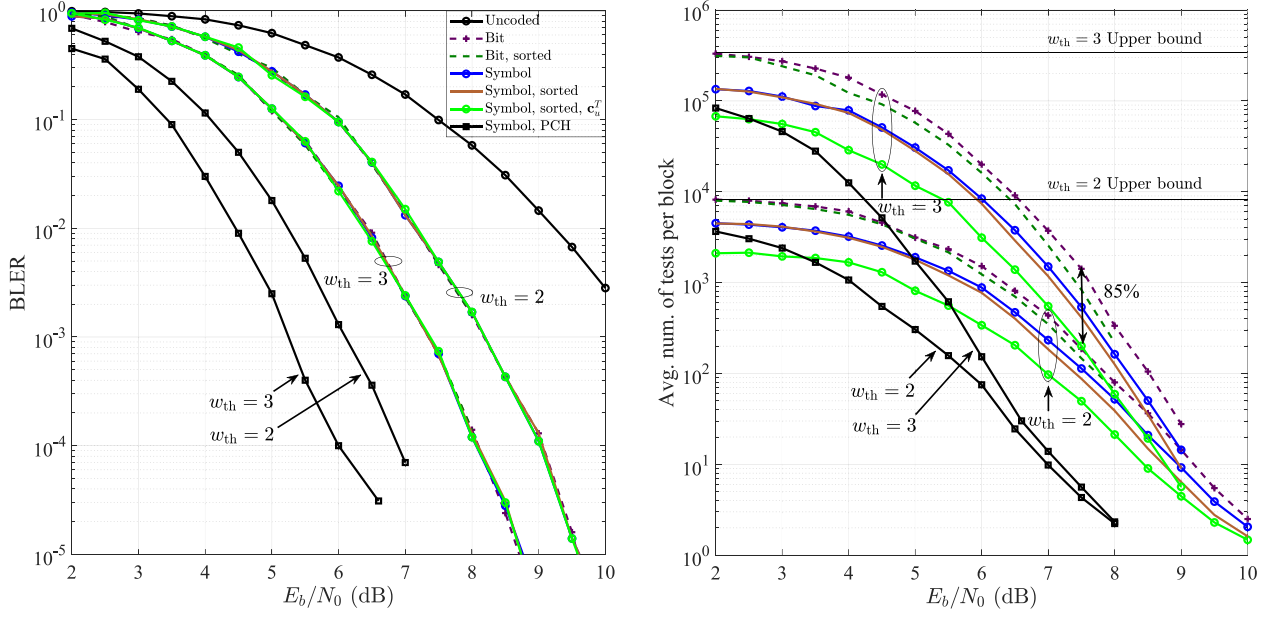


FIGURE 11. BLER performance (left) and decoding complexity (right) for different thresholds $w_{th} = 2, 3$, for different decoders, using RLC (128,103), with $N_T = 32$ and $N_R = 100$, and 16-QAM. The corresponding PCH lower bounds are also plotted.

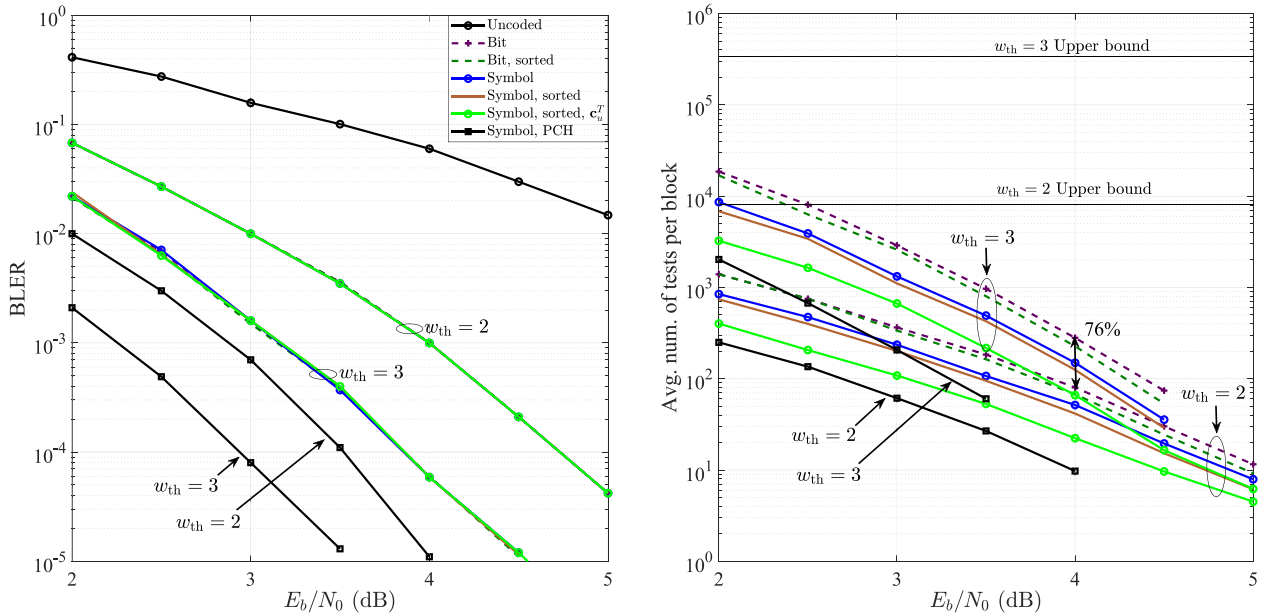


FIGURE 12. BLER performance (left) and decoding complexity (right) for different thresholds $w_{th} = 2, 3$, for different decoders, using RLC (128,103), with $N_T = 32$ and $N_R = 200$, and 16-QAM. The corresponding PCH lower bounds are also plotted.

complexity results for 64-QAM. The block error rate (BLER) has been used to evaluate the system's performance as a function of E_b/N_0 , as it is commonly used in recent works evaluating GRAND. The decoding complexity has been expressed in terms of the expected number of membership tests needed at each E_b/N_0 . All figures include the curves for uncoded transmission, bit-level GRAND decoding, sorted-bit-level decoding, symbol-level GRAND decoding, sorted-symbol-level decoding, and sorted-symbol-level decoding with extra "filtering" provided by the application of the lightweight membership test. Each system

configuration is assessed with two different thresholds for the number of bits in error in the error pattern, $w_{th} = 2, 3$. The figures also include performance and complexity results when using symbol-level GRAND for PCH. Recall that, in the ideal scenario of PCH, antenna sorting prior to GRAND has no impact on the overall decoding complexity because all streams experience the same SNR after ZF detection, as seen in (14).

As expected, the BLER performance greatly improves as one tests error patterns with larger weights, but this is achieved at the cost of a considerably larger number of

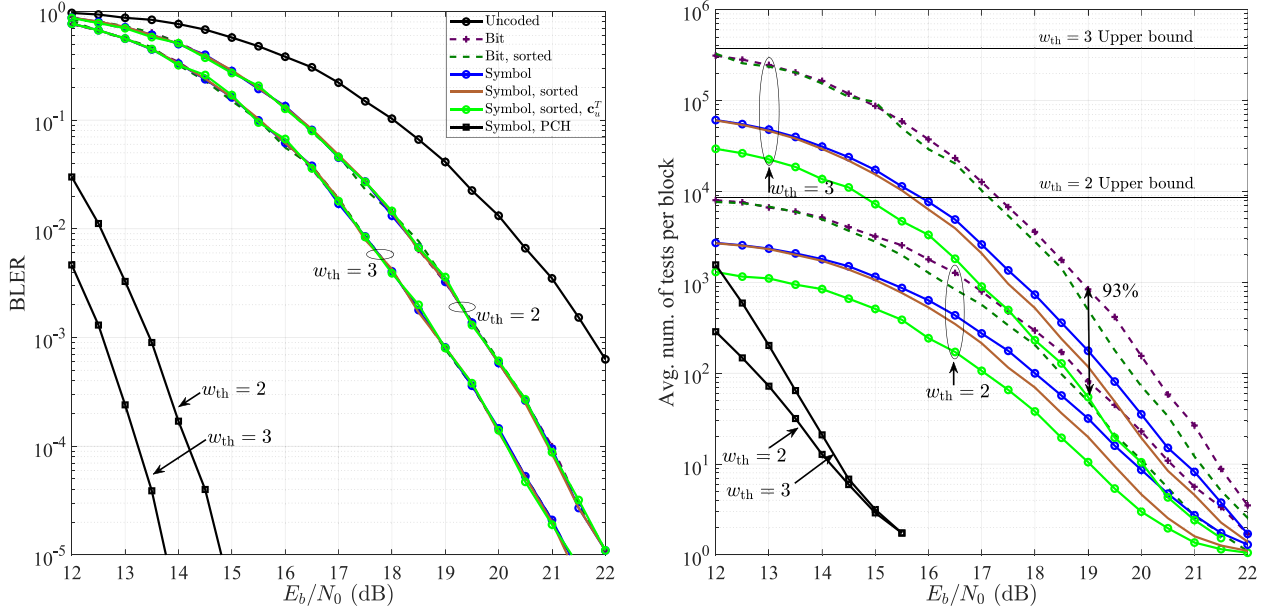


FIGURE 13. BLER performance (left) and decoding complexity (right) for different thresholds $w_{th} = 2, 3$, for different decoders, using RLC (132,106), with $N_T = 22$ and $N_R = 34$, and 64-QAM. The corresponding PCH lower bounds are also plotted.

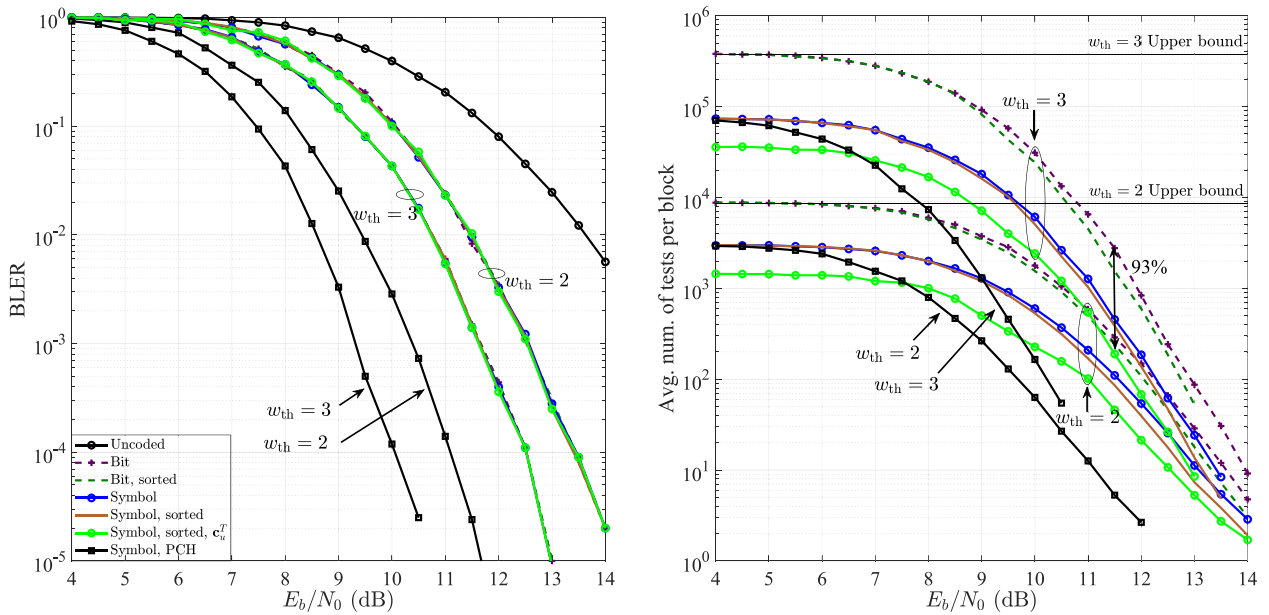


FIGURE 14. BLER performance (left) and decoding complexity (right) for different thresholds $w_{th} = 2, 3$, for different decoders, using RLC (132,106), with $N_T = 22$ and $N_R = 69$, and 64-QAM. The corresponding PCH lower bounds are also plotted.

membership tests. When considering a given w_{th} threshold, the worst-case upper bound on the number of membership tests occurs in the case of bit-level GRAND, which is $\sum_{w=0}^{w_{th}} \binom{n}{w}$. The upper-bound value for $w_{th} = 2, 3$ has been plotted in the figures that show complexity results. The results show that the average number of membership tests is much lower than the upper bound for $M = 16$ with $N_R = 200$ and $M = 64$ with $N_R = 138$. Nevertheless, when the noise is too large, the decoding complexity can get close to the upper bound due to the sheer number of erroneous symbols. As one would expect, when the noise vanishes, the average

number of membership tests always tends to be one; in that case, all the received words are valid codewords and the only membership test performed is to check, and confirm that the error pattern is $\hat{\mathbf{e}}_b = \mathbf{0}$.

One should note that the BLER performance results for bit-level GRAND, symbol-level GRAND, sorted-bit-level, sorted-symbol-level, and sorted-constrained-symbol-level decoding for the analyzed range of E_b/N_0 are all the same (all four curves overlap). However, the complexity comparison illustrates that the sorting antenna schemes and then symbol-level sorted with the lightweight test involving

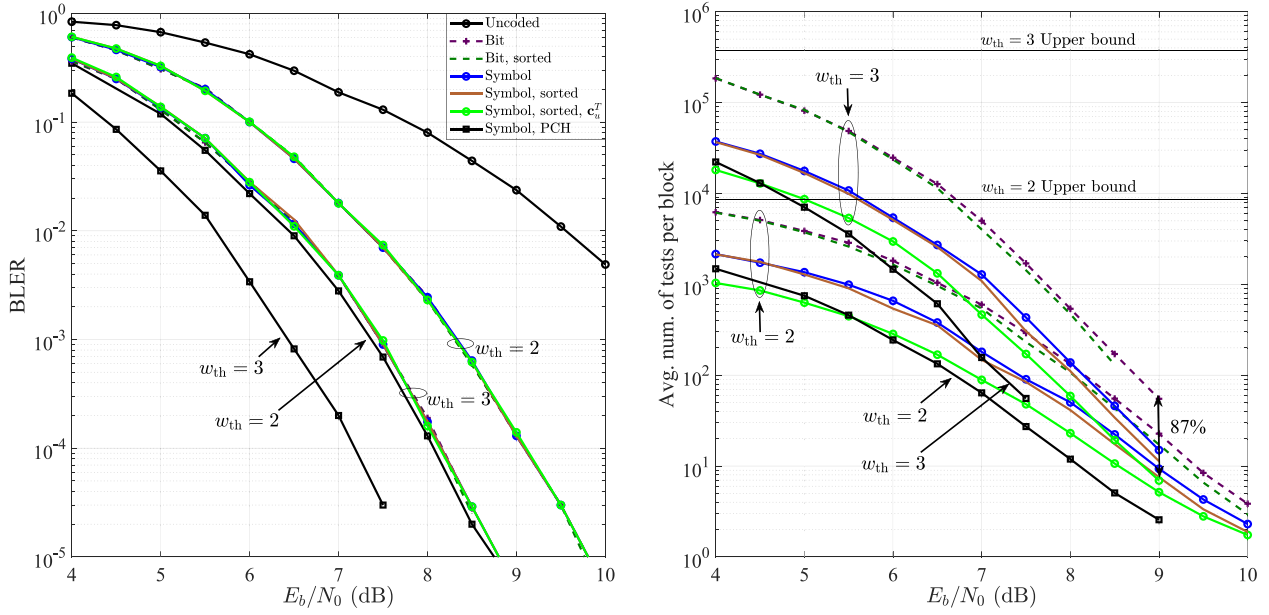


FIGURE 15. BLER performance (left) and decoding complexity (right) for different thresholds $w_{th} = 2, 3$, for different decoders, using RLC (132,106), with $N_T = 22$ and $N_R = 138$, and 64-QAM. The corresponding PCH lower bounds are also plotted.

$\mathbf{h}_u^T$ remarkably outperform bit-level GRAND. The extra complexity reduction added by the sorting preprocessing becomes more significant when N_R becomes smaller. This is due to a less strong channel hardening effect so that the equivalent SNR at each of the N_T streams becomes more uneven. As seen in the complexity results, the decoding time is reduced by $\approx 80\%$ when $M = 16$, and by $\approx 90\%$ in the system with 64-QAM.

VIII. CONCLUSION

This paper proposes a coded mMIMO transmission scheme for high-throughput, high-reliability, and low-latency, in accordance to the URLLC desiderata. This is accomplished using RLCs and ordered reliability symbol-level GRAND. Symbol-level GRAND has been generalized from M -QAM to any M -ary modulation scheme, and extended from SISO channels to mMIMO channels.

When the channel conditions are far from ideal, the paper shows that linear detectors can provide a soft-metric for the reliability of each spatial stream, which in the proposed setup corresponds to a symbol reliability. The orthogonality defect of the mMIMO lattice is related to the variance of the reliability of the symbols. The disparity between the reliability of the symbols gets larger when N_R decreases; in that case, the proposed antenna sorting can provide a significant reduction of the decoding complexity.

The results show that symbol-level GRAND provides much faster decoding than the bit-level GRAND counterpart in the same mMIMO setup, throughout the SNR range of interest. The proposed antenna sorting mechanism further speeds up the decoding process in scenarios where channel-hardening is far from perfect. Furthermore, the error patterns generated by the symbol-level decoder with antenna sorting decoding stages undergo a “filtering” constraint based

on a partial membership test. This proposed lightweight membership test employs a linear complexity constraint $\mathcal{O}(n)$, rather than the quadratic one, $\mathcal{O}(n^2)$, involved in a full membership test. This technique has been optimized to maximize the number of discarded error patterns. In the worst scenario of “filtering” with a balanced row, half of the error patterns are discarded.

Without performance degradation, the number of membership tests can be dramatically reduced when comparing the results of the GRAND receiver combining symbol-level GRAND, antenna sorting and the lightweight membership test, with the results of using conventional bit-level GRAND.

APPENDIX: ANTENNA SNR WITH PERFECT AND IMPERFECT CSI

This appendix explores the impact of imperfect CSI on the output SNR at each antenna stream for ZF. For a clearer and more streamlined presentation, the derivation of (6) for perfect CSI is first presented and a similar line of thought is then followed to obtain an expression for the output SNR in the case of imperfect CSI.

A. PERFECT CSI

While the case with perfect CSI is well-known (e.g., [33]), we start by deriving (6) with the aim of obtaining useful equalities for the following subsection. Let us consider the model in (5). After applying (3) (and dropping the ZF notation), one gets $\mathbf{W}\mathbf{y} = \mathbf{x} + \mathbf{W}\mathbf{n}$. The post-detection noise covariance matrix is

$$\begin{aligned} \mathbf{R}_n &= \mathbb{E}[\mathbf{W}\mathbf{n}\mathbf{n}^H\mathbf{W}^H] \\ &= \mathbf{W} \underbrace{\mathbb{E}[\mathbf{n}\mathbf{n}^H]}_{=\sigma_n^2\mathbf{I}} \mathbf{W}^H = \sigma_n^2 \mathbf{W}\mathbf{W}^H. \end{aligned} \quad (24)$$

It is possible to show that, using (3),

$$\mathbf{W}\mathbf{W}^H = (\mathbf{H}^H\mathbf{H})^{-1}, \quad (25)$$

and therefore, considering $\mathbb{E}\{|s_i|^2\} = 1$, the resulting SNR for the i -th stream is

$$\text{snr}_i^{(\text{perfect CSI})} = \frac{1}{\sigma_n^2 [\mathbf{H}^H\mathbf{H}]^{-1}_{ii}} = \frac{\text{snr}}{[\mathbf{G}^{-1}]_{ii}} = g_i \text{ snr}. \quad (26)$$

B. IMPERFECT CSI

In the case of imperfect CSI, the estimated channel matrix can be modeled as

$$\hat{\mathbf{H}} = \mathbf{H} + \mathbf{H}_\varepsilon, \quad (27)$$

where the elements of $\mathbf{H}_\varepsilon$ are i.i.d. $\mathcal{CN}(0, \sigma_\varepsilon^2)$. The received signal can be written as

$$\mathbf{y} = (\hat{\mathbf{H}} - \mathbf{H}_\varepsilon)\mathbf{x} + \mathbf{n}. \quad (28)$$

The detection filter constructed at the receiver will then be

$$\hat{\mathbf{W}} = (\hat{\mathbf{H}}^H\hat{\mathbf{H}})^{-1}\hat{\mathbf{H}}^H. \quad (29)$$

Hence, the detected signal is

$$\begin{aligned} \hat{\mathbf{W}}\mathbf{y} &= (\hat{\mathbf{H}}^H\hat{\mathbf{H}})^{-1}\hat{\mathbf{H}}^H(\hat{\mathbf{H}} - \mathbf{H}_\varepsilon)\mathbf{x} + \hat{\mathbf{W}}\mathbf{n} \\ &= \mathbf{x} - \hat{\mathbf{W}}\mathbf{H}_\varepsilon\mathbf{x} + \hat{\mathbf{W}}\mathbf{n}. \end{aligned} \quad (30)$$

One has now to compute the power of the two noise terms, starting by obtaining the correlation matrices

$$\begin{aligned} \mathbf{R}_\varepsilon &= \mathbb{E}[\hat{\mathbf{W}}\mathbf{H}_\varepsilon\mathbf{H}_\varepsilon^H\hat{\mathbf{W}}^H] \\ &= \hat{\mathbf{W}}\underbrace{\mathbb{E}[\mathbf{H}_\varepsilon\mathbf{H}_\varepsilon^H]}_{=\sigma_\varepsilon^2\mathbf{I}}\hat{\mathbf{W}}^H \\ &= \sigma_\varepsilon^2\hat{\mathbf{W}}\hat{\mathbf{W}}^H = \sigma_\varepsilon^2(\hat{\mathbf{H}}^H\hat{\mathbf{H}})^{-1}, \end{aligned} \quad (31)$$

where the last equality uses (25). The covariance matrix of the $\hat{\mathbf{W}}\mathbf{n}$ noise term is obtained similarly to (24) and (25)

$$\mathbf{R}_{\hat{\mathbf{n}}} = \mathbb{E}[\hat{\mathbf{W}}\mathbf{n}\mathbf{n}^H\hat{\mathbf{W}}^H] \quad (32)$$

$$= \sigma_n^2\hat{\mathbf{W}}\hat{\mathbf{W}}^H = \sigma_n^2(\hat{\mathbf{H}}^H\hat{\mathbf{H}})^{-1}. \quad (33)$$

Therefore, the SNR for the i -th stream under imperfect CSI is

$$\begin{aligned} \text{snr}_i^{(\text{imperfect CSI})} &= \frac{1}{[\hat{\mathbf{H}}^H\hat{\mathbf{H}}]^{-1}(\sigma_\varepsilon^2 + \sigma_n^2)_{ii}} \\ &= \frac{\frac{1}{\sigma_n^2}}{\hat{g}_i^{-1}\left(1 + \frac{\sigma_\varepsilon^2}{\sigma_n^2}\right)} \\ &= \frac{\text{snr}}{\hat{g}_i^{-1}(1 + \sigma_\varepsilon^2 \text{snr})}, \end{aligned} \quad (34)$$

where $\text{snr} = \frac{1}{\sigma_n^2}$. Also, $\hat{g}_i = [\hat{\mathbf{G}}]_{ii}$, and $\hat{\mathbf{G}} = \hat{\mathbf{H}}^H\hat{\mathbf{H}}$ stands for the imperfectly estimated Gram matrix.

Note that when $\sigma_\varepsilon^2 = 0$, one recovers (6), as in that case $\hat{g}_i = g_i$. The impact of a small σ_ε^2 on the ordering of the antennas will be negligible, however for large perturbations of the channel estimate, with $\sigma_\varepsilon^2 \gg \sigma_n^2$, the estimates in (34) are defined by the random variables $\hat{g}_i$. While under perfect CSI, the sorting order is deterministically obtained from $\mathbf{H}$, when only a very poor estimate of the channel is available, the sorting becomes random. For near-perfect CSI, the sorting order should be nearly identical or even equal to the optimal order. The evaluation of how the decoding performance is affected and how robust the complexity reduction remains for a given level of CSI uncertainty is suggested for future work.

REFERENCES

- [1] P. Nouri, H. Alves, M. A. Uusitalo, O. Alcaraz López, and M. Latva-aho, "Machine-type wireless communications enablers for beyond 5G: Enabling URLLC via diversity under hard deadlines," *Comput. Netw.*, vol. 174, no. 3, Jun. 2020, Art. no. 107227.
- [2] 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, May 2022.
- [3] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Techn. J.*, vol. 27, pp. 379–423, Jul. 1948.
- [4] R. Gallager, "The random coding bound is tight for the average code (corresp.)," *IEEE Trans. Inf. Theory*, vol. 19, no. 2, pp. 244–246, Mar. 1973.
- [5] M. Shirvanimoghaddam et al., "Short block-length codes for ultra-reliable low latency communications," *IEEE Commun. Mag.*, vol. 57, no. 2, pp. 130–137, Feb. 2019.
- [6] A. Becker, A. Joux, A. May, and A. Meurer, "Decoding random binary linear codes in $2^{n/20}$: How $1 + 1 = 0$ improves information set decoding," in *Proc. 31st Int. Conf. Theory Appl. Crypt. Techn. (EUROCRYPT)*, Cambridge, U.K., Apr. 2012, pp. 520–536.
- [7] J. Wolf, "Efficient maximum likelihood decoding of linear block codes using a trellis," *IEEE Trans. Inf. Theory*, vol. 24, no. 1, pp. 76–80, Jan. 1978.
- [8] T. Kasami, T. Takata, T. Fujiwara, and S. Lin, "On complexity of trellis structure of linear block codes," *IEEE Trans. Inf. Theory*, vol. 39, no. 3, pp. 1057–1064, May 1993.
- [9] F. R. Kschischang and V. Sorokine, "On the trellis structure of block codes," *IEEE Trans. Inf. Theory*, vol. 41, no. 6, pp. 1924–1937, Nov. 1995.
- [10] G. D. Forney, "Coset codes—Part II: Binary lattices and related codes," *IEEE Trans. Inf. Theory*, vol. 34, no. 5, pp. 1152–1187, Sep. 1988.
- [11] A. H. Banihashemi, "Decoding complexity and trellis structure of lattices," Ph.D. dissertation, Dept. Electr. Eng., Univ. Waterloo, Waterloo, ON, Canada, 1997.
- [12] F. A. Monteiro and F. R. Kschischang, "Trellis detection for random lattices," in *Proc. 8th Inter. Symp. Wireless Commun. Syst. (ISWCS)*, Aachen, Germany, Nov. 2011, pp. 755–759.
- [13] B. Honary, *Trellis Decoding of Block Codes: A Practical Approach*. New York, NY, USA: Kluwer Academic Publ., 1997.
- [14] S. Lin, T. Kasami, and M. Fossorier, *Trellises and Trellis-Based Decoding Algorithms for Linear Block Codes*. New York, NY, USA: Kluwer Academic Publ., 1998.
- [15] J. T. Coffey and R. M. Goodman, "The complexity of information set decoding," *IEEE Trans. Inf. Theory*, vol. 36, no. 5, pp. 1031–1037, Sep. 1990.
- [16] 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, Jul. 2019.
- [17] K. R. Duffy, A. Solomon, K. M. Konwar, and M. Médard, "5G NR CA-polar maximum likelihood decoding by GRAND," in *Proc. 54th Annu. Conf. Inf. Sci. Syst. (CISS)*, May 2020, pp. 1–5.

- [18] K. R. Duffy, "Ordered reliability bits Guessing random additive noise decoding," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2021, pp. 8268–8272.
- [19] S. M. Abbas, M. Jalaleddine, and W. J. Gross, "GRAND for Rayleigh fading channels," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Rio de Janeiro, Brazil, Jan. 2022, pp. 504–509.
- [20] Y. Polyanskiy, H. V. Poor, and S. Verdú, "Channel coding rate in the finite blocklength regime," *IEEE Trans. Inf. Theory*, vol. 56, no. 5, pp. 2307–2359, Apr. 2010.
- [21] W. Yang, G. Durisi, T. Koch, and Y. Polyanskiy, "Quasi-static multiple-antenna fading channels at finite blocklength," *IEEE Trans. Inf. Theory*, vol. 60, no. 7, pp. 4232–4265, Jul. 2014.
- [22] C. Yue, V. Miloslavskaya, M. Shirvanimoghaddam, B. Vucetic, and Y. Li, "Efficient decoders for short block length codes in 6G URLLC," *IEEE Commun. Mag.*, vol. 61, no. 4, pp. 84–90, Apr. 2023.
- [23] S. Allahkaram, F. A. Monteiro, and I. Chatzigeorgiou, "URLLC with Coded Massive MIMO via Random Linear Codes and GRAND," in *Proc. IEEE 96th Veh. Technol. Conf. (VTC-Fall)*, London, U.K., Sep. 2022, pp. 1–5.
- [24] K. R. Duffy, M. Médard, and W. An, "Guessing random additive noise decoding with symbol reliability information (SRGRAND)," *IEEE Trans. Commun.*, vol. 70, no. 1, pp. 3–18, Sep. 2022.
- [25] D. Cruz, F. A. Monteiro, and B. C. Coutinho, "Quantum error correction via noise guessing decoding," *IEEE Access*, vol. 11, pp. 19446–119461, 2023.
- [26] D. Cruz, F. A. Monteiro, A. Roque, and B. C. Coutinho, "Fault-tolerant noise guessing decoding of quantum random codes," *IEEE Trans. Quantum Eng.*, vol. 6, pp. 2689–1808, 2025.
- [27] D. Chandra, Z. B. Kaykac Egilmez, Y. Xiong, S. X. Ng, R. G. Maund, and L. Hanzo, "Universal decoding of quantum stabilizer codes via classical guesswork," *IEEE Access*, vol. 11, pp. 19059–19072, 2023.
- [28] X. Ma, "Guessing what, noise or codeword?" in *Proc. IEEE Inf. Theory Workshop (ITW)*, Shenzhen, China, Nov. 2024, pp. 460–465.
- [29] R. Hadavian, D. Truhachev, K. El-Sankary, H. Ebrahimzad, and H. Najafi, "Ordered reliability direct error pattern testing (ORDEPT) algorithm," in *Proc. IEEE Global Commun. Conf.*, Kuala Lumpur, Malaysia, Dec. 2023, pp. 6983–6988.
- [30] A. Solomon, K. R. Duffy, and M. Médard, "Soft Maximum Likelihood Decoding using GRAND," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2020, pp. 1–6.
- [31] P. Yuan, M. Médard, 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, Apr. 2025.
- [32] I. Chatzigeorgiou and F. A. Monteiro, "Symbol-level GRAND for high-order modulation over block fading channels," *IEEE Commun. Lett.*, vol. 27, no. 2, pp. 447–451, Feb. 2023.
- [33] Y. Jiang, M. K. Varanasi, and J. Li, "Performance analysis of ZF and MMSE equalizers for MIMO systems: An in-depth study of the high SNR regime," *IEEE Trans. Inf. Theory*, vol. 57, no. 4, pp. 2008–2026, Apr. 2011.
- [34] F. A. Monteiro, "Lattices in MIMO spatial multiplexing: Detection and geometry," Ph.D. dissertation, Dept. Eng., Univ. Cambridge, Cambridge, U.K., 2012.
- [35] K. Su and I. Wassell, "A new ordering for efficient sphere decoding," in *Proc. IEEE Int. Conf. Commun. (ICC)*, vol. 3, May 2005, pp. 1906–1910.
- [36] J. Griffin, P. Yuan, K. R. Duffy, and M. Médard, "Using a single-parity-check to reduce the guesswork of guessing codeword decoding," Nov. 2024, *arXiv:2411.09803*.
- [37] M. Rowshan and J. Yuan, "Constrained error pattern generation for GRAND," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Espoo, Finland, Jun. 2022, pp. 1767–1772.
- [38] D. J. C. MacKay, *Information Theory, Inference and Learning Algorithms*. Cambridge, U.K.: Cambridge Univ., 2003.
- [39] R. W. Heath Jr. and A. Lozano, *Foundations of MIMO Communication*. Cambridge, U.K.: Cambridge Univ., 2019.
- [40] F. A. Monteiro and I. J. Wassell, "Recovery of a lattice generator matrix from its Gram matrix for feedback and precoding in MIMO," in *Proc. 4th Int. Symp. Commun., Control Signal Process. (ISCCSP)*, Limassol, Cyprus, 2010, pp. 1–6.
- [41] M. Taherzadeh, A. Mobasher, and A. K. Khandani, "LLL reduction achieves the receive diversity in MIMO decoding," *IEEE Trans. Inf. Theory*, vol. 53, no. 12, pp. 4801–4805, Dec. 2007.
- [42] E. Björnson, J. Hoydis, and L. Sanguinetti, "Massive MIMO networks: Spectral, energy, and hardware efficiency," *Found. Trends® Signal Process.*, vol. 11, nos. 3–4, pp. 154–655, 2017.
- [43] S.-J. Park, "Triangular quadrature amplitude modulation," *IEEE Commun. Lett.*, vol. 11, no. 4, pp. 292–294, Apr. 2007.
- [44] S.-J. Park and M.-K. Byeon, "Irregularly distributed triangular quadrature amplitude modulation," in *Proc. IEEE 19th Int. Symp. Pers., Indoor Mobile Radio Commun.*, Cannes, France, 2008, pp. 1–5.
- [45] T. K. Oikonomou, S. A. Tegos, D. Tyrovolas, P. D. Diamantoulakis, and G. K. Karagiannidis, "On the error analysis of hexagonal-QAM constellations," *IEEE Commun. Lett.*, vol. 26, no. 8, pp. 1764–1768, Aug. 2022.
- [46] J. Lu, K. B. Letaief, J.-I. Chuang, and M. L. Liou, "M-PSK and M-QAM BER computation using signal-space concepts," *IEEE Trans. Commun.*, vol. 47, no. 2, pp. 181–184, Feb. 1999.
- [47] K. Cho and D. Yoon, "On the general BER expression of one- and two-dimensional amplitude modulations," *IEEE Trans. Commun.*, vol. 50, no. 7, pp. 1074–1080, Jul. 2002.



SAHAR ALLAHKARAM (Student Member, IEEE) received the B.Sc. degree in electronic engineering from the Azad University of Tehran in 2015, and the M.Sc. degrees in aerospace engineering from the Sapienza University of Rome in 2020. She is currently pursuing the Ph.D. degree with the Department of Information Science and Technology, Iscte—University Institute of Lisbon, Lisbon, Portugal. She is working on signal processing and coding techniques for 6G ultra-reliable low-latency wireless machine-type communications.



FRANCISCO A. MONTEIRO (Member, IEEE) received the Ph.D. degree from the University of Cambridge, Cambridge, U.K., and the Licenciatura and M.Sc. degrees in electrical and computer engineering from Instituto Superior Técnico, University of Lisbon, where he also became a Teaching Assistant. He is an Associate Professor with the Department of Information Science and Technology, Iscte—University Institute of Lisbon, and a Researcher with Instituto de Telecomunicações, Lisbon, Portugal. He held visiting research positions with the Universities of Toronto, Canada, Lancaster, U.K., Oulu, Finland, and Pompeu Fabra, Barcelona, Spain. He has won two best paper prizes awards at IEEE conferences in 2004 and 2007, respectively, a Young Engineer Prize (3rd place) from the Portuguese Engineers Institution (Ordem dos Engenheiros) in 2002, and for two years in a row was a recipient of Exemplary Reviewer Awards from the IEEE Wireless Communications Letters in 2014 and 2015. He co-edited the book *MIMO Processing for 4G and Beyond: Fundamentals and Evolution*, published by CRC Press in 2014. In 2016, he was the Lead Guest Editor of a special issue on Network Coding of the EURASIP Journal on Advances in Signal Processing. He was a General Chair of ISWCS 2018—The 15th International Symposium on Wireless Communication Systems, an IEEE major conference in wireless communications.



IOANNIS CHATZIGEORGIOU (Senior Member, IEEE) received the Dipl.Eng. degree in electrical engineering from Democritus University of Thrace, Greece, the M.Sc. degree in satellite communication engineering from the University of Surrey, U.K., and the Ph.D. degree from the University of Cambridge, U.K. He is currently a Senior Lecturer with Lancaster University, U.K. Prior to his appointment, he held postdoctoral positions with the University of Cambridge and also with the Norwegian University of Science and Technology, supported by the Engineering and Physical Sciences Research Council and the European Research Consortium for Informatics and Mathematics, respectively. His research interests include signal processing, coding theory and performance analysis of communication systems.

Chapter 4

Largely Constrained Noise-guessing Decoding of Short LDPC codes

Journal: IEEE Open Journal of the Communications Society

Status: Accepted

Pages: 1-20

Largely Constrained Noise-guessing Decoding of Short LDPC codes

SAHAR ALLAHKARAM^{1,2} (Student Member, IEEE), FRANCISCO A. MONTEIRO^{1,2} (Member, IEEE), JOÃO RIBEIRO^{1,3}, AND CARLOS SALEMA^{1,3} (Life Member, IEEE)

¹Instituto de Telecomunicações, Portugal

²ISCTE – Instituto Universitário de Lisboa, Portugal

³Instituto Superior Técnico, Universidade de Lisboa, Portugal

Corresponding author: Francisco A. Monteiro (e-mail: francisco.monteiro@lx.it.pt).

This work is funded by national funds through FCT – Fundação para a Ciência e a Tecnologia, I.P., and, when eligible, co-funded by EU funds under project/support UID/50008/2025 – Instituto de Telecomunicações, with DOI identifier <https://doi.org/10.54499/UID/50008/2025>. João Ribeiro is also supported by FCT – Fundação para a Ciência e a Tecnologia, I.P., under project 2024.12595.CMU with DOI <https://doi.org/10.54499/2024.12595.CMU>, and by the European Union (LESYNCH, 101218842). Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them. Sahar Allakharam is funded by a merit scholarship from ISCTE – Instituto Universitário de Lisboa.

ABSTRACT When using guessing random additive noise decoding (GRAND) to decode linear block codes (LBCs), the list of error patterns to be tested is blind to any structure in the parity-check matrix. This paper starts by proposing simple and general set partitioning (SP) constraints that reduce the number of possible patterns to less than half when decoding random linear codes (RLCs). Sparser parity-check matrices induce faster decoding when using SP, but at the expense of a severe performance loss. This loss can be mostly recovered by considering regular or quasi-regular low-density parity-check (LDPC) codes. Given their structure, one can apply the proposed Sieving-GRAND that, for a given support of the syndrome, “filters” possible error positions. Subsequently, a guessing approach is applied to those few positions. The filtering threshold can be successively relaxed to extend the set of candidate positions. We further propose a largely constrained GRAND mechanism (LC-GRAND), which takes advantage of zeroed syndrome positions to efficiently eliminate incompatible error patterns, while not requiring quasi-regular LDPC codes. The two main proposed techniques lead to similar sharp pruning of the search space. For example, LC-GRAND, in combination with SP, reduces the number of membership tests to $\simeq 1\%$. All the proposed hard-decoding techniques retain the same decoding performance of unconstrained GRAND.

INDEX TERMS Constrained GRAND, LDPCs, RLCs, set partitioning, short codes, sieving error positions, URLLC.

I. INTRODUCTION

FOR error protection, binary messages with a length of k bits should be mapped to codewords n bits long, taken from a finite codebook $\mathcal{C}$, with a code rate $R = \frac{k}{n}$. Shannon showed that it is possible to communicate with an error probability that tends to zero when using a code with sufficiently long codewords ($n \rightarrow \infty$), as long as the code rate is below capacity [1]. When employing an error correction code (ECC), the asymptotic bit error probability (BER) decays as $\text{BER} \propto e^{-n\epsilon(R)}$ [2]. The capacity corresponds to the highest code rate for which $\epsilon(R) > 0$. The statistics of the error exponent of random codes have been very recently analyzed in [3] when $n \rightarrow \infty$, proving that it converges to its expected

value for both high and low code rates. Shannon’s non-constructive proof showed that *random* codes could reach capacity [1], [4], [5], however, a simple maximum-likelihood (ML) decoder does not exist for such codes [6]. Decades-long research looked for codes capable of achieving capacity with some embedded *structure*, for which a feasible decoder could exist [7]. The focus was on reaching capacity, and the length of the codewords was not a major constraint. Low-density parity-check (LDPC) codes, also known as Gallager codes [8], later rediscovered by MacKay and Neal (MN) [9], [10], reach capacity for extremely large codewords.

Short codes became relevant for ultra-reliable low-latency communications (URLLC) [11], [12] in 6G [13], [14], even

more than in 5G [15]. The capacity for the *finite* blocklength regime was found by Polyanskiy et al. [16], recovering Shannon's capacity when $n \rightarrow \infty$. The authors pointed out that very few prior works had looked at that problem, as considered by Salema [17, sec. 6.12-6.13], and by Dolinar et al. [18], who revisited Shannon's work considering finite codewords [2].

Ordered statistics decoding (OSD), which has similarities with the concept of noise-guessing decoding, can be used to decode any linear block code (LBC) with quasi-ML performance [19], [20], has been proposed for short codes [21], [22]. Although OSD involves costly gaussian elimination for general LBCs, a low-latency OSD has recently been proposed for the particular case of Bose-Chaudhuri-Hocquenghem (BCH) codes, which eliminated the need of gaussian elimination [23]. The concepts of noise-guessing decoding can be traced back to [24]. Using a constrained serial list Viterbi algorithm, a very low number of codewords needs to be tested [25], however, this comes at the expense of a Gaussian elimination preprocessing for each received codeword, with a complexity that scales with $\approx \mathcal{O}(n^3)$.

High-speed decoding of LDPC codes (with $n = 1027$) has been presented in [26], which includes a list of other fast decoders implemented for n between 672 and 1994 bits. With a focus on fast decoding, short protographs LDPC codes have been proposed for URLLC, unfortunately enforcing rather resource-intensive very low code rates (from $\frac{1}{3}$ to $\frac{1}{12}$) [12].

Guessing random additive noise decoding (GRAND) is a *universal* ML decoder for codes with moderate redundancy, i.e., with high rate. This includes short or high-rate random linear codes (RLCs), which attain the maximum possible rates of the finite blocklength regime [27]. It should be noted that, while OSD appeared as a solution to decode short block codes (including LDPCs) with ML decoding performance, OSD has limited decoding throughput and is not adequate to meet the low-latency decoding aimed for 6G [28]. In addition to the Gaussian elimination required for each received codeword, OSD operates by flipping the k most reliable bits of the n -bits received words, its complexity exponentially increases due to the binomial coefficient $\binom{n}{k}$. For that reason, OSD is only a low-complexity viable alternative when low-rate codes are considered. For the high-rate codes considered in this work (for higher spectral efficiency), OSD is not competitive with noise-guessing approaches regarding decoding latency.

GRAND-type decoders [29] were originally proposed by Duffy et al. as a hard ML decoder [11], [27], [30]. The concept shifted the decoding paradigm from finding the transmitted codeword to finding the error pattern that corrupted that codeword. This is done by successively testing error patterns in decreasing order of likelihood. In the case of RLCs, as for any LBC, the membership test depends entirely on the observed syndrome [4], [29]. GRAND has been extended to soft-decoding [31], [32] and GRAND with joint detection has been analyzed in [33]. GRAND has been applied to fading channels in single-antenna [34]

and multi-antenna systems [35] [36]. GRAND has been implemented in hardware [37], [38], showing record energy-efficient decoding [39]. The noise-guessing concept has also been applied to network coding [40], to quantum ECCs [41] [42], [43], and even to the purification of quantum links [44]. With GRAND, error-*detecting* codes, such as CRCs [45], or even the AES cryptosystem [46], can be turned into an ECC. GRAND can also compromise physical-layer security [47].

Although the statistics of the error patterns are central in determining the *order* in which they should be tested [11], [31], one should also restrict the *number* of possible error patterns that are worth testing. This was done in [48] by taking into account the modulation symbols. One can also take advantage of the parity-check matrix, $\mathbf{H}$. Given a syndrome, not all error patterns are compatible with the observed syndrome, which can greatly reduce the list of possible error patterns $\mathcal{L}$. Other recent proposals have appeared to reduce the decoding complexity of GRAND-type algorithms while preserving their near-ML decoding performance. In [49], the authors make use of a single parity-check constraint to eliminate guesswork in GRAND-based decoding, demonstrating that decoding complexity can be halved without compromising performance, particularly at lower signal-to-noise ratios (SNRs). Taking into account $\mathbf{H}$ and the observed syndrome, some constrained versions of GRAND have been proposed, achieving ML performance and requiring many fewer membership tests. In [50], constraints associated with the rows of $\mathbf{H}$ are used: for each considered row, the average number of tests is halved. However, processing those constraints becomes an extra burden, and the authors recommend using a single row. The work in [51] proposes splitting $\mathbf{H}$ in two matrices: one is used to generate low-weight error patterns via a (partial) trellis search, and the other is used to check whether the candidate error patterns can generate the remaining partial syndrome. In [52] the idea is to start with a list of partial error patterns ordered in decreasing order of likelihood, as defined by the soft metrics of the bits, and complete the remaining error position by testing which position, associated with a column of $\mathbf{H}$, is compatible with the observed syndrome. A unified analytical framework was introduced in [53] to distinguish between guessing noise decoding (GND) and guessing codeword decoding (GCD) algorithms for linear block codes. In this work, GND, representing the family of GRAND-type decoders, was contrasted with GCD approaches, such as OSD, which explicitly exploit the algebraic structure of the code. Both decoding paradigms were shown to achieve ML decoding performance; however, GCD demonstrated greater efficiency in terms of the number of required guesses, particularly for low-rate or highly redundant codes. In [54], [55], a segmented ORBGRAND method was proposed, in which the noise-guessing search space is partitioned into syndrome-consistent sub-patterns and recombined using logistic-weighted compositions. This segmentation reduces redundancy, cutting the average number of decoding queries

by over 66% across all SNR while maintaining near-ML accuracy. In [56], a soft-output GRAND (SOGRAND) decoder was introduced to extend the GRAND framework to soft-information processing. This method generates accurate a posteriori reliability estimates for both single and list decodings, supporting iterative architectures that outperform standardized LDPC codes from 5G New Radio in reliability and latency.

Collectively, these works mark an important evolution of the GRAND family from purely combinatorial noise-guessing toward adaptive, probabilistic, and structure-aware decoding methods. A recent overview and comparison between GRAND, OSD, and GCD is provided in [57], where the authors point out in which regions of coding rate and block length each decoding approach is preferable. Building upon these foundations, the present work proposes a largely constrained GRAND framework that makes use of two newly proposed techniques specifically designed for LDPCs, and later combined with one general purpose complexity-reduction technique. Our proposals introduce parity-check-driven structural constraints into the noise-guessing process. One of our main proposals leverages the inherent sparsity and quasi-regularity of LDPC parity-check matrices to perform syndrome-guided sieving of candidate error positions. The other main proposal looks at the zero-syndrome “filtering”, thereby eliminating incompatible error patterns. In doing that, one achieves ML decoding performance for *any* short LDPC code with significantly reduced computational overhead and latency, extending the practicality of GRAND-type decoders for URLLC scenarios.

Note that the proposed decoders are distance-bounded decoders, which do not try to decode beyond the number of errors t dictated by the minimum Hamming distance, $d_{\min}$. Distance-bounded decoders do not reach capacity, with $n \rightarrow \infty$, as to reach capacity one has to be able to decode errors beyond $t = \frac{d_{\min}-1}{2}$ [10] [4, Ch.13]. However, in the finite blocklength regime imposed in URLLC, this distinction has no practical meaning.

A. Contributions:

Belief-propagation (BP) decoders are known to offer a low-complexity solution suitable for long block lengths, with a complexity that grows linearly with n . However, in the case of short LDPCs, BP falls short from the ML decoding performance [13], [14], and near-ML decoding of short LDPCs has been classified as an open problem in [58]. Noise-guessing decoders are seen as an efficient decoding approach to that goal [13], however none of them takes advantage of the sparsity of the LDPCs. The decoder in [50], which introduced the idea of “pre-filtering” error patterns by applying one or more constraints associated with one or more rows (or combinations of rows) of the parity-check matrix is a state-of-the-art decoding technique that fits well the hard-decoding of sparse codes and will be compared with our proposals. This paper focuses on low-complexity noise-

guessing hard-decoders fit for quasi-regular LDPC codes, while some of the proposed techniques are not restricted to LDPCs. The contributions of this work are the following:

1) Short RLCs with GRAND have shown faster decoding than CRC-aided polar (CA-Polar) codes of similar length or than standalone polar codes using the Tal-Vardy list decoder [31], [35], [59]. This paper starts by proposing in Section III simple constraints based on a set partitioning (SP) of the positions of the errors, which can be applied to *any* LBC, and is tested with RLCs. The existence of a very unbalanced number of ones and zeros in some row of $\mathbf{H}$ determines the reduction of complexity. Sparse RLCs would lead to faster decoding, however, at the expense of the codes’ performance. By considering short LDPC codes, one can retain a good performance.

2) To mitigate the performance degradation observed with sparse random codes, we use short quasi-regular low-density parity-check (LDPC) codes that maintain high reliability while enabling efficient constrained decoding. By relaxing the strict overlap constraints of Gallager and MacKay–Neal constructions from one to two overlaps allowed between any two columns [8]–[10], the proposed LDPC matrices retain uniformly weighted columns suitable for rapid error-pattern filtering in GRAND decoders that is crucial for the Sieving-GRAND technique proposed in Section IV. This is a syndrome-guided “sieving mechanism” that uses the structure of the LDPC matrix to iteratively identify and prioritize likely error positions. This filtering process dynamically enlarges the candidate set of positions in a controlled manner, allowing the decoder to find the ML codeword solution with minimal computational effort.

3) Next, in Section VI we develop the largely constrained GRAND (LC-GRAND) algorithm, which introduces a novel constraint based on zero-valued syndrome components. This additional layer of structural restriction enables the early elimination of incompatible error patterns, providing sharper search-space pruning and faster convergence toward the correct error pattern. Likewise SP, this technique is applicable to both regular and irregular LDPCs. Without loss of generality, for a fair performance comparison, LC-GRAND is assessed with the quasi-regular LDPC code used to assess Sieving-GRAND.

4) Finally, by applying SP to the reduced candidate positions obtained from the Sieving-GRAND and LC-GRAND stages one obtains the most efficient decoding strategy, as shown in the results in Section VII. The integration achieves maximum-likelihood performance while reducing the number of membership tests to approximately 2% of those required by unconstrained GRAND, enabling decoding speeds up to $50\times$ faster.

To improve the manuscript’s readability, Table 1 summarizes the proposed GRAND-based decoding variants and clarifies their relationships. Each variant introduces a distinct constraint or filtering mechanism, and the combined schemes apply multiple mechanisms jointly to further reduce

TABLE 1. Overview of the GRAND-based decoding variants.

Decoder	Mechanism
GRAND	Unconstrained noise-guessing decoding.
SP-GRAND	Classifies the n potential error positions in disjoint subsets where the errors can lie.
Sieving-GRAND	Syndrome-guided filtering of candidate error positions using column-weight thresholding of a reduced parity-check matrix.
LC-GRAND	From the zero-valued syndrome components only tests feasible combinations of error positions based on the weight of the columns of a reduced parity-check matrix.
SPS-GRAND	Applies set partitioning over the reduced candidate-position set obtained from the Sieving technique.
SPLC-GRAND	Applies set partitioning and additionally enforces the LC constraints for sharper pruning.

the search space while always preserving the same ML performance obtained by GRAND [27].

Notation: we use MATLAB's notation to denote rows and columns of a matrix: the i -th row is $\mathbf{H}(i,:)$ and the j -th column is $\mathbf{H}(:,j)$. The Hamming weight of a vector is denoted by its 1-norm, $\|\cdot\|_1$. A set of positions is denoted as $\mathcal{P} \subseteq \{p_1, p_2, \dots, p_n\}$, with size $|\mathcal{P}| \leq n$. This set of positions can be written as $\mathcal{P} = \{p_{j_1}, p_{j_2}, \dots, p_{j_{|\mathcal{P}|}}\}$, where $j_1, j_2, \dots, j_{|\mathcal{P}|}$ are all different. A set can be a set of positions (integers) or a set of binary vectors (error patterns). The addition in GF(2) is denoted by $\oplus$. The *overlap* between two binary column vectors is the number of common positions holding 1s. $\lceil x \rceil$ denotes the smallest integer larger than x . $\mathbf{1}_n$ is the row-vector of ones of size n , $\mathbf{0}_n$ is a column-vector of zeros of size n , $\mathbf{a}^T$ denotes the transpose of vector $\mathbf{a}$, and $\times$ denotes the Cartesian product of sets.

II. Encoding schemes

A. Random Linear Codes: dense and sparse

A column vector $\mathbf{a}$ with k independent and identically distributed (i.i.d.) information bits is encoded by an LBC onto n -bits long codewords, $\mathbf{x}_b$, and then mapped onto binary phase-shift keying (BPSK) or quadrature phase-shift keying (QPSK), subjected to additive white Gaussian noise (AWGN).

We start by considering short (systematic) RLCs, given their performance guarantees, mentioned in Section I. The elements of the generator matrix $\mathbf{G} \in \mathbb{F}_2^{n \times k}$ can be chosen uniformly at random from a Galois field $\mathbb{F}_q = \{0, 1, \dots, q-1\}$, where in this work, $q = 2$. The $\mathbf{G}$ matrix defines a codebook $\mathcal{C} = \{\mathbf{x}_b = \mathbf{G}\mathbf{a} : \mathbf{a} \in \mathbb{F}_2^k\}$ with $2^k = 2^{nR}$ codewords of length n , which is a linear subspace of the discrete vector space $\mathbb{F}_2^n$. For systematic codes, $\mathbf{G}$ is of the form $\mathbf{G} = \begin{bmatrix} \mathbf{P} \\ \mathbf{I}_k \end{bmatrix}$, where $\mathbf{P} \in \mathbb{F}_2^{(n-k) \times k}$ is a random binary matrix, whose elements have probabilities: $P(1) = p$ and

$P(0) = 1 - p$. When $p \geq 0.5$, we say that the RLC is dense, and for $p < 0.5$ we say that the RLC is p -sparse. $\mathbf{I}_k$ is the $k \times k$ identity matrix associated with the systematic part of the encoding.

Any LBC can be also defined by the $(n - k) \times n$ parity-check matrix, $\mathbf{H} = [\mathbf{I}_{n-k} | \mathbf{P}_{(n-k) \times k}] \in \mathbb{F}_2^{(n-k) \times n}$, which spans the null space of $\mathbf{G}$, so that $\mathbf{H}\mathbf{G} = \mathbf{0}_{n-k}$, and thus $\mathbf{H}\mathbf{x}_b = \mathbf{0}_{n-k}, \forall \mathbf{x}_b \in \mathcal{C}$. A received block, after hard-detection, can be written as $\mathbf{y}_b = \mathbf{x}_b \oplus \mathbf{e}$, where $\mathbf{e}$ is the binary error vector after detection. The syndrome, $\mathbf{s} = [s_1, s_2, \dots, s_{(n-k)}]^T \in \mathbb{F}_2^{n-k}$, is computed as $\mathbf{s} = \mathbf{H}\mathbf{y}_b = \mathbf{H}(\mathbf{x}_b \oplus \mathbf{e}) = \mathbf{H}\mathbf{e}$. GRAND uses the syndrome for the membership test, given that $\mathbf{s} = \mathbf{0}_{n-k}$ if and only if $\mathbf{y}_b \oplus \hat{\mathbf{e}} \in \mathcal{C}$, for some candidate error pattern $\hat{\mathbf{e}}$.

B. Quasi-regular LDPC codes with relaxed overlapping

Although short RLCs exist for any length and rate, the rules to design the parity-check matrices of MN codes and LDPC codes may not be feasible to fulfill for some desired lengths or rates. In its simplest form, a quasi-regular MN or LDPC code is constructed via its parity-check matrix with two simple rules [9], [10]:

- 1) randomly generate different sparse columns for $\mathbf{H}$, all with equal weight w_c ;
- 2) any new column, $\mathbf{h}_{\text{new}}$, can only be allowed to have an overlap ≤ 1 with all previously accepted columns.

The resulting code is a non-systematic one. To construct short LDPC codes we relax the second rule and tolerate overlaps ≤ 2 . The overlaps can be easily checked by the inner product between columns of the extended $\mathbf{H}_{\text{ext}} = [\mathbf{H} | \mathbf{h}_{\text{new}}]$. The acceptance of a new column must check the off-diagonal elements of the Gram matrix: a new column $\mathbf{h}_{\text{new}}$ is rejected if any element in of the matrix constructed as $\mathbf{H}_{\text{ext}}^T \mathbf{H}_{\text{ext}} - w_c \mathbf{I}$ is > 2 .

As noted earlier, short RLC codes outperform short LDPC codes. However, for reasons that will become apparent in Section IV.B, having an $\mathbf{H}$ with equal column weights (quasi-regular LDPC codes), will allow to strongly reduce the number of candidate error positions. Regular LDPC codes would additionally have an equal weight, w_r , in all their rows and could be used equally. Given its simpler construction, and similar performance when the codes are short (operating far from capacity), quasi-regular codes will be considered.

III. Set partitioning for Linear Block Codes

We start by proposing the partition of the n positions of a received block $\mathbf{y}_b$ into sets. This allows us to generate a shorter list of admissible error patterns, $\mathcal{L}$.

Definitions: the positions of the syndrome containing a 1 are called the *flagged positions* of the syndrome, and their number is $F = \|\mathbf{s}\|_1$. In all of them, $s(f) = 1$, for $f \in \{1, 2, \dots, n - k\}$. The rows of $\mathbf{H}$ that are associated with these positions are called the *flagged rows*. The flagged

rows are stacked to form the matrix $\mathbf{H}$ -reduced, denoted as $\mathbf{H}_{\text{red}}$. The *support* of a binary vector $\mathbf{a}$ of length n is defined as $\text{supp}(\mathbf{a}) \triangleq \{i \in \{1, \dots, n\} : a_i \neq 0\}$, as in [50]. Accordingly, the i -th row of $\mathbf{H}$ has a support $\text{supp}(\mathbf{H}(i, :))$, whose cardinality (i.e., number of 1s) is its Hamming weight: $\|\mathbf{H}(i, :)\|_1 = |\text{supp}(\mathbf{H}(i, :))|$. Because $\mathbf{s} = \mathbf{H}\mathbf{e}$, we say that an error pattern $\mathbf{e}$ *activates* the j -th column of $\mathbf{H}$ when $e(j) = 1$. Each row i of $\mathbf{H}_{\text{red}}$ has a *set of ones*, whose elements are the positions of the row that contain a 1, denoted by $\mathcal{S}_1^{(i)}$ (corresponding to the support of the row), and a *set of zeros*, which comprises the positions of the row holding a 0, denoted by $\mathcal{S}_0^{(i)}$. The size of these sets is given by $|\mathcal{S}_1^{(i)}|$ and $|\mathcal{S}_0^{(i)}|$, respectively.

A. Construction of the admissible error sets

For a clearer explanation, we start by independently describing how the error patterns of weights $t = \|\mathbf{e}\|_1 = 1, 2, 3$ are generated, and later we generalize to any even or odd weight.

1) One error case

In the case of a single error corrupting the block of n bits, there is no need to run membership tests by successively flipping single bits. Given that $\mathbf{s} = \mathbf{H}\mathbf{e}$, this situation corresponds to the case where only one column of $\mathbf{H}$ is active. This amounts to identifying whether the observed $\mathbf{s}$ is a copy of some column of $\mathbf{H}$. If only one error exists in position $p_1 = j$, then $\mathbf{s} = \mathbf{H}(:, j)$.

2) Two errors case

With $t = 2$ errors, any flagged position, f , in $\mathbf{s}$ can only result from two active columns $\mathbf{H}(:, p_1)$ and $\mathbf{H}(:, p_2)$, where the flagged row $\mathbf{H}(f, :)$ has in the positions $\mathbf{H}(f, p_1)$ and $\mathbf{H}(f, p_2)$ different bits. This means that the position of one of the errors must be in the *set of ones*, $\mathcal{S}_1^{(i)}$, of the row $\mathbf{H}(f, :)$ while the other error must necessarily be contained in the *set of zeros*, $\mathcal{S}_0^{(i)}$, of that same row.

The flagged rows are used to build the smaller matrix $\mathbf{H}_{\text{red}}$. The row of $\mathbf{H}_{\text{red}}$ that discloses more information about the location of the two errors is the one with the highest unbalance between the number of 0s and the number of 1s, i.e., the one where the sizes of the sets $\mathcal{S}_1^{(i)}$ and $\mathcal{S}_0^{(i)}$ that are most different. While the location of two errors among the n positions, all drawn from the same set, entails $|\mathcal{L}_2| = \binom{n}{2}$ possible error patterns, the list of all possible error patterns when the erroneous positions come from two disjoint sets is defined by the Cartesian product $\mathcal{S}_1^{(i)} \times \mathcal{S}_0^{(i)}$, whose size is $|\mathcal{S}_1^{(i)} \times \mathcal{S}_0^{(i)}| = |\mathcal{S}_1^{(i)}| \cdot |\mathcal{S}_0^{(i)}|$. Note that $\mathcal{S}_1^{(i)} \times \mathcal{S}_0^{(i)}$ generates the list of candidate error patterns $\hat{\mathcal{E}}_1, \hat{\mathcal{E}}_2, \dots, \hat{\mathcal{E}}_{|\mathcal{S}_1^{(i)} \times \mathcal{S}_0^{(i)}|}$, where each error pattern is defined by a set of two positions, i.e., $\hat{\mathcal{E}}_i = \{p_{j_1}, p_{j_2}\}$. The associated error pattern $\hat{\mathbf{e}}_i$ is the binary vector with 1s in the two positions in $\hat{\mathcal{E}}_i$.

The number of candidate error patterns can be minimized by choosing the row of $\mathbf{H}_{\text{red}}$ that has the minimum *or*

the maximum weight of the support. From an information-theoretic perspective, that is the most informative row about the locations of the errors. The remaining entropy is subsequently dealt with by applying a noise-guessing approach that only considers the reduced set of candidate error patterns. The two extreme cases correspond to: a) $|\mathcal{S}_1^{(i)}| = 1$, where the number of error patterns with two errors is $|\mathcal{L}_2| = 1 \cdot (n-1)$, which is the case where $\mathcal{S}_1^{(i)}$ fully specifies the location of one of the two errors, leaving the other error position to be guessed among the remaining $n-1$ positions; b) $|\mathcal{S}_1^{(i)}| = |\mathcal{S}_0^{(i)}| = \frac{n}{2}$, leading to $|\mathcal{L}_2| = \frac{n}{2} \cdot \frac{n}{2} = \frac{n^2}{4}$. For the benefit of the algorithmic complexity of the decoder, we opt for only looking at the minimum support row of $\mathbf{H}_{\text{red}}$, i.e., the row of $\mathbf{H}_{\text{red}}$ that has the fewest 1s, defined in the following manner:

$$\mathbf{h}_m^T = \mathbf{H}(m, :), \quad m = \arg \min_{i \in \mathcal{I}} |\text{supp}(\mathbf{H}(i, :))|, \quad (1)$$

where $\mathcal{I} \subseteq \{1, \dots, n-k\}$ denotes the set of row indices selected from $\mathbf{H}$ to form $\mathbf{H}_{\text{red}}$, so that $|\mathcal{I}| = F$.

This option becomes optimal when sparse codes are considered, as $\mathbf{h}_m^T$ will always be the most unbalanced row [36]. In the case of $t = 2$, two other sets should also be formed: the *full set*, $\mathcal{F}$, which contains the positions p_j where the columns $\mathbf{H}_{\text{red}}(:, p_j)$, $p_j \in \mathcal{F}$, are column-vectors of 1s with size F :

$$\mathcal{F} = \{p_j : \mathbf{H}_{\text{red}}(:, p_j) = \mathbf{1}_F^T, j \in \{1, \dots, n\}\}, \quad (2)$$

and the *null set*, $\mathcal{N}$, which contains the positions p_j where the columns $\mathbf{H}_{\text{red}}(:, p_j)$, $p_j \in \mathcal{N}$, are a zero column-vector with $(n-k) - F$ elements:

$$\mathcal{N} = \{p_j : \mathbf{H}_{\text{red}}(:, p_j) = \mathbf{0}_{(n-k)-F}^T, j \in \{1, \dots, n\}\}. \quad (3)$$

If one error $p_1 \in \mathcal{F}$, then it is easy to see that, necessarily, $p_2 \in \mathcal{N}$. Therefore, we have a new set of possible error patterns given by another Cartesian product: $\mathcal{F} \times \mathcal{N}$.

Finally, the complete list of admissible error patterns is $\mathcal{L}_2 = ((\mathcal{S}_1^{(m)} \setminus \mathcal{F}) \times (\mathcal{S}_0^{(m)} \setminus \mathcal{N})) \cup (\mathcal{F} \times \mathcal{N})$, with size $|\mathcal{L}_2| = |\mathcal{S}_1^{(m)} \setminus \mathcal{F}| \cdot |\mathcal{S}_0^{(m)} \setminus \mathcal{N}| + |\mathcal{F}| \cdot |\mathcal{N}|$. This further splitting of the sets in the particular case of $\|\mathbf{e}\|_1 = 2$ allows for an additional reduction of the number of error patterns to be tested, which is more accentuated when the codes are very sparse and more zero-columns appear in $\mathbf{H}_{\text{red}}$.

3) Three errors case

In the case of $t = 3$ errors, the flagged rows, and in particular the minimum support row, $\mathbf{h}_m^T$, can only trigger the flag if one of the two situations happens:

- a) all three positions in error $p_1, p_2, p_3 \in |\mathcal{S}_1^{(m)}|$ and thus one has to search through all the possible error patterns with three positions in $\mathcal{S}_1^{(m)}$;
- b) one error activated a 1 in the row and the other two lie “hidden” in the 0s, i.e., $p_1 \in \mathcal{S}_1^{(m)}$ and $p_2, p_3 \in \mathcal{S}_0^{(m)}$. Thus, one has to search through error patterns resulting

from the Cartesian product of $\mathcal{S}_1^{(m)} \times \mathcal{H}$, where $\mathcal{H}$ is the set of error patterns having all the combinations of two error positions “hiding” in $\mathcal{S}_0^{(m)}$.

With these constraints, the number of admissible error patterns with weight three is $|\mathcal{L}_3| = \binom{|\mathcal{S}_1^{(m)}|}{3} + |\mathcal{S}_1^{(m)}| \cdot \binom{|\mathcal{S}_0^{(m)}|}{2} \ll \binom{n}{3}$.

4) Any even or odd number of errors

Given that only an odd number of error positions in $\mathcal{S}_1^{(m)}$ can cause the syndrome flag associated with row $\mathbf{h}_m^T$, when considering t errors, the set split is performed considering $l = 2a + 1 \leq t$ positions in $\mathcal{S}_1^{(m)}$, for $a \in \mathbb{Z}_0^+$, and the remaining $t - l$ positions are errors “hidden” in $\mathcal{S}_0^{(m)}$. The cases detailed for $t = 1, 2, 3$ are particular examples of this simple rule. For simplicity, the full set and the null set are recommended to be used only in the $t = 2$ case. Without those two sets, when applying set-partitioning, the maximum number of error patterns of weight up to t having to be tested is:

$$|\mathcal{L}| = \sum_{l=0}^t |\mathcal{L}_l| = 1 + \sum_{l=1}^t \sum_{\substack{i=1 \\ i \text{ odd}}}^l \binom{|\mathcal{S}_1^{(m)}|}{i} \cdot \binom{|\mathcal{S}_0^{(m)}|}{l-i}. \quad (4)$$

The following section will show that, for the parity-check matrices of RLCs or LDPCs, $|\mathcal{L}|$ is much smaller than the number of all error patterns of weight $\|\mathbf{e}\|_1 < t$, which amounts to $\sum_{i=0}^t \binom{n}{i}$.

B. Probability mass function of the minimum weight of the rows of $\mathbf{H}_{\text{red}}$

The impact of splitting the n positions into the set of ones, $\mathcal{S}_1^{(m)}$, and the set of zeros, $\mathcal{S}_0^{(m)}$, depends on how low the weight of the minimum weight of the support row of $\mathbf{H}_{\text{red}}$ is. Therefore, it is important to look at the probability distribution of this weight $\|\mathbf{h}_m^T\|_1$, which corresponds to the size $|\mathcal{S}_1^{(m)}|$ in the minimum support row. The probability mass function (PMF) of $|\mathcal{S}_1^{(m)}|$ is plotted in Fig. 1 for RLCs (128,103) with SP-GRAND. Given that $\mathbf{H} = [\mathbf{I}_{n-k} \mid \mathbf{P}_{(n-k) \times k}] \in \mathbb{F}_2^{(n-k) \times n}$, dense RLCs (128,103) have on average 65 1s’ in each row of $\mathbf{H}$: $\frac{n}{2}$ coming from the elements in the $\mathbf{P} \in \mathbb{F}_2^{(n-k) \times k}$ submatrix, plus one coming from the identity. The average weight of the minimum support row, $\mathbb{E}\{\|\mathbf{h}_m^T\|_1\} = \mathbb{E}\{|\mathcal{S}_1^{(m)}|\}$, must be lower than $\frac{n}{2} + 1$. As observed in Fig. 1, the mean value $\mathbb{E}\{|\mathcal{S}_1^{(m)}|\} = 45$, and therefore $\mathbb{E}\{|\mathcal{S}_0^{(m)}|\} = n - \mathbb{E}\{|\mathcal{S}_1^{(m)}|\} = 83$. In this case, and considering $t = 3$, the number of possible error patterns is $|\mathcal{L}_3| = \binom{45}{3} + 45 \cdot \binom{83}{2} \ll \binom{128}{3}$.

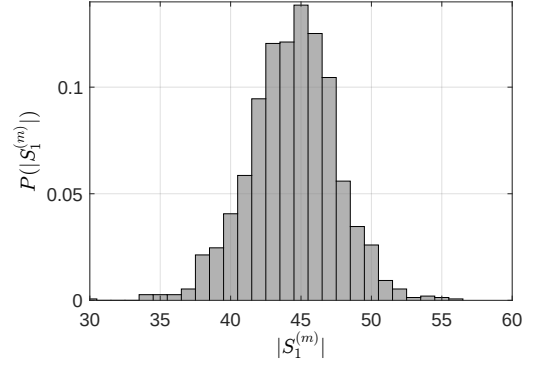


FIGURE 1. Simulated PMF for the number of 1s in the most unbalanced row of $\mathbf{H}_{\text{red}}$, using RLC (128,103).

Fig. 2 compares the number of error patterns required by unconstrained GRAND with the number of error patterns generated by the Cartesian products when set partitioning is applied, considering different weights of the minimum support row, $|\mathcal{S}_1^{(m)}|$. The number of error patterns is plotted as a function of the codewords’ length $2 \leq n \leq 128$, and in the case when one is trying to guess the locations of $t = 2$ errors. In the case of unconstrained GRAND (blue curve), the number of error patterns is $|\mathcal{L}_2| = \binom{n}{2}$. When applying set partitioning GRAND, the worst-case scenario (black curve) corresponds to the case where $|\mathcal{S}_1^{(m)}| = |\mathcal{S}_0^{(m)}| = \frac{n}{2}$, in which case $|\mathcal{L}_2| = \frac{n}{2} \cdot \frac{n}{2} = \frac{n^2}{4}$. This upper bound for SP-GRAND is always much lower than $\binom{n}{2}$. The figure also depicts several curves for a varying number of $|\mathcal{S}_1^{(m)}|$ and $|\mathcal{S}_0^{(m)}| = n - |\mathcal{S}_1^{(m)}|$ pairs. In all cases, $|\mathcal{L}_2| = |\mathcal{S}_1^{(m)}| \times |\mathcal{S}_0^{(m)}| = |\mathcal{S}_1^{(m)}| \cdot |\mathcal{S}_0^{(m)}| \ll \binom{n}{2}$.

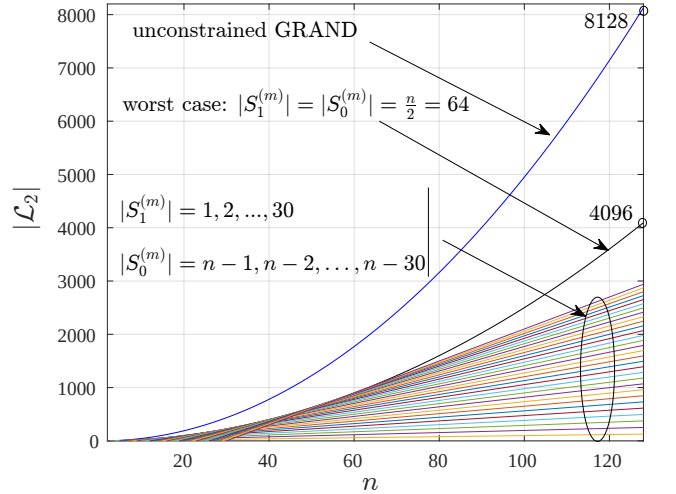


FIGURE 2. Number of error patterns considered in unconstrained GRAND and in set partitioning GRAND (SP-GRAND) for different weights of the minimum support row, $|\mathcal{S}_1^{(m)}|$, when looking for $t=2$ errors in codewords of length n .

IV. Sieving-GRAND for quasi-regular LDPC codes

When the parity-check matrix is both sparse and all columns have equal Hamming weight, such as the ones of quasi-

regular LDPC codes, the number of candidate error positions that need to be tested can be greatly reduced using a simple constraint (or “filter”), capable of gradually sieving candidate positions into a set, $\mathcal{P}$, with more positions being added to the set at different iterations. (See sieve algorithms for the closest vector problem in lattices [60], [61]). At each iteration, a GRAND approach only considers error patterns formed by error positions that belong to that reduced set of positions. We will next define that set and then explain the two proposed filtering mechanisms.

After the l -th iteration, the set of candidate error positions is $\mathcal{P}^{(l)} = \{p_{j_1}, p_{j_2}, \dots, p_{j_{|\mathcal{P}^{(l)}|}}\}$. The size of the set increases monotonically, with more positions added to it in each iteration. The number of iterations is upper bounded by $w_c - 1$. When the algorithm reaches iteration $l = w_c - 1$ without finding the ML error pattern in any of the previous iterations, all the n positions will be in the set in that last iteration. After the first iteration, the set of candidate positions $\mathcal{P}^{(1)}$ is usually very small and, although $|\mathcal{P}^{(l)}|$ grows rapidly with each iteration, the number of error patterns to be tested for t errors in the l -th iteration is $\binom{|\mathcal{P}^{(l)}|}{t} \ll \binom{n}{t}$, for $k = 1, 2, \dots, w_c - 1$. Note that the maximum number of iterations depends on the column weights of the LDPC code; nevertheless, the decoding can successfully terminate at a much earlier iteration if the ML $\hat{\mathbf{e}}$ has been found.

The proposed Sieving-GRAND algorithm implements two techniques involving a) the weight of the syndrome, and b) a Sieving mechanism of candidate positions where the errors may be. They are described in the following.

A. Discarding error patterns by weight

In the case of regular or quasi-regular-LDPC codes, by looking at the weight of the error syndrome, $\|\mathbf{e}\|_1$ it is possible to immediately discard some error patterns. For an $\mathbf{H}$ where all columns have constant weight w_c , the maximum weight of the syndrome can bear when t errors affect a codeword and is $\max\{\|\mathbf{s}\|_1\} = t \cdot w_c$. This happens when all the 1s in the t activated columns of $\mathbf{H}$ are in disjoint rows, i.e., with none of them overlapping. When overlaps of columns $\mathbf{H}_{\text{red}}(:, j)$ take place, $\|\mathbf{s}\|_1 \leq t \cdot w_c$. Therefore, when $\|\mathbf{s}\|_1 > m \cdot w_c$, for some $m \in \mathbb{Z}^+$, the decoder should only test error pattern with weights $\|\hat{\mathbf{e}}\|_1 > m$. Therefore, the first t to be tested should be $t = \lceil \frac{\|\mathbf{s}\|_1}{w_c} \rceil$.

B. Filtering candidate error positions

We start by constructing $\mathbf{H}_{\text{red}} \in \mathbb{F}_2^{F \times n}$, consisting of the F flagged rows of $\mathbf{H}$, as defined in Section III. The Sieving (or filtering) mechanism depends on the Hamming weights of the columns of $\mathbf{H}_{\text{red}}$:

$$\begin{aligned} \Sigma(j) &= \|\mathbf{H}_{\text{red}}(:, j)\|_1 \\ &= \sum_{i=1}^F \mathbf{H}_{\text{red}}(i, j), \quad \text{for } j = 1, \dots, n, \end{aligned} \quad (5)$$

corresponding to the number of 1s in each column of $\mathbf{H}_{\text{red}}$. These sums form a row vector that can be directly obtained via the inner product of the syndrome vector with each column of the parity-check matrix:

$$\Sigma = \mathbf{s}^\top \cdot \mathbf{H}. \quad (6)$$

The iterations begin with an empty set of candidate positions $\mathcal{P}^{(0)} = \emptyset$. Next, more candidate positions are added such that $\mathcal{P}^{(1)} = \emptyset \cup \{p_i\}_{\text{new}}$. The mechanism for accepting positions into the candidate set at the l -th iteration is the following:

$$\mathcal{P}^{(l)} = \{p_j : \Sigma(j) \geq w_{\text{th}}(l), j \in \{1, \dots, n\}\}, \quad (7)$$

where j indexes the columns of $\mathbf{H}_{\text{red}}$ and $w_{\text{th}}(l)$ is the value of a weight threshold during the l -th iteration, starting at $w_{\text{th}}(1) = w_c$. While the number of possible positions is $|\mathcal{P}^{(l)}| < t$, w_{th} is lowered by one unit: $w_{\text{th}}(l+1) = w_{\text{th}}(l) - 1$ until it reaches a pre-defined minimum value $w_{\text{th}_{\min}}$. By choosing $w_{\text{th}_{\min}} = 1$ in Sieving-GRAND, the decoding performance must coincide with the one of GRAND (i.e., ML); a choice of $w_{\text{th}_{\min}} > 1$ will lead to faster but sub-optimal decoding, leaving some errors uncorrected. At this point, the error patterns are tested by running GRAND on the reduced set of positions $\mathcal{P}^{(l)}$, involving only $|\mathcal{L}| = \binom{|\mathcal{P}^{(l)}|}{t} \ll \binom{n}{t}$ candidate error patterns. The threshold is successively reduced in unit steps until the set of candidate positions $\mathcal{P}^{(l)}$ includes *all* the t positions where the ML errors lie, in which case the decoding of the ML error pattern $\hat{\mathbf{e}}$ will take place during the guessing stage. The need to have equal-weight columns in $\mathbf{H}$ assures an unbiased selection of positions to be added to $\mathcal{P}^{(l)}$ at iteration l , meaning that they all hold an equal *a priori* probability. If the columns of $\mathbf{H}$ were to have different weights, the columns with larger weights would have more matches with the syndrome flags and would be selected for inclusion in $\mathcal{P}^{(l)}$ at early iterations, even when that column has not been activated by an error position.

For a visualization of these calculations, consider the example in Fig. 3 with $n = 30$. An overview of the procedure is provided through the flowchart in Fig. 4, and detailed in Algorithm 1.

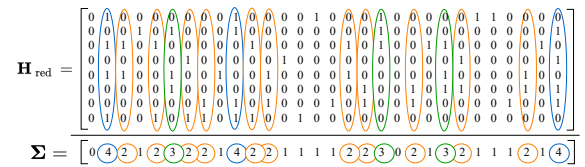


FIGURE 3. Hamming weights of the columns of $\mathbf{H}_{\text{red}}$ and formation of the candidate set $\mathcal{P}^{(l)}$, for iterations $l = 1, 2, 3$, which respectively correspond to the column-weight thresholds $w_{\text{th}} = 4, 3, 2$.

In this example, the different colors indicate the set of new positions that are added to the pool at each iteration. In the first iteration, the selected candidate positions are $\mathcal{P}^{(1)} = \{p_1, p_2, p_3\} = \{2, 10, 30\}$ (colored in blue) for the first threshold $w_{\text{th}}(1) = 4$. We denote such

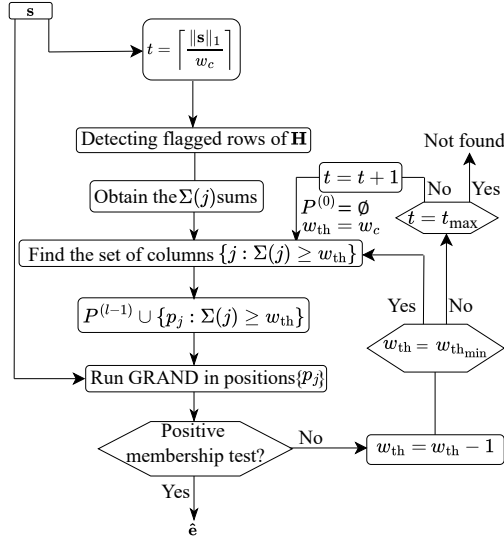


FIGURE 4. Sieving (or filtering) candidate positions using i) a column weight threshold, and ii) the weight of the syndrome to discard error weights.

Algorithm 1 Sieving (filtering) candidate positions using (i) a column-weight threshold and (ii) the syndrome weight to discard error weights

```

1: Input: syndrome  $s$ 
2: Construct  $\mathbf{H}_{\text{red}}$  from the flagged rows of  $\mathbf{H}$ 
3: Compute  $\Sigma(j) = \|\mathbf{H}_{\text{red}}(:, j)\|_1$ ,  $j = 1, \dots, n$ 
4:  $w_{\text{th\_min}} \leftarrow 1$ 
5: for  $t = \lceil \frac{\|s\|_1}{w_c} \rceil$  to  $t_{\text{max}}$  do
6:    $w_{\text{th}} \leftarrow w_c$ 
7:    $P^{(0)} \leftarrow \emptyset$ 
8:    $l \leftarrow 0$ 
9:   while  $w_{\text{th}} \geq w_{\text{th\_min}}$  do
10:     $l \leftarrow l + 1$ 
11:     $P^{(l)} \leftarrow P^{(l-1)} \cup \{p_j : \Sigma(j) \geq w_{\text{th}}(l)\}$ 
12:    Run GRAND restricted to positions in  $P^{(l)}$ 
13:    if positive membership test then
14:      Output:  $\hat{e}$ 
15:      return
16:    end if
17:     $w_{\text{th}} \leftarrow w_{\text{th}} - 1$ 
18:  end while
19: end for
20: Output: Not found
    
```

columns as columns of type $\mathbb{S}_4$, defined by their weight. The set of candidate positions is subsequently extended to $\mathcal{P}^{(2)} = \{p_1, p_2, p_3, p_4, p_5\} = \{2, 6, 10, 19, 23, 30\}$ (colored in green), for $w_{\text{th}}(2) = 3$; similarly, we denote such columns as columns of type $\mathbb{S}_3$. Finally, the candidate positions are extended (if needed) to include the remaining positions where $\|\mathbf{H}_{\text{red}}(:, j)\|_1 \geq 2$ (colored orange), which are denoted as columns of type $\mathbb{S}_2$.

Error patterns that have been tested in a previous iteration do not need to be retested after adding new positions to $\mathcal{P}$, as only error patterns that involve *new* positions added during each iteration need to be considered.

One should note that the proposed Sieving mechanism is rather different from majority logic decoding (MLD) for LDPC codes, first proposed by Gallager as one-shot hard-decoder [8], and which later evolved to the iterative (hard or soft) bit-flipping algorithm [62, Ch. 8], [63, Ch. 8]. In MLD, parity-check equations cast votes to iteratively determine and flip symbol values, directly modifying the received word during decoding. The Sieving mechanism does not perform bit decisions nor iterative correction. Instead, it leverages syndrome-parity-check consistency to rank candidate error positions that are successively added to the pool of candidate locations for the errors. This preserves maximum-likelihood decoding (the ML error pattern is never discarded without testing) while greatly reducing the search space in comparison to GRAND.

C. Probability mass function of the set of candidate positions $|\mathcal{P}^{(l)}|$ containing the t errors

The Sieving approach adds at each iteration l a new set of positions to the set $\mathcal{P}^{(l-1)}$, leading to $\mathcal{P}^{(l)}$. Here we analyze the probability distribution of the increasing set of candidate positions, containing the t errors after l iterations. A positive membership test takes place at the l -th iteration if and only if the correct t error positions $\{p_1, p_2, \dots, p_t\} \in \mathcal{P}^{(l)}$. Fig. 5a) shows the continuous-variable approximated PMF of the size $|\mathcal{P}^{(l)}|$ that contains *all* the $t = 3$ errors for an LDPC code (128,104) with the Sieving-GRAND decoder. We get that the set sizes $|\mathcal{P}^{(l)}| \leq 10$ concentrate $\approx 74\%$ of the probability mass function (as seen in the cumulative distribution function (CDF) plotted in Fig. 5b). This very small pool of candidate positions translates into an extreme reduction of the average number of error patterns to be tested.

V. Mixing set partitioning and Sieving-GRAND

By combining set partitioning and the Sieving technique, one obtains set partitioning Sieving-GRAND (SPS-GRAND). To shed light on the concept, an example with a small parity-check matrix, with $n = 30$ and $n - k = 20$, will be discussed in the following (and illustrated in Fig. 6).

- Let us consider a number of flags in the syndrome $F = \|s\|_1 = 8$.
- both the *flags* and the *flagged rows* are highlighted by red arrows (rows $\{1, 3, 4, 5, 7, 9, 14, 16\}$).
- the flagged rows are stacked to form $\mathbf{H}_{\text{red}}$.
- the columns' Hamming weights are obtained: $\Sigma(j) = \|\mathbf{H}_{\text{red}}(:, j)\|_1$, and the positions $\mathcal{P}^{(1)} = \{2, 10, 30\}$ are first selected, for a threshold $w_{\text{th}} = 4$, $\mathcal{P}^{(2)} = \{2, 6, 10, 19, 23, 30\}$, for $w_{\text{th}} = 3$, and finally extended (if needed) to include the remaining positions where $\|\mathbf{H}_{\text{red}}(:, j)\|_1 \geq 2$.

zero-valued syndrome positions and column-wise Hamming weight properties of the parity-check matrix.

For a binary syndrome vector $\mathbf{s}$, define the set F_0 as the positions in the syndrome vector where the value is zero, which will be called *zero-flagged positions*. The number of such positions is $|F_0| = (n - k) - F$, and the corresponding rows are called *the zero-flagged rows* of $\mathbf{H}$, which satisfy $\mathbf{H}(f_0, :) \cdot \mathbf{e} = 0$, for $f_0 \in \{1, 2, \dots, n - k\}$. These positions are stacked to form matrix $\mathbf{H}_{\text{red}} \in \mathbb{F}_2^{|F_0| \times n}$. The decoding strategy depends on analyzing the column-wise weights of $\mathbf{H}_{\text{red}}$. Similarly to vector Σ , defined in Subsection IV.B, we define the row vector $\bar{\Sigma}$, whose elements are the sums of the columns of matrix $\mathbf{H}_{\text{red}}$:

$$\begin{aligned} \bar{\Sigma}(j) &= \|\mathbf{H}_{\text{red}}(:, j)\|_1 \\ &= \sum_{i=1}^{|F_0|} \mathbf{H}_{\text{red}}(i, j), \quad \text{for } j = 1, \dots, n. \end{aligned} \quad (8)$$

This “sums vector” can be obtained via the product of the syndrome vector with the parity-check matrix, noting the following relation between $\mathbf{H}_{\text{red}}$ and $\mathbf{H}$:

$$\begin{aligned} \bar{\Sigma} &= w_c \mathbf{1}_n - \Sigma \\ &= w_c \mathbf{1}_n - \mathbf{s}^T \cdot \mathbf{H}, \end{aligned} \quad (9)$$

where $\mathbf{1}$ is the row-vector of ones of size n . The relation captured in (9) comes from the construction of the considered quasi-regular LDPC codes. Considering that all columns of $\mathbf{H}$ have weight w_c , when the j -th column of $\mathbf{H}_{\text{red}}$ has weight $\Sigma(j)$, then the corresponding j -th column of $\mathbf{H}_{\text{red}}$ necessarily holds the remaining $\bar{\Sigma}(j) = w_c - \Sigma(j)$ 1s.

The columns of $\mathbf{H}_{\text{red}}$ are classified according to their weights. Specifically, $\mathbb{C}_i$ denotes the set of columns having $\|\mathbf{H}_{\text{red}}(:, j)\|_1 = i$, for $i = 0, 1, \dots, w_{\text{th}}^{(\text{LC})}$. For a fair comparison between Sieving-GRAND and LC-GRAND, we use the same quasi-regular LDPC and therefore the maximum weight a column of $\mathbf{H}_{\text{red}}$ can take is

$$w_{\text{th}}^{(\text{LC})} = w_c - w_{\text{thmin}}, \quad (10)$$

where w_{thmin} has been defined in Section IV.B. An error pattern is valid only if the activated columns from these sets yield even parity across all flagged zero rows. This principle allows many patterns to be eliminated in advance. To illustrate how the proposed LC-GRAND method narrows down viable error patterns, we examine cases of error patterns with weights $t = 2, 3, 4$. In each example, we analyze whether the given Cartesian product of sets $\mathbb{C}_i$ can generate a valid error pattern. Each set $\mathbb{C}_i$ contributes exactly i ones in the corresponding column, i.e., $\mathbb{C}_0$ contains no 1s, $\mathbb{C}_1$ contains one 1, $\mathbb{C}_2$ contains two 1s, and so on, up to $\mathbb{C}_{w_{\text{th}}^{(\text{LC})}}$. We aim to cancel 1s across rows so that each row reaches even parity. If all 1s can be paired or grouped in such a way that every row ends up with an even number of 1s (or dashes after cancellation), the result is a valid error pattern (even). Otherwise, if any row has an unpaired 1 (odd parity), it cannot lead to a valid error pattern.

This rule enables us to identify feasible error patterns in combinations of sets by examining how the 1s align and whether full cancellation is achievable. The error sets of weight t have a structure $\mathbb{C}_{i_1} \times \mathbb{C}_{i_2} \cdots \times \mathbb{C}_{i_t} = \prod_{a=1}^t \mathbb{C}_{i_a}$, where $i_a \in \{1, 2, \dots, w_{\text{th}}^{(\text{LC})}\}$. Determining whether one of such structures is feasible, is an algorithmic problem that can be generally framed and solved by making use of the following simple rules:

- 1) Place the sets $\mathbb{C}_i$ with increasing cardinality from left to right;
- 2) group horizontal even numbers of 1s and eliminate them;
- 3) if there is a remaining 1 in a row, it descends to a row below where it can turn that row even; if no such row is found, that 1 drops to an extra row at the bottom.

Algorithm 2 formally describes how these rules are applied, and to make this procedure clearer some examples will be explored in detail in the following.

Algorithm 2 Checking feasibility of an error pattern with LC-GRAND constraints

- 1: **Input:** A collection of t sets $\{\mathbb{C}_i\}$
 - 2: Sort the sets $\mathbb{C}_i$ in increasing order of their cardinality
 - 3: Place the sets from left to right according to this order
 - 4: **for** each row r **do**
 - 5: **while** exists a horizontal pair of 1s in row r **do**
 - 6: Group two 1s and eliminate them
 - 7: **end while**
 - 8: **if** exactly one 1 remains in row r **then**
 - 9: Search for a row $r' > r$ that becomes even by adding this 1
 - 10: **if** such a row r' exists **then**
 - 11: Move remaining 1 to row r'
 - 12: Eliminate the pair
 - 13: **else**
 - 14: Move remaining 1 to extra row at the bottom
 - 15: **end if**
 - 16: **end if**
 - 17: **end for**
 - 18: **if** all rows have no 1 left unpaired **then**
 - 19: **Output:** Feasible error pattern type
 - 20: **else**
 - 21: **Output:** Infeasible error pattern type
 - 22: **end if**
-

Feasible example for $t = 2$:

$$\begin{array}{c} \mathbb{C}_1 \times \mathbb{C}_1 \\ 1 \quad 1 \rightarrow - \end{array}$$

- each $\mathbb{C}_1$ contributes a single 1. These two 1s cancel each other out, leaving no unmatched 1s, so the resulting parity is even, meaning this set is a valid error pattern.

TABLE 3. Valid and invalid error patterns for two errors ($t = 2$) under LC-GRAND constraints.

$\mathbf{H}_{\text{red}}$	$\mathbf{H}_{\text{red}}^{\perp}$	Possible patterns
$\mathbb{S}_4 \times \mathbb{S}_4$	$\mathbb{C}_0 \times \mathbb{C}_0$	Yes
$\mathbb{S}_4 \times \mathbb{S}_3$	$\mathbb{C}_0 \times \mathbb{C}_1$	No
$\mathbb{S}_4 \times \mathbb{S}_2$	$\mathbb{C}_0 \times \mathbb{C}_2$	No
$\mathbb{S}_4 \times \mathbb{S}_1$	$\mathbb{C}_0 \times \mathbb{C}_3$	No
$\mathbb{S}_3 \times \mathbb{S}_3$	$\mathbb{C}_1 \times \mathbb{C}_1$	Yes
$\mathbb{S}_3 \times \mathbb{S}_2$	$\mathbb{C}_1 \times \mathbb{C}_2$	No
$\mathbb{S}_3 \times \mathbb{S}_1$	$\mathbb{C}_1 \times \mathbb{C}_3$	No
$\mathbb{S}_2 \times \mathbb{S}_2$	$\mathbb{C}_2 \times \mathbb{C}_2$	Yes
$\mathbb{S}_2 \times \mathbb{S}_1$	$\mathbb{C}_2 \times \mathbb{C}_3$	No
$\mathbb{S}_1 \times \mathbb{S}_1$	$\mathbb{C}_3 \times \mathbb{C}_3$	No, Overlap > 2

TABLE 4. Feasible and infeasible error patterns for three errors ($t = 3$) using LC-GRAND constraints.

$\mathbf{H}_{\text{red}}$	$\mathbf{H}_{\text{red}}^{\perp}$	Possible patterns
$\mathbb{S}_4 \times \mathbb{S}_4 \times \mathbb{S}_4$	$\mathbb{C}_0 \times \mathbb{C}_0 \times \mathbb{C}_0$	Yes
$\mathbb{S}_4 \times \mathbb{S}_4 \times \mathbb{S}_3$	$\mathbb{C}_0 \times \mathbb{C}_0 \times \mathbb{C}_1$	No
$\mathbb{S}_4 \times \mathbb{S}_3 \times \mathbb{S}_3$	$\mathbb{C}_0 \times \mathbb{C}_1 \times \mathbb{C}_1$	Yes
$\mathbb{S}_4 \times \mathbb{S}_4 \times \mathbb{S}_2$	$\mathbb{C}_0 \times \mathbb{C}_0 \times \mathbb{C}_2$	No
$\mathbb{S}_4 \times \mathbb{S}_3 \times \mathbb{S}_2$	$\mathbb{C}_0 \times \mathbb{C}_1 \times \mathbb{C}_2$	No
$\mathbb{S}_4 \times \mathbb{S}_2 \times \mathbb{S}_2$	$\mathbb{C}_0 \times \mathbb{C}_2 \times \mathbb{C}_2$	Yes
$\mathbb{S}_4 \times \mathbb{S}_4 \times \mathbb{S}_1$	$\mathbb{C}_0 \times \mathbb{C}_0 \times \mathbb{C}_3$	No
$\mathbb{S}_4 \times \mathbb{S}_3 \times \mathbb{S}_1$	$\mathbb{C}_0 \times \mathbb{C}_1 \times \mathbb{C}_3$	No
$\mathbb{S}_4 \times \mathbb{S}_2 \times \mathbb{S}_1$	$\mathbb{C}_0 \times \mathbb{C}_2 \times \mathbb{C}_3$	No
$\mathbb{S}_4 \times \mathbb{S}_1 \times \mathbb{S}_1$	$\mathbb{C}_0 \times \mathbb{C}_3 \times \mathbb{C}_3$	Yes
$\mathbb{S}_3 \times \mathbb{S}_3 \times \mathbb{S}_3$	$\mathbb{C}_1 \times \mathbb{C}_1 \times \mathbb{C}_1$	No
$\mathbb{S}_3 \times \mathbb{S}_3 \times \mathbb{S}_2$	$\mathbb{C}_1 \times \mathbb{C}_1 \times \mathbb{C}_2$	Yes
$\mathbb{S}_3 \times \mathbb{S}_2 \times \mathbb{S}_2$	$\mathbb{C}_1 \times \mathbb{C}_2 \times \mathbb{C}_2$	No
$\mathbb{S}_3 \times \mathbb{S}_3 \times \mathbb{S}_1$	$\mathbb{C}_1 \times \mathbb{C}_1 \times \mathbb{C}_3$	No
$\mathbb{S}_3 \times \mathbb{S}_2 \times \mathbb{S}_1$	$\mathbb{C}_1 \times \mathbb{C}_2 \times \mathbb{C}_3$	Yes
$\mathbb{S}_3 \times \mathbb{S}_1 \times \mathbb{S}_1$	$\mathbb{C}_1 \times \mathbb{C}_3 \times \mathbb{C}_3$	No
$\mathbb{S}_2 \times \mathbb{S}_2 \times \mathbb{S}_2$	$\mathbb{C}_2 \times \mathbb{C}_2 \times \mathbb{C}_2$	No
$\mathbb{S}_2 \times \mathbb{S}_2 \times \mathbb{S}_1$	$\mathbb{C}_2 \times \mathbb{C}_2 \times \mathbb{C}_3$	No
$\mathbb{S}_2 \times \mathbb{S}_1 \times \mathbb{S}_1$	$\mathbb{C}_2 \times \mathbb{C}_3 \times \mathbb{C}_3$	No
$\mathbb{S}_1 \times \mathbb{S}_1 \times \mathbb{S}_1$	$\mathbb{C}_3 \times \mathbb{C}_3 \times \mathbb{C}_3$	No

TABLE 5. Feasible and infeasible error patterns for four errors ($t = 4$) using LC-GRAND constraints.

$\mathbf{H}_{\text{red}}$	$\mathbf{H}_{\text{red}}^{\perp}$	Possible patterns
$\mathbb{S}_4 \times \mathbb{S}_4 \times \mathbb{S}_4 \times \mathbb{S}_4$	$\mathbb{C}_0 \times \mathbb{C}_0 \times \mathbb{C}_0 \times \mathbb{C}_0$	Yes
$\mathbb{S}_4 \times \mathbb{S}_4 \times \mathbb{S}_4 \times \mathbb{S}_3$	$\mathbb{C}_0 \times \mathbb{C}_0 \times \mathbb{C}_0 \times \mathbb{C}_1$	No
$\mathbb{S}_4 \times \mathbb{S}_4 \times \mathbb{S}_3 \times \mathbb{S}_3$	$\mathbb{C}_0 \times \mathbb{C}_0 \times \mathbb{C}_1 \times \mathbb{C}_1$	Yes
$\mathbb{S}_4 \times \mathbb{S}_3 \times \mathbb{S}_3 \times \mathbb{S}_3$	$\mathbb{C}_0 \times \mathbb{C}_1 \times \mathbb{C}_1 \times \mathbb{C}_1$	No
$\mathbb{S}_4 \times \mathbb{S}_4 \times \mathbb{S}_4 \times \mathbb{S}_2$	$\mathbb{C}_0 \times \mathbb{C}_0 \times \mathbb{C}_0 \times \mathbb{C}_2$	Yes
$\mathbb{S}_4 \times \mathbb{S}_4 \times \mathbb{S}_3 \times \mathbb{S}_2$	$\mathbb{C}_0 \times \mathbb{C}_0 \times \mathbb{C}_1 \times \mathbb{C}_2$	No
$\mathbb{S}_4 \times \mathbb{S}_3 \times \mathbb{S}_3 \times \mathbb{S}_2$	$\mathbb{C}_0 \times \mathbb{C}_1 \times \mathbb{C}_1 \times \mathbb{C}_2$	Yes
$\mathbb{S}_4 \times \mathbb{S}_4 \times \mathbb{S}_2 \times \mathbb{S}_2$	$\mathbb{C}_0 \times \mathbb{C}_0 \times \mathbb{C}_2 \times \mathbb{C}_2$	Yes
$\mathbb{S}_4 \times \mathbb{S}_3 \times \mathbb{S}_2 \times \mathbb{S}_2$	$\mathbb{C}_0 \times \mathbb{C}_1 \times \mathbb{C}_2 \times \mathbb{C}_2$	No
$\mathbb{S}_4 \times \mathbb{S}_2 \times \mathbb{S}_2 \times \mathbb{S}_2$	$\mathbb{C}_0 \times \mathbb{C}_2 \times \mathbb{C}_2 \times \mathbb{C}_2$	Yes
$\mathbb{S}_4 \times \mathbb{S}_4 \times \mathbb{S}_4 \times \mathbb{S}_1$	$\mathbb{C}_0 \times \mathbb{C}_0 \times \mathbb{C}_0 \times \mathbb{C}_3$	No
$\mathbb{S}_4 \times \mathbb{S}_4 \times \mathbb{S}_3 \times \mathbb{S}_1$	$\mathbb{C}_0 \times \mathbb{C}_0 \times \mathbb{C}_1 \times \mathbb{C}_3$	Yes
$\mathbb{S}_4 \times \mathbb{S}_3 \times \mathbb{S}_3 \times \mathbb{S}_1$	$\mathbb{C}_0 \times \mathbb{C}_1 \times \mathbb{C}_1 \times \mathbb{C}_3$	No
$\mathbb{S}_4 \times \mathbb{S}_4 \times \mathbb{S}_2 \times \mathbb{S}_1$	$\mathbb{C}_0 \times \mathbb{C}_0 \times \mathbb{C}_2 \times \mathbb{C}_3$	No
$\mathbb{S}_4 \times \mathbb{S}_3 \times \mathbb{S}_2 \times \mathbb{S}_1$	$\mathbb{C}_0 \times \mathbb{C}_1 \times \mathbb{C}_2 \times \mathbb{C}_3$	Yes
$\mathbb{S}_4 \times \mathbb{S}_2 \times \mathbb{S}_2 \times \mathbb{S}_1$	$\mathbb{C}_0 \times \mathbb{C}_2 \times \mathbb{C}_2 \times \mathbb{C}_3$	No
$\mathbb{S}_4 \times \mathbb{S}_4 \times \mathbb{S}_1 \times \mathbb{S}_1$	$\mathbb{C}_0 \times \mathbb{C}_0 \times \mathbb{C}_3 \times \mathbb{C}_3$	Yes
$\mathbb{S}_4 \times \mathbb{S}_3 \times \mathbb{S}_1 \times \mathbb{S}_1$	$\mathbb{C}_0 \times \mathbb{C}_1 \times \mathbb{C}_3 \times \mathbb{C}_3$	No
$\mathbb{S}_4 \times \mathbb{S}_2 \times \mathbb{S}_1 \times \mathbb{S}_1$	$\mathbb{C}_0 \times \mathbb{C}_2 \times \mathbb{C}_3 \times \mathbb{C}_3$	Yes
$\mathbb{S}_4 \times \mathbb{S}_1 \times \mathbb{S}_1 \times \mathbb{S}_1$	$\mathbb{C}_0 \times \mathbb{C}_3 \times \mathbb{C}_3 \times \mathbb{C}_3$	No
$\mathbb{S}_3 \times \mathbb{S}_3 \times \mathbb{S}_3 \times \mathbb{S}_3$	$\mathbb{C}_1 \times \mathbb{C}_1 \times \mathbb{C}_1 \times \mathbb{C}_1$	Yes
$\mathbb{S}_3 \times \mathbb{S}_3 \times \mathbb{S}_3 \times \mathbb{S}_2$	$\mathbb{C}_1 \times \mathbb{C}_1 \times \mathbb{C}_1 \times \mathbb{C}_2$	No
$\mathbb{S}_3 \times \mathbb{S}_3 \times \mathbb{S}_2 \times \mathbb{S}_2$	$\mathbb{C}_1 \times \mathbb{C}_1 \times \mathbb{C}_2 \times \mathbb{C}_2$	Yes
$\mathbb{S}_3 \times \mathbb{S}_2 \times \mathbb{S}_2 \times \mathbb{S}_2$	$\mathbb{C}_1 \times \mathbb{C}_2 \times \mathbb{C}_2 \times \mathbb{C}_2$	No
$\mathbb{S}_3 \times \mathbb{S}_3 \times \mathbb{S}_3 \times \mathbb{S}_1$	$\mathbb{C}_1 \times \mathbb{C}_1 \times \mathbb{C}_1 \times \mathbb{C}_3$	Yes
$\mathbb{S}_3 \times \mathbb{S}_3 \times \mathbb{S}_2 \times \mathbb{S}_1$	$\mathbb{C}_1 \times \mathbb{C}_1 \times \mathbb{C}_2 \times \mathbb{C}_3$	No
$\mathbb{S}_3 \times \mathbb{S}_2 \times \mathbb{S}_2 \times \mathbb{S}_1$	$\mathbb{C}_1 \times \mathbb{C}_2 \times \mathbb{C}_2 \times \mathbb{C}_3$	Yes
$\mathbb{S}_3 \times \mathbb{S}_3 \times \mathbb{S}_1 \times \mathbb{S}_1$	$\mathbb{C}_1 \times \mathbb{C}_1 \times \mathbb{C}_3 \times \mathbb{C}_3$	Yes
$\mathbb{S}_3 \times \mathbb{S}_2 \times \mathbb{S}_1 \times \mathbb{S}_1$	$\mathbb{C}_1 \times \mathbb{C}_2 \times \mathbb{C}_3 \times \mathbb{C}_3$	No
$\mathbb{S}_3 \times \mathbb{S}_1 \times \mathbb{S}_1 \times \mathbb{S}_1$	$\mathbb{C}_1 \times \mathbb{C}_3 \times \mathbb{C}_3 \times \mathbb{C}_3$	Yes
$\mathbb{S}_2 \times \mathbb{S}_2 \times \mathbb{S}_2 \times \mathbb{S}_2$	$\mathbb{C}_2 \times \mathbb{C}_2 \times \mathbb{C}_2 \times \mathbb{C}_2$	Yes
$\mathbb{S}_2 \times \mathbb{S}_2 \times \mathbb{S}_2 \times \mathbb{S}_1$	$\mathbb{C}_2 \times \mathbb{C}_2 \times \mathbb{C}_2 \times \mathbb{C}_3$	No
$\mathbb{S}_2 \times \mathbb{S}_2 \times \mathbb{S}_1 \times \mathbb{S}_1$	$\mathbb{C}_2 \times \mathbb{C}_2 \times \mathbb{C}_3 \times \mathbb{C}_3$	Yes
$\mathbb{S}_2 \times \mathbb{S}_1 \times \mathbb{S}_1 \times \mathbb{S}_1$	$\mathbb{C}_2 \times \mathbb{C}_3 \times \mathbb{C}_3 \times \mathbb{C}_3$	No
$\mathbb{S}_1 \times \mathbb{S}_1 \times \mathbb{S}_1 \times \mathbb{S}_1$	$\mathbb{C}_3 \times \mathbb{C}_3 \times \mathbb{C}_3 \times \mathbb{C}_3$	Yes

construction of the LDPC requires $k = 104$). For bit energy E_b and noise power spectral density N_0 , we use the block error rate (BLER) as the performance metric, and the average number of error patterns tested as the complexity metric, both assessed as a function of E_b/N_0 . In the case of RLCs, each transmitted message was assigned a randomly generated code so that the average performance of RLCs is measured. Let $t = 1, 2, 3$ be the maximum number of errors that a distance-bounded GRAND-based decoder aims to correct.

Fig. 10 depicts the results using RLCs with set partitioning GRAND (SP-GRAND) compared to unconstrained GRAND. All BLERs are identical (all curves overlap), and the complexity curves show that SP-GRAND is substantially faster than unconstrained GRAND. For $t = 1$, only one membership check is needed, as seen in Section 1, as it is

well known. For the more interesting $t = 2, 3$, SP-GRAND achieves 68% and 55% reductions if the average number of membership tests, respectively. We note that the results for $t = 2$ considered the sets $\mathcal{F}$ and $\mathcal{N}$, however it was observed that the extra reduction attained by considering the $\mathcal{F} \times \mathcal{N}$ is negligible. Given the extra pre-processing complexity they induce, we recommend these extra partitions to be implemented only for very sparse codes, where a larger $\mathcal{N}$ can arise in $\mathbf{H}_{\text{red}}$.

Fig. 11 compares Sieving-GRAND and unconstrained GRAND for the LDPC code (128,104). The overlapping BLER curves confirm that Sieving-GRAND attains the same performance of unconstrained GRAND, given that all possible error patterns can eventually be tested. It is important to

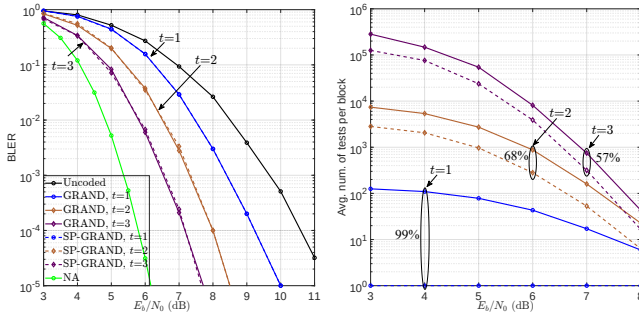


FIGURE 10. BLER performance (left) and decoding complexity (right) for $t=1,2,3$, using RLC (128,103) with set partitioning GRAND (SP-GRAND) compared with unconstrained GRAND. The normal approximation bound for codewords (128, 103) is also included.

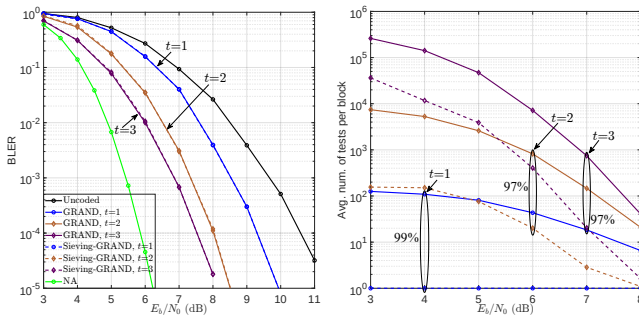


FIGURE 11. BLER performance (left) and decoding complexity (right) for $t=1,2,3$, using an LDPC (128,104) with Sieving-GRAND compared with unconstrained GRAND. The normal approximation bound for codewords (128, 104) is also included.

note that the RLCs outperform the similar-size LDPC code (comparing the left figures of Fig. 10 and Fig. 11). The 0.5 dB loss is due to the LDPC code itself rather than to the filtering mechanism applied in the decoding. The code's structure allows for much faster decoding, which is traded off by its slightly less good performance.

The normal approximation (NA) is a metric that is commonly used in the finite blocklength regime [64], [65], [66, Sec.1.5, App.C.1]. NA bounds have been included in both Fig. 10 and Fig. 11 for the respective code sizes. In Fig. 10 one can observe that the RLC (128,103) gets 1.5 dB close to the NA bound with GRAND or with SP-GRAND (both of which are hard-decoders). In [13], the performance results for RLC (128,106) with soft-GRAND gets 1 dB close to the respective NA. In Fig. 11, one can see that the LDPC (128,104) exhibits a gap of ≈ 2 dB to the respective NA.]

As seen, the Sieving-GRAND approach sharply reduces the number of membership tests needed by unconstrained GRAND, purging $\approx 97\%$ of the original $\sum_{i=0}^t \binom{n}{i}$ tests for $t=2,3$. This is achieved by considering a set of admissible error positions $\mathcal{P}^{(l)}$, of size $|\mathcal{P}^{(l)}| \ll n$, which includes the correct error positions with a monotonically increasing probability as the iterations unfold. Although in some cases, to find the ML $\hat{e}$, the threshold has to be lowered down to $w_{th}=2$, this happens with low probability, thus the average

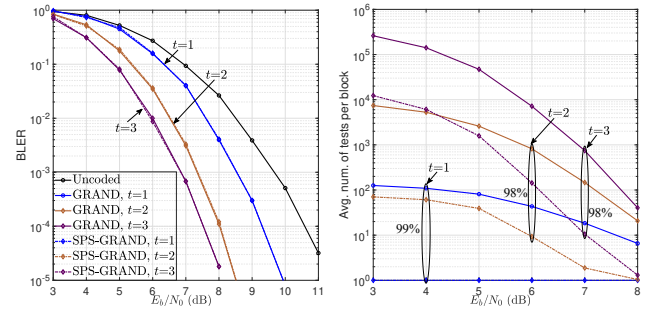


FIGURE 12. BLER performance (left) and decoding complexity (right) for $t=1,2,3$, using an LDPC (128,104) with combined set partitioning and Sieving-GRAND (SPS-GRAND) compared with unconstrained GRAND.

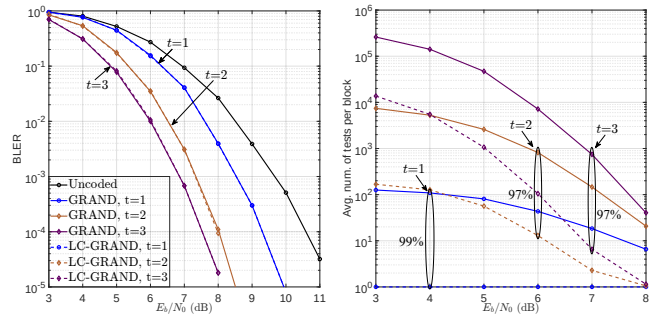


FIGURE 13. BLER performance (left) and decoding complexity (right) for $t=1,2,3$, using an LDPC (128,104) with largely constrained GRAND (LC-GRAND) compared with unconstrained GRAND.

number of positions considered in the guessing stage remains consistently $\ll n$.

The SP and Sieving techniques can be easily combined by applying SP to the candidate positions filtered at each iteration, leading to the here proposed set partitioning and Sieving-GRAND (SPS-GRAND). The resulting decoding of the LDPC code (128,104) is shown in Fig. 12. As expected, SPS-GRAND has the same BLER as unconstrained GRAND. The complexity comparison reveals that SPS-GRAND outstandingly outperforms GRAND, eliminating on average over 98% of the membership tests, even when looking for $t=3$ errors.

As shown in Fig. 13, the proposed LC-GRAND constrain for LDPC codes (128,104) significantly enhances the efficiency of error pattern selection by exploiting zero-valued positions in the syndrome. The results show that LC-GRAND also keeps the same decoding performance of GRAND and, even though the mechanism to attain a reduction of error patterns tested being quite different, the number of membership tests is virtually the same as the one attained by Sieving-GRAND. This constraint allows the decoder to eliminate a large portion of infeasible error patterns, thereby reducing the number of membership tests by 97% compared with the unconstrained GRAND approach, while maintaining near-maximum likelihood decoding accuracy.

For a fair comparison with Sieving-GRAND, the first error weight tested by LC-GRAND is also $t = \lceil \frac{\|s\|_1}{w_c} \rceil$, given that

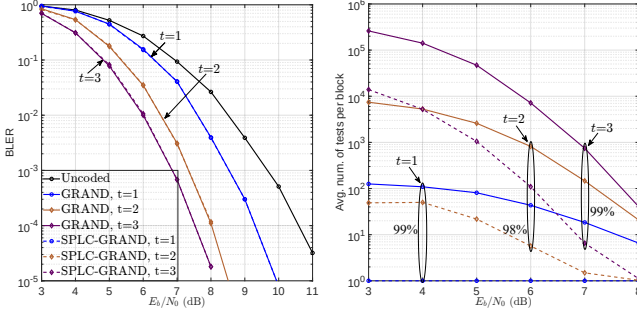


FIGURE 14. BLER performance (left) and decoding complexity (right) for $t = 1, 2, 3$, using an LDPC (128,104) with combined set partitioning and largely constrained GRAND (SPLC-GRAND) compared with unconstrained GRAND.

this is a mechanism that can be applied to any quasi-regular (or regular) LDPC code, as presented in Section IV.A.

Furthermore, when LC-GRAND is combined with set partitioning constraints (referred to as SPLC-GRAND), additional complexity gains are observed. As illustrated in Fig. 14, SPLC-GRAND achieves a dramatic reduction in decoding complexity $\approx 99\%$, leading to a much larger decoding speed without compromising reliability.

As described in Sections III, IV and VI, SP-GRAND, Sieving-GRAND and LC-GRAND have a very lightweight pre-processing per incoming word when compared with a competitor such as OSD, which would involve computing Gaussian elimination for each received word, to obtain the positions of the k most reliable bits. In SP-GRAND one has to obtain $\mathbf{h}_m^T$ from (1). In Sieving-GRAND and LC-GRAND, the pre-processing amounts to computing a vector-matrix product as in (6) or (9), and in the case of Sieving-GRAND, this is complemented by adding more positions to $\mathcal{P}^{(l)}$ at each iteration. In the main three proposed techniques (and also in the syndrome-weight technique applicable to quasi-regular LDPCs), there is no extra complexity when generating the error patterns, and no error pattern is ever generated to be discarded before any membership test is made. All these techniques use light pre-processing to reduce the number of positions where the errors may be and consequently they generate the error patterns by taking Cartesian products of t positions taken from a set of filtered candidate positions, $\hat{\mathcal{E}}_i = \{p_{j_1}, p_{j_2} \dots p_{j_t}\}$, while in unconstrained GRAND one would generate all Cartesian products of up to t sets, each set containing *all* the n positions (expunged from repetitions). Note that, as stated in Subsection III.A.2), each binary error pattern $\hat{\mathbf{e}}_i$ to be tested has 1s in the positions listed in $\hat{\mathcal{E}}_i$.

A. Noise-guessing vs belief-propagation decoding

The belief-propagation (BP) algorithm has been identified quite early as a possible decoder for LDPCs [10]. BP is an iterative message-passing algorithm operating on the Tanner graph of the code, where soft reliability information is exchanged between variable nodes and check nodes. Its decod-

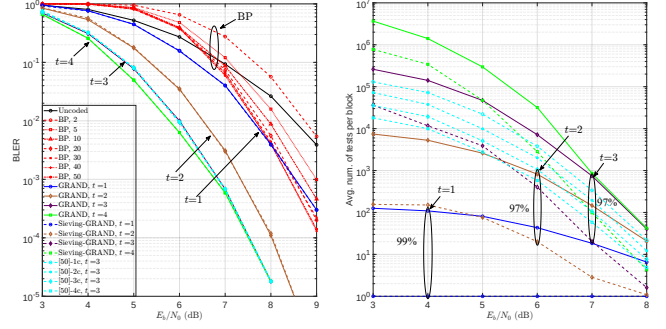


FIGURE 15. BLER performance (left) and decoding complexity (right) for $t = 1, 2, 3, 4$ using an LDPC (128,104) code, comparing BP decoding with different maximum iteration counts (2, 5, 10, 20, 30, 40, 50), with the proposed Sieving-GRAND (the results for LC-GRAND overlap those and are omitted) and with GRAND, for reference. The constrained decoder proposed in [50] is also included when decoding the same LDPC, and tested with a different number of row-constraints (1, 2, 3, 4).

ing performance improves with the number of iterations, at the expense of increased decoding latency and computational effort, and it can reach the ML decoding performance when the LDPC codes are cycle-free. However, short LDPC codes tend to have many short cycles and many small trapping sets [62, Sec.2.7]. For that reason, the performance of short LDPC codes decoded via BP may lie far from the normal approximation (NA), as mentioned in [13], [14]

The performance of BP decoding has been assessed for different numbers of iterations (between 2 and 50) and compared with the one of GRAND and Sieving-GRAND in Fig. 15. Note that all the proposed noise-guessing receivers hold the same performance, and LC-GRAND and Sieving-GRAND lead to undistinguishable complexities; for that reason LC-GRAND is omitted in Fig. 15. In the same figure, we also show the results attained by the decoding technique proposed in [50] for an increasing number of constraints (1, 2, 3, and 4). While the number of *full* membership tests is halved each time an additional constraint is included, one should note that is due to a “pre-filtering” of error patterns by the considered constraints that entails a lighter complexity than a *full* membership test. When considering four constraints that approach can be competitive with Sieving-GRAND and LC-GRAND in at very low E_b/N_0 when many columns of $\mathbf{H}$ are activated by many errors. When looking at the average decoding time in the next Subsection, one can observe that the “pre-filtering” lighter operations add up in a non-negligible manner, contributing to a larger latency.

As expected, the BP algorithm converges extremely fast to a codeword (see next Subsection) when decoding the LDPC (128,104) code. However, it often converges to an incorrect codeword and consequently Fig. 15 exhibits a performance gap of ≈ 1.5 dB to the performance of any of the noise-guessing decoders when they look for error patterns of up to $t = 3$ errors. This performance loss happens even though the BP decoder makes use of soft detection metrics. This is a well-known phenomena when applying BP decoding to

short LDPC codes [28]. For LDPC codes of similar length to the tested LDPC, the performance distance in respect to the NA bound has been found to be between 1 dB and 1.75 dB, while OSD can perform 0.5 dB off from the NA curve [13].

A larger number of iterations improves the error-rate performance, but beyond 10 iterations the performance gain is low. Noticeably, even with a rather large number of iterations, BP does not fully achieve the performance attained by GRAND, which attains the ML performance [27], and which is seen to be kept by all the proposed decoders.

For the reference LDPC, with $d_{\min} = 7$, testing error patterns with weight up to $t = 3$ suffices to reach the asymptotic ML hard-decoding performance. Nevertheless, Fig. 15 also includes the performance results for GRAND and Sieving-GRAND testing error patterns up to weight $t = 4$. As expected, the asymptotic performance $t = 3$ and $t = 4$ coincide. However, interestingly, at low SNR, correcting up to $t = 4$ errors slightly reduces the BLER. This is an expected effect: while a minimum distance $d_{\min} = 2t+1$ ensures that a ball of Hamming distance t can exist around any two codewords containing only a codeword at its center and with the two balls not intercepting, in some error instances it is still possible to correct beyond $t = \frac{d_{\min}-1}{2}$ [4, Ch.13].

B. Overall decoding time

The proposed decoders rely on some pre-processing based on the syndrome and the parity-check matrix, while the original concept of GRAND in [27] only requires the noise-guessing stage. It is therefore relevant to compare the average total decoding time per codeword taken by the proposed techniques, i.e., including the pre-processing stage, rather than only looking at the number of membership tests. We start by summarizing in Table 6 the complexities of both decoding stages of the most relevant proposed techniques by means of the big-O notation. Note that the types of feasible error patterns in LC-GRAND are entirely known in advance, and fixed for any LDPC code, according to the tables given in Section VI.

For a fair assessment, the total decoding time, including pre-processing and guessing stages, of the main proposed noise-guessing techniques have been measured, along with the decoding time of BP decoding (with a varying number of iterations). The decoding time of the decoder inspired by the idea of segmentation, as presented in the previous subsection, is also included. Such results are shown in Fig. 16. The tests were run using MATLAB in a 12-th gen Intel core i7 2.10 GHz computer (with no explicit parallelization in the code).

As mentioned in the previous Subsection, BP is known to converge very rapidly to a codeword and is therefore significantly faster than GRAND-based decoders at low E_b/N_0 ; however, it often does *not* converge to the ML codeword. Increasing the maximum number of iterations up to 20 in BP improves the error-rate performance, however, beyond that number no performance gain is noticeable, as

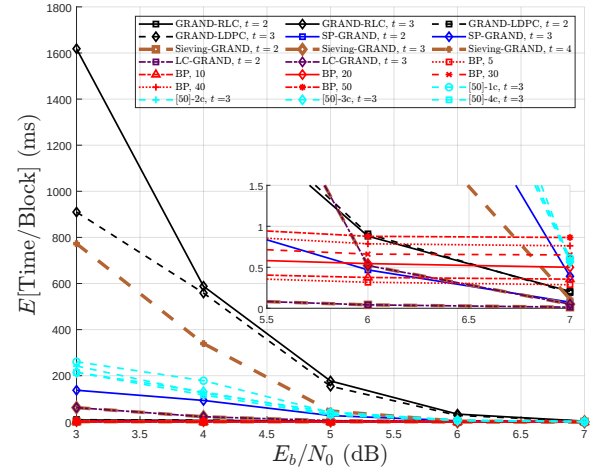


FIGURE 16. Average decoding time per block versus E_b/N_0 for LDPC(128,104) and RLC(128,104) codes for different decoders: the proposed SP-GRAND, Sieving-GRAND and LC-GRAND are compared with unconstrained GRAND and, in the case of the LDPC, also compared with BP decoding for different maximum iteration limits (2, 5, 10, 20, 30, 40, and 50). The noise-guessing decoders are plotted for $t = 2, 3$ (single errors amount to a column search in H), while Sieving-GRAND is evaluated up to $t = 4$ (beyond its correction capability, as in Fig. 15). Note that for $t = 2$ and $t = 3$ the LC-GRAND curves fully overlap the corresponding curves with Sieving-GRAND. The constrained decoder proposed in [50] is also included and tested with a different number of row-constraints (1, 2, 3, 4).

seen in Fig. 15. Also as expected, at low E_b/N_0 more iterations are required before convergence or termination. Even so, for $E_b/N_0 > 6$ dB, Sieving-GRAND and LC-GRAND (considering error patterns up to $t = 3$) can even decode faster than the BP algorithm when it runs more than 20 iterations, and beyond 7 dB they even get faster than BP running 5 iterations. This is due to the very low number of error patterns tested when the noise entropy is low (and also because the proposed decoders have a minimal pre-processing per codeword). In Fig. 16 one can also observe that the Sieving- and LC-GRAND decoders hold the same decoding times. This happens due to i) their very similar pre-processing and ii) due to an indistinguishable number of membership tests, as observed in the complexity curves in the previous Subsection. Interestingly, the average decoding time of the proposed Sieving-GRAND and LC-GRAND is lower than the decoding times achieved by [50] for all the tested number of constraints. This happens even though, as seen in Fig. 15, when applying 4 constraints, the number of queries attained by the technique proposed in [50] being lower when the E_b/N_0 is very low. However, the computation of the “pre-filtering” of error patterns via the constraints do not translate into a lower decoding time. This is a clear example that complexity comparisons should not only rely on the number of membership tests.

C. Constrained GRAND techniques with coded 16-QAM

The proposed set partitioning and Sieving-GRAND (SPS-GRAND) operates at a bit level on the received LDPC codewords; nevertheless, the M -arity of the modulation mod-

TABLE 6. Worst-case complexity comparison (big-O per received block) for the main decoding techniques correcting up to t errors.

Decoder	Preprocessing	Membership tests (worst-case)
Unconstrained GRAND	None	$\sum_{i=0}^t \binom{n}{i} = O(n^t)$
SP-GRAND	$O(n(n-k))$ (find min-weight row of $\mathbf{H}_{\text{red}}$)	$\sum_{i=1}^t \binom{ S_1^{(m)} }{i} \binom{ S_0^{(m)} }{t-i}$ with $ S_1^{(m)} + S_0^{(m)} = n$, $O(n^t)$ worst-case
Sieving-GRAND	$O(n(n-k))$ (compute $\mathbf{s}^T \mathbf{H}$, as in (6))	$\sum_l \binom{ \mathcal{P}^{(l)} }{t}$, $ \mathcal{P}^{(l)} \leq n$ avg. $ \mathcal{P}^{(l)} \ll n$
LC-GRAND	$O(n(n-k))$ (compute $w_c \mathbf{1}_n - \mathbf{s}^T \mathbf{H}$, as in (9))	Feasible t -fold products from $\mathbb{C}_i$ sets (see Tables 3–5), $O(n^t)$ bound

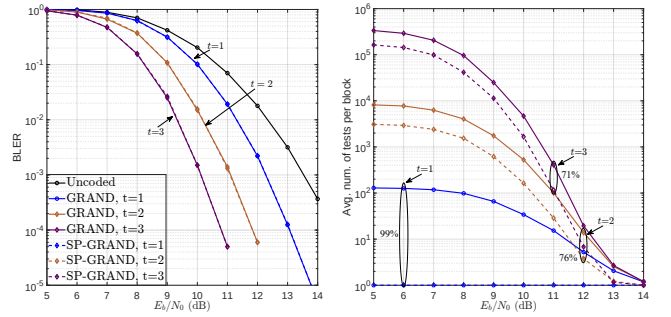
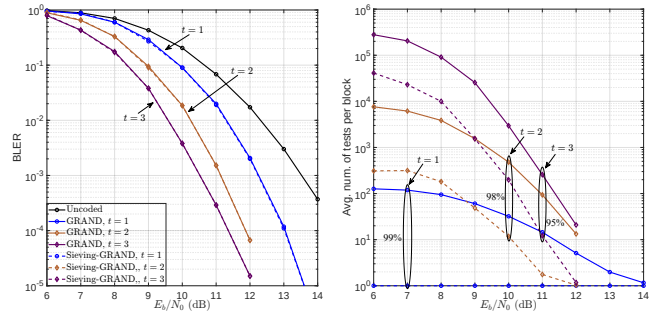
Note: $|S_1^{(m)}|$ and $|S_0^{(m)}|$: no. of ones/zeros in min-weight row of $\mathbf{H}_{\text{red}}$; $|\mathcal{P}^{(l)}|$: candidate position set size at iteration l ; w_c : constant column weight.

ifies the performance of a coded system. Here, we present the BLER and complexity results when using the same coding schemes, now with 16-QAM (quadrature amplitude modulation; $M = 16$) with Gray mapping. The results are shown in Fig. 17, Fig. 18, Fig.19, Fig.20, and Fig. 21 in the same order as in the main text. Fig. 17 illustrates the performance and decoding complexity of RLC (128,103) using set partitioning GRAND (SP-GRAND), in comparison to unconstrained GRAND. The BLER remains identical for SP-GRAND and unconstrained GRAND across the analyzed range of E_b/N_0 , with the curves overlapping. As with $M = 2$ or $M = 4$ (BPSK or QPSK), with 16-QAM SP-GRAND also significantly reduces the number of tests needed with unconstrained GRAND: for $t = 2, 3$ the savings are over 70%. (In the case with $t = 1$, as seen in Section 1, only a direct comparison of $\mathbf{s}$ with all columns $\mathbf{H}(:, j)$ is needed, and only one membership test is involved.)

Fig. 18 compares performances and decoding complexities of Sieving-GRAND with the original unconstrained GRAND, when decoding the LDPC code (128,104). The BLER curves for both Sieving-GRAND and GRAND perfectly overlap, indicating that in this setup Sieving-GRAND also achieves the ML decoding performance. As observed, Sieving-GRAND significantly reduces the number of membership tests required by unconstrained GRAND, eliminating over 95% of the original $\sum_{i=0}^t \binom{n}{i}$ membership tests.

The decoded output of mixing set partitioning GRAND and Sieving-GRAND for an LDPC code (128,104) is shown in Fig. 19. As anticipated, within the specified E_b/N_0 range, SPS-GRAND holds the same BLER as unconstrained GRAND. Regarding complexity savings, the results also show the superiority of SPS-GRAND, ditching over 98% of the membership tests needed on average.

Fig. 20 compares the performances and decoding complexity of the LC-GRAND and the original unconstrained GRAND for $t = 1, 2, 3$, using an LDPC code (128, 104). Without sacrificing performance, LC-GRAND significantly reduces the number of error patterns that are tested.


FIGURE 17. BLER performance (left) and decoding complexity (right) for $t = 1, 2, 3$, using RLC (128,103) and 16-QAM, with set partitioning GRAND (SP-GRAND) compared with unconstrained GRAND.

FIGURE 18. BLER performance (left) and decoding complexity (right) for $t = 1, 2, 3$, using an LDPC (128,104) and 16-QAM, with Sieving-GRAND compared with unconstrained GRAND.

The performance and complexity results for $t = 1, 2, 3$, using an LDPC code (128, 104) with combined set partitioning and LC-GRAND (i.e., SPLC-GRAND), are compared with unconstrained GRAND in Fig. 21. The BLER performance remains identical, while the decoding complexity is reduced by approximately 99% compared with unconstrained GRAND.

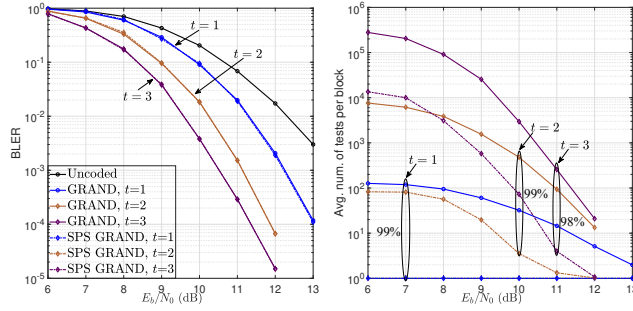


FIGURE 19. BLER performance (left) and decoding complexity (right) for $t=1,2,3$, using an LDPC (128,104) and 16-QAM, with combined set partitioning and Sieving GRAND (SPS-GRAND) compared with unconstrained GRAND.

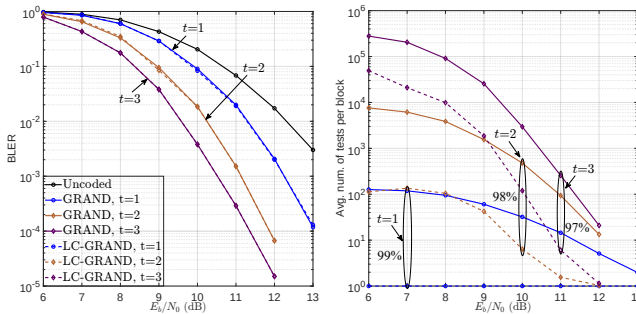


FIGURE 20. BLER performance (left) and decoding complexity (right) for $t=1,2,3$, using an LDPC (128,104) and 16-QAM with largely constrained GRAND compared with unconstrained GRAND.

VIII. CONCLUSION

This paper starts by proposing a simple constraint for the decoding of any LBC. The constraint splits the positions of the errors into sets, allowing one to construct the error patterns by means of the Cartesian product between sets. With this set partitioning the complexity reduction is more accentuated with sparser codes. By using short quasi-regular LDPC codes, one can retain the benefits of sparsity without incurring a severe performance loss. For such codes, a simple filtering technique is used to greatly reduce the candidate positions where the error positions may lie. The technique is based on the “fingerprints” that the active columns of the parity-check matrix leave on the observed syndrome. This technique starts by selecting a small pool of positions where the errors most likely lie, and then GRAND runs on those few positions only. This is done iteratively, allowing more positions to be added to that pool until the ML error pattern is found. The number of these iterations is bounded by the weight of the columns of the parity-check matrix.

The third decoding approach proposed, the largely constrained mechanism, applies a more straightforward constraint to any short LDPC code (not necessarily quasi-regular), and can be extended to any short linear block code – which is suggested as a natural extension of the present work. In this case, the decoder avoids testing infeasible error patterns. It only tests error patterns derived from valid Cartesian products that are computed *a priori*, ditching

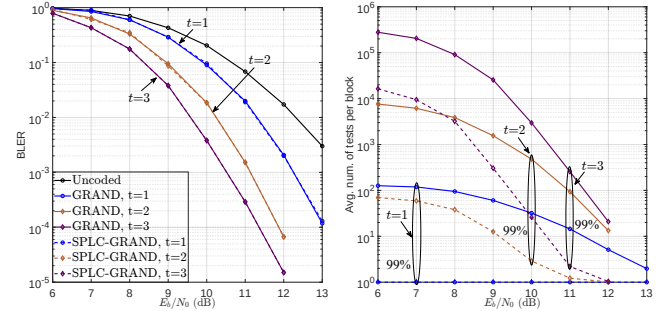


FIGURE 21. BLER performance (left) and decoding complexity (right) for $t=1,2,3$, using an LDPC (128,104) and 16-QAM with combined set partitioning and largely constrained GRAND compared with unconstrained GRAND.

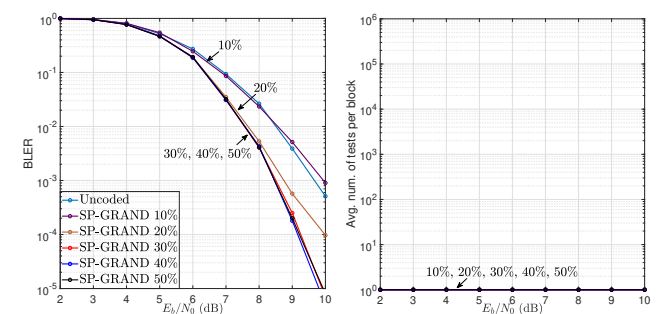


FIGURE 22. BLER performance (left) and decoding complexity (right) for the number of error $t=1$, using RLC (128,103) and BPSK, with p -sparse (indicated as a percentage) parity-check matrices, using set partitioning GRAND.

the need of an iterative process as the one in the Sieving approach. This enhanced decoding strategy reduces computational complexity also without compromising performance, particularly when combined with set partitioning constraints. It reduced the membership tests to $\approx 1\%$ of those required by GRAND. We conclude by highlighting the pressing need for ML-performance decoders for short LDPC codes in the context of quantum error correction, where LDPC-based quantum codes are emerging as a practical solution [67]–[69], and where noise-guessing decoders may play a relevant role [41], [43]. The presented techniques can be of interest to tackle the present challenge of decoding quantum LDPCs.

Appendix: Set partitioning GRAND with sparse RLCs

The number of candidate error patterns constructed via Cartesian products of two position sets gets smaller when the entropy of one of the sets is lower; fewer elements in a set provide more information about the errors’ locations. Sparse RLCs necessarily verify the objective of having a low minimum support row $\mathbf{h}_m^T$. We assessed the performance and the complexity of SP-GRAND for RLCs (128, 103) with different sparsity values in their parity-check matrix, as shown in Fig. 22, Fig. 23, and Fig. 24 for bounded-decoders with $t=1, 2, 3$, respectively. All figures include the performances of uncoded transmissions as well as the BLER and complexity results with SP-GRAND. With a sparser

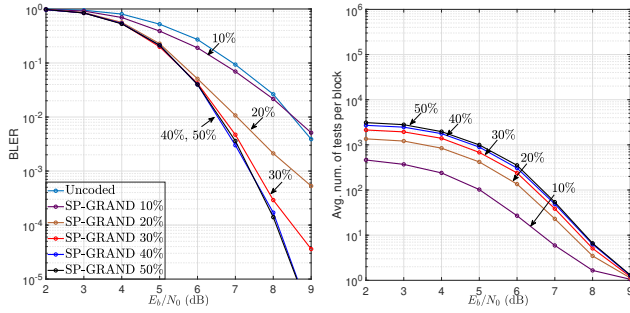


FIGURE 23. BLER performance (left) and decoding complexity (right) for the number of errors $t=2$, using RLC (128,103) and BPSK, with p -sparse (indicated as a percentage) parity-check matrices, using set partitioning GRAND.

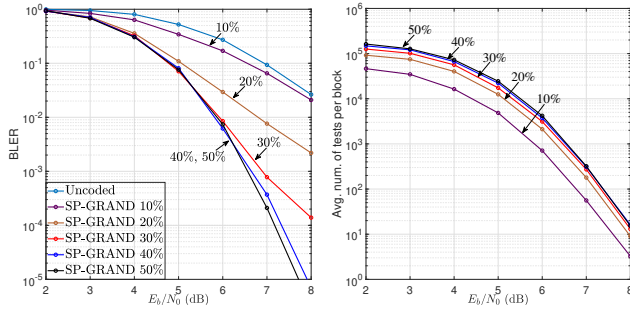


FIGURE 24. BLER performance (left) and decoding complexity (right) for the number of errors $t=3$, using RLC (128,103) and BPSK, with p -sparse (indicated as a percentage) parity-check matrices, using set partitioning GRAND.

parity-check matrix, the weight of the minimum support row is lower and faster decoding is possible, however, at the expense of a severe deterioration of the codes' performance, as one would expect (as they lack the proper construction of LDPC codes). It should be noted that the very sparse RLC codes can lead to a BLER even higher than the one of the uncoded transmission. This is because the added redundant bits contribute to more errors and the energy spent in their transmission is assigned to E_b , while this extra redundancy does not contribute to a codeword correction that compensates for that extra spent energy.

Acknowledgment

Francisco Monteiro thanks Dr. Bruno Coutinho (German Aerospace Center – DLR), and Prof. Marco Gomes and Prof. Vitor Silva (University of Coimbra) for discussions on set partitioning and sieving techniques. Carlos Salema thanks Prof. Paulo Mateus (IST, University of Lisbon) for discussions on low-complexity decoding techniques of block codes.

REFERENCES

- [1] C. E. Shannon, "A mathematical theory of communication," *The Bell Syst. Tech. J.*, vol. 27, no. 3, pp. 379–423, Jul. 1948. doi: 10.1002/j.1538-7305.1948.tb01338.x
- [2] —, "Probability of error for optimal codes in a gaussian channel," *The Bell Syst. Tech. J.*, vol. 38, no. 3, pp. 611–656, May. 1959. doi: 10.1002/j.1538-7305.1959.tb03905.x

- [3] L. V. Truong, G. Cocco, J. Font-Segura, and A. Guillén i Fàbregas, "Concentration properties of random codes," *IEEE Trans. Inf. Theory*, vol. 69, no. 12, pp. 7499–7537, Dec. 2023. doi: 10.1109/TIT.2023.3312326
- [4] D. MacKay, *Inf. Theory, Inference and Learning Algorithms*. Cambridge, UK: Cambridge Univ. Press, 2003.
- [5] R. Gallager, "The random coding bound is tight for the average code (corresp.)," *IEEE Trans. Inf. Theory*, vol. 19, no. 2, p. 244–246, Mar. 1973. doi: 10.1109/TIT.1973.1054971
- [6] J. Coffey and R. Goodman, "Any code of which we cannot think is good," *IEEE Trans. Inf. Theory*, vol. 36, no. 6, pp. 1453–1461, Nov. 1990. doi: 10.1109/18.59944
- [7] D. J. Costello and G. D. Forney, "Channel coding: The road to channel capacity," *Proc. of the IEEE*, vol. 95, no. 6, pp. 1150–1177, Jun. 2007. doi: 10.1109/JPROC.2007.895188
- [8] R. Gallager, "Low-density parity-check codes," *IEEE Trans. Inf. Theory*, vol. 8, no. 1, pp. 21–28, Jan. 1962. doi: 10.1109/TIT.1962.1057683
- [9] D. MacKay and R. Neal, "Good codes based on very sparse matrices," in *Cryptography and Coding 5th IMA Conf.*, ser. Lecture Notes in Computer Science, C. Boyd, Ed., no. 1025, 1995, pp. 100–111. doi: 10.1007/3-540-60693-9_13
- [10] D. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Trans. Inf. Theory*, vol. 45, no. 2, pp. 399–431, Mar. 1999. doi: 10.1109/18.748992
- [11] 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, Jun. 2022. doi: 10.1109/TCOMM.2022.3171798
- [12] X. Wu, M. Jiang, C. Zhao, L. Ma, and Y. Wei, "Low-rate PBRL-LDPC codes for URLLC in 5G," *IEEE Wireless Commun. Lett.*, vol. 7, no. 5, pp. 800–803, Oct. 2018. doi: 10.1109/LWC.2018.2825988
- [13] C. Yue, V. Miloslavskaya, M. Shirvanimoghaddam, B. Vucetic, and Y. Li, "Efficient decoders for short block length codes in 6G URLLC," *IEEE Commun. Mag.*, vol. 61, no. 4, pp. 84–90, Apr. 2023. doi: 10.1109/MCOM.001.2200275
- [14] M. Shirvanimoghaddam, M. S. Mohammadi, R. Abbas, A. Minja, C. Yue, B. Matuz, G. Han, Z. Lin, W. Liu, Y. Li, S. Johnson, and B. Vucetic, "Short block-length codes for ultra-reliable low latency communications," *IEEE Commun. Mag.*, vol. 57, no. 2, pp. 130–137, Feb. 2019. doi: 10.1109/MCOM.2018.1800181
- [15] K. Arora, J. Singh, and Y. S. Randhawa, "A survey on channel coding techniques for 5G wireless networks," *Telecommun. Syst.*, vol. 73, no. 4, p. 637–663, Apr. 2020. doi: 10.1007/s11235-019-00630-3
- [16] Y. Polyanskiy, H. V. Poor, and S. Verdú, "Channel coding rate in the finite blocklength regime," *IEEE Trans. Inf. Theory*, vol. 56, no. 5, pp. 2307–2359, May. 2010. doi: 10.1109/TIT.2010.2043769
- [17] C. Salema, *Microwave Radio Links: From Theory to Design*. London, UK: Wiley, 2008.
- [18] S. Dolinar, D. Divsalar, and F. Pollara, "Code performance as a function of block size," TMO Progress Report, JPL-NASA, techreport 42-133, May. 1998.
- [19] M. Fossorier and S. Lin, "Soft-decision decoding of linear block codes based on ordered statistics," *IEEE Trans. Inf. Theory*, vol. 41, no. 5, pp. 1379–1396, Sep. 1995. doi: 10.1109/18.412683
- [20] S. Alnawayseh and P. Loskot, "Ordered statistics-based list decoding techniques for linear binary block codes," *EURASIP J. Wireless Commun. Netw.*, vol. 2012, no. 314, pp. 1–12, Oct. 2012. doi: 10.1186/1687-1499-2012-314
- [21] C. Kim, D. Kam, S. Kim, G. Park, and Y. Lee, "Simplified ordered statistic decoding for short-length linear block codes," *IEEE Commun. Lett.*, vol. 26, no. 8, pp. 1720–1724, Aug. 2022. doi: 10.1109/LCOMM.2022.3176646
- [22] J. Liang, Y. Wang, S. Cai, and X. Ma, "A low-complexity ordered statistic decoding of short block codes," *IEEE Commun. Lett.*, vol. 27, no. 2, pp. 400–403, Feb. 2023. doi: 10.1109/LCOMM.2022.3222819
- [23] L. Yang, J. Zhao, X. Li, L. Chen, H. Zhang, and J. Tong, "Efficient ordered statistics decoding of BCH codes without Gaussian elimination," *IEEE Trans. Inf. Theory*, vol. 71, no. 11, pp. 8294–8311, Nov. 2025. doi: 10.1109/TIT.2025.3613748
- [24] B. Dorsch, "A decoding algorithm for binary block codes and J-ary output channels (corresp.)," *IEEE Trans. Inf. Theory*, vol. 20, no. 3, pp. 391–394, May 1974. doi: 10.1109/TIT.1974.1055217

- [25] Y. Wang, J. Liang, and X. Ma, "Local constraint-based ordered statistics decoding for short block codes," in *2022 IEEE Inf. Theory Workshop (ITW)*, Mumbai, India, Nov. 2022, pp. 107–112. doi: 10.1109/ITW54588.2022.9965916
- [26] M. Li, V. Derudder, K. Bertrand, C. Desset, and A. Bourdoux, "High-speed LDPC decoders towards 1 Tb/s," *IEEE Trans. Circuits Syst. I: Regul. Pap.*, vol. 68, no. 5, pp. 2224–2233, May. 2021. doi: 10.1109/TCSI.2021.3060880
- [27] 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, Jul. 2019. doi: 10.1109/TIT.2019.2896110
- [28] G. Li and X. Yu, "Boosting ordered statistics decoding of short LDPC codes with simple neural network models," *IEEE Commun. Lett.*, vol. 28, no. 12, pp. 2714–2718, Dec. 2024. doi: 10.1109/LCOMM.2024.3475874
- [29] S. M. Abbas, M. Jalaeddine, and W. J. Gross, *Guessing Random Additive Noise Decoding: A Hardware Perspective*. Switzerland: Springer, 2023.
- [30] K. R. Duffy, J. Li, and M. Médard, "Guessing noise, not code-words," in *Proc. of IEEE Int. Symp on Inf. Theory (ISIT)*, Vail, CO, USA, Jun. 2018, pp. 671–675. doi: 10.1109/ISIT.2018.8437648
- [31] K. R. Duffy, W. An, and M. Médard, "Ordered reliability bits guessing random additive noise decoding," *IEEE Trans. Signal Process.*, vol. 70, pp. 4528–4542, Aug. 2022. doi: 10.1109/TSP.2022.3203251
- [32] S. M. Abbas, M. Jalaeddine, and W. J. Gross, "List-GRAND: A practical way to achieve maximum likelihood decoding," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 31, no. 1, pp. 43–54, Jan. 2023. doi: 10.1109/TVLSI.2022.3223692
- [33] H. Sameddeen, P. Yuan, M. Médard, and K. R. Duffy, "Soft-input, soft-output joint data detection and GRAND: A performance and complexity analysis," in *Proc. of IEEE Int. Symp on Inf. Theory (ISIT)*, Taipei, Taiwan, Jun. 2023, pp. 1090–1095. doi: 10.1109/ISIT54713.2023.10206823
- [34] S. M. Abbas, M. Jalaeddine, and W. J. Gross, "GRAND for Rayleigh fading channels," in *Proc. of IEEE Globecom Workshops (GC Wkshps)*, Rio de Janeiro, Brazil, Dec. 2022, pp. 504–509. doi: 10.1109/GCWkshps56602.2022.10008698
- [35] S. Allahkaram, F. A. Monteiro, and I. Chatzigeorgiou, "URLLC with coded massive MIMO via random linear codes and GRAND," in *Proc. of IEEE 96th Vehic. Tech. Conf. (VTC2022-Fall)*, London, UK, Sep. 2022, pp. 1–5. doi: 10.1109/VTC2022-Fall57202.2022.10012803
- [36] —, "Constrained symbol-level noise-guessing decoding with antenna sorting for massive MIMO," *IEEE Open J. Commun. Soc.*, vol. 6, pp. 9486–9503, Nov. 2025. doi: 10.1109/OJCOMS.2025.3630551
- [37] A. Riaz, V. Bansal, A. Solomon, W. An, Q. Liu, K. Galligan, K. R. Duffy, M. Médard, and R. T. Yazicigil, "Multi-code multi-rate universal maximum likelihood decoder using GRAND," in *In Proc. of Eur. Solid State Circuits Conf. (ESSCIRC)*, Grenoble, France, Sep. 2021, pp. 239–246. doi: 10.1109/ESSCIRC53450.2021.9567867
- [38] C. Condo, "A fixed latency ORBGRAND decoder architecture with LUT-aided error-pattern scheduling," *IEEE Trans. Circuits Syst. I: Regul. Pap.*, vol. 69, no. 5, pp. 2203–2211, May. 2022. doi: 10.1109/TCSI.2022.3150583
- [39] A. Riaz, A. Yasar, F. Ercan, W. An, J. Ngo, K. Galligan, M. Médard, K. R. Duffy, and R. T. Yazicigil, "A sub-0.8pj/b 16.3gbps/mm2 universal soft-detection decoder using ORBGRAND in 40nm CMOS," in *2023 IEEE Int. Solid-State Circuits Conf. (ISSCC)*, San Francisco, CA, USA, Feb. 2023, pp. 432–434. doi: 10.1109/ISSCC42615.2023.10067519
- [40] I. Chatzigeorgiou, "Transversal GRAND for network coded data," in *Proc. of IEEE Int. Symp on Inf. Theory (ISIT)*, Espoo, Finland, Jun. 2022, pp. 1773–1778. doi: 10.1109/ISIT50566.2022.9834692
- [41] D. Cruz, F. A. Monteiro, and B. C. Coutinho, "Quantum error correction via noise guessing decoding," *IEEE Access*, vol. 11, pp. 119446–119461, Oct. 2023. doi: 10.1109/ACCESS.2023.3327214
- [42] D. Chandra, Z. B. Kaykac Egilmez, Y. Xiong, S. X. Ng, R. G. Maunder, and L. Hanzo, "Universal decoding of quantum stabilizer codes via classical guesswork," *IEEE Access*, vol. 11, pp. 19059–19072, Feb. 2023. doi: 10.1109/ACCESS.2023.3247966
- [43] D. Cruz, F. A. Monteiro, A. Roque, and B. C. Coutinho, "Fault-tolerant noise guessing decoding of quantum random codes," *IEEE Trans. Quantum Eng.*, vol. 6, pp. 1–26, Aug. 2025. doi: 10.1109/TQE.2025.3595778
- [44] A. Roque, D. Cruz, F. A. Monteiro, and B. C. Coutinho, "Efficient entanglement purification based on noise guessing decoding," *Quantum*, vol. 8, p. 1476, Sep. 2024. doi: 10.22331/q-2024-09-19-1476
- [45] W. An, M. Médard, and K. R. Duffy, "CRC codes as error correction codes," in *IEEE Int. Conf. on Comm. (ICC)*, Montreal, QC, Canada, Jul. 2021, pp. 1–6. doi: 10.1109/ICC42927.2021.9500279
- [46] A. Cohen, R. G. L. 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, Aug. 2023. doi: 10.1109/LCOMM.2023.3285404
- [47] M. Médard and K. R. Duffy, "Physical layer insecurity," in *Proc. 57th Annu. Conf. Inf. Sci. Syst. (CISS)*, Baltimore, MD, USA, Mar. 2023. doi: 10.1109/CISS56502.2023.10089749
- [48] I. Chatzigeorgiou and F. A. Monteiro, "Symbol-level GRAND for high-order modulation over block fading channels," *IEEE Commun. Lett.*, vol. 27, no. 2, pp. 447–451, Feb. 2023. doi: 10.1109/LCOMM.2022.3227593
- [49] J. Griffin, P. Yuan, K. R. Duffy, and M. Médard, "Using a single parity-check to reduce the guesswork of guessing codeword decoding," in *Proc. of 59th Annual Conf. on Information Sciences and Systems (CISS)*, Baltimore, MD, USA, Mar. 2025, pp. 1–6. doi: 10.1109/CISS64860.2025.10944685
- [50] M. Rowshan and J. Yuan, "Constrained error pattern generation for GRAND," in *Proc. of IEEE Int. Symp on Inf. Theory (ISIT)*, Espoo, Finland, Jun. 2022, pp. 1767–1772. doi: 10.1109/ISIT50566.2022.9834343
- [51] Y. Wang, J. Liang, and X. Ma, "Partially constrained GRAND of linear block codes," in *Proc. of 9th International Conference on Computer and Communication Systems (ICCCS)*, Apr. 2024, pp. 617–622. doi: 10.1109/ICCCS61882.2024.10602848
- [52] R. Hadavian, D. Truhachev, K. El-Sankary, H. Ebrahimzad, and H. Najafi, "Ordered reliability direct error pattern testing ORDEPT algorithm," in *Proc. of IEEE Global Commun. Conf.*, Dec. 2023, pp. 6983–6988. doi: 10.1109/GLOBECOM54140.2023.10437894
- [53] X. Ma, "Guessing what, noise or codeword?" in *Proc. of IEEE Inf. Theory Workshop (ITW)*, Shenzhen, China, Nov. 2024, pp. 460–465. doi: 10.1109/ITW61385.2024.10806987
- [54] M. Rowshan and J. Yuan, "Low-complexity GRAND by segmentation," in *Proc. of 2023 IEEE Global Communications Conference (GLOBECOM)*, Kuala Lumpur, Malaysia, Dec. 2023, pp. 6145–6151. doi: 10.1109/GLOBECOM54140.2023.10436895
- [55] —, "Segmented GRAND: Complexity reduction through sub-pattern combination," *IEEE Trans. Commun.*, vol. 73, no. 8, pp. 5607–5620, Aug. 2025. doi: 10.1109/TCOMM.2025.3541094
- [56] P. Yuan, M. Médard, 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, Jan. 2025. doi: 10.1109/TWC.2025.3530880
- [57] Q. Wang, J. Liang, P. Yuan, K. R. Duffy, M. Médard, and X. Ma, "Guessing decoding of short blocklength codes," *arXiv:2511.12108 [cs.IT]*, Nov. 2025. [Online]. Available: <https://arxiv.org/abs/2511.12108>
- [58] M. Geiselhart, F. Krieg, J. Clausius, D. Tandler, and S. ten Brink, "6G: A welcome chance to unify channel coding?" *IEEE BITS the Information Theory Magazine*, vol. 3, no. 1, pp. 67–80, Mar. 2023. doi: 10.1109/MBITS.2023.3322974
- [59] I. Tal and A. Vardy, "List decoding of polar codes," *IEEE Trans. Inf. Theory*, vol. 61, no. 5, pp. 2213–2226, May. 2015. doi: 10.1109/TIT.2015.2410251
- [60] F. Monteiro, "Lattices in MIMO spatial multiplexing: Detection and geometry," Ph.D. dissertation, Univ. of Cambridge, UK, 2012.
- [61] Z. Sun, C. Gu, and Y. Zheng, "A review of sieve algorithms in solving the shortest lattice vector problem," *IEEE Access*, vol. 8, pp. 190475–190486, Oct. 2020. doi: 10.1109/ACCESS.2020.3031276
- [62] T. Richardson and R. Urbanke, *Modern Coding Theory*. Cambridge, UK: Cambridge Univ. Press, 2012.
- [63] K. D. Rao, *Channel Coding Techniques for Wireless Communications*. London, UK: Springer, 2016.
- [64] T. Erseghe, "Coding in the finite-blocklength regime: Bounds based on laplace integrals and their asymptotic approximations," *IEEE Transactions on Information Theory*, vol. 62, no. 12, pp. 6854–6883, Dec. 2016. doi: 10.1109/TIT.2016.2616900
- [65] M. Rowshan, M. Qiu, Y. Xie, X. Gu, and J. Yuan, "Channel coding toward 6G: Technical overview and outlook," *IEEE Open Journal of*

- the *Communications Society*, vol. 5, pp. 2585–2685, Apr. 2024. doi: 10.1109/OJCOMS.2024.3390000
- [66] M. Rowshan and E. Viterbo, *Polar Codes: From Theory to Practice*. New Jersey, USA: Wiley, 2026.
- [67] Google Quantum AI and Collaborators, “Quantum error correction below the surface code threshold,” *Nature*, vol. 638, no. 8052, pp. 920–926, Feb. 2025. doi: 10.1038/s41586-024-08449-y
- [68] S. Bravyi, A. W. Cross, J. M. Gambetta, D. Maslov, P. Rall, and T. J. Yoder, “High-threshold and low-overhead fault-tolerant quantum memory,” *Nature*, vol. 627, no. 8005, pp. 778–782, Mar. 2024. doi: 10.1038/s41586-024-07107-7
- [69] O. Ferraz, B. Coutinho, G. Falcão, M. Gomes, F. A. Monteiro, and V. Silva, “GPU-accelerated syndrome decoding for quantum LDPC codes below the 63 μ s latency threshold,” in *Proc. of the 59th Asilomar Conf. on Signals, Systems, and Computers*, Pacific Grove, CA, USA, Oct. 2025, pp. 1878–1882. doi: 10.1109/IEEECONF67917.2025.11443687



Sahar Allahkaram is a PhD student in the Dep. of Information Science and Technology at Iscte - University Institute of Lisbon, Portugal. She is working on Signal Processing and Coding Techniques for 6G Ultra-reliable Low-latency Wireless Machine-type Communications. She obtained her MSc in Aerospace Engineering from Sapienza University of Rome, Italy, in 2020 and her BSc in Electronic Engineering from Azad University of Tehran, Iran, in 2015.



Francisco A. Monteiro (M’07) is an Associate Professor in the Dep. of Information Science and Technology at Iscte - University Institute of Lisbon, Portugal, and a researcher at Instituto de Telecomunicações, Lisbon, Portugal. He holds a PhD from the University of Cambridge, U.K., and the Licenciatura and MSc degrees in Electrical and Computer Engineering from Instituto Superior Técnico (IST), University of Lisbon, Portugal, where he also became a Teaching Assistant. He held visiting research positions at the Universities of Toronto (Canada), Lancaster (U.K.), Oulu (Finland), and Pompeu Fabra (Barcelona, Spain). He has won two best paper prizes awards at IEEE conferences (2004 and 2007), a Young Engineer Prize (3rd place) from the Portuguese Engineers Institution (Ordem dos Engenheiros) in 2002, and for two years in a row was a recipient of Exemplary Reviewer Awards from the IEEE Wireless Communications Letters (in 2014 and 2015). He co-edited the book “MIMO Processing for 4G and Beyond: Fundamentals and Evolution”, published by CRC Press in 2014. In 2016 he was the Lead Guest Editor of a special issue on Network Coding of the EURASIP Journal on Advances in Signal Processing. He was a general chair of ISWCS 2018 - The 15th International Symposium on Wireless Communication Systems, an IEEE major conference in wireless communications.



João Ribeiro is currently an Assistant Professor with the Department of Mathematics, Instituto Superior Técnico (IST), University of Lisbon, Portugal, and a Researcher at the Instituto de Telecomunicações, Lisbon, Portugal. He received the B.Sc. degree in applied mathematics and computation from IST, Universidade de Lisboa in 2015, the M.Sc. degree in computer science from the Swiss Federal Institute of Technology (ETH), Zürich, Switzerland, in 2017, and the Ph.D. degree in computing from Imperial College London, U.K., in 2021. Previously, he was an Assistant Professor with the Department of Computer Science, NOVA School of Science and Technology, Caparica, Portugal; a Researcher at NOVA LINCS, Caparica; and a Post-Doctoral Fellow with the Computer Science Department, Carnegie Mellon University, Pittsburgh, PA, USA. His research interests include coding and information

theory, cryptography, and pseudorandomness. In 2025 he was the recipient of a Starting Grant from the European Research Council (ERC) for research on coding against synchronization errors.



Carlos Salema (Life Member, IEEE) He is a Professor Emeritus at Instituto Superior Técnico (IST), University of Lisbon, Portugal. He graduated in electrical engineering from the IST, University of Lisbon, in 1965. He received the Ph.D. degree in electrical engineering from Queen Mary College, University of London, U.K., in 1972, and the Aggregate degree from IST in 1978, where he became a Full Professor in 1979. He acted as Chairperson of the National Foundation for Scientific Computing from 1986 to 1989, and from 1989 to 1992, he was the President of the Portuguese National Science and Technology Research Council (JNICT). From 1993 until 2024 he was the Chairperson of the Board of Directors of the Instituto de Telecomunicações, Portugal. He was twice distinguished by the students as the best professor in electrical engineering at IST. He is the author or a coauthor of several conference and journal articles, and of six textbooks. Carlos Salema is a Counselor Member of the Portuguese Institute of Engineers (“Ordem dos Engenheiros”), a fellow (and former President) of the Lisbon Academy of Sciences, and a fellow (and former President) of the Portuguese Academy of Engineering. He has served as a member of the Selection Committee of the Institute of Engineering and Technology (IET) A. F. Harvey Research Prize from 2015 to 2020. In 1974, he was awarded the Marconi Prize by the Institute of Electrical and Electronic Engineers (IEEE).

Chapter 5

Uniquely Decodable Signature Sequences Over the Noisy Adder Channel

Journal: IEEE Access

Status: Under review

Pages: 1-13

Conference: 2025 International Symposium on Networks, Computers and Communications (ISNCC)

Date: 21 November 2025

Location: Paris, France

Pages: 1-4

DOI:10.1109/ISNCC66965.2025.11250456 URL:

<https://ieeexplore.ieee.org/document/11250456>

Note: A preliminary version of this work was published in the above conference. The journal article is an extended version.

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2024.0429000

Uniquely Decodable Bose-Chowla-Lindström Signature Sequences Over the Multiple Access Adder Channel

SAHAR ALLAHKARAM.^{1,2}, (Student Member, IEEE), FRANCISCO A. MONTEIRO.^{1,2} (Member, IEEE)

¹Instituto de Telecomunicações, Portugal

²ISCTE – Instituto Universitário de Lisboa, Portugal

Corresponding author: Francisco A. Monteiro (e-mail: francisco.monteiro@lx.it.pt).

This work is funded by national funds through FCT – Fundação para a Ciência e a Tecnologia, I.P., and, when eligible, co-funded by EU funds under project/support UID/50008/2025 – Instituto de Telecomunicações, with DOI identifier <https://doi.org/10.54499/UID/50008/2025>. Sahar Allahkaram is funded by a merit scholarship from ISCTE – Instituto Universitário de Lisboa.

ABSTRACT Signature sequences that are uniquely decodable allow one to identify which terminals are active by observing their sum. They solve the problem of uniquely identifying from a total of t users which $m \leq t$ users are active: for example, they want to fire an alarm, or want to transmit information, as happens in a grant-free access scheme that works with physical layer network coding (PLNC) principles. Although such sequences were known to exist, and some previous works on random access assumed their existence, they have never been explicitly constructed, nor was their performance studied when considering a noisy-sum observation. Such sequences are defined by sets of integers that verify the unique-sum property. The construction of the sets is algebraically defined depending on two Galois fields with related dimensions. Using the Bose-Chowla-Lindström (BCL) construction, the paper finds the sets of integer sequences that verify the unique-sum property and that can be generated with reasonable computational complexity. From these integer sets one obtains the corresponding UD binary sequences, which are seen to require fewer bits than Hadamard orthogonal sequences for similar size setups. Subsequently, the binary signature sequences are mapped to some binary antipodal modulation and assessed in the multiple access adder channel (MAAC) disturbed by noise. This work assesses the detection performance of BCL sequences under additive white Gaussian noise (AWGN). The error performance is shown to be perfectly described by a general model that considers uniformly distributed random binary sequences, where the received sums can be interpreted as non-uniform PAM signals. We can then obtain the closed-form expressions for the probability of having an incorrect sum detection as a function of the signal-to-noise (SNR) ratio, which is then numerically verified. Both hard and soft maximum likelihood detectors are applied to the equivalent-PAM sum signals, allowing one to assess the probability of not correctly identifying the subset of active users in the random access system.

INDEX TERMS Uniquely decodable sequences, non-orthogonal signature sequences, multiple access adder channel, noisy sums, grant-free, random access, IoT.

I. INTRODUCTION

IN sensor networks and machine-type communications that will be at the core of Internet of Things (IoT) services, it is common to have a large number of terminals deployed, but their activity pattern is such that they only sporadically transmit information [1], [2]. The use of random access protocols is common in standards such as LTE/5G [3]. Random access has been analyzed in the context of massive MIMO

[4]–[8] and even cell-free massive MIMO (CF-mMIMO) [9], which aims to provide uniform service and improved spectral efficiency for dense, distributed deployments. This work derives terminal identification sequences holding the unique-sum property, and then focuses on their application to the multiple access adder channel (MAAC) with additive white Gaussian noise (AWGN).

Polyanskiy formalized the multiple-access problem for

a massive number of uncoordinated users (also called unsourced users), as encountered in modern wireless scenarios such as massive machine-type communication (mMTC) and low-power wide-area networks (LP-WANs) [10]. That work proposes a formal definition of random-access codes for the Gaussian multiple-access channel (MAC), enabling reliable decoding of a subset of active users' messages in the presence of noise. It presents an achievability bound for such codes and benchmarks their theoretical performance against practical schemes, including ALOHA, coded slotted ALOHA, CDMA, and interference-as-noise strategies. A framework for understanding such random access scenarios has been proposed in [11], under the name unsourced multiple access (UMAC), restating the problem of massive uncoordinated communication as a coding-theoretic problem, in which all users transmit using a common codebook and the base station (BS) attempts to decode a subset of transmitted messages without knowing which users are active.

Random access schemes are traditionally grant-based, meaning that one terminal can only start transmitting its message after some access grant is conveyed by the BS. The 1ms latency goal that is sought for such systems becomes hard to accomplish [12]. In a grant-free protocol, a terminal starts transmitting its message without having to wait for this access grant, which contributes to achieving the ultra-reliable low-latency communication goals (URLLC) in terrestrial networks [12], [13], as well as in satellite-enabled IoT [14]. These grant-free protocols use different flavors of hybrid automatic repeat request (HARQ) retransmissions after collisions occur, leading to discarding information that could be otherwise retrieved from the resulting collision (i.e., from the sum of the colliding packets). In order to take advantage of this partial information, the use of physical layer network coding (PLNC) [15] for the grant-free random access channel has been proposed by Goseling et al. [16], [17]. In their work, the identification of *how many* and *which* terminals are involved in a collision is crucial for the implementation of PLNC in the message-bearing part of the packets, and this knowledge was assumed to be available at the BS when decoding the packets. When that information is not available, successive interference cancellation (SIC) techniques have been used when uniquely decodable sequences (UDSs) are considered [2], [18]. In [19], [20] the authors proposed non-orthogonal multiple access (NOMA) uplink random access system, taking advantage of the power-domain [21] to have an extra dimension such that two users that collide in time can still be decoders applying SIC.

In the random access channel, within the synchronization time slots defined by the BS, some terminals may access the channel, transmitting a packet containing the terminal's identification, possibly also followed by a message. Orthogonal sequences, such as Hadamard sequences or Zadoff-Chu sequences, and also pseudo-orthogonal ones such as Gold sequences or M-sequences, can be used to solve the terminal identification problem when several terminals become active simultaneously [22]–[24]. However, the problem can

be more efficiently solved by using non-orthogonal sequences that, given the sum of m sequences, allow to uniquely identify which m out of t terminals transmitted the colliding sequences. In short, from a sum one can uniquely determine the terms involved in it. This is known as the *signature coding problem*, as discussed in [25]–[27]. There are two types of UDSs: those obtained by the Chang-Weldon construction [26] (related to construction by Cantor and Mills [28]), and those obtained by the Lindström construction [25], [26], [29]. The former is not optimal, but the latter are asymptotically optimal [26] with respect to the length of the UDSs it generates. In fact, Lindström sequences are entirely dependent on the algebraic construction based on Galois fields, which were first proposed by Bose and Chowla in [30], hence we will refer to these sequences as Bose-Chowla-Lindström (BCL) sequences. To our knowledge, [2] has so far been the only engineering work exploring the existence of sequences with such properties, precisely in the context of a random channel applying PLNC principles. The authors assumed that the problem of identifying m -out-of- t terminals had a solution (using “Lindström sequences”) when considering collisions of up to m sequences, and focused on proposing an algorithm to deal with the collisions that are non-resolvable by the UDSs in the grant-free random access channel. In a related but distinct approach, [31] proposed a physical-layer collision resolution method using FM0 delay codes combined with a finite-symbol separation Viterbi decoder (FSVD). Their technique, designed for short-range random multi-access networks, enables the separation and decoding of colliding signals without prior knowledge of fading coefficients. Information-theoretic wise, this type of collision resolution is treated under the framework of the multiple-access adder channel (MAAC). In [32], the authors proposed uniquely decodable multi-amplitude sequences in association with bit interleaving coded modulation for better error performance.

Signature sequences are related to uniquely decodable codes (UDCs) over the MAAC. UDCs apply to the *permanent activity case*, where all users are permanently active in transmitting information, and from the sum of the codewords transmitted, it is possible to infer the messages transmitted by each terminal. In that case, the set of codewords assigned to a user can be associated with different messages, but the zero codeword cannot be assigned to more than one user. Such codes are known for the two-users MAAC [33], [34], and for a rather limited number of simultaneous T users in a MAAC [35], [36]. It is known that the construction of UDCs can be related to the coin-weighing problem, as described in [26], [29], [37]. When only a subset of users is active and one wants to identify that subset, the problem is known as the *partial activity case* (or the m -out-of- t model). In this case, no more than m users out of t total users are active at any given time: inactive users transmit a zero vector, representing silence, while active users send codewords that correspond to their unique identification. The receiver collects the sum of these codewords and from it should determine which users

were active. In this case, we have UDSs, even though some literature also calls them UDCs (and are also known as alarming codes [38]). In short, UDSs are UDCs that contain a zero codeword in the set of codewords of each user, and which is associated with their inactivity periods [25], [29].

In [39] the author provided a UDC for T users for the permanent activity case, that can also function as UDSs in the context of the partial activity case for up to $T/2$ users, even though only for a noiseless MAAC. UDSs other than the BCL ones, have been addressed in: [38], considering the columns of the parity-check matrix of linear block error correction codes (ECCs) as signature codes (which due to the properties of their syndrome calculation allows having a number of columns with unique sums); in [40], using detection matrices, and not providing error correcting capability for noisy sum-sequences; and in [41], allowing for error correcting capability but using non-binary sequences with exponential length on the number of users (due to an embedded Hadamard construction); in [42], with signature sequences constructed from Pascal's triangles; and in [43], where the authors construct UDSs from linear ECCs and extend those sequences via an Hadamard construction.

Up until now we have discussed the problem of dealing with random access to the physical channel. Once the random users gain access, different problems arise, depending on the terminals' objectives. For example, [44] deals with the problem of having massive random access (MRA) to a centralized that stores a compressed version of the information of correlated sources, where numerous users randomly request different data segments. They introduced the so-called incremental coding based extractable compression (IBEC), which enables a server to store everything once and then quickly extract just what each user needs without re-encoding the data, saving time and resources.

To our knowledge, the BCL integer sets have never been generated and made easily available, so that they could be tested in a range of systems. This paper constitutes a first step towards the use of BCL sequences in the MAAC. Our goal is to provide explicit integer sets, characterize their properties, and evaluate their behavior under AWGN. The $BCL(t, m)$ integer sets are computationally obtained and listed in the present work for the first time. We present lists of $BCL(t, m)$ integer sets which were feasible to obtain with reasonable computational complexity (less than 48 hours on CPU Intel Core I7-12700 at 3.2 GHz). Traditional orthogonal signature codes, such as Hadamard codes, are known for their robust error detection capabilities but lack the flexibility needed to handle dynamic and variable user populations. In contrast, BCL sequences offer shorter sequence lengths and greater adaptability, especially when the number of active users is much smaller than the total user count. In this work, the identification error performance of BCL sequences are also analyzed for the first time considering that the receiver only has access to noisy sums. The failure probability of systems using UDSs mapped onto a binary antipodal signal such as binary phase-shift keying (BPSK) is analytically derived and

numerically verified, considering additive white Gaussian noise (AWGN) at the BS.

II. SYSTEM MODEL

We consider a system with t terminals that apply a random access protocol to access the channel via (non-orthogonal) BCL sequences. As will be seen later, such sequences require the system to be designed for a number of users $r \geq t$, which is required to be prime. Without loss of generality, we consider systems with a prime number of users $t = r$. At a given time slot, m -out-of- r terminals want to simultaneously access the channel to signal an alarm or declare that they want to initiate the transmission of the message. Thus, we always have $m \leq t \leq r$, and when $t < r$, we just assume that $r - t$ users always remain inactive.

A system where m -out-of- r users are active is associated with a BCL set $BCL(r, m)$ containing r integer elements, generated via the BCL construction method. The system operates at three levels or layers:

- i) the *Galois field level*, over $GF(r^m)$, where the integer sets $BCL(r, m)$ have the property that any sums of m $d_i \in BCL(r, m)$ are different and the sum $S(\mathcal{A})$ uniquely identifies the terms d_i and therefore their indexes $i \in \mathcal{A}$, defining $\mathcal{A}$ as the set containing the indexes of the active users;
- ii) the *logic level*, with binary UDSs $\mathbf{u}_i$ representing the d_i integers by $1 + n$ bits in the adder channel. These n -bits sequences $\mathbf{u}_i$ are also UD, and their sums can be converted to the unique $S(\mathcal{A})$. The extra bit serves to allow the BS to know the number of active users;
- iii) the *physical-layer level*, where the signature sequences are mapped onto the $\{-1, +1\}$ alphabet, resulting in uniquely decodable sum-vectors where the $1 + n$ channel symbols after the adder channel are isomorphic to the sums of the $\mathbf{u}_i$ and can also be converted back to the BCL integer sums $S(\mathcal{A})$.

Fig. 1 illustrates the system model for transmitting BCL signature sequences over a multiple access adder channel (MAAC), considering a slotted-time system, similar to the one in [16], [41], [43]. Each terminal can be active or inactive, which is determined by the variable $a \in \{0, 1\}$. The probability $P(a = 1)$ is considered to be low, such that the number of m terminals colliding is small in comparison to the total number of t terminals. This model can encompass the case of the wireless access channel, where each terminal has access to perfect channel state information (CSI) and power control to compensate for the channel gain. Time division duplex (TDD) with channel reciprocity is assumed, such that the channel can be estimated from a reference signal sent from the BS and inverted, as considered in [2], [19], [20]. Before transmission, the signature sequences are mapped onto binary phase shift-keying (BPSK), or, equivalently, to amplitude shift-keying (ASK).

In this system, each user is assigned a unique integer, $d_i \in BCL(r, m)$, which is then converted to binary, leading to the terminal's signature sequence $\mathbf{u}_i$ (a n -bit long vector).

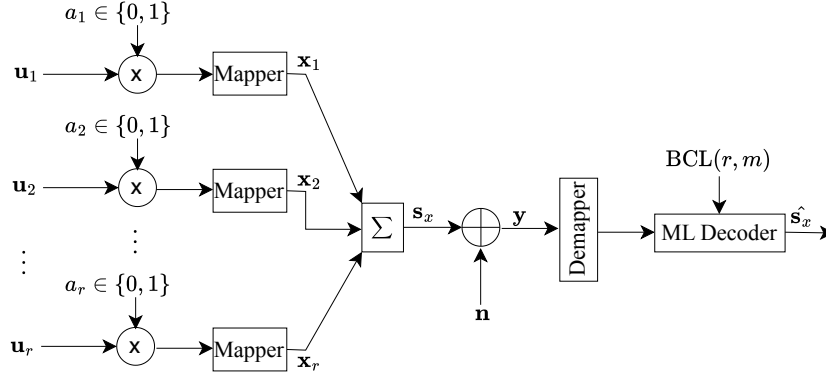


FIGURE 1. System model for the adder channel using uniquely decodable signature sequences.

This sequence is then mapped to the physical-layer, using the $\mathcal{M}$ mapper, producing vector $\mathbf{x}_i = \mathcal{M}(\mathbf{u}_i)$, containing n symbols. Crucially, the unique decodability property of the d_i over the adder channel is preserved by both $\mathbf{u}_i$ and $\mathbf{x}_i$ sequences. In fact, the sum $S(\mathcal{A}) = \sum_{i \in \mathcal{A}} d_i$ can also be obtained either from $\mathbf{s}_u = \sum_{i \in \mathcal{A}} \mathbf{u}_i$ or from $\mathbf{s}_x = \sum_{i \in \mathcal{A}} \mathbf{x}_i$. Without loss of generality, we assume that the m active users are the first ones, i.e.: $a_i = 1$, for $i = 1, 2, \dots, m$, and $a_i = 0$, for $i = m + 1, m + 2, \dots, t$, holding the respective signature sequences $\mathbf{u}_i$, with $i = 1, 2, \dots, m$. At the receiver, the sum-signal $\mathbf{s}_x$ is corrupted by additive white Gaussian noise (AWGN) with zero mean and noise power σ_n^2 , and can be written as:

$$\mathbf{y} = \sum_{i=1}^m \mathbf{x}_i + \mathbf{n} = \mathbf{s}_x + \mathbf{n}. \quad (1)$$

Given as input the (logic) binary sequences $\mathbf{u}_i$, the adder channel leads to the non-binary vector $\mathbf{s}_u = \sum_{i=1}^m \mathbf{u}_i$, whose elements $s_u(i) \in \mathbb{S}_u = \{0, 1, 2, \dots, m-2, m-1, m\}$, as exemplified in Figure 2. Hence, the sequences $\mathbf{s}_u$ are in $[\text{GF}(m+1)]^{1+n}$.

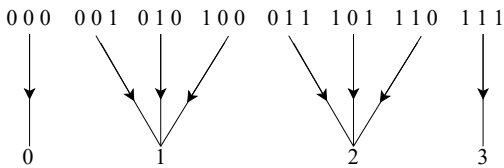


FIGURE 2. Outputs of the adder channel when adding $m = 3$ bits, leading to $\mathbb{S}_u = \{0, 1, 2, 3\}$.

It is easy to see that $\mathbb{S}_u$ is isomorphic to the set $\mathbb{S}_x$, consisting of the possible values for the elements of the sum-vector $\mathbf{s}_x = \sum_{i=1}^m \mathbf{x}_i$, with $s_x(i) \in \mathbb{S}_x = \{-m, -m+2, -m+4, \dots, m-4, m-2, m\}$. Hence, each $s_u(j)$ can be obtained from the individual symbols in the adder channel, $s_x(i)$, by means of the one-to-one mapping $s_u(i) \leftrightarrow s_x(j)$, for $j = 1, 2, \dots, n$. Eventually, the sum-signal vector $\mathbf{s}_u$ at the logic-level can be converted into the integer sum by applying

$$S(\mathcal{A}) = \sum_{j=1}^n 2^{n-j} s_u(j). \quad (2)$$

A maximum-likelihood (ML) decoder is used to detect the sum-signal $\mathbf{s}_x$ at the receiver. The ML detector performs an exhaustive search over all possible sums of codewords, selecting the one with the smallest Euclidean distance to the received signal:

$$\hat{\mathbf{s}}_x = \arg \min_{\mathbf{s}_x \in \mathcal{C}} \|\mathcal{Q}(\mathbf{y}) - \mathbf{s}_x\|^2, \quad (3)$$

where $\mathcal{C}$ is the set of all possible sum-vectors $\mathbf{s}_x$ associated with the unique sums generated by the BCL set $\text{BCL}(r, m)$. The operator $\mathcal{Q}(\cdot)$ denotes the quantization to the set $\mathbb{S}_x$, applied in a hard-decision ML decoder. By removing the quantizing operator $\mathcal{Q}(\cdot)$ in (3), one obtains a soft-decision ML decoder.

III. BOSE-CHOWLA-LINDSTRÖM SIGNATURE SEQUENCES

The known most efficient family of UDSs are the BCL sequences [25], [26]. For a system with t users, out of which m are active, the sequence's length is $n(t, m) \sim m \log t$ bits [25], [29], [30], denoting that

$$\lim_{m \rightarrow \infty} \lim_{t \rightarrow \infty} \frac{n(t, m)}{m \log t} \leq 1.$$

Note that BCL sequences are not necessarily binary, as they are constructed from integers, that can be mapped onto different Galois fields [25].

The BCL construction aims at obtaining a set of $r \geq t$ integers such that m -out-of- r have uniquely decodable sums, which are denoted by $\text{BCL}(r, m)$. The construction requires r to be a prime power (i.e., an integer of the form p^i , where p is prime and i is an integer). To construct a code for t users, one first generates a code for $r \geq t$ users, choosing r as a prime power, and subsequently, one discards $r - t$ integers (i.e., $r - t$ terminals are considered to be always muted) to produce the set of integers to be used by the t users. As a consequence of the prime number theorem, the upper bound $r \leq 2^{\lceil \log_2 t \rceil}$ can be established, where $\lceil x \rceil$ denotes the first integer that is larger than x .

For construction, let α be a primitive element of $\text{GF}(r^m)$, that is, $\text{GF}(r^m) = \{0, \alpha^0, \alpha^1, \dots, \alpha^{r^m-1}\}$, meaning that all elements in $\text{GF}(r^m)$ can be generated by α^i . Note that any primitive element of any Galois field verifies $\alpha > 1$.

Let $\beta_1, \beta_2, \dots, \beta_r$ be elements of $\text{GF}(r)$. Define integers $d_1, d_2, \dots, d_r$ such that [30], [25]:

$$\alpha^{d_i} = \alpha + \beta_i \quad \text{for } i = 1, 2, \dots, r. \quad (4)$$

The primitive elements α and d_i can be computed using Python's Galois fields package. Since α is one of the primitive elements of $\text{GF}(r^m)$ and $\alpha + \beta_i$ is a nonzero element, such d_i exists for every i . Given that $\alpha > 1$, the right side of (4) is > 1 , and therefore $d_i \neq 0$. Moreover, recalling the definition of primitive power, one gets that $0 < d_i \leq r^m - 1$.

In our implementation, the Galois field arithmetic is handled using the Galois library in Python, which allows efficient computations within finite fields. It begins by defining the Galois field $\text{GF}(r)$, where r is a user-specified prime integer. This field includes elements $\beta_1, \beta_2, \dots, \beta_r$, which are stored in a list for further processing. Next, the procedure defines the extended Galois field $\text{GF}(r^m)$ and identifies its primitive elements α . These primitive elements are critical, as they form the basis for calculating the d_i values for the BCL sets. For each primitive element α , the library iteratively searches for integers d_i such that (4) holds for each element β_i in $\text{GF}(r)$. This solution is stored in a dictionary, mapping each primitive element to its corresponding list of d_i values.

One crucial characteristic of the numbers $d_1, d_2, \dots, d_r$ is that different sums of exactly m of them cannot coincide. Defining

$$S(\mathcal{A}) = \sum_{i=1}^m d_{a_i} \quad ; \quad S(\mathcal{B}) = \sum_{i=1}^m d_{b_i} \quad (5)$$

then

$$S(\mathcal{A}) = S(\mathcal{B}) = S \iff \{a_1, a_2, \dots, a_m\} = \{b_1, b_2, \dots, b_m\}. \quad (6)$$

This means that if the sums of two different subsets of m integers are the same, then these sets must consist of the same elements and, given the construction in (4), they also appear in the same order. In short, the sets must be identical.

Let the binary representations of the integers d_i be denoted as $\mathbf{u}_i$. They have one extra digit 1 to allow the BS to count the number of users colliding, as mentioned in Section II. It requires $\lceil \log_2(\ell+1) \rceil$ bits to express integers up to ℓ in binary. Given $d_i < r^m - 1$ and $r \leq 2^{\lceil \log_2 t \rceil}$, consequently:

$$n(t, m) = \lceil \log_2(r^m - 1) \rceil + 1. \quad (7)$$

An upper bound for this number is:

$$\begin{aligned} n(t, m) &= m \log_2 r + 2 \\ &\leq m \lceil \log_2 t \rceil + 2 \\ &\leq (1 + \log_2 t)m + 2. \end{aligned} \quad (8)$$

The property in (6) translates to binary sequences, such that any sum of two binary sequences over the adder channel leads to a unique sequence that corresponds to the unique sum in (6) [30], [25]. This is easy to confirm: each bit in the $\mathbf{x}_i$ is associated with a weight 2^i , depending on its position. These vectors are added, as defined in (1), and the addition,

performed element-wise, preserves the sums at each weight 2^i , such that the sum of the integers, $S(\mathcal{A})$, can also be obtained using (2).

As an example, let us consider a system with $r=5$ and $m=2$; consequently, the β 's come from the field $\text{GF}(5)=\{0, 1, 2, 3, 4\}$ and the d_i integers belong to $\text{GF}(5^2)$. Table 1 provides the lists of the d_i integers associated with each of the primitive elements α of $\text{GF}(5^2)$, that result from (4), which were computed using Python's Galois field package:

TABLE 1. BCL ($r = 5, m = 2$) in $\text{GF}(r^m)$

r	m	α	d_i
5	2	5	{1, 22, 15, 2, 17}
		9	{1, 17, 14, 15, 10}
		10	{1, 8, 4, 17, 3}
		13	{1, 3, 17, 16, 20}
		15	{1, 3, 17, 16, 20}
		17	{1, 8, 4, 17, 3}
		20	{1, 17, 14, 15, 10}
		21	{1, 22, 15, 2, 17}

In order to preserve the order of the calculation of the d_i resulting from the algebraic equation (4) in $\text{GF}(r^m)$, the integers d_i are not sorted in Table 1, but rather listed preserving the order dictated by the β_i . For instance, when considering the primitive element $\alpha = 5$ and $\beta = \{0, 1, 2, 3, 4\}$, the integer set comes ordered as $d_i = \{1, 22, 15, 2, 17\}$. In detail, operating in $\text{GF}(5^2)$, one has the following equalities:

$$\begin{aligned} \alpha^1 &= \alpha + 0, & \alpha^{22} &= \alpha + 1, & \alpha^{15} &= \alpha + 2, \\ \alpha^2 &= \alpha + 3, & \alpha^{17} &= \alpha + 4. \end{aligned}$$

In the Appendix we list the integer sets $\text{BCL}(r, m)$ that are computationally attainable. A summary of those system configurations that can be obtained with reasonable computational complexity is given in Table 2.

TABLE 2. BCL (r, m) in $\text{GF}(r^m)$

r	m
5	3
7	2, 3, 4, 5
11	2, 3, 4, 5, 6
13	2, 3
17	2, 3
31	2, 3, 4, 5
67	2, 3, 4

IV. BCL VS ORTHOGONAL SIGNATURES

The well-known Hadamard sequences are a set of orthogonal binary vectors with size 2^k bits, $k \in \mathbb{Z}^+$ [45]. A Hadamard matrix $\mathbf{H}$ of order $n_H = 2^k$ is a square matrix with elements in $\{-1, +1\}$, with all rows mutually orthogonal. Given a Hadamard matrix $\mathbf{H} \in \{-1, +1\}^{n_H \times n_H}$, one has $\mathbf{H} \cdot \mathbf{H}^T = n_H \mathbf{I}$, where $\mathbf{I}$ is the $n_H \times n_H$ identity matrix. For any pair of rows in $\mathbf{H}$, $\mathbf{h}_i \cdot \mathbf{h}_j = 0$ for $i \neq j$, and $\mathbf{h}_i \cdot \mathbf{h}_i = n_H$. We consider the set of Hadamard signature sequences of length n_H to correspond to the rows of a Hadamard matrix (the columns could be similarly considered).

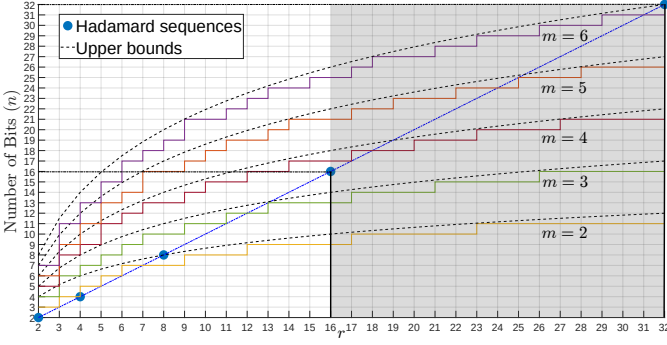


FIGURE 3. Comparison of the number of bits required for signature sequences using BCL and Hadamard constructions as a function of the total number of users r and the number of active users m . The stair-step curves represent n_{BCL} , the bit length of BCL sequences computed using (7), for $m = 2, 3, 4, 5, 6$. The black dots indicate $n_{\text{H}} = r$, the length of Hadamard sequences for $r = 2^k$. The shaded region highlights the range $16 \leq r \leq 32$ where BCL sequences require fewer bits up to $m \leq 6$ active users. Similar advantages of the BCL sequences in comparison with the Hadamard sequences (i.e., requiring shorter sequences) can also be seen for systems with $r < 16$ users.

In a MAAC for random access based on orthogonal sequences, each terminal has a physical-layer sequence $\mathbf{x}_i$, corresponding to the i -th row of $\mathbf{H}$. The identification process of the active terminals makes use of the column vector $\mathbf{s}_x$ to compute the n_{H} inner products with all the signature sequences, given by the n_{H} elements of column vector

$$\mathbf{D} = \mathbf{H} \cdot \mathbf{s}_x. \quad (9)$$

A basic example is provided for the order $n_{\text{H}} = 4$ Hadamard matrix

$$\mathbf{H} = \begin{bmatrix} +1 & +1 & +1 & +1 \\ +1 & -1 & +1 & -1 \\ +1 & +1 & -1 & -1 \\ +1 & -1 & -1 & +1 \end{bmatrix}$$

Suppose that it is the sum of the 2nd and 3rd columns. The MAAC will generate vector $\mathbf{s}_x = [+2, 0, 0, -2]^T$. To identify which columns were involved in the sum one computes

$$\mathbf{D} = (\mathbf{H} \cdot \mathbf{s}_x) / n_{\text{H}} = [0, 1, 1, 0]^T,$$

which identifies user 2 and user 3 as the active users.

BCL sequences can be constructed as long as r is a prime number and m is a positive integer. BCL sequences require n_{BCL} bits, given by (7) when expressed as a binary sequence. On the other hand, a system based on Hadamard sequences needs to be designed for a number of users that is rounded up to the next power of two ($r = 2^k$), and the length of those sequences is $n_{\text{H}} = r = 2^k$.

As seen in Fig. 3, the stair curves display BCL signature codes for varying numbers of active users, ranging from two to six by using (7), while the black points represent the first five existing Hadamard signature codes. The dashed lines show the upper bounds of the BCL sequences, as derived in (8).

There are two reasons why BCL signature sequences are preferable to Hadamard ones:

- the set containing the possible maximum number of users that a Hadamard-based system can accommodate is sparser than the one designed based on BCL sequences. This is a consequence of the set of prime numbers being denser than the set of the powers of two, for system sizes of practical interest.
- BCL sequences are shorter than Hadamard sequences when the number of active users is low compared to the number of total users in the system ($m \ll r$), which is also expected in practical scenarios.

For instance, consider systems with a number of users in the shaded region of Fig. 3; if one requires a Hadamard-based system with more than 16 users, then one is forced to design the system for a maximum number of users $r = 32$ users, implying sequences of length $n_{\text{H}} = 32$. In contrast, BCL signature codes provide in that region a large selection of values of r to choose from (i.e., the prime numbers between 16 and 32). Moreover, for various numbers of active users, m , the sequences require fewer bits, i.e., $n_{\text{BCL}} < n_{\text{H}}$. For $m = 2, 3, 4, 5, 6$, one has, respectively, $n_{\text{BCL}} = 10, 15, 19, 23, 28$ bits, which translates to a significant saving when compared with the required $n_{\text{H}} = 32$.

V. PERFORMANCE ANALYSIS OF NOISY-SUM DETECTION

We consider a system with m active users transmitting over a MAAC, where each user independently transmits either $+1$ or -1 with equal probability. The received sum signal $\mathbf{y}$ is the superposition of these m signals and can be modeled as a non-uniform M -PAM (M -ary pulse amplitude modulation) signal, where we make $M \triangleq m + 1$. The corresponding constellation at the physical layer is symmetric and defined as:

$$\{-m, -m + 2, -m + 4, \dots, m - 2, m\}.$$

This set, denoted as $\mathbb{S}_x$, corresponds to the symbol set of an M -PAM constellation. Thus, the output sum vectors $\mathbf{s}_x$ produced by the adder channel can be interpreted as M -PAM signals. Each symbol s_i in the constellation has an associated energy $|s_i|^2$, and a transmission probability $P(s_i)$ determined by how many binary combinations of ± 1 inputs sum to that value. Therefore, the average symbol energy $\mathbb{E}\{s_x^2\}$ is computed as

$$\mathbb{E}\{s_x^2\} = \sum_{i=0}^M |s_i|^2 \cdot P(s_i). \quad (10)$$

To illustrate this, consider the case where $m = 3$. Each of the three users transmits $x_i \in \{-1, +1\}$, resulting in $2^3 = 8$ combinations. All possible combinations of three users accessing the MAAC are as follows:

$$\begin{bmatrix} -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 \\ -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 \\ -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Summing column-wise one gets all the possible outcomes at the logic level:

$$\mathbf{s}_x = [-3 \quad -1 \quad -1 \quad +1 \quad -1 \quad +1 \quad +1 \quad +3].$$

The amplitudes -1 and $+1$ appear in three cases, while -3 and $+3$ appear once. Thus, the corresponding probabilities are:

$$P(-3) = P(3) = \frac{1}{8}, \quad P(-1) = P(1) = \frac{3}{8}.$$

More generally, the number of ways a specific sum can occur is governed by a binomial coefficient. Let us look at the example with $m = 3$ active users, in which the maximum sum attainable is $+3$, when the set of transmitted signals by the three terminals is $\{+1, +1, +1\}$. In fact, only that combination can reach $s_x = \sum_{i=1}^m = +3$. The sum $s_x = \sum_{i=1}^m = +1$ can be obtained from any of the sets $\{-1, +1, +1\}$, $\{+1, -1, +1\}$, $\{+1, +1, -1\}$. In general, the number of ways a given sum (and thus a symbol) can be generated depends on a binomial coefficient. In the end, the probability of each sum symbol in the non-uniform M-PAM is normalized by the total number of possible user transmission combinations, 2^m , to ensure that the sum of the probabilities equals 1.

The resulting non-uniform PAM signal are described by the following expressions:

- With m odd:

$$\begin{cases} P(s_i) = \binom{m}{\frac{m-1}{2} - i} \cdot \frac{1}{2^m} & \text{for } i = 0, \dots, \frac{m-1}{2} \\ s_i = (2i+1)^2 \end{cases} \quad (11)$$

- With m even:

$$\begin{cases} P(s_i) = \binom{m}{\frac{m}{2} - i} \cdot \frac{1}{2^m} & \text{for } i = 1, \dots, \frac{m}{2} \\ s_i = (2i)^2 \end{cases} \quad (12)$$

Due to symmetry in the M -PAM constellation, it is sufficient to compute the contribution to the average energy from only one half of the spectrum (e.g., the positive amplitudes) and then double those contributions. That symmetry accounts for the factor of 2 in both (13) and (14). Accordingly, the average symbol energy $\mathbb{E}\{s_x^2\}$ is given by:

- With m odd:

$$\mathbb{E}\{s_x^2\} = 2 \sum_{i=0}^{\frac{m-1}{2}} (2i+1)^2 \cdot \binom{m}{\frac{m-1}{2} - i} \cdot \frac{1}{2^m}. \quad (13)$$

- With m even:

$$\mathbb{E}\{s_x^2\} = 2 \sum_{i=1}^{\frac{m}{2}} (2i)^2 \cdot \binom{m}{\frac{m}{2} - i} \cdot \frac{1}{2^m}. \quad (14)$$

Theorem 1: Let m be the number of active users in a BCL-based non-uniform M -PAM modulation system, where each active user transmits ± 1 independently with equal probability. Then, the average symbol energy $\mathbb{E}\{s_x^2\}$ satisfies

$$\mathbb{E}\{s_x^2\} = \sum_{i=0}^M |s_i|^2 \cdot P(s_i) = m,$$

regardless of m being even or odd.

Proof: Let $X \sim \text{Binomial}(m, \frac{1}{2})$ represent the number of users transmitting $+1$. The received symbol can be written as:

$$s_i = 2X - m,$$

yielding the $M = m + 1$ constellation points $\{-m, -m + 2, \dots, m\}$. Let $D = |X - \frac{m}{2}|$ be the deviation from the center.

When m is even:

$$|s_i| = 2D \Rightarrow |s_i|^2 = 4D^2.$$

Therefore, the average symbol energy is

$$\mathbb{E}\{s_x^2\} = \mathbb{E}[|s_i|^2] = \mathbb{E}[4D^2] = 4 \cdot \mathbb{E}[D^2]. \quad (15)$$

To compute $\mathbb{E}[D^2]$, note that:

$$D^2 = \left(X - \frac{m}{2}\right)^2,$$

so

$$\mathbb{E}[D^2] = \mathbb{E}\left[\left(X - \frac{m}{2}\right)^2\right] = \mathbb{E}[X^2] - m \cdot \mathbb{E}[X] + \frac{m^2}{4}. \quad (16)$$

One now computes the mean and variance of X . Since $X \sim \text{Binomial}(m, \frac{1}{2})$, we have:

$$\mathbb{E}[X] = m \cdot \frac{1}{2} = \frac{m}{2}, \quad \text{Var}(X) = m \cdot \frac{1}{2} \cdot \left(1 - \frac{1}{2}\right) = \frac{m}{4}.$$

By the definition of variance:

$$\text{Var}(X) = \mathbb{E}[X^2] - (\mathbb{E}[X])^2 = \mathbb{E}[X^2] - \frac{m^2}{4},$$

which implies that

$$\mathbb{E}[X^2] = \text{Var}(X) + \frac{m^2}{4} = \frac{m}{4} + \frac{m^2}{4}.$$

Substituting this into the earlier expression for $\mathbb{E}[D^2]$ given in (16):

$$\begin{aligned} \mathbb{E}[D^2] &= \left(\frac{m}{4} + \frac{m^2}{4}\right) - m \cdot \frac{m}{2} + \frac{m^2}{4} \\ &= \left(\frac{m}{4} + \frac{m^2}{4}\right) - \frac{m^2}{2} + \frac{m^2}{4} = \frac{m}{4}. \end{aligned}$$

Finally, using (15), the average symbol energy comes as:

$$\mathbb{E}\{s_x^2\} = 4 \cdot \mathbb{E}[D^2] = 4 \cdot \frac{m}{4} = m.$$

For odd m , symbol values are $\pm(2i+1)$ with similar binomial symmetry and an analogous reasoning can be applied. $\square$

Interestingly, recovering the direct (and traditional) way of deriving the average signal power, as done in (14) and (13), one can now ascertain that, for even m :

$$\mathbb{E}\{s_x^2\} = \sum_{i=1}^{\frac{m}{2}} (2i)^2 \cdot 2 \binom{m}{\frac{m}{2} - i} \cdot \frac{1}{2^m} = m, \quad (17)$$

and for odd m ,

$$\mathbb{E}\{s_x^2\} = \sum_{i=0}^{\frac{m-1}{2}} (2i+1)^2 \cdot 2 \binom{m}{\frac{m-1}{2} - i} \cdot \frac{1}{2^m} = m. \quad (18)$$

These expressions allow us to derive performance metrics, such as symbol error rate (SER) and block error rate (BLER), in the presence of AWGN, as presented in Section VI. Obtaining the SER depends on the probabilities of each symbol and on the distance between the amplitude of the symbols, thus depending on the Q-function, defined as $Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-\frac{\lambda^2}{2}} d\lambda$. The signal-to-noise ratio (SNR) per symbol is $SNR = \frac{\mathbb{E}\{s_x^2\}}{\sigma^2} = \frac{m}{\sigma^2}$, where σ^2 is the noise variance of a zero mean AWGN, leading to

$$\frac{1}{\sigma} = \sqrt{\frac{SNR}{m}}. \quad (19)$$

The symbol error rate (SER), P_s , is the weighted sum of the individual error contributions from each constellation point s_i :

$$P_s = \sum_{i=1}^M P_e(s_i) \cdot P(s_i), \quad (20)$$

where $P_e(s_i)$ is the error probability of symbol s_i .

The symbols in the physical layer of our MAAC constitute a constellation of M symbols, separated by a distance $\Delta = 2$. The outermost symbols in the “constellation” are $s_1 = -m$ and $s_M = +m$, corresponding to the limit cases of having all the m active users transmitting -1 or $+1$, correspondingly. These symbols have only one adjacent neighbor and thus contribute with a single-sided error probability of the form $Q(\frac{\Delta/2}{\sigma})$. In contrast, the inner symbols (i.e., $-m + 2 \leq s_i \leq m - 2$) have two adjacent neighbors, resulting in two possible error directions and a corresponding contribution of $2Q(\frac{\Delta/2}{\sigma})$.

The probability of the largest symbol in the sum-constellation is $P(s_i = +m) = \frac{1}{2^m}$. Therefore, the probability of the two outermost symbols, $P(s_i = \pm m) = \frac{2}{2^m}$. The probability associated to the remaining (inner) symbols in the constellation is thus $1 - \frac{2}{2^m}$. By using both (19) and (20), we obtain the symbol error rate

$$P_s = \frac{2}{2^m} Q\left(\sqrt{\frac{SNR}{m}}\right) + \left(1 - \frac{2}{2^m}\right) 2Q\left(\sqrt{\frac{SNR}{m}}\right). \quad (21)$$

The block error rate (BLER) for the non-uniform M -PAM signals at the receiver is defined as the probability that at least one symbol in a block of n symbols are incorrectly received. The probability that a symbol is received correctly is $(1 - P_s)$ and the probability that all n symbols in a block are correctly received is $(1 - P_s)^n$. Therefore,

$$BLER = 1 - (1 - P_s)^n \quad (22)$$

Finally, the probability density function (pdf) for the random variable that makes the M -PAM signal-type observed in the physical layer of the MAAC is

$$p_S(s) = \sum_{i=0}^{M-1} P(s_i) \delta(s - s_i). \quad (23)$$

VI. SIMULATION RESULT

In this section, we evaluate the performance of BCL signature sequences under noisy channel for various system configurations, such that the non-uniform M -PAM arising in the MAAC are corrupted by AWGN. We investigate three key performance metrics as functions of the SNR, SER, BLER, and identification error rate (IDER), which is defined as the probability that at least one active user is incorrectly detected. Two (r, m) example systems are numerically evaluated via Monte Carlo simulation with 10^{p+2} simulation events run when measuring probabilities of the order 10^{-p} .

Expressions (11) and (12) are applied in two representative cases, one with an odd number of active users and another with an even number. The first case gives rise to a non-uniform 4-PAM signal, while the second leads to a 7-PAM signal. Performance is assessed using both hard and soft-ML detection techniques, as explained in Section II: in the hard-ML system, the received signal is first quantized before detection, while in the soft-ML case, the unquantized received values are utilized directly in the distance calculation.

A. ODD ACTIVE USER CONFIGURATION: $r = 5, m = 3$

In this configuration, the MAAC leads to a non-uniform 4-PAM signal with values $\{-3, -1, +1, +3\}$. These arise from all possible ± 1 combinations among three users. The symbol probabilities are $P(\pm 1) = 3/8$ and $P(\pm 3) = 1/8$, which, from (13), yield an average symbol energy $\mathbb{E}\{s_x^2\} = 3$. The PDF of this constellation is depicted in Fig. 4. Fig. 5 shows the SER, BLER, and IDER versus SNR for both hard and soft-ML detection. As illustrated in Fig. 4, the SER is computed step by step by summing the individual error contributions from each constellation point, weighted by their respective probabilities:

$$\begin{aligned} P_s &= \left(\frac{1}{8} + \frac{1}{8}\right) Q\left(\sqrt{\frac{SNR}{3}}\right) + \left(\frac{3}{8} + \frac{3}{8}\right) \cdot 2Q\left(\sqrt{\frac{SNR}{3}}\right) \\ &= \frac{7}{4} Q\left(\sqrt{\frac{SNR}{3}}\right). \end{aligned} \quad (24)$$

This result aligns with the general SER formulation in (21).

The BLER is calculated from (22) with a block length $n_{BCL} = 8$. The simulation results match the derived theoretical expressions and also demonstrate the performance benefit of soft-ML decoding.

B. EVEN ACTIVE USER CONFIGURATION: $r = 11, m = 6$

In this scenario, the MAAC creates a non-uniform 7-PAM signal with constellation points $\{-6, -4, -2, 0, +2, +4, +6\}$ with probabilities: $P(0) = \frac{20}{64}$, $P(\pm 2) = \frac{15}{64}$, $P(\pm 4) = \frac{6}{64}$, $P(\pm 6) = \frac{1}{64}$, given by (12). The average symbol energy is $\mathbb{E}\{s_x^2\} = 6$. Fig. 6 illustrates the resulting distribution. Based on the constellation depicted in Fig. 6, the SER is below obtained by aggregating the error contributions of each

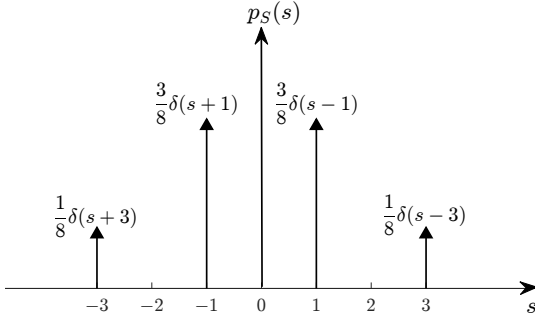


FIGURE 4. PDF of the non-uniform 4-PAM signal resulting from a BCL-based MAAC system with parameters $r = 5$, $m = 3$, and sequence length $n_{\text{BCL}} = 8$. The constellation symbols are $\{-3, -1, +1, +3\}$ with corresponding probabilities obtained from (11).

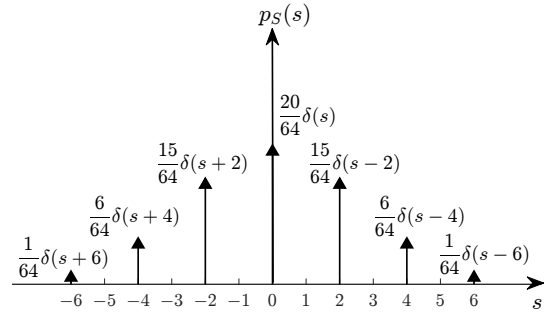


FIGURE 6. PDF of the non-uniform 7-PAM signal resulting from a BCL-based MAAC system with $r = 11$, $m = 6$, and sequence length $n_{\text{BCL}} = 22$. The constellation symbols are $\{-6, -4, -2, 0, +2, +4, +6\}$ with associated probabilities obtained from (12).

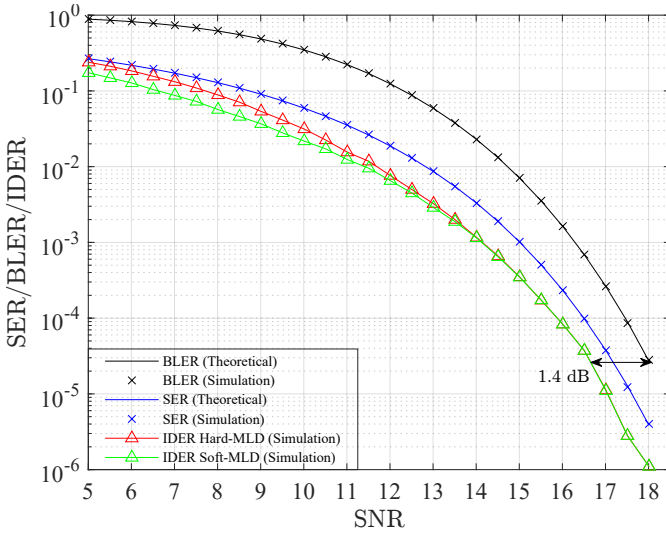


FIGURE 5. Symbol error rate (SER), block error rate (BLER), and identification error rate (IDER) as a function of signal-to-noise ratio (SNR) for the non-uniform 4-PAM modulation system with BCL sequences ($r = 5$, $m = 3$, $n_{\text{BCL}} = 8$). The theoretical SER is calculated using (24) and the theoretical BLER using (22).

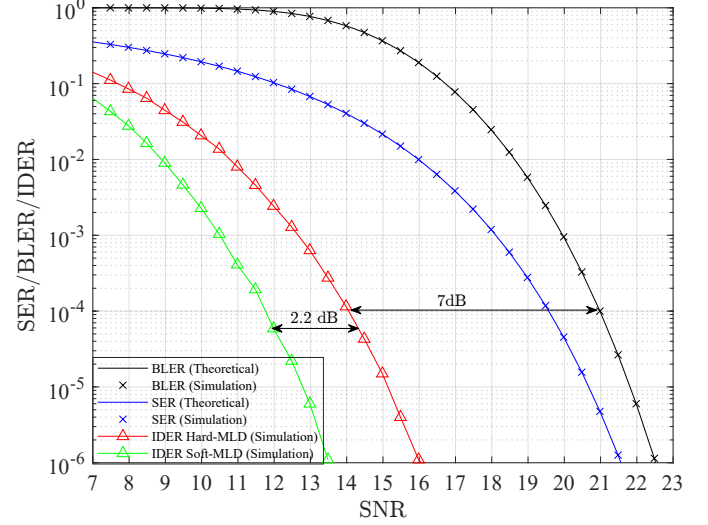


FIGURE 7. Symbol error rate (SER), block error rate (BLER) and identification error rate (IDER) as a function of signal-to-noise ratio (SNR) for the non-uniform 7-PAM modulation system with BCL sequences ($r = 11$, $m = 6$, $n_{\text{BCL}} = 22$). The theoretical SER is calculated using (25) and the theoretical BLER using (22).

symbol, weighted by their probabilities:

$$\begin{aligned}
 P_s &= \left(\frac{1}{64} + \frac{1}{64} \right) Q \left(\sqrt{\frac{\text{SNR}}{6}} \right) \\
 &+ \left(\frac{6}{64} + \frac{6}{64} + \frac{15}{64} + \frac{15}{64} + \frac{20}{64} \right) \cdot 2Q \left(\sqrt{\frac{\text{SNR}}{6}} \right) \\
 &= \frac{63}{32} Q \left(\sqrt{\frac{\text{SNR}}{6}} \right).
 \end{aligned} \tag{25}$$

This particular expression is also consistent with the general expression (21).

The BLER can follow accordingly, using (22) with $n_{\text{BCL}} = 22$. Fig. 7 presents the performances, confirming again that soft decoding significantly improves detection accuracy.

The results confirm that the analytical expressions for SER and BLER accurately model the performance of the system under AWGN. Both evaluated cases, with odd and even numbers of active users, illustrate that BCL sequences provide

reliable user identification in noisy conditions. As expected, soft-ML detection outperforms hard detection. Furthermore, IDER remains low across practical SNR values, demonstrating the feasibility of using BCL sequences in grant-free random access systems.

It is crucial to note that the BLER measures the errors caused by noise in the received sequence while the IDER measures the error after comparing the received sequence y with all the possible sum-sequences s_x , as stipulated in (3). Because not all sum-sequences s_x are valid and they constitute a finite set, the BCL sequences can in fact be seen as a non-linear error correction code operating with maximum likelihood decoding. One can see in Fig. 5 a 1.4 dB (asymptotic) gain, measured between the BLER and the two IDER curves. In this case, the soft decoder is only able to add an extra gain in the very-low SNR regime. The reason for soft MLD not offering a significant gain in this case is due to the shortness of the sequences, with only $n_{\text{BCL}} = 8$ symbols long. In contrast, Fig. 7 reveals a much larger 7 dB gain between the BLER and the hard-decoded IDER curves, demonstrating

the effectiveness of hard decoding in significantly reducing the identification error rate when longer sequences are used. In the same scenario, soft MLD also achieves a 2.2 dB improvement over hard MLD, taking advantage of the increased sequence length $n_{\text{BCL}} = 22$, which leads to finding the correct sum-sequence by averaging out the 22 distance coordinates in (3).

VII. CONCLUSIONS

This paper presents the construction and analysis of Bose-Chowla-Lindström (BCL) uniquely decodable signature sequences for use in the noisy multiple access adder channel (MAAC). The paper presents for the first time the BCL sets of unique-sum integers and, by mapping them onto binary sequences, applied them to the MAAC with AWGN. The error performance was studied using a general analytical framework for uniformly distributed random sequences, with the simulation results of systems with BCL sequences entirely matching those analytical expressions. This study can serve as a basis for future extensions considering fading channels and combined with random-access protocols. These sequences are particularly well suited for grant-free random access scenarios with partial user activity (m out of t), as they enable a reliable identification of active users from the sum-in-the-air of their transmitted signals. We showed that these sequences give rise to symbol sequences that can be interpreted as a non-uniform M -PAM signal, and we derived expressions for performance metrics, including symbol error rate (SER), block error rate (BLER), and identification error rate (IDER). The simulation results, covering examples with an odd and an even numbers of active users, confirmed the analytical expressions and demonstrated the benefits of soft maximum likelihood detection (MLD) over hard detection when the sequences are longer. Compared to traditional orthogonal codes, such as Hadamard sequences, BCL signature sequences are shorter and provide improved flexibility, particularly in sparse user activity scenarios. This makes them well-suited for ultra-reliable low-latency communications (URLLC), massive machine-type communications (mMTC), and grant-free random access protocols for IoT. An important open research problem is to find the error correcting capability of these sequences over the MAAC, interpreting the valid sum-sequences as a non-linear code. Most importantly, the derived BCL integer sets that are explicitly listed in this work, given their unique-sum property, can potentially find applications in various contexts, much beyond that of communication systems.

APPENDIX A

In the appendix, we present the sets of BCL integers d_i that have been computationally generated for different fields $\text{GF}(r^m)$, where r is a prime number and m is a positive integer. A Galois field $\text{GF}(r^m)$ can have numerous primitive elements in the range $r < \alpha < r^m$. For concision, in this appendix we present the sets for the ten first primitive elements of each

field, except in the cases where the computational complexity became infeasible.

TABLE 3. BCL ($r = 5, m = 3$) in $\text{GF}(r^m)$

r	m	α	d_i
5	3	5	{1, 96, 76, 119, 103}
		8	{1, 29, 99, 80, 84}
		9	{1, 59, 84, 20, 77}
		12	{1, 86, 77, 52, 50}
		14	{1, 20, 86, 29, 14}
		16	{1, 76, 29, 24, 82}
		18	{1, 112, 114, 77, 24}
		20	{1, 103, 119, 14, 34}
		21	{1, 77, 82, 22, 59}
		22	{1, 22, 18, 99, 29}

TABLE 4. BCL ($r = 7, m = 2$) in $\text{GF}(r^m)$

r	m	α	d_i
7	2	7	{1, 5, 38, 36, 2, 11, 31}
		8	{1, 46, 36, 10, 31, 35, 29}
		12	{1, 29, 35, 31, 34, 12, 22}
		13	{1, 31, 11, 26, 12, 14, 5}
		14	{1, 18, 21, 27, 6, 31, 20}
		19	{1, 44, 31, 30, 27, 21, 42}
		22	{1, 20, 31, 45, 46, 10, 3}
		24	{1, 3, 34, 22, 45, 31, 44}
		32	{1, 20, 31, 45, 46, 10, 3}
		34	{1, 3, 34, 22, 45, 31, 44}

TABLE 5. BCL ($r = 7, m = 3$) in $\text{GF}(r^m)$

r	m	α	d_i
7	3	7	{1, 83, 72, 22, 291, 273, 55}
		8	{1, 108, 128, 123, 267, 149, 239}
		13	{1, 199, 101, 306, 274, 111, 291}
		14	{1, 63, 311, 45, 186, 55, 22}
		16	{1, 153, 336, 263, 242, 11, 9}
		19	{1, 274, 199, 225, 329, 63, 78}
		23	{1, 110, 88, 78, 309, 70, 186}
		28	{1, 300, 177, 55, 197, 22, 159}
		31	{1, 215, 274, 177, 199, 192, 339}
		32	{1, 14, 39, 125, 222, 237, 35}

TABLE 6. BCL ($r = 7, m = 4$) in $\text{GF}(r^m)$

r	m	α	d_i
7	4	7	{1, 419, 2395, 47, 264, 1367, 2313}
		8	{1, 905, 613, 1656, 1693, 1827, 779}
		10	{1, 312, 1561, 1479, 1583, 877, 1685}
		12	{1, 639, 1703, 757, 1085, 841, 792}
		16	{1, 1693, 105, 227, 613, 2379, 56}
		17	{1, 285, 1439, 841, 903, 1592, 757}
		19	{1, 119, 1977, 328, 1163, 959, 2115}
		20	{1, 783, 1112, 877, 1561, 885, 2279}
		21	{1, 567, 1647, 2019, 713, 2264, 2395}
		25	{1, 1528, 515, 377, 159, 1719, 363}

TABLE 7. BCL ($r = 7, m = 5$) in $\text{GF}(r^m)$

r	m	α	d_i
7	5	7	{1, 12071, 12380, 9244, 8408, 13714, 4038}
		8	{1, 4576, 12914, 5812, 15788, 1536, 2843}
		14	{1, 14010, 867, 13714, 1176, 15242, 9244}
		25	{1, 13097, 995, 15504, 747, 5811, 8372}
		26	{1, 10879, 10644, 3303, 2409, 7456, 10331}
		27	{1, 14136, 12297, 14937, 9940, 10793, 655}
		32	{1, 1710, 4584, 8445, 4576, 5812, 12740}
		36	{1, 1367, 10083, 1585, 10842, 16413, 7204}
		37	{1, 6597, 11951, 13974, 1893, 9902, 209}
		39	{1, 7938, 11589, 2878, 8737, 11219, 14925}

TABLE 8. BCL ($r = 11, m = 2$) in $\text{GF}(r^m)$

r	m	α	d_i
11	2	11	{1, 27, 58, 80, 100, 98, 117, 71, 65, 66, 55}
		18	{1, 55, 6, 65, 71, 117, 38, 40, 20, 118, 27}
		22	{1, 93, 3, 71, 70, 89, 44, 6, 64, 79, 110}
		25	{1, 50, 79, 4, 66, 104, 89, 10, 71, 3, 93}
		27	{1, 76, 54, 56, 71, 70, 89, 117, 27, 79, 110}
		31	{1, 50, 79, 27, 117, 89, 10, 71, 116, 114, 16}
		33	{1, 28, 113, 99, 2, 66, 82, 69, 103, 8, 71}
		35	{1, 3, 34, 42, 74, 93, 71, 16, 7, 17, 116}
		41	{1, 56, 17, 7, 76, 71, 93, 14, 102, 94, 3}
		43	{1, 71, 68, 103, 69, 22, 6, 62, 99, 113, 88}

TABLE 9. BCL ($r = 11, m = 3$) in $\text{GF}(r^m)$

r	m	α	d_i
11	3	11	{1, 1214, 238, 21, 523, 324, 65, 1137, 111, 100, 535}
		15	{1, 1028, 1175, 849, 697, 580, 975, 857, 338, 476, 707}
		18	{1, 723, 220, 645, 1013, 862, 364, 1323, 459, 1032, 675}
		19	{1, 600, 1215, 671, 634, 98, 791, 1143, 614, 1055, 837}
		24	{1, 1170, 413, 562, 322, 1109, 508, 274, 731, 712, 794}
		29	{1, 762, 1076, 99, 1068, 526, 999, 1090, 287, 178, 1148}
		32	{1, 172, 274, 131, 320, 903, 472, 952, 309, 48, 1314}
		34	{1, 165, 604, 1294, 1112, 476, 377, 177, 175, 583, 325}
		40	{1, 131, 385, 439, 1285, 1203, 325, 432, 582, 1086, 728}
		41	{1, 805, 379, 355, 263, 835, 582, 1172, 1186, 168, 731}

TABLE 10. BCL ($r = 11, m = 4$) in $\text{GF}(r^m)$

r	m	α	d_i
11	4	11	{1, 12130, 2803, 5442, 1164, 5677, 9463, 7043, 8166, 502, 7175}
		13	{1, 9654, 7908, 3679, 5101, 1841, 12162, 7234, 2285, 13867, 7750}
		16	{1, 12019, 14159, 4878, 766, 3395, 9013, 10810, 9439, 4746, 8892}
		17	{1, 7781, 6282, 4714, 13745, 13687, 5710, 7861, 10974, 3348, 6619}
		18	{1, 14082, 13634, 12685, 8747, 5030, 10481, 6534, 6708, 12479, 12941}
		23	{1, 10102, 7781, 7861, 1890, 12438, 6178, 1884, 10817, 763, 1975}
		25	{1, 10481, 9690, 7998, 458, 5244, 9757, 6623, 11675, 12941, 9422}
		26	{1, 7770, 11118, 8698, 6444, 13997, 12463, 2155, 5101, 12142, 1841}
		31	{1, 9179, 2615, 7073, 8246, 3253, 12773, 1650, 2214, 11234, 3132}
		32	{1, 6085, 307, 12274, 2447, 12367, 9270, 12066, 8086, 7428, 12179}

TABLE 11. BCL ($r = 11, m = 5, 6$) in $\text{GF}(r^m)$

r	m	α	d_i
11	5	11	{1, 149272, 31984, 145800, 113093, 121759, 129602, 139759, 15255, 79817, 49461}
	6	11	{1, 101410, 703772, 279323, 1053498, 1565140, 1622352, 633225, 820891, 1765613, 1042680}

TABLE 12. BCL ($r = 13, m = 2$) in $\text{GF}(r^m)$

r	m	α	d_i
13	2	13	{1, 110, 87, 48, 164, 65, 21, 89, 32, 36, 39, 2, 97}
		18	{1, 21, 97, 16, 60, 135, 106, 17, 137, 118, 159, 24, 124}
		20	{1, 40, 108, 159, 34, 137, 17, 22, 135, 144, 100, 97, 21}
		25	{1, 97, 86, 39, 120, 116, 89, 21, 65, 80, 132, 87, 26}
		27	{1, 148, 82, 107, 38, 74, 131, 142, 160, 97, 81, 77, 129}
		30	{1, 82, 142, 97, 50, 80, 59, 51, 7, 75, 11, 116, 62}
		33	{1, 146, 32, 11, 75, 7, 51, 59, 164, 134, 97, 58, 166}
		36	{1, 129, 77, 81, 97, 76, 58, 131, 158, 122, 107, 166, 64}
		41	{1, 118, 23, 77, 47, 106, 97, 80, 129, 16, 124, 81, 116}
		43	{1, 91, 97, 158, 143, 40, 159, 8, 20, 135, 100, 95, 122}

TABLE 13. BCL ($r = 13, m = 3$) in $\text{GF}(r^m)$

r	m	α	d_i
13	3	13	{1, 1713, 1560, 1688, 1064, 1811, 908, 651, 802, 1887, 207, 1516, 918}
		18	{1, 1840, 717, 266, 1461, 1665, 344, 414, 1175, 1239, 1536, 412, 676}
		29	{1, 519, 1206, 479, 1182, 1223, 1219, 818, 1749, 311, 1864, 906, 780}
		32	{1, 438, 1057, 461, 1102, 375, 1321, 1040, 1038, 2028, 1247, 1569, 1818}
		34	{1, 1301, 2062, 339, 2137, 296, 2010, 1236, 1871, 417, 1134, 241, 150}
		35	{1, 1622, 1911, 665, 88, 2022, 2148, 1951, 213, 990, 1229, 282, 1217}
		38	{1, 1856, 822, 984, 1271, 849, 18, 517, 258, 1861, 1169, 970, 627}
		39	{1, 423, 1079, 249, 939, 176, 96, 1516, 1383, 956, 1650, 802, 332}
		41	{1, 507, 933, 1840, 804, 1076, 2181, 412, 1878, 998, 676, 1907, 729}
		52	{1, 1671, 2115, 698, 981, 418, 1900, 347, 1926, 1284, 1155, 542, 1322}

TABLE 14. BCL ($r = 17, m = 2$) in $\text{GF}(r^m)$

r	m	α	d_i
17	2	17	{1, 229, 178, 150, 191, 152, 87, 266, 171, 58, 183, 136, 223, 102, 2, 5, 161}
		18	{1, 202, 222, 275, 56, 267, 98, 207, 82, 75, 232, 211, 174, 122, 161, 173, 205}
		21	{1, 40, 201, 214, 117, 38, 105, 248, 161, 186, 190, 187, 31, 95, 155, 206, 138}
		29	{1, 282, 62, 155, 95, 31, 187, 46, 42, 161, 104, 105, 182, 117, 70, 201, 184}
		32	{1, 205, 173, 161, 266, 30, 211, 88, 75, 226, 207, 242, 267, 200, 275, 78, 58}
		33	{1, 161, 5, 146, 246, 223, 280, 183, 202, 171, 122, 87, 8, 191, 6, 34, 229}
		34	{1, 238, 85, 111, 70, 244, 186, 151, 119, 138, 260, 182, 15, 149, 158, 161, 243}
		36	{1, 147, 238, 196, 114, 283, 59, 66, 20, 158, 51, 161, 134, 29, 135, 61, 118}
		41	{1, 161, 102, 284, 26, 105, 83, 146, 167, 261, 103, 34, 115, 201, 10, 76, 150}
		42	{1, 6, 220, 154, 201, 115, 178, 103, 261, 167, 2, 83, 105, 170, 140, 246, 161}

TABLE 15. BCL ($r = 17, m = 3$) in $\text{GF}(r^m)$

r	m	α	d_i
17	3	17	{1, 4597, 4660, 4120, 1345, 1385, 4284, 644, 836, 4067, 3475, 1278, 2962, 3873, 2459, 3203, 2800}
		18	{1, 2948, 2248, 1789, 1477, 3916, 2836, 356, 4135, 4823, 838, 474, 2701, 959, 3015, 2720, 4413}
		21	{1, 841, 1548, 3700, 2820, 675, 2979, 4734, 802, 3969, 1291, 2179, 4768, 1249, 4437, 4532, 3016}
		29	{1, 4908, 1412, 3572, 4539, 1563, 2558, 2162, 361, 3027, 2187, 800, 3465, 3901, 1156, 1528, 3849}
		32	{1, 4065, 138, 758, 3739, 4601, 1153, 2816, 715, 727, 1564, 3512, 3835, 2963, 2884, 3644, 3388}
		33	{1, 3578, 4134, 1515, 1369, 1953, 3888, 1275, 1159, 2892, 2072, 587, 2467, 4868, 796, 4908, 3265}
		34	{1, 3723, 3571, 2016, 3697, 4501, 1636, 4440, 1521, 2041, 1660, 3460, 1044, 67, 2195, 4334, 1666}
		36	{1, 249, 4208, 819, 2351, 4828, 4648, 1267, 4555, 1428, 1852, 1916, 537, 1977, 426, 4262, 3747}
		41	{1, 4416, 891, 4231, 1420, 1656, 4779, 1123, 420, 4012, 3164, 3553, 1665, 4026, 1398, 2619, 217}
		42	{1, 2839, 2141, 2247, 362, 664, 2278, 4804, 1345, 3873, 1385, 1231, 4898, 1975, 1258, 958, 1450}

TABLE 16. BCL ($r = 31, m = 2, 3, 4, 5$) in $\text{GF}(r^m)$

r	m	α	d_i
31	2	31	{1, 655, 23, 146, 573, 508, 38, 408, 779, 332, 345, 557, 726, 709, 194, 734, 379, 906, 467, 615, 212, 629, 648, 698, 868, 3, 206, 233, 625, 511, 624}
		31	{1, 3080, 19301, 20334, 4892, 25142, 6162, 14029, 2283, 6436, 5549, 10189, 10468, 24439, 16644, 2992, 19437, 1488, 17470, 5175, 15130, 26436, 1742, 6314, 7120, 6759, 19156, 20086, 14898, 3133, 22500}
	4	31	{1, 446567, 400155, 219870, 146135, 471241, 476282, 266729, 421472, 262179, 479455, 254531, 658887, 226374, 923004, 389815, 250709, 795186, 494451, 785358, 868213, 324006, 709018, 478267, 79516, 525951, 76542, 514785, 850158, 782610, 468627}
		31	{1, 10342133, 9604646, 15264964, 16223190, 20012610, 18106802, 7238195, 11511369, 3708129, 18108920, 14120023, 4186988, 26216728, 26092301, 14986702, 24669807, 18097153, 1939695, 13891837, 4963721, 25706100, 5121522, 2907072, 15948053, 354646, 2686880, 7697960, 19164236, 20302054, 21358055}
	5	31	

TABLE 17. BCL ($r = 67, m = 2, 3, 4$) in $\text{GF}(r^m)$

r	m	α	d_i
67	2	67	{1, 1673, 4380, 292, 1385, 1568, 1038, 2153, 3413, 2859, 3567, 626, 2630, 654, 189, 868, 1504, 4068, 629, 1936, 349, 3543, 1661, 3449, 398, 2359, 3327, 38, 1371, 3560, 2242, 1638, 4278, 4354, 2900, 4341, 302, 753, 3217, 1982, 4439, 1331, 1761, 3187, 1804, 3995, 1032, 4276, 2056, 1443, 1182, 3422, 3794, 3369, 813, 2027, 2879, 4470, 4076, 791, 3856, 3984, 2135, 2311, 33, 34, 4455}
		67	{1, 285066, 165390, 104198, 1282, 165094, 51537, 188715, 95522, 33373, 54763, 34554, 205910, 262779, 5284, 221197, 258474, 4590, 258881, 61865, 67506, 183360, 268866, 193607, 220175, 217607, 281189, 111066, 81815, 279519, 108950, 135751, 221068, 28356, 14702, 23734, 262271, 97194, 294451, 124908, 178556, 90582, 29362, 283426, 272544, 187782, 286818, 120344, 97356, 74346, 56009, 111587, 141038, 80792, 264873, 180305, 49013, 279336, 261982, 204363, 47134, 196759, 290047, 173293, 65844, 236458, 264576}
	4	67	{1, 5157084, 540552, 1575080, 2130098, 3868256, 597656, 10183419, 2122602, 15801420, 7487495, 2843820, 2605983, 7463940, 1204340, 12511143, 11869423, 13442311, 2796618, 2174047, 5760790, 10028153, 16363831, 5352422, 2573368, 16441438, 12936522, 5004529, 3911067, 17356877, 6057327, 4394639, 5770568, 12270345, 6526180, 11680883, 15705025, 7884744, 7705300, 2967533, 4664805, 8258716, 3991615, 14462977, 17409065, 2075047, 2097608, 19543030, 8535596, 12077407, 17035334, 3716205, 11052286, 13727578, 7002238, 11442704, 14763519, 9885160, 10496742, 14054764, 15447672, 7860863, 16407366, 972411, 5627734, 171642, 7536112}

ACKNOWLEDGMENT

The authors are thankful to Prof. Danyo Danev (Linköping University) for his help on the algebraic operations in finite fields, and to Prof. Čedomir Stefanović (Aalborg University) for clarifications regarding the application of signature sequences in random access.

REFERENCES

- [1] N. Mahmood, S. Böcker, and I. Moerman et al., "Machine type communications: key drivers and enablers towards the 6G era," *EURASIP Journal on Wireless Communications and Networking*, no. 134, pp. 1–25, Jun. 2021.
- [2] J. Goseling, Č. Stefanović, and P. Popovski, "Sign-compute-resolve for tree splitting random access," *IEEE Trans. on Information Theory*, vol. 64, no. 7, pp. 5261–5276, Jul. 2018.
- [3] J. Thota and A. Aijaz, "On performance evaluation of random access enhancements for 5G uRLLC," in *Proc. of IEEE Wireless Communications and Networking Conf. (WCNC)*, Marrakesh, Morocco, Apr. 2019, pp. 1–7.
- [4] J. Ding, D. Qu, H. Jiang, and T. Jiang, "Success probability of grant-free random access with massive MIMO," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 506–516, Feb. 2019.
- [5] L. Valentini, M. Chiani, and E. Paolini, "Interference cancellation algorithms for grant-free multiple access with massive MIMO," *IEEE Trans. on Communications*, vol. 71, no. 8, pp. 4665–4677, Aug. 2023.
- [6] L. Liu and W. Yu, "Massive connectivity with massive MIMO — Part I: Device activity detection and channel estimation," *IEEE Trans. on Signal Processing*, vol. 66, no. 11, pp. 2933–2946, Mar. 2018.
- [7] S. Allahkaram, F. A. Monteiro, and I. Chatzigeorgiou, "uRLLC with coded massive MIMO via random linear codes and GRAND," in *Proc. of IEEE 96th Vehicular Technology Conf. (VTC2022-Fall)*, London, UK, Sep. 2022, pp. 1–5.
- [8] L. Liu, E. G. Larsson, W. Yu, P. Popovski, C. Stefanovic, and E. de Carvalho, "Sparse signal processing for grant-free massive connectivity: A future paradigm for random access protocols in the internet of things," *IEEE Signal Processing Magazine*, vol. 35, no. 5, pp. 88–99, Sep. 2018.
- [9] M. Mohammadi, Z. Mobini, H. Quoc Ngo, and M. Matthaiou, "Next-generation multiple access with cell-free massive MIMO," *Proceedings of the IEEE*, vol. 112, no. 9, pp. 1372–1420, Sep. 2024.
- [10] Y. Polyanskiy, "A perspective on massive random-access," in *Proc. of IEEE Int. Symp. on Inf. Theory (ISIT)*, Jun. 2017, pp. 2523–2527.
- [11] G. Liva and Y. Polyanskiy, "Unsourced multiple access: A coding paradigm for massive random access," *Proceedings of the IEEE*, vol. 112, no. 9, pp. 1214–1229, Aug. 2024.
- [12] Y. Liu, Y. Deng, M. ElKashlan, A. Nallanathan, and G. K. Karagiannis, "Analyzing grant-free access for uRLLC service," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 3, pp. 741–755, Mar. 2021.
- [13] P. Popovski, Č. Stefanović, J. J. Nielsen, E. de Carvalho, M. Angelichinoski, K. F. Trillingsgaard, and A.-S. Bana, "Wireless access in ultra-reliable low-latency communication (uRLLC)," *IEEE Trans. on Communications*, vol. 67, no. 8, pp. 5783–5801, May 2019.
- [14] Z. Zhang, Y. Li, C. Huang, Q. Guo, L. Liu, C. Yuen, and Y. L. Guan, "User activity detection and channel estimation for grant-free random access in LEO satellite-enabled internet of things," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 8811–8825, Sep. 2020.
- [15] J. S. Lemos, F. A. Monteiro, I. Sousa, and F. E. Ferreira, "Efficient message exchange protocols exploiting state-of-the-art phy layer," *EURASIP Journal on Wireless Communications and Networking*, vol. 2017, pp. 1–14, May 2017.
- [16] J. Goseling, M. Gastpar, and J. H. Weber, "Random access with physical-layer network coding," *IEEE Trans. on Information Theory*, vol. 61, no. 7, pp. 3670–3681, Jul. 2015.
- [17] J. Goseling, "A random access scheme with physical-layer network coding and user identification," in *Proc. of the IEEE Int. Conf. on Communications Workshops (ICC)*, Sydney, NSW, Australia, Jun. 2014, pp. 507–512.
- [18] Y. Yu and G. B. Giannakis, "High-throughput random access using successive interference cancellation in a tree algorithm," *IEEE Trans. on Information Theory*, vol. 53, no. 12, pp. 4628–4639, Dec. 2007.
- [19] J.-B. Seo, B. C. Jung, and H. Jin, "Nonorthogonal random access for 5G mobile communication systems," *IEEE Trans. on Vehicular Technology*, vol. 67, no. 8, pp. 7867–7871, Aug. 2018.
- [20] —, "Performance analysis of NOMA random access," *IEEE Communications Letters*, vol. 22, no. 11, pp. 2242–2245, Nov. 2018.

- [21] R. Alberto and F. A. Monteiro, "Downlink MIMO-NOMA With and Without CSI: A short survey and comparison," in *Proc. of the Int. Symp. on Communication Systems, Networks and Digital Signal Processing (CSNDSP)*, Porto, Portugal, Jul. 2020, pp. 1–6.
- [22] R. Tuninato, D. Riviello, R. Garelli, B. Melis, and R. Fantini, "A comprehensive study on the synchronization procedure in 5G NR with 3GPP-compliant link-level simulator," *EURASIP Journal on Wireless Communications and Networking*, no. 111, pp. 1–29, Oct. 2023.
- [23] J. H. Sørensen, E. de Carvalho, Č. Stefanović, and P. Popovski, "Coded pilot random access for massive MIMO systems," *IEEE Trans. on Wireless Communications*, vol. 17, no. 12, pp. 8035–8046, Dec. 2018.
- [24] R.-A. Pitaval, B. M. Popovic, F. Berggren, and P. Wang, "Overcoming 5G PRACH capacity shortfall by combining Zadoff-Chu and M-sequences," in *Proc. of the IEEE Int. Conf. on Communications (ICC)*, Kansas City, MO, USA, May 2018, pp. 1–6.
- [25] D. Danev, Z. Füredi, and M. Ruzinkó, "Multiple access channels: theory and practice," in *Multiple Access Channels*, E. Biglieri and L. Györfi, Eds. Amsterdam, Netherlands: IOS press, 2007, pp. 26–53.
- [26] B. Laczay, "General theory of information transfer and combinatorics," in *General Theory of Information Transfer and Combinatorics*, R. Ahlswede, L. Bäumer, N. Cai, H. Aydinian, V. Blinovskiy, C. Deppe, and H. Mashurian, Eds. Berlin, Germany: Springer, 2006, ch. Coding for the Multiple-Access Adder Channel, pp. 165–195.
- [27] G. H. Khachatrian, "A survey of coding methods for the adder channel," in *Numbers, Information and Complexity*, I. Althöfer, N. Cai, G. Dueck, L. Khachatrian, M. S. Pinsker, A. Sárközy, I. Wegener, and Z. Zhang, Eds. Boston, MA, USA: Springer, 2000, pp. 181–196.
- [28] D. G. Cantor and W. H. Mills, "Determination of a subset from certain combinatorial properties," *Canadian Journal of Mathematics*, vol. 18, pp. 42–48, 1966.
- [29] B. Lindström, "Determining subsets by unramified experiments," in *A Survey of Statistical Design and Linear Models*, J. N. Srivastava, Ed. Amsterdam, Netherlands: North-Holland, 1975, pp. 407–418.
- [30] R. C. Bose and S. Chowla, "Theorems in the additive theory of numbers," *Commentarii Mathematici Helvetici*, vol. 37, no. 1, pp. 141–147, Dec. 1962.
- [31] H. Wu, X. Wu, Y. Li, and Y. Zeng, "Collision resolution with FM0 signal separation for short-range random multi-access wireless network," *IEEE Trans. on Signal and Information Processing over Networks*, vol. 7, pp. 438–450, Jun. 2021.
- [32] Q.-Y. Yu and K.-X. Song, "Uniquely decodable multi-amplitude sequence for grant-free multiple-access adder channels," *IEEE Trans. on Wireless Communications*, vol. 22, no. 12, pp. 8999–9012, Dec. 2023.
- [33] P. van den Braak and H. van Tilborg, "A family of good uniquely decodable code pairs for the two-access binary adder channel," *IEEE Trans. on Information Theory*, vol. 31, no. 1, pp. 3–9, Jan. 1985.
- [34] R. Ahlswede and V. Balakirsky, "Construction of uniquely decodable codes for the two-user binary adder channel," *IEEE Trans. on Information Theory*, vol. 45, no. 1, pp. 326–330, Jan. 1999.
- [35] L. Kiviluoto and P. R. J. Ostergard, "New uniquely decodable codes for the T -user binary adder channel with $3 \leq T \leq 5$," *IEEE Trans. on Information Theory*, vol. 53, no. 3, pp. 1219–1220, Feb. 2007.
- [36] B. Hughes and A. Cooper, "Nearly optimal multiuser codes for the binary adder channel," *IEEE Trans. on Information Theory*, vol. 42, no. 2, pp. 387–398, Mar. 1996.
- [37] B. Lindström, "On a combinatory detection problem I," *Publications of the Mathematical Institute of the Hungarian Academy of Science*, vol. 9, pp. 195–207, 1964.
- [38] V. Gritsenko, G. Kabatiansky, V. Lebedev, and A. Maevskiy, "Signature codes for noisy multiple access adder channel," *Designs, Codes and Cryptography*, vol. 82, no. 1, pp. 293–299, Jan. 2017.
- [39] P. Mathys, "A class of codes for a T active users out of N multiple-access communication system," *IEEE Trans. on Information Theory*, vol. 36, no. 6, pp. 1206–1219, Nov. 1990.
- [40] W. H. Mow, "Recursive constructions of detecting matrices for multiuser coding: A unifying approach," *IEEE Transactions on Information Theory*, vol. 55, no. 1, pp. 93–98, Jan. 2009.
- [41] S. Lu, J. Cheng, W. Hou, and Y. Watanabe, "Generalized construction of signature code for multiple-access adder channel," in *Proc. of the IEEE Int. Symp. on Inf. Theory*, Istanbul, Turkey, Jul. 2013, pp. 1655–1659.
- [42] G. Erdoğan, G. Maringer, and N. Polyanskiy, "Signature codes for a noisy adder multiple access channel," in *Proc. of the IEEE Inf. Theory Workshop (ITW)*, Mumbai, India, Nov. 2022, pp. 476–481.

- [43] S. Lu, W. Hou, and J. Cheng, "A family of $(k + 1)$ -ary signature codes for noisy multiple-access adder channel," *IEEE Trans. on Information Theory*, vol. 61, no. 11, pp. 5848–5853, Nov. 2015.
- [44] T. Maugey, A. Roumy, E. Dupraz, and M. Kieffer, "Incremental coding for extractable compression in the context of massive random access," *IEEE Trans. on Signal and Information Processing over Networks*, vol. 6, pp. 251–260, Mar. 2020.
- [45] K. J. Horadam, *Hadamard matrices and their applications*. USA: Princeton University Press, 2012.



SAHAR ALLAHKARAM is a PhD student in the Dep. of Information Science and Technology at Iscte - University Institute of Lisbon, Portugal. She is working on Signal Processing and Coding Techniques for 6G Ultra-reliable Low-latency Wireless Machine-type Communications. She obtained her MSc in Aerospace Engineering from Sapienza University of Rome in 2020 and her BSc in Electronic Engineering from Azad University of Tehran in 2015.



FRANCISCO A. MONTEIRO (M'07) is an Associate Professor in the Dep. of Information Science and Technology at Iscte - University Institute of Lisbon, Portugal, and a researcher at Instituto de Telecomunicações, Lisbon, Portugal. He holds a PhD from the University of Cambridge, U.K., and the Licenciatura and MSc degrees in Electrical and Computer Engineering from Instituto Superior Técnico (IST), University of Lisbon, Portugal, where he also became a Teaching Assistant. He

held visiting research positions at the Universities of Toronto (Canada), Lancaster (U.K.), Oulu (Finland), and Pompeu Fabra (Barcelona, Spain). He has won two best paper prizes awards at IEEE conferences (2004 and 2007), a Young Engineer Prize (3rd place) from the Portuguese Engineers Institution (Ordem dos Engenheiros) in 2002, and for two years in a row was a recipient of Exemplary Reviewer Awards from the IEEE Wireless Communications Letters (in 2014 and 2015). He co-edited the book "MIMO Processing for 4G and Beyond: Fundamentals and Evolution", published by CRC Press in 2014. In 2016 he was the Lead Guest Editor of a special issue on Network Coding of the EURASIP Journal on Advances in Signal Processing. He was a general chair of ISWCS 2018 - The 15th International Symposium on Wireless Communication Systems, an IEEE major conference in wireless communications.

...

Chapter 6

Conclusions and future research

6.1 Main conclusions

This dissertation examined advanced signal processing and coding strategies designed to enable ultra-reliable and low-latency communications (URLLC) in emerging wireless machine-type communication (MTC) systems, where the simultaneous demands of sub-millisecond latency and extremely high reliability render conventional long-blocklength and iterative coding techniques inadequate. The central motivation of this work was therefore to search for physical-layer structures both in coding and in signal processing to develop a cohesive family of finite-blocklength coding and signal-processing methodologies including coded massive MIMO (mMIMO) with random linear code (RLC) and guessing random additive noise decoding (GRAND), symbol-level GRAND with antenna sorting, constrained noise-guessing for low-density parity-check (LDPC) codes, and uniquely decodable BCL signature sequences that collectively represent the feasibility of achieving close to maximum-likelihood (ML) detection performance at short blocklengths without incurring the prohibitive complexity typically associated with ML decoding.

The first contribution of this dissertation is that RLCs combined with the GRAND framework offer a powerful and practically implementable solution for URLLC, particularly in spatially multiplexed mMIMO systems. By reframing decoding around the noise rather than the code structure, GRAND achieves fast, maximum-likelihood performance in the short-blocklength regime, where traditional long-code asymptotics break down, and classical decoders incur prohibitive complexity. The research demonstrates 3.5 – 4.7 dB BLER improvements over polar codes and significantly reduced average decoding complexity. Moreover, the synergy between mMIMO’s spatial regularization and GRAND’s noise-centric philosophy reveals that massive MIMO inherently acts as a reliability amplifier, enabling short-block coding to operate far beyond its conventional limits, even under high-order QAM and single-shot transmission conditions.

The second contribution advanced the decoding strategy by introducing constrained symbol-level noise-guessing coupled with antenna sorting, designed for mMIMO systems. The proposed symbol-level GRAND framework, augmented with zero-forcing (ZF) or minimum mean square error (MMSE)-detection with antenna sorting, shows that carefully exploiting channel-state information significantly reduces the search space of symbol-error structures. A precise analytical characterization of symbol-error structures, combined with SNR-indexed lookup tables, yields a principled ordering of admissible error patterns, while a newly developed lightweight membership test efficiently filters out infeasible candidates before they enter the guessing sequence. Collectively, these mechanisms enable search-space reductions exceeding 90% without sacrificing ML optimality and establishing symbol-level GRAND as a practical, high-performance solution for next-generation URLLC systems.

The third contribution addressed the challenge of noise-guessing decoding to short codes by beginning with set-partitioning GRAND (SP-GRAND), introduced as a general decoding mechanism applicable to any linear block code. When first applied to random linear codes, SP-GRAND reduced complexity by constructing error patterns from Cartesian products of partitioned position sets. Although sparsifying RLC parity-check matrices accelerates this process, the results showed that such sparsity severely harms minimum distance, causing substantial BLER degradation, sometimes worse than uncoded transmission, thereby revealing a fundamental performance limitation of sparse RLCs. This motivated a transition to short, regular, and quasi-regular LDPC codes, with fixed column weights in the parity-check matrix. Two LDPC-driven mechanisms were then developed: sieving-GRAND, which uses the “fingerprints” of the columns of the $\mathbf{H}$ on the syndrome to filter likely error positions, and largely constrained GRAND (LC-GRAND), which eliminates the infeasible error patterns using zero-syndrome constraints. When combined with the general set-partitioning framework, these constraints reduce over 99% of membership tests while preserving maximum-likelihood performances. Overall, this strand showed that while sparse RLCs cannot sustain reliability under constrained guessing, LDPC codes paired with SP-GRAND, sieving, and large constraints enable practical ML decoding at short blocklengths, making them strong candidates for URLLC applications.

The fourth contribution focuses on the construction and analysis of Bose–Chowla–Lindström (BCL) uniquely decodable signature sequences for the noisy multiple-access adder channel (MAAC). These sequences guarantee that the sum of up to m active users produces a unique signature, enabling reliable identification of active users even in the presence of noise. For the first time, explicit algebraic BCL integer sets with the required sum of m uniqueness properties were constructed and mapped to short non-uniform M -PAM-like sequences. Analytical expressions for symbol error rate (SER), block error rate (BLER), and identification error rate (IDER) revealed significant performance benefits, including SNR gains of up to 7 dB for longer sequences. These results confirm that BCL sequences serve as inherently collision-resolving, making them strong candidates for practical wireless communication systems.

Taken together, these contributions form an integrated set of physical-layer tools that bring short-blocklength, high-reliability, and low-latency operation into alignment. The thesis shows that noise-centric decoding (via GRAND and its constrained variants), when matched with appropriate code structures (RLCs and short LDPC) and with signal-processing techniques (ZF / MMSE mMIMO and antenna-sorting), can reshape the complexity-reliability tradeoff in short-blocklength communication. Likewise, the BCL-based UD sequences demonstrate that algebraic combinatorial designs can play a decisive role in enabling and reliable grant-free access in next-generation MTC systems.

Overall, this dissertation provides a unified perspective on how decoding algorithms, coding structures, and multi-antenna signal processing can be co-designed to meet the specifications of URLLC and massive machine-type communications. The resulting techniques not only outperform existing 5G-NR baselines but also expand the theoretical understanding of what is achievable within finite-blocklength and multi-user regimes. These contributions lay the groundwork for future physical-layer designs that can fully exploit short-packet communications while ensuring stringent reliability and latency requirements.

6.2 Future research proposals

While this dissertation establishes noise-centric decoding and constrained GRAND frameworks as viable physical-layer solutions for short-blocklength URLLC and massive machine-type communications, several promising research directions remain open and warrant further investigation.

1) Hardware-Aware and Low-Power Implementations of GRAND Decoders

A critical next step is the translation of constrained GRAND algorithms into hardware-efficient architectures suitable for real-time deployment in 6G base stations and edge devices. Although the proposed methods significantly reduce average decoding complexity, future work should focus on memory access patterns and energy-per-decode metrics. In particular, mapping antenna-sorted and symbol-level GRAND to massively parallel architectures could reduce the average decoding latency while preserving ML optimality under URLLC constraints.

2) GRAND Decoding under Imperfect Channel State Information

The symbol-level GRAND and antenna-sorting framework developed in Chapter 3 relies on symbol-reliability metrics derived from post-detection effective SNRs under ZF or MMSE processing. This approach is well justified under strong channel hardening and reliable channel estimation. However, in practical 6G deployments, particularly under short sequences, high mobility, or massive connectivity, channel state information (CSI) is inevitably imperfect. Future work should therefore focus on extending antenna sorting and constraint design to explicitly account for imperfect CSI. Rather than relying on point estimates of post-detection

SNR, uncertainty-aware or probabilistic reliability metrics could be incorporated into the noise-guessing order and constraint selection, allowing GRAND to maintain near-ML performance while preserving low-latency under imperfect CSI conditions.

3) Extension to Non-Orthogonal and Cell-Free Massive MIMO Architectures

Future 6G systems are expected to rely heavily on cell-free massive MIMO and non-orthogonal multiple access (NOMA) paradigms. Extending the proposed antenna-sorting and constrained symbol-level GRAND techniques to distributed antenna systems and cooperative multi-point reception remains an open challenge. In such settings, decoding must operate under spatially distributed observations and heterogeneous link qualities, requiring new reliability metrics and decentralized constraint mechanisms within the noise-guessing framework.

4) Extension to Larger Blocklengths and Higher-Order Error Events

While this dissertation has primarily focused on short blocklengths (typically $n = 128$) and low error-weight regimes (up to $t = 3$), an important direction for future research is the extension of the proposed decoding frameworks to larger blocklengths, such as $n = 256, n = 512$, and beyond, together with the consideration of higher-order error events (e.g., $t = 4$ and above as long as the code holds a $d_{min} \geq 2t + 1$). Investigating this regime would provide deeper insight into the scalability of constrained GRAND techniques and their associated complexity–performance trade-offs as both blocklength and error multiplicity increase. Such an extension is particularly relevant for emerging 6G scenarios in which slightly longer packets may be acceptable, provided that ultra-high reliability requirements are maintained.

5) Generalization to Soft-Input and Soft-Output Decoding

All GRAND-based decoding strategies developed and evaluated in this dissertation are based on hard-decision inputs. While hard-input decoding offers simplicity and deterministic latency, it discards valuable reliability information available at the receiver. A natural and promising extension of this work is the generalization of the proposed GRAND-based and constrained decoding techniques to soft-input, and, potentially, soft-output [69]. By incorporating soft information such as symbol reliabilities, log-likelihood ratios, or Euclidean distance metrics, the decoder may further improve error performance and reduce the number of required noise, particularly in ultra-reliable operating regimes. Exploring how the proposed constraints, antenna ordering strategies, and noise-guessing mechanisms can be adapted to exploit soft information constitutes an important way for future research.

6) Comparative and Cross-Layer Analysis of BCL Signature Sequences

A natural extension of this work is the comparative evaluation of the proposed BCL-based uniquely decodable signature sequences against alternative non-orthogonal signature designs within an integrated communication system. While the present study focuses on the physical-layer properties of BCL sequences under noisy adder-channel models, future research should

investigate their performance in fading channels and end-to-end grant-free random access architectures, considering interactions with higher-layer mechanisms. In particular, comparative studies that incorporate medium access control (MAC) layer procedures, such as user activity detection, collision resolution, scheduling policies, and retransmission strategies, would provide deeper insight into the system-level trade-offs between reliability, latency, and scalability. Such a cross-layer perspective is essential to assess the practical viability of BCL-based signatures in realistic 6G massive machine-type communication scenarios.

References

- [1] A. E. Kalor et al., “Wireless 6G connectivity for massive number of devices and critical services”, *Proceedings of the IEEE*, pp. 1–23, Nov. 2024. DOI: [10.1109/JPROC.2024.3484529](https://doi.org/10.1109/JPROC.2024.3484529).
- [2] N. H. Mahmood et al., “Machine type communications: Key drivers and enablers towards the 6G era”, *EURASIP Journal on Wireless Communications and Networking*, vol. 2021, no. 1, p. 134, Jul. 2021. DOI: [10.1186/s13638-021-02010-5](https://doi.org/10.1186/s13638-021-02010-5).
- [3] M. Chafii, L. Bariah, S. Muhaidat, and M. Debbah, “Twelve scientific challenges for 6G: Rethinking the foundations of communications theory”, *IEEE Communications Surveys & Tutorials*, vol. 25, no. 2, pp. 868–904, Feb. 2023. DOI: [10.1109/COMST.2023.3243918](https://doi.org/10.1109/COMST.2023.3243918).
- [4] S. R. Pokhrel, J. Ding, J. Park, O.-S. Park, and J. Choi, “Towards enabling critical mMTC: A review of URLLC within mMTC”, *IEEE Access*, vol. 8, pp. 131 796–131 813, Jul. 2020. DOI: [10.1109/ACCESS.2020.3010271](https://doi.org/10.1109/ACCESS.2020.3010271).
- [5] T. Q. Duong, S. R. Khosravirad, C. She, P. Popovski, M. Bennis, and T. Q. Quek, *Ultra-reliable and Low-Latency Communications (URLLC) theory and practice: Advances in 5G and beyond*. John Wiley & Sons, Mar. 2023.
- [6] M. Geiselhart, F. Krieg, J. Clausius, D. Tandler, and S. ten Brink, “6G: A welcome chance to unify channel coding?”, *IEEE BITS the Information Theory Magazine*, vol. 3, no. 1, pp. 67–80, Mar. 2023. DOI: [10.1109/MBITS.2023.3322974](https://doi.org/10.1109/MBITS.2023.3322974).
- [7] S. Lu et al., “Integrated sensing and communications: Recent advances and ten open challenges”, *IEEE Internet of Things Journal*, vol. 11, no. 11, pp. 19 094–19 120, Feb. 2024. DOI: [10.1109/JIOT.2024.3361173](https://doi.org/10.1109/JIOT.2024.3361173).
- [8] S. R. Sabuj, A. Ahmed, Y. Cho, K.-J. Lee, and H.-S. Jo, “Cognitive UAV-aided URLLC and mMTC services: Analyzing energy efficiency and latency”, *IEEE Access*, vol. 9, pp. 5011–5027, Dec. 2021. DOI: [10.1109/ACCESS.2020.3048436](https://doi.org/10.1109/ACCESS.2020.3048436).
- [9] I. Marić, S. Shamai, and O. Simeone, *Information Theoretic Perspectives on 5G Systems and Beyond*. Cambridge University Press, 2022.

- [10] M. Shirvanimoghaddam et al., “Short block-length codes for ultra-reliable low latency communications”, *IEEE Communications Magazine*, vol. 57, no. 2, pp. 130–137, Dec. 2019. DOI: [10.1109/MCOM.2018.1800181](https://doi.org/10.1109/MCOM.2018.1800181).
- [11] D. J. MacKay, *Information theory, inference and learning algorithms*. United Kingdom: Cambridge University Press, 2003.
- [12] N. H. Mahmood, I. Atzeni, E. A. Jorswieck, O. L. A. López, et al., “Ultra-reliable low-latency communications: Foundations, enablers, system design, and evolution towards 6G”, *Foundations and Trends in Communications and Information Theory*, vol. 20, no. 5-6, pp. 512–747, 2023. DOI: [10.1561/01000000129](https://doi.org/10.1561/01000000129).
- [13] J. Östman, A. Lancho, G. Durisi, and L. Sanguinetti, “URLLC with massive MIMO: Analysis and design at finite blocklength”, *IEEE Transactions on Wireless Communications*, vol. 20, no. 10, pp. 6387–6401, Apr. 2021. DOI: [10.1109/TWC.2021.3073741](https://doi.org/10.1109/TWC.2021.3073741).
- [14] A. Tarek, A. Abdelaziz, and H. Al-Dahshan, “Channel coding challenges for short and moderate block lengths-based applications”, in *2023 International Telecommunications Conference (ITC-Egypt)*, Alexandria, Egypt, Jul. 2023, pp. 266–271. DOI: [10.1109/ITC-Egypt58155.2023.10206234](https://doi.org/10.1109/ITC-Egypt58155.2023.10206234).
- [15] A. Morello and U. Reimers, “DVB-S2, the second generation standard for satellite broadcasting and unicasting”, *International Journal of Satellite Communications and Networking*, vol. 22, no. 3, pp. 249–268, 2004. DOI: [10.1002/sat.788](https://doi.org/10.1002/sat.788).
- [16] M. Rowshan, M. Qiu, Y. Xie, X. Gu, and J. Yuan, “Channel coding toward 6G: Technical overview and outlook”, *IEEE Open Journal of the Communications Society*, vol. 5, pp. 2585–2685, Apr. 2024. DOI: [10.1109/OJCOMS.2024.3390000](https://doi.org/10.1109/OJCOMS.2024.3390000).
- [17] A. Gautam, P. Thakur, and G. Singh, “Analysis of universal decoding techniques for 6G ultra-reliable and low-latency communication scenario”, *Future Internet*, vol. 17, no. 4, Apr. 2025. DOI: [10.3390/fi17040181](https://doi.org/10.3390/fi17040181).
- [18] L. Lu, G. Y. Li, A. L. Swindlehurst, A. Ashikhmin, and R. Zhang, “An overview of massive MIMO: Benefits and challenges”, *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 5, pp. 742–758, 2014. DOI: [10.1109/JSTSP.2014.2317671](https://doi.org/10.1109/JSTSP.2014.2317671).
- [19] F. A. Monteiro, “Lattices in MIMO spatial multiplexing: Detection and geometry”, Ph.D. dissertation, University of Cambridge, United Kingdom, 2012.
- [20] 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, Jul. 2019, ISSN: 1557-9654. DOI: [10.1109/TIT.2019.2896110](https://doi.org/10.1109/TIT.2019.2896110).
- [21] S. M. Abbas, M. Jalaeddine, and W. J. Gross, *Guessing Random Additive Noise Decoding: A Hardware Perspective*. Switzerland: Springer, 2023. DOI: [10.1007/978-3-031-31663-0](https://doi.org/10.1007/978-3-031-31663-0).

- [22] K. R. Duffy, M. Médard, and W. An, “Guessing random additive noise decoding with symbol reliability information (SRGRAND)”, *IEEE Transactions on Communications*, vol. 70, no. 1, pp. 3–18, 2022. DOI: [10.1109/TCOMM.2021.3114315](https://doi.org/10.1109/TCOMM.2021.3114315).
- [23] K. R. Duffy, W. An, and M. Médard, “Ordered reliability bits guessing random additive noise decoding”, *IEEE Transactions on Signal Processing*, vol. 70, pp. 4528–4542, Aug. 2022. DOI: [10.1109/TSP.2022.3203251](https://doi.org/10.1109/TSP.2022.3203251).
- [24] K. R. Duffy and M. Médard, “Guessing random additive noise decoding for digital data communication”, *Foundations and Trends in Integrated Circuits and Systems*, vol. 5, no. 2, pp. 105–215, 2026. DOI: [10.1108/FTICS-09-2025-0017](https://doi.org/10.1108/FTICS-09-2025-0017).
- [25] S.-I. Chu, S.-A. Ke, S.-J. Liu, and Y.-W. Lin, “An efficient hard-detection GRAND decoder for systematic linear block codes”, *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 31, no. 11, pp. 1852–1864, Nov. 2023. DOI: [10.1109/TVLSI.2023.3300568](https://doi.org/10.1109/TVLSI.2023.3300568).
- [26] F. A. B. Cercas, “A new family of codes for simple receiver implementation”, Ph.D. dissertation, Technical University of Lisbon, Lisbon, Portugal, 1996.
- [27] N. L. Biggs, “Linear codes”, in *Codes: An Introduction to Information Communication and Cryptography*. London: Springer London, 2008, pp. 1–18. DOI: [10.1007/978-1-84800-273-9_8](https://doi.org/10.1007/978-1-84800-273-9_8).
- [28] W. Ryan and S. Lin, *Channel codes: classical and modern*. United Kingdom: Cambridge University Press, 2009.
- [29] J. C. Moreira and P. G. Farrell, *Essentials of error-control coding*. United Kingdom: John Wiley & Sons, 2006.
- [30] D. J. MacKay and R. M. Neal, “Good codes based on very sparse matrices”, in *IMA International Conference on Cryptography and Coding*, Springer, Berlin, Oct. 1995, pp. 100–111. DOI: [10.1007/3-540-60693-9_13](https://doi.org/10.1007/3-540-60693-9_13).
- [31] D. MacKay, “Good error-correcting codes based on very sparse matrices”, *IEEE Transactions on Information Theory*, vol. 45, no. 2, pp. 399–431, Mar. 1999. DOI: [10.1109/18.748992](https://doi.org/10.1109/18.748992).
- [32] N. Alon and M. Luby, “A linear time erasure-resilient code with nearly optimal recovery”, *IEEE Transactions on Information Theory*, vol. 42, no. 6, pp. 1732–1736, Nov. 1996. DOI: [10.1109/18.556669](https://doi.org/10.1109/18.556669).
- [33] J. W. Byers, M. Luby, M. Mitzenmacher, and A. Rege, “A digital fountain approach to reliable distribution of bulk data”, *ACM SIGCOMM Computer Communication Review*, vol. 28, no. 4, pp. 56–67, Oct. 1998.
- [34] R. Gallager, “Low-density parity-check codes”, *IRE Transactions on Information Theory*, vol. 8, no. 1, pp. 21–28, Jan. 1962. DOI: [10.1109/TIT.1962.1057683](https://doi.org/10.1109/TIT.1962.1057683).

- [35] L. Lu, G. Y. Li, A. L. Swindlehurst, A. Ashikhmin, and R. Zhang, “An overview of massive MIMO: Benefits and challenges”, *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 5, pp. 742–758, Apr. 2014. DOI: [10.1109/JSTSP.2014.2317671](https://doi.org/10.1109/JSTSP.2014.2317671).
- [36] T. Marzetta, “Multi-cellular wireless with base stations employing unlimited numbers of antennas”, in *Proc. UCSD Inf. Theory Applicat. Workshop*, 2010.
- [37] T. L. Marzetta, “Noncooperative cellular wireless with unlimited numbers of base station antennas”, *IEEE Transactions on Wireless Communications*, vol. 9, no. 11, pp. 3590–3600, Oct. 2010. DOI: [10.1109/TWC.2010.092810.091092](https://doi.org/10.1109/TWC.2010.092810.091092).
- [38] E. G. Larsson, O. Edfors, F. Tufvesson, and T. L. Marzetta, “Massive MIMO for next generation wireless systems”, *IEEE Communications Magazine*, vol. 52, no. 2, pp. 186–195, Feb. 2014. DOI: [10.1109/MCOM.2014.6736761](https://doi.org/10.1109/MCOM.2014.6736761).
- [39] C. Cahn, “Combined digital phase and amplitude modulation communication systems”, *IRE Transactions on Communications Systems*, vol. 8, no. 3, pp. 150–155, Jan. 1960. DOI: [10.1109/TCOM.1960.1097623](https://doi.org/10.1109/TCOM.1960.1097623).
- [40] C. Campopiano and B. Glazer, “A coherent digital amplitude and phase modulation scheme”, *IRE Transactions on Communications Systems*, vol. 10, no. 1, pp. 90–95, Mar. 1962. DOI: [10.1109/TCOM.1962.1088634](https://doi.org/10.1109/TCOM.1962.1088634).
- [41] J. Salz, J. Sheehan, and D. Paris, “Data transmission by combined AM and PM”, *Bell System Technical Journal*, vol. 50, no. 7, pp. 2399–2419, 1971.
- [42] M. Simon and J. Smith, “Hexagonal multiple phase-and-amplitude-shift-keyed signal sets”, *IEEE Transactions on Communications*, vol. 21, no. 10, pp. 1108–1115, Oct. 1973. DOI: [10.1109/TCOM.1973.1091549](https://doi.org/10.1109/TCOM.1973.1091549).
- [43] G. Foschini, R. Gitlin, and S. Weinstein, “On the selection of a two-dimensional signal constellation in the presence of phase jitter and gaussian noise”, *Bell System Technical Journal*, vol. 52, no. 6, pp. 927–965, 1973.
- [44] P. K. Singya, P. Shaik, N. Kumar, V. Bhatia, and M.-S. Alouini, “A survey on design and performance of higher-order QAM constellations”, *arXiv preprint arXiv:2004.14708*, 2020. [Online]. Available: <https://doi.org/10.48550/arXiv.2004.14708>.
- [45] P. K. Singya, P. Shaik, N. Kumar, V. Bhatia, and M.-S. Alouini, “A survey on higher-order QAM constellations: Technical challenges, recent advances, and future trends”, *IEEE Open Journal of the Communications Society*, vol. 2, pp. 617–655, Mar. 2021. DOI: [10.1109/OJCOMS.2021.3067384](https://doi.org/10.1109/OJCOMS.2021.3067384).
- [46] C. E. Shannon, “A mathematical theory of communication”, *The Bell system technical journal*, vol. 27, no. 3, pp. 379–423, 1948. DOI: [10.1002/j.1538-7305.1948.tb01338.x](https://doi.org/10.1002/j.1538-7305.1948.tb01338.x).

- [47] C. Berrou, A. Glavieux, and P. Thitimajshima, “Near shannon limit error-correcting coding and decoding: Turbo-codes. 1”, in *Proceedings of ICC '93 - IEEE International Conference on Communications*, vol. 2, May 1993, 1064–1070 vol.2. DOI: [10.1109/ICC.1993.397441](https://doi.org/10.1109/ICC.1993.397441).
- [48] E. Şaşoğlu, E. Telatar, and E. Arikan, “Polarization for arbitrary discrete memoryless channels”, in *2009 IEEE Information Theory Workshop*, Oct. 2009, pp. 144–148. DOI: [10.1109/ITW.2009.5351487](https://doi.org/10.1109/ITW.2009.5351487).
- [49] E. Arikan, “Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels”, *IEEE Transactions on information Theory*, vol. 55, no. 7, pp. 3051–3073, Jun. 2009. DOI: [10.1109/TIT.2009.2021379](https://doi.org/10.1109/TIT.2009.2021379).
- [50] M. C. Coşkun et al., “Efficient error-correcting codes in the short blocklength regime”, *Physical Communication*, vol. 34, pp. 66–79, 2019. DOI: <https://doi.org/10.1016/j.phycom.2019.03.004>.
- [51] K. R. Duffy, J. Li, and M. Médard, “Guessing noise, not code-words”, in *2018 IEEE International Symposium on Information Theory (ISIT)*, Aug. 2018, pp. 671–675. DOI: [10.1109/ISIT.2018.8437648](https://doi.org/10.1109/ISIT.2018.8437648).
- [52] K. R. Duffy, A. Solomon, K. M. Konwar, and M. Médard, “5G NR CA-polar maximum likelihood decoding by GRAND”, *54th Annual Conf. on Info. Sciences and Systems (CISS)*, 2020. DOI: [10.1109/CISS48834.2020.1570617412](https://doi.org/10.1109/CISS48834.2020.1570617412).
- [53] W. An, M. Médard, and K. R. Duffy, “Keep the bursts and ditch the interleavers”, *IEEE Transactions on Communications*, vol. 70, no. 6, pp. 3655–3667, 2022. DOI: [10.1109/TCOMM.2022.3171798](https://doi.org/10.1109/TCOMM.2022.3171798).
- [54] S. M. Abbas, M. Jalaleddine, and W. J. Gross, *Guessing random additive noise decoding: A hardware perspective*. Springer Nature, 2023.
- [55] J. Goseling, Č. Stefanović, and P. Popovski, “Sign-compute-resolve for tree splitting random access”, *IEEE Transactions on Information Theory*, vol. 64, no. 7, pp. 5261–5276, Jul. 2018.
- [56] Y. Liu, Y. Deng, M. El Kashlan, A. Nallanathan, and G. K. Karagiannidis, “Analyzing grant-free access for URLLC service”, *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 3, pp. 741–755, Mar. 2021. DOI: [10.1109/JSAC.2020.3018822](https://doi.org/10.1109/JSAC.2020.3018822).
- [57] M. B. Shahab, R. Abbas, M. Shirvanimoghaddam, and S. J. Johnson, “Grant-free non-orthogonal multiple access for IoT: A survey”, *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 1805–1838, May 2020. DOI: [10.1109/COMST.2020.2996032](https://doi.org/10.1109/COMST.2020.2996032).
- [58] P. Popovski et al., “Wireless access in ultra-reliable low-latency communication (URLLC)”, *IEEE Transactions on Communications*, vol. 67, no. 8, pp. 5783–5801, May 2019. DOI: [10.1109/TCOMM.2019.2914652](https://doi.org/10.1109/TCOMM.2019.2914652).

- [59] F. A. Monteiro et al., “Special issue on network coding”, *EURASIP Journal on Advances in Signal Processing*, vol. 2017, no. 29, pp. 1–3, Apr. 2017. doi: [10.1186/s13634-017-0463-2](https://doi.org/10.1186/s13634-017-0463-2).
- [60] J. Goseling, M. Gastpar, and J. H. Weber, “Random access with physical-layer network coding”, *IEEE Transactions on Information Theory*, vol. 61, no. 7, pp. 3670–3681, Jul. 2015. doi: [10.1109/TIT.2015.2425879](https://doi.org/10.1109/TIT.2015.2425879).
- [61] J. Goseling, “A random access scheme with physical-layer network coding and user identification”, in *Proc. of the IEEE International Conference on Communications Workshops (ICC)*, Sydney, NSW, Australia, Jun. 2014, pp. 507–512. doi: [10.1109/ICCW.2014.6881249](https://doi.org/10.1109/ICCW.2014.6881249).
- [62] R.-A. Pitaval, B. M. Popovic, F. Berggren, and P. Wang, “Overcoming 5G PRACH capacity shortfall by combining Zadoff-Chu and M-sequences”, in *Proc. of the IEEE International Conference on Communications (ICC)*, Kansas City, MO, USA, May 2018, pp. 1–6. doi: [10.1109/ICC.2018.8422146](https://doi.org/10.1109/ICC.2018.8422146).
- [63] D. Danev, Z. Füredi, and M. Ruzinkó, “Multiple access channels: Theory and practice”, in *Multiple Access Channels*, E. Biglieri and L. Györfi, Eds., Amsterdam, Netherlands: IOS press, 2007, pp. 26–53.
- [64] B. Laczay, “General theory of information transfer and combinatorics”, in *General Theory of Information Transfer and Combinatorics*, R. Ahlswede et al., Eds., Berlin, Germany: Springer, 2006, ch. Coding for the Multiple-Access Adder Channel, pp. 165–195, ISBN: 978-3-540-46245-3. doi: [10.1007/11889342_9](https://doi.org/10.1007/11889342_9).
- [65] G. H. Khachatrian, “A survey of coding methods for the adder channel”, in *Numbers, Information and Complexity*, Boston, MA, USA: Springer, 2000, pp. 181–196.
- [66] D. G. Cantor and W. H. Mills, “Determination of a subset from certain combinatorial properties”, *Canadian Journal of Mathematics*, vol. 18, pp. 42–48, 1966. doi: [10.4153/CJM-1966-007-2](https://doi.org/10.4153/CJM-1966-007-2).
- [67] B. Lindström, “Determining subsets by unramified experiments”, in *A Survey of Statistical Design and Linear Models*, J. N. Srivastava, Ed., Amsterdam, Netherlands: North-Holland, 1975, pp. 407–418.
- [68] R. C. Bose and S. Chowla, “Theorems in the additive theory of numbers”, *Commentarii Mathematici Helvetici*, vol. 37, no. 1, pp. 141–147, Dec. 1962.
- [69] K. R. Duffy, P. Yuan, J. Griffin, and M. Médard, “Soft-output guessing codeword decoding”, *IEEE Communications Letters*, vol. 29, no. 2, pp. 328–332, Dec. 2025. doi: [10.1109/LCOMM.2024.3516486](https://doi.org/10.1109/LCOMM.2024.3516486).