Unsourced Random Access

Kirill Andreev k.andreev@skoltech.ru
Pavel Rybin, p.rybin@skoltech.ru
Alexey Frolov, al.frolov@skoltech.ru

Skolkovo Institute of Science and Technology

Contents

Acronyms

3GPP Third-Generation Partnership Project

5G NR Fifth-generation New Radio

ACK acknowledgment

AMP approximate message passing

AoA angle-of-arrival

ARQ Automatic Repeat reQuest

ASR approximate support recovery

BAC binary adder channel

BCH Bose-Chaudhuri-Hocquenghem

BP belief propagation

BPSK binary phase-shift keying

BS base station

CCS coded compressed sensing

CDMA code division multiple-access

CNOP check-node operation

CRC cyclic redundancy check

CRDSA contention resolution diversity slotted ALOHA

CS compressed sensing

CSI channel state information

CSMA carrier-sense multiple-access

CSMA/CA carrier-sense multiple-access with collision avoidance

CSS chirp spread spectrum

DBPSK differential binary phase-shift keying

DE density evolution

DFT discrete Fourier transform

DSSS direct-sequence spread spectrum

FAR false alarm rate

FBL finite blocklength

FCFS first-come, first-serve

FDMA frequency division multiple-access

FFT fast Fourier transform

FT Fourier transform

GMAC Gaussian MAC

G-URA URA over Gaussian MAC

IDMA interleave division multiple-access

IoT Internet of Things

IRSA irregular repetition slotted ALOHA

IRSA-B basic IRSA protocol

IRSA-F IRSA with per-frame preambles

IRSA-S IRSA with per-slot preambles

JSC joint successive cancellation

LDPC low-density parity check

LDS low-density signature

LLR log-likelihood ratio

LMMSE linear MMSE

LPWAN low-power wide area network

MAC multiple-access channel

MF matched filter

MIMO multiple input, multiple output

ML maximum likelihood

MMSE minimum mean squared error

MMV multiple measurement vector

MPA message passing algorithm

MSE mean squared error

MTC machine-type communications

MUD multi-user detector

NNLS non-negative least squares

ODMA on-off division multiple access

OFDM orthogonal frequency-division multiplexing

OMP orthogonal matching pursuit

PAN personal-area network

PEXIT protograph extrinsic information transfer

PO physical uplink shared channel occasion

PUPE per-user probability of error

PUSCH physical uplink shared channel

QoS quality of service

QPSK quadrature phase-shift keying

RA random access

RACH random access channel

RCB random coding bound

RFID radio-frequency identification

RIP restricted isometry property

RM Reed-Muller

RS Reed-Solomon

RSMA rate-splitting multiple-access

SA slotted ALOHA

SC successive cancellation

SCL successive cancellation list

SCMA sparse-coded multiple-access

SF spreading factor

SIC successive interference cancellation

SINR signal-to-interference-plus-noise ratio

SoIC soft SIC

SPARCs sparse regression codes

SPC single-parity-check

TDMA time-division multiple-access

TIN treat interference as noise

TIN-SIC TIN followed by SIC

UE user equipment

URA unsourced random access

VNOP variable-node operation

Abstract

Current wireless networks are designed to optimize spectral efficiency for human users, who typically require sustained connections for high-data-rate applications like file transfers and video streaming. However, these networks are increasingly inadequate for the emerging era of MTC. With a vast number of devices exhibiting sporadic traffic patterns consisting of short packets, the grant-based multiple access procedures utilized by existing networks lead to significant delays and inefficiencies. To address this issue the URA paradigm has been proposed. This paradigm assumes the devices to share a common encoder thus simplifying the reception process by eliminating the identification procedure. The URA paradigm not only addresses the computational challenges but it also considers the random access (RA) as a coding problem, i.e., takes into account both medium access protocols and physical layer effects. In this monograph we provide a comprehensive overview of the URA problem in noisy channels, with the main task being to explain the major ideas rather than to list all existing solutions.

Notation

Throughout the monograph we use the following notations and abbreviations:

x, X	deterministic scalar value
$\boldsymbol{x}$	deterministic column-vector
$\boldsymbol{X}$	deterministic matrix
$\boldsymbol{I}_n$	$n \times n$ identity matrix
diag	diagonal matrix
$\|\boldsymbol{x}\|_2$	Euclidean norm of vector $\boldsymbol{x}$
$\operatorname{supp}(\boldsymbol{x})$	support of vector $\boldsymbol{x}$
$\operatorname{wt}(\boldsymbol{x})$	the number of nonzero components in vector $\boldsymbol{x}$
$\mathbb{N}$	set of natural numbers
$\mathbb{C}$	set of complex numbers
$\mathbb{F}_q$	finite field with q elements
$\mathcal{X}$	set
$\boldsymbol{\mathcal{X}}$	sequence of sets
$\bigsqcup$	disjoint union
$[n]$	$[n] = \{1, \ldots, n\}$, where $n \in \mathbb{N}$
$\boldsymbol{a}_{\mathcal{I}}$	$\boldsymbol{a}_{\mathcal{I}} = (a_{i_1}, \ldots, a_{i_s})$, where $\boldsymbol{a} = (a_1, \ldots, a_n)$ and $\mathcal{I} = \{i_1, \ldots, i_s\} \subseteq [n]$ with $i_1 < \ldots < i_s$
$\boldsymbol{A}_{\mathcal{I}}$	$\boldsymbol{A}_{\mathcal{I}} = (\boldsymbol{a}_{i_1}, \ldots, \boldsymbol{a}_{i_s})$, where $\boldsymbol{A} = (\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n)$ and $\mathcal{I} = \{i_1, \ldots, i_s\} \subseteq [n]$ with $i_1 < \ldots < i_s$
$\mathcal{CN}(\boldsymbol{0}, \boldsymbol{I}_n)$	circularly symmetric complex standard normal distribution
$\operatorname{Bern}(p)$	Bernoulli distribution with parameter p
$\operatorname{Unif}([Q])$	uniform distribution on $[Q]$
x, X	random scalar value
$\boldsymbol{\mathsf{x}}$	random column-vector
$\boldsymbol{\mathsf{X}}$	random matrix
E	event
E^c	complementary event to E
$\mathbb{1}_E$	indicator of the event E
$\Pr[E]$	probability of the event E
$\mathbb{E}$	expectation operator
$H(\mathsf{x})$	entropy of discrete random variable x
$I(\mathsf{x}, \mathsf{y})$	mutual information of random variables x and y

$h(p)$	$h(p) = -p\log_2(p) - (1-p)\log_2(1-p)$, $0 \leq p \leq 1$, where $h(1) = h(0) = 0$
$\tau(x)$	binary phase-shift keying (BPSK) modulation $\tau(x) = (1-2x)\sqrt{P}$
w.l.o.g.	"without loss of generality"
r.v.	"random variable"
i.i.d.	"independent identically distributed"
p.m.f.	"probability mass function"

Chapter 1

Massive machine-type communications

Machine-type communications (MTC) dramatically change traffic patterns. Instead of focusing on peak data rates and low latencies, massive connectivity becomes a key requirement. The MTC concept involves a massive number of autonomous devices and sensors being connected to a gateway: a node (or a set of nodes) responsible for data collection. MTC is a crucial component of the IoT paradigm, which defines the infrastructure and scenarios for interconnecting devices rather than humans. IoT encompasses various tasks, including monitoring, remote and automated control, data collection, and data-related services. MTC is a communication technology specifically designed to support this paradigm, enabling connectivity for a vast number of devices. In this monograph, we focus on the communication aspects and use the terms IoT and MTC interchangeably.

IoT applications encompass a wide range of use cases, including:

- Environmental and health monitoring.
- Smart homes, cities, and industries.
- Road traffic monitoring and tracking to improve efficiency and safety.

Typical MTC transmissions involve short measurement reports generated either regularly or sporadically, resulting in additional requirements.

1. Improved *battery life* is essential. Wiring a large number of devices to the electricity grid would require expensive cabling, making it preferable for these devices to be autonomous. Moreover, monitoring the battery status of thousands (or even millions) of devices may also be prohibitively costly. Therefore, the battery lifetime must match the device lifetime (approximately 10 years). As a result, energy-efficient solutions must utilize simple radio-frequency devices with low-complexity signal processing algorithms.

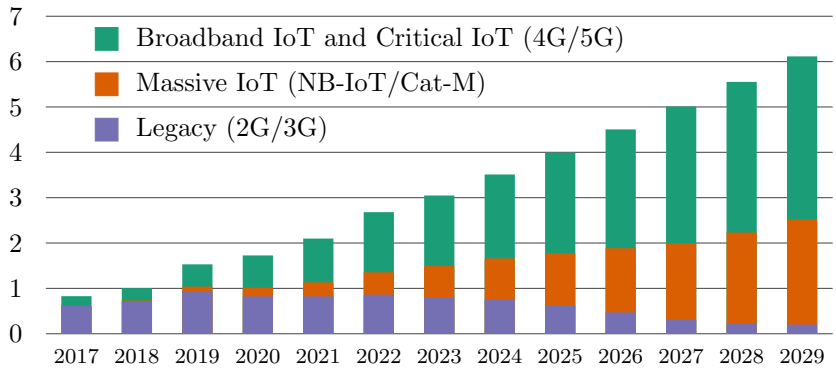

Figure 1.1: Predicted number of worldwide IoT cellular subscriptions (billions) in accordance with Ericsson mobility report [1] (Nov. 2023).

2. There is a need for *improved coverage*. Many sensors may be located in so-called deep indoor environments (e.g., building basements), leading to significant signal loss between the transmitter and receiver.

3. *Low cost* is a critical factor. The challenge of low cost is twofold:

   - since IoT devices generate only small amounts of data, subscription fees should be much lower compared to those for ordinary smartphones, minimizing operational expenses;
   - the massive deployment of IoT devices necessitates a low cost per device, reducing capital expenses.

According to Ericsson's forecast [1], the number of cellular IoT devices (see Fig. 1.1) is expected to reach 6.1 billion by 2029, while the total number of connected devices across all IoT technologies is projected to reach approximately 39 billion. Currently, the IoT industry exhibits a compound annual growth rate of approximately 16%. Given this rapid growth, standardization plays a crucial role in ensuring sustainable IoT development.

1.1 Internet of Things standardization

There are various standards for IoT, each fulfilling different requirements specified above. We distinguish three main branches of IoT technologies: short-range, wide-area, and cellular.

Short-range IoT encompasses a variety of technologies, including radio-frequency identification (RFID) and personal-area networks (PANs) such as Bluetooth and Zigbee. These technologies typically operate in unlicensed spectrum and within a very short range. Massive connectivity in this case is limited by the small coverage area (on the order of several meters) of the corresponding radio devices [2].

In contrast, wide-area technologies have the potential to connect millions of autonomous devices to a single base station (BS). In this short overview, we focus on Sigfox and LoRa technologies, which

are described in Section 1.1.2.

1.1.1 Short-range IoT

Short-range, or PAN, provides a communication environment for various IoT applications, such as communication between wearable devices, short-range location tags, home controllers, and more. A typical network consists of at most a few dozen devices, making communication between them relatively simple in terms of channel access. The low communication range is beneficial for energy efficiency and security.

An extreme example of energy efficiency (particularly regarding remote device battery life) is RFID technology, where the energy required to transmit data is induced by an interrogation pulse from a nearby reader device. The short communication range also prevents signals from being detected over large distances between the transmitter and receiver, which significantly simplifies security protocols.

Due to the absence of massive connectivity issues and the wide variety of PAN technologies, we will not consider them in the remainder of this monograph.

1.1.2 Wide-area IoT

To fulfill the requirements described above, several solutions are commonly employed in wide-area IoT systems – low-power wide area networks (LPWANs). To improve system range, narrowband signals are typically utilized. Since thermal noise power is proportional to the processed bandwidth, narrowband signals can tolerate a greater link loss for the same transmitter power and hence enable coverage of a wider area. Additionally, due to the extended communication distances, narrowband signals benefit from the reduced frequency selectivity of the wireless channel. Consequently, there is no need for complex multi-carrier modulations or high-complexity signal processing algorithms at the transmitter.

To simplify remote devices, wide-area IoT communication systems often have limited or even completely absent downlink functionality. This limitation reduces the ability to coordinate transmitting devices and typically eliminates the possibility of employing Automatic Repeat reQuest (ARQ) mechanisms. Additionally, the use of ARQ in scenarios involving sporadic data transmission by a massive number of devices can overwhelm the control channel.

To address this issue, transmissions can be repeated multiple times, potentially at different frequencies, to exploit both frequency and time diversity, assuming the retransmission delay exceeds the channel coherence time. Further transmitter simplifications often include the use of constant-envelope modulation formats, which do not require expensive or power-inefficient signal amplifiers.

At the receiver, additional solutions are applied to enhance performance. The simplest way to increase diversity is through *multiple receptions*. When multiple nodes collect the transmitted data, many can detect, demodulate, and decode the message, thereby enhancing *spatial diversity*.

These typical solutions are implemented in Sigfox and LoRa, the most popular wide-area IoT technologies, which are described below. Both approaches rely on distributed resource coordination mechanisms[1] based on ALOHA and carrier-sense multiple-access with collision avoidance (CSMA/CA), rather than the centralized coordination employed in cellular systems.

Sigfox – an ultra-narrowband IoT technology

The main feature of Sigfox is its extremely narrow transmission bandwidth of just 100 Hz. This narrow bandwidth significantly reduces in-band thermal noise to very low levels (approximately -154 dBm), which is a key enabler of its long-range communication. Downlink functionality is extremely limited, with a complete absence of synchronization, coordination, and ARQ mechanisms.

To send a packet, a device selects a transmission frequency within a 192 kHz band and transmits the packet, followed by two replicas at different randomly selected frequencies to enhance diversity. Multiple reception is achieved by deploying multiple BSs, which continuously scan the entire 192 kHz bandwidth to detect uplink messages.

The transmitted packets are exceptionally short. Each packet consists of:

- a 4-byte preamble,
- a 2-byte frame-synchronization sequence,
- a 4-byte device identifier,
- a payload of up to 12 bytes,
- a variable-length hash code for packet authentication within the Sigfox network, and
- a 2-byte cyclic redundancy check (CRC).

Uplink packets are typically modulated using differential binary phase-shift keying (DBPSK). Differential modulation is chosen to allow for non-coherent detection. The combination of simple modulation and coding, along with a very low sampling rate, results in a highly cost-effective solution. Additionally, the high link budget enables a reduction in transmit power, thereby fulfilling battery efficiency requirements with ease.

Downlink messages (if configured) are triggered by a transmitting device in the form of a callback. The BS's response has a fixed delay and is transmitted at the reception frequency of the request plus a predefined frequency shift. The payload size for a downlink message is fixed at 8 bytes.

[1]These mechanisms are often referred to as medium access control (MAC). In this monograph, however, we interpret the abbreviation MAC as multiple-access channel.

LoRa

The LoRa (Long Range) protocol is another IoT alternative. Similar to the previously described ultra-narrowband solutions, this communication standard assumes a BS and end devices connected to it, managed by a simple medium access protocol. LoRa supports wider communication bandwidths (125 or 500 kHz) based on the chirp spread spectrum (CSS) technique, which utilizes linear frequency modulation.

Let B denote the total transmission bandwidth at a carrier frequency f_c, and let T represent the symbol duration. The instantaneous frequency $f(t), \quad t \in [0, T]$, of the CSS signal changes linearly with a rate of B/T, wrapping around when it reaches the edges of the transmitted bandwidth:

$$f(t) = f_c - \frac{B}{2} + \left\{ \frac{B}{T} t + \frac{B}{M} i \right\} \bmod B,$$

where $i \in [M]$ is the transmitted message index, and the transmission rate is $\log_2 M$ bits per symbol. Different values of M correspond to different spreading factor (SF) values [3].

Different SFs are supported by LoRa, and signals corresponding to different SFs are almost orthogonal [3]. This property reduces interference between transmissions with different SFs, thereby improving the communication range. The medium access mechanism also allows for downlink transmissions. The LoRa standard supports three different classes of devices.

Devices of the first class (A) behave similarly to Sigfox: they send data as it becomes available, and downlink messages can be sent during *receive windows*, whose timings are configurable. After an uplink transmission, the device waits for the start of a downlink message within two receive windows. If no downlink message is detected, the device enters sleep mode. For this type of device, the sender spends most of its time in sleep mode, enabling long battery life. Class-A functionality is basic and must be supported by all devices.

Devices of the second class (B) extend the downlink functionality by opening periodic receive windows (or ping slots). To manage this functionality, periodic beacons are sent by the network to maintain synchronization.

Finally, devices of the third class (C) enhance the capabilities of Class-A devices by keeping the receive windows open unless transmitting an uplink message. As a result, Class-C devices can receive downlink messages almost any time, offering very low latency for downlink transmissions.

Typical use cases for Class-A devices include sensors that periodically report measurements or send data triggered by an alarm event. Class-B devices are useful for applications requiring measurements on request, while Class-C devices are ideal for remote control mechanisms powered by a continuous power source.

1.1.3 Cellular IoT

Cellular systems are highly attractive for massive deployments due to their wide coverage, straightforward subscription procedures, operation over licensed spectrum with effective interference man-

agement, and robust security protocols.

However, cellular systems employ centralized coordination algorithms (as opposed to distributed CSMA/CA), which are better suited for high data rates among a fixed and relatively small number of active users within the coverage area of a single BS.

Cellular networks did not support machine-type devices prior to Third-Generation Partnership Project (3GPP) Release 12. Earlier releases assumed that any device connecting to the BS would support the full bandwidth of 20 MHz. This large bandwidth requirement was not suitable for achieving the long battery life needed by IoT devices.

Different device types

The 3GPP standards define various categories of devices based on their capabilities. These categories are represented by numbers, where higher numbers indicate devices that support higher peak uplink and downlink rates, a greater number of supported antennas, and so on. However, these categories primarily pertain to devices operated by humans, while the requirements for IoT devices can differ significantly.

3GPP Release 13 introduced the Cat-M category (with "M" denoting MTC). This user equipment category was the first narrowband device type, supporting a bandwidth of 6 resource blocks (1.08 MHz). This new device type required novel approach to control channel design.

Further optimization for MTC devices was achieved in Release 13 with the introduction of the NB (narrowband) device category. Devices in this category reduced the total supported bandwidth to 200 kHz. Additionally, optimizations enabled narrowband transmissions with bandwidths reduced to a single subcarrier, referred to as single-tone transmission. The subcarrier bandwidth for these transmissions is either 15 kHz or 3.75 kHz.

The Fifth-generation New Radio (5G NR) standard introduced broadband IoT, allowing sensing devices to transmit larger amounts of data. In Release 17, the RedCap (Reduced Capabilities) network type was introduced. The RedCap standard aims to support all industrial applications by enabling broadband communication services for machine-type devices. This standard assumes the utilization of up to 20 MHz bandwidth in the frequency range below 6 GHz.

Random access procedures in cellular systems

To initiate a connection with a cellular network, each device must proceed with random access (RA) procedure. The introduction of IoT devices and their massive deployments could overload the control channel. To address this issue, a new RA procedure should be developed, aiming to reduce the overall communication overhead.

As specified in 3GPP TS 138.321, the original RA procedure follows a four-way handshake, as illustrated in Fig. 1.2. This handshake consists of the following phases:

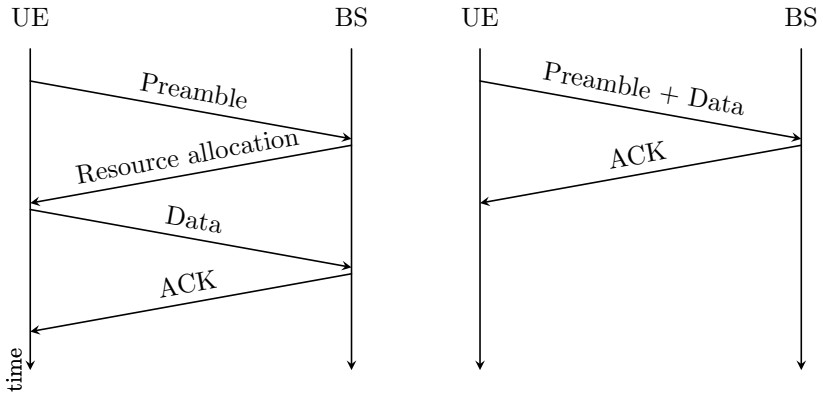

Figure 1.2: Four-step RA (left) and two-step RA procedures specified in 3GPP TS 138.321. A time diagram corresponds to message exchange between user equipment (UE) and base station (BS).

Table 1.1: Simulation parameters for numerical comparison presented in Fig. 1.4

Parameter	Value
Preamble length	2×139 (A1 configuration)
Error-correcting code	$(500, 100)$ LDPC (5G NR base graph 2)
Modulation	Quadrature phase-shift keying (QPSK)
Decoding algorithm	TIN / TIN-SIC
Pilot configuration	Pilot-free
Number of POs	64
Overall frame length	16278 channel uses

1. RA through preamble transmission to identify users.

2. Resource allocation provided by the BS.

3. Data transmission using orthogonal resources assigned to the identified users.

4. Final acknowledgment (ACK).

The described above procedure separates the RA phase from data transmission. Starting with 3GPP Release 16, this procedure was simplified, requiring only a two-way handshake. In this updated procedure, the preamble transmission also announces the resources to be used for data transmission, which follows immediately. Users select preambles from a predefined orthogonal set. Data is then transmitted during specified positions of the physical uplink shared channel occasion (PO)s. If the BS successfully receives the data, it sends an acknowledgment. Otherwise, the traditional four-way handshake is performed. This transmission scheme is depicted in Fig. 1.3 (see the detailed description in [4]).

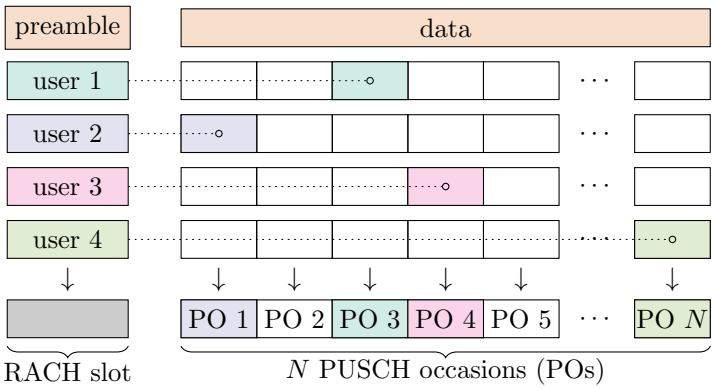

Figure 1.3: Two-step RA procedure with data transmission.

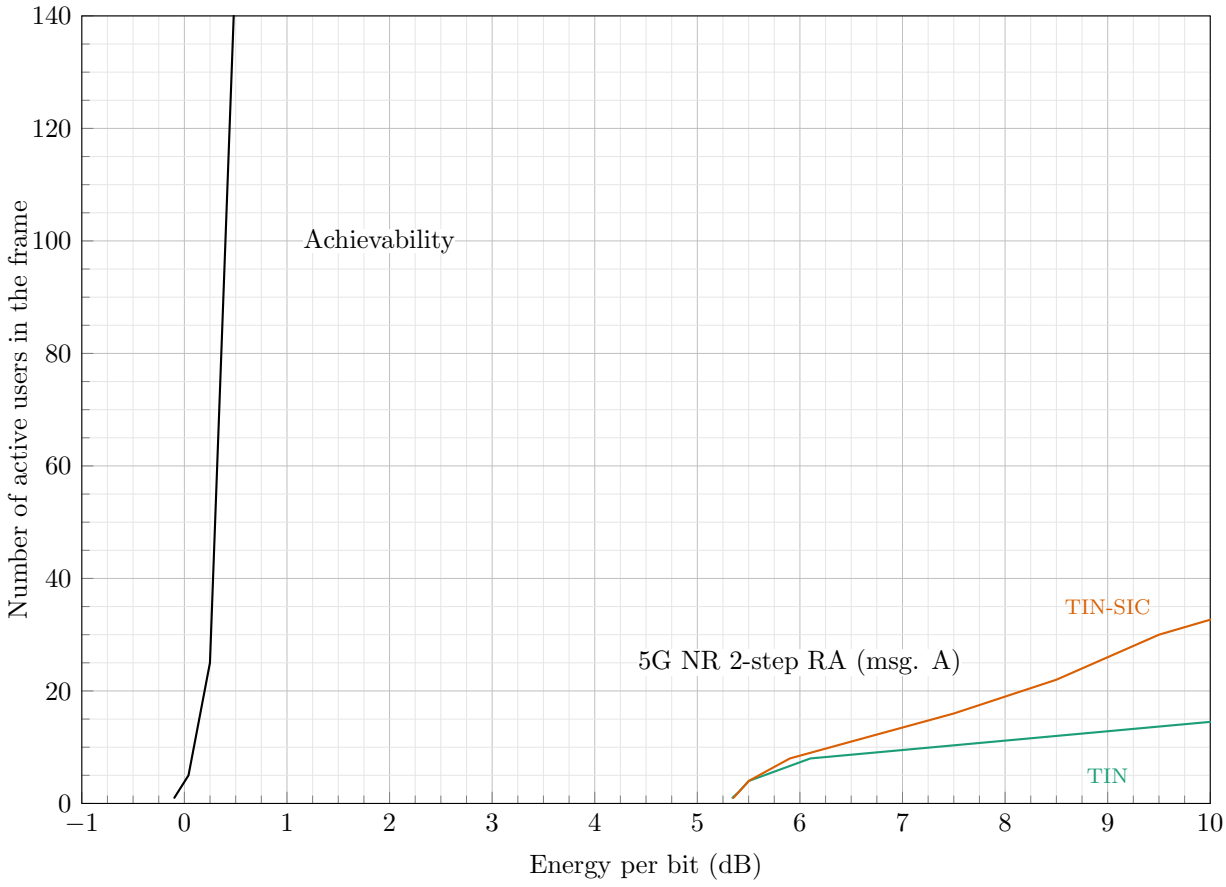

Figure 1.4: Energy efficiency of the proposed 2-step RA procedure in 5G NR versus achievability bound on energy efficiency [5]. AWGN channel model, reference signals are not considered [4].

1.2 Challenges for the next-generation cellular systems

The key challenge for the next generation of radio-access networks is managing the massive number of infrequently communicating sensors. Current solutions are inadequate due to their reliance on centralized resource allocation, which orthogonalizes access for different users. For MTC, this approach is unacceptable as it results in significant control-layer overhead and latency. Therefore, a new communication solution is required.

As evidence, we will first demonstrate the performance of the two-step RA procedure presented above. Following the exposition in [4], we reproduce several numerical results from the referenced manuscript. The objective of the numerical setup is to highlight the significant gap between the energy efficiency of the two-step RA procedure and the achievable energy efficiency in a massive RA scenario (see Theorem 4.1). Energy efficiency is defined as the minimum energy required to transmit a single information bit (or energy per bit) under certain quality of service (QoS) constraints. The exact definition of energy efficiency will be provided in Chapter 3.

The technical details of this numerical experiment are as follows. The preamble dictionary consists of 64 Zadoff-Chu sequences with varying lengths. For short preambles, the length is 139, while for long preambles, it is 839. Each preamble corresponds to a PO. Within a PO, data is transmitted using low-density parity check (LDPC) codes as specified in 5G NR standards. The additive white Gaussian noise (AWGN) channel model is considered, and reference signals are not required in this setup. The system parameters outlined in Table 1.1, and the energy efficiency as a function of the number of simultaneously active users is presented in Fig. 1.4.

Current schemes that are part of existing standards exhibit significantly lower energy efficiency compared to theoretical bounds. Moreover, the two-step RA procedure proposed by 3GPP remains optional. The lack of energy-efficient schemes has motivated many researchers to extensively study this new massive MTC scenario.

In this monograph, we outline the core ideas behind these theoretical bounds and provide a brief overview of the challenges in designing low-complexity schemes. We demonstrate that some of these schemes closely approach the achievability bounds.

In our introductory example in Fig. 1.4, we considered the simple case of a Gaussian channel. However, real wireless channels are affected by multipath propagation, which introduces additional random effects during signal transmission. In the subsequent chapters, we address various challenges posed by real propagation environments.

1.3 Monograph organization

The monograph is organized as follows:

- Chapter 2 explores MAC problems and their formulations, emphasizing the differences between classical MAC scenarios and the unsourced random access (URA) setup.

- Chapter 3 defines the URA problem and introduces per-user probability of error (PUPE), the primary measure of the system's operational quality. It also revisits the definition of energy efficiency (see (3.3)) previously discussed in Fig. 1.4. Additionally, this chapter establishes a connection between the URA problem and the well-known CS problem.
- Chapter 4 investigates the fundamental limits of energy efficiency under PUPE constraints in the Gaussian channel.
- Chapter 5 focuses on low-complexity schemes for the Gaussian channel.
- Chapter 6 examines more realistic channels with fading effects, specifically the quasi-static Rayleigh fading channel. It covers scenarios where the BS is equipped with either a single antenna or multiple antennas.
- Chapter 7 concludes the monograph by highlighting the remaining challenges and open problems.

Chapter 2

Classical multiple-access problem statements

The study of MACs originated in Shannon's paper [6]. Today, various variants of the MAC problem exist. These variants differ in terms of noise models (noiseless, noisy, or worst-case/adversarial), probability of error (zero-error, vanishing, per-user, or joint) the presence of feedback, user activity, and other factors. Following [7, 5], in this chapter, we classify MAC problems based on user activity.

We focus on the vanishing probability of error case and briefly describe the zero-error results in Section 2.1. We consider two main setups from the literature: all-active users (information-theoretic approach) and partial user activity. It is important to distinguish between cases where users utilize different encoders or codebooks (e.g.,[8, 9, 10]) and cases where users share the same encoder. The scenario with all-active users utilizing different encoders is typical for *coordinated multiple access*, i.e., a form of transmission where channel access is managed by a centralized scheduler that assigns resources to the users. The scenario of partial user activity and same encoders is typical for *uncoordinated multiple access* or *random access* [11, 12, 13, 14, 15, 16]. In the latter case, the total number of users is irrelevant, and it is common to assume it to be infinite, which aligns well with the massive access scenario. Note that for the case of partial user activity, we distinguish between the total number of users and the number of active users.

The classification and, consequently, the organization of this chapter is depicted in Fig. 2.1.

We start with zero-error decoding, as presented in Section 2.1. Then, we consider two different cases depending on the user activity. We present the case of all-active users in Section 2.2. Another important branch is partial user activity, as described in Section 2.3. Next, we consider the so-called many-access channel, as presented in Section 2.4. This is an extreme asymptotic case, where the number of active users grows simultaneously with the block length. This setup is the closest to the URA model. Finally, we discuss the main differences and justify the need for a new information-theoretic framework.

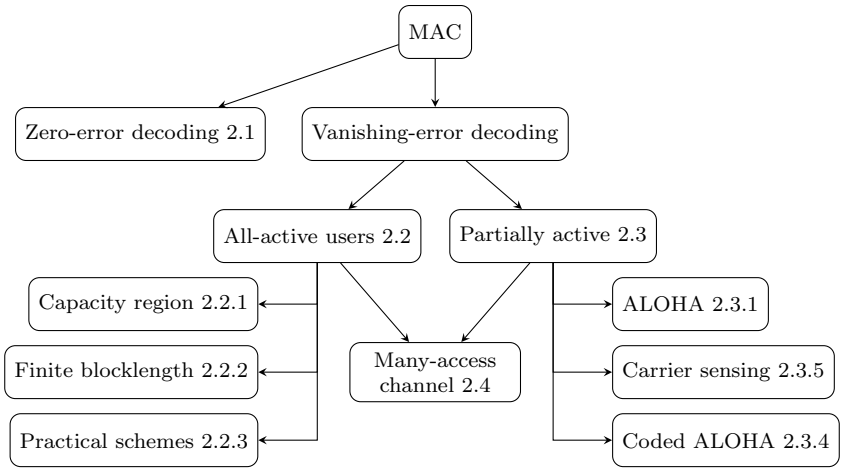

Figure 2.1: Classification of MAC problems based on user activity. References to corresponding section of this chapter are added.

2.1 Zero-error decoding

A significant amount of effort has been devoted to evaluating the zero-error capacity for various natural and relatively simple MAC models. One of the most popular models is the noiseless adder MAC, where users send 0 and 1, and the channel output equals their ordinary sum. This problem was first studied in the context of group testing [17, 18, 19], and was rediscovered nearly twenty years later within MAC theory [20]. Notably, one of the most intriguing topics in MAC theory is determining the zero-error capacity for the simplest case of the two-user BAC, with the current best upper bound being 0.5753, as reported in [21].

The majority of zero-error coding schemes focus on the case of partially active users (T-of-N schemes for activity detection). These solutions can be viewed as Sidon sets [22], specifically for scenarios where the input alphabet consists of binary or q-ary vectors ($\{0,1\}^n$ or $\{0,1,\ldots,q-1\}^n$). Relevant work includes B_2-sequences [23, 24] and subsequent studies [25, 26, 27, 28]. The construction proposed in [25] has been utilized as a component of several low-complexity URA schemes (see Chapter 5).

For an overview of the latest results, we refer the reader to [29].

2.2 MAC with all-active users

Classical information theory provides an exact solution for the case of a constant number of users, an infinite blocklength $n \to \infty$, and a vanishing probability of error, as described in [9, 8].

Consider a fixed number of transmitters, K, sending their messages over a shared channel. The receiver observes a noisy linear combination of the transmitted signals. Assuming a discrete-time

memoryless channel:

$$\mathbf{y} = \sum_{k=1}^{K} \mathbf{x}_k + \mathbf{z}, \quad \mathbf{z} \sim \mathcal{N}(0, \mathbf{I}_n), \quad \|\mathbf{x}_k\|^2 \leq nP_k \tag{2.1}$$

where P_k is the transmit power of the k-th user, and the terms $\mathbf{x}_k$ and $\mathbf{z}$[1] represent the transmitted signal of the k-th user and the noise, respectively. This setup is known as the GMAC.

2.2.1 Capacity region

The most straightforward way to manage K simultaneously active users in a shared channel is to allocate them orthogonal resources. Orthogonality can be achieved in the time, frequency, code, or spatial domains. The corresponding strategies are known as time-division multiple-access (TDMA), frequency division multiple-access (FDMA), code division multiple-access (CDMA), and spatial multiplexing, respectively.

As we see in Section 2.2.3, some of these strategies may, in certain cases, achieve optimal performance. Nevertheless, in many scenarios, an orthogonalization strategy is suboptimal.

In 1973, Ahlswede [8] and Liao [9] derived a coding theorem for the MAC. These theorems state that given a K-user MAC channel with transmit powers $P_1, \ldots, P_K$, any tuple of rates $R_1, \ldots, R_K$ is achievable if

$$\sum_{k \in \mathcal{K}} R_k \leq C\left(\sum_{k \in \mathcal{K}} P_k\right) \quad \forall \mathcal{K} \subseteq [K], \tag{2.2}$$

where $C(P)$ is Shannon's capacity, given by

$$C(P) = \frac{1}{2}\log_2(1+P). \tag{2.3}$$

An illustration for the case $K = 2$ is provided in Fig. 2.2. It is worth noting that user 1 can increase their data rate by increasing P_1 without affecting user 2's data rate until point B is reached. The line A–B of the resulting capacity region corresponds to the condition

$$R_1 + R_2 = C(P_1 + P_2), \tag{2.4}$$

which represents a point-to-point communication channel with a total transmit power of $P_1 + P_2$ and is referred to as the *dominant facet*.

2.2.2 Finite blocklength

In this section, we examine a normal approximation introduced in [30, 31] and applied to the capacity region of the K-user MAC, as given in (2.2). The core concept of the normal approximation is *channel dispersion*. For an AWGN channel with transmit power P, the dispersion is given by [30, Theorem 54]:

[1]For simplicity, we consider real degrees of freedom.

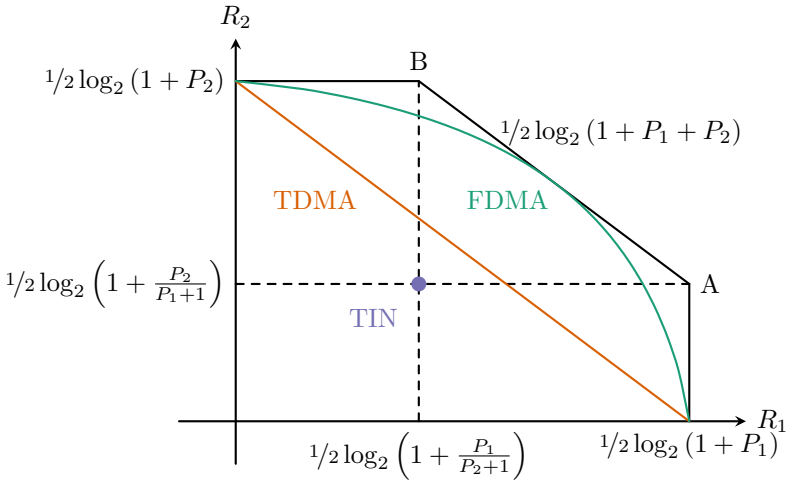

Figure 2.2: A 2-user MAC for an AWGN channel with real degrees of freedom. The information-theoretic capacity region is represented by a black polygon passing through points A and B. The capacity region of the TDMA scheme without power control is shown by an orange line, while the FDMA capacity region is depicted by a green line. The capacity region of TIN is a single point, denoted by a blue dot.

$$V(P) = \frac{\log^2 e}{2} \cdot \frac{P(P+2)}{(1+P)^2}. \tag{2.5}$$

The following theorem [32, Theorem 3] considers the joint probability of error ε for a K-user MAC in the finite blocklength (FBL) regime. The authors consider the codebook generated from the uniform distribution on the n-dimensional sphere, which is typically referred to as a *power shell* (see Appendix B.2.2).

Theorem 2.1. *A second-order achievable rate region with power shell inputs for the K-user GMAC with a power constraint P_u for each user $u \in [K]$ is the set of all rate tuples $(R_1, \ldots, R_K)$ satisfying*

$$R_{\mathcal{K}} < C(P_{\mathcal{K}}) - \sqrt{\frac{V(P_{\mathcal{K}}) + V_c(P_{\mathcal{K}})}{n}} Q^{-1}(\lambda_{\mathcal{K}} \varepsilon) + \mathcal{O}\left(\frac{1}{n}\right), \tag{2.6}$$

for all subsets $\mathcal{K} \subseteq [K]$ and any choice of non-negative coefficients $\lambda_{\mathcal{K}}$ that sum to one:

$$\sum_{\mathcal{K} \subseteq [K]} \lambda_{\mathcal{K}} = 1.$$

Here, $P_{\mathcal{K}}$ is the total transmit power of users in subset $\mathcal{K}$:

$$P_{\mathcal{K}} = \sum_{u \in \mathcal{K}} P_u,$$

$V(\cdot)$ is the channel dispersion defined in (2.5), and $V_c(\cdot)$ is the channel cross-dispersion given by:

$$V_c(P_{\mathcal{K}}) = \frac{\log^2 e}{2} \cdot \frac{(P_{\mathcal{K}})^2 - \sum_{u \in \mathcal{K}} P_u^2}{(1 + P_{\mathcal{K}})^2}.$$

To gain some intuition about this theorem, let us consider the sum-rate case, which corresponds to the dominant facet, $\mathcal{K} = [K]$, with equal power allocation $P_u = P$ for all u. In this case, the sum-rate is given by:

$$R^\star = C\left(KP\right) - \sqrt{\frac{V\left(KP\right) + V_c\left(K,P\right)}{n}} Q^{-1}\left(\varepsilon\right) + \mathcal{O}\left(\frac{1}{n}\right),$$

where

$$V_c\left(K,P\right) = \frac{\log_2 e}{2} \cdot \frac{K\left(K-1\right)P^2}{\left(1+KP\right)^2}.$$

One can observe an increased dispersion value over the dominant facet.

2.2.3 Practical schemes

In this section, we consider practical coding schemes that achieve points on (or the entirety of) the dominant facet of the capacity region and may be implemented with low computational complexity through appropriate single-user code selection.

Orthogonal schemes

Let us begin with orthogonal schemes by considering TDMA, FDMA, and the orthogonal case of CDMA. To proceed with our discussion, recall the capacity region presented in Fig. 2.2.

TDMA. Given P_i as the transmission power of the i-th user ($i = 1, 2$), the corresponding capacities of the point-to-point channels are $C\left(P_1\right)$ and $C\left(P_2\right)$. In TDMA, orthogonalization is performed in the time domain. Let α be the fraction of resources allocated to user 1, meaning that $1-\alpha$ is the fraction allocated to user 2. For a MAC with TDMA, the set of achievable sum rates is given by:

$$\alpha C\left(P_1\right) + \left(1-\alpha\right) C\left(P_2\right), \tag{2.7}$$

where $\alpha C\left(P_1\right)$ and $\left(1-\alpha\right) C\left(P_2\right)$ form a pair of achievable individual transmission rates. Equation (2.7) represents a straight line, as illustrated in Fig. 2.2. In terms of the capacity region, this scheme is strictly suboptimal for all $\alpha \in (0,1)$. Suboptimality means that the achievable rate pairs lie strictly inside the capacity region.

FDMA. In the TDMA strategy, each transmission occupies the entire bandwidth allocated for the shared channel. In contrast, FDMA allows for better utilization of transmit power spectral density by occupying only a fraction of the allocated bandwidth. Given a bandwidth fraction α, the channel capacity for a fraction α of the total frequency resources is:

$$C_f\left(P,\alpha\right) = \frac{\alpha}{2}\log_2\left(1+\frac{P}{\alpha}\right).$$

For the previously discussed two-user MAC with the FDMA strategy, the achievable individual transmission rate pairs are:

$$R_1 = C_f\left(P_1, \alpha\right), \quad R_2 = C_f\left(P_2, 1-\alpha\right),$$

which corresponds to the green curve shown in Fig. 2.2.

This transmission scheme becomes optimal (touching the dominant facet of the capacity region) when:

$$\alpha^\star = \frac{P_1}{P_1 + P_2}, \tag{2.8}$$

due to the increased transmit power spectral density, compared to TDMA without power control as considered previously.

Remark 2.1 (TDMA with power control)**.** *In the previous discussion, we considered TDMA without power control. According to* (2.1)*, the energy transmitted by the first user is limited to* $\alpha n P_1$*, and by the second user to* $(1-\alpha)\, nP_2$*. Hence, to fulfill the energy constraint from* (2.1)*, and given a fraction of allocated resources equal to* α*, the first user can transmit with an instantaneous power of* P_1/α*, and the second user with* $P_2/1-\alpha$*. As a result, the TDMA scheme with power control (the term described in [33]) has the same capacity region as FDMA.*

CDMA. To achieve orthogonality in the *code* domain, the transmitted signal can be *spread* using a spreading sequence $\mathbf{a}^{(i)}$. We refer the reader to (5.23) and Section 5.4.1 for details. To detect a user's signal from the received sequence, the receiver employs a matched filter (MF) matched to the spreading sequence $\mathbf{a}^{(i)}$ assigned to the i-th user. Given orthogonal spreading sequences, i.e., $\left(\mathbf{a}^{(i)}\right)^T \cdot \mathbf{a}^{(j)} = 0$, applying a MF converts the two-user MAC channel into a TDMA channel, as described above, with $\alpha = 1/2$[2]. Thus, the capacity region of CDMA with orthogonal spreading sequences reduces to a single point on the TDMA capacity region. In the case of $P_1 = P_2$, and following (2.8), this point lies on the dominant facet of the capacity region. Note that the maximum number of orthogonal sequences is limited by their length, meaning that the maximum number of users that can be orthogonalized equals the length of the spreading sequence.

Additionally, the spreading technique also helps to perform time-domain channel equalization in the presence of multipath propagation with delay spread, using a RAKE receiver [34].

Treat interference as noise

To recover all K transmitted codewords from the received signal (2.1) in a non-orthogonal regime, a joint decoder is required. However, the complexity of joint decoding can be prohibitively high for practical applications. In practice, particularly in a non-symmetrical MAC – where the received powers P_i are unequal – the user with the strongest received power is more likely to be successfully

[2]This can be easily verified by expressing the spread signal in the form (5.25) and (5.24), keeping in mind that the K spreading sequences form a $K \times K$ orthogonal matrix. Then, a simple basis change transforms (5.25) into a TDMA channel (identity spreading matrix) without altering the noise properties.

decoded by a *single-user* decoder, even when treating other users' signals as noise. This approach is known as treat interference as noise (TIN).

Applying an information-theoretic analysis to TIN in the two-user MAC, the noise and interference[3] power experienced by the first user's TIN decoder is $1 + P_2$, resulting in a maximum achievable communication rate of R_1:

$$R_1 = \frac{1}{2}\log_2\left(1 + \frac{P_1}{1+P_2}\right), \quad R_2 = \frac{1}{2}\log_2\left(1 + \frac{P_2}{1+P_1}\right),$$

where R_2 is obtained similarly. Note that in this case, the relative received power of each user does not affect the result. This rate pair is represented by the blue dot in Fig. 2.2. This scenario was previously discussed in the context of CDMA with pseudo-orthogonal spreading sequences.

Notably, depending on the values of P_1 and P_2, the TIN point can lie either inside or outside the capacity region of the TDMA scheme. The reference case in Fig. 2.2 corresponds to equal power allocation.

Now, let us analyze a K-user symmetric MAC and consider the achievable sum rate as $K \to \infty$ in the case where a TIN strategy is applied to the codewords of each user:

$$\lim_{K\to\infty} \frac{K}{2}\log_2\left(1 + \frac{P}{(K-1)\,P+1}\right) \approx \frac{\log_2 e}{2}.$$

This sum rate is limited and falls far below the dominant facet of the capacity region, where the maximum sum rate is equal to $C\,(KP)$, which is unbounded.

Successive interference cancellation

Can the previously discussed TIN strategy be improved without significantly increasing computational complexity?

After performing the TIN step and applying capacity-achieving codes, the receiver successfully decodes a codeword. This codeword can then be subtracted from the received sequence. For instance, if user 1 is decoded using the TIN approach, the residual signal – after subtracting the first user's codeword – will consist only of user 2's signal and noise. This effectively reduces the problem to a single-user decoding scenario in the AWGN channel, where user 2 can achieve the rate $C(P_2)$. The corresponding rate pair is given by:

$$R_1 = \frac{1}{2}\log_2\left(1 + \frac{P_1}{1+P_2}\right), \quad R_2 = C\,(P_2)\,.$$

This rate pair corresponds to point B in the capacity region shown in Fig. 2.2. Similarly, point A can be achieved by reversing the order in which the TIN step and subtraction are applied.

[3]Under the assumption of random coding with i.i.d. Gaussian codebooks, the sum of noise and interference will also follow a Gaussian distribution.

For the K-user MAC, the TIN step followed by subtraction can be repeated multiple times. This method is known as successive interference cancellation (SIC). After each SIC step, the dimensionality of the problem is reduced.

In the two-user MAC example, applying SIC allows us to achieve points A and B on the dominant facet of the capacity region. To reach any point along the line segment A–B, a TDMA strategy can be employed, alternating between different decoding orders of the TIN step followed by SIC. As a result, we obtain a low-complexity scheme capable of achieving any point on the dominant facet of the capacity region.

Non-orthogonal CDMA

Orthogonal sequences allow us to use a single-user decoder to solve a multiple-access problem. However, the spreading sequences may not always be orthogonal. In this case, the system load may be higher compared to orthogonal spreading, at the cost of additional interference. In the presence of interference, a multi-user detector (MUD) may outperform a single-user one. The problem of MUD in CDMA has been studied in [35, 36]. In [37], the idea of blind detection was proposed – a scenario in which the receiver knows the target spreading sequence but does not know the interfering spreading sequences.

Regarding the relationship between CDMA and the capacity region, the authors of [38] proposed the design of spreading sequences for synchronous overloaded CDMA (where the number of active users exceeds the length of the spreading sequence) in an equal-power setting and estimated the achievable sum rate. The unequal-power case was considered in [39]. Finally, MUD combined with SIC, as proposed in [40], achieved full capacity region.

For randomly generated sequences, the authors of [41, 42] analyzed capacity for random binary and spherical sequences under joint decoding and linear MUDs. Additionally, an iterative decoding scheme for turbo codes with spreading was proposed in [43]. Lastly, let us highlight the work of Montanari and Tse in [44], where they proposed message-passing algorithms to design near-optimal MUDs. This approach later evolved into low-density signature (LDS) CDMA [45] and sparse-coded multiple-access (SCMA), as proposed in [10].

Rate-splitting multiple-access

In the previous discussion, we found that the dominant facet of the capacity region can be achieved using the TIN-SIC strategy with time-sharing. However, is there a simpler strategy that can be achieved using only single-user codes? As proposed in [46], this can be accomplished through RSMA.

To explain the main idea, let us redefine Shannon's capacity (2.3) by explicitly considering the noise power σ^2:

$$C_\sigma\left(P, \sigma^2\right) = \frac{1}{2}\log_2\left(1 + \frac{P}{\sigma^2}\right)$$

As we observed previously, the maximum sum-rate R_Σ of the two-user MAC is achieved on the dominant facet, resulting in

$$R_\Sigma = C_\sigma\left(P_1 + P_2, \sigma^2\right) = C_\sigma\left(P_2, \sigma^2\right) + C_\sigma\left(P_1, P_2 + \sigma^2\right), \tag{2.9}$$

where the last equation is referred to as the *chain rule*. The physical meaning of this chain rule is as follows: In the first step, we apply a TIN step, resulting in $R_1 = C_\sigma\left(P_1, P_2 + \sigma^2\right)$, followed by a SIC step, yielding $R_2 = C_\sigma\left(P_2, \sigma^2\right)$. This strategy allows us to achieve one corner point of the capacity region pentagon.

An interesting observation is that the chain rule (2.9) allows us to split a high-rate code into two codes with lower rates. Then, the TIN-SIC strategy can be used to decode two superimposed codes. As proposed in [46], any point in the two-user MAC capacity region can be achieved by splitting the higher-rate code of the user with the larger received power into two superimposed codes. Then, a TIN-SIC strategy can be applied to decode all three codes.

To illustrate this, let us consider the case where $P_1 > P_2$, and let us choose some $0 < \delta < P_1$. Next, we define

$$\begin{aligned} p_1 &= \delta, \\ p_2 &= P_2, \\ p_3 &= P_1 - \delta. \end{aligned}$$

Then, let $r_i = C_\sigma\left(p_i, \sigma^2\right)$ be the rate achievable by the i-th code. The sum rate of the three codes equals

$$r_1 + r_2 + r_3 = C_\sigma\left(p_1 + p_2 + p_3, \sigma^2\right) = C_\sigma\left(P_1 + P_2, \sigma^2\right) = R_\Sigma.$$

Note also that for any δ, there is a unique r_1 value. Hence, any point on the dominant facet can be achieved.

Furthermore, the result above was generalized to the case of a K-user MAC in [46]. It was shown that a total of $2K-1$ superimposed codes are required to achieve an arbitrary point on the dominant facet.

Schemes based on polar codes

Several works present theoretical results on the achievability of the capacity region using polar coding. In [47], it was shown that two users with a uniform data source, using Arıkan's construction, may achieve a point on the dominant facet of the capacity region. The resulting MAC polarizes into five extreme channels, with the corresponding capacity regions illustrated in Fig. 2.3. Next, it was shown that a polar coding technique allows the achievement of any point in the capacity region of a two-user MAC [48].[4] Finally, a method for improving the performance of two-user MAC polar coding through list decoding was described in [49].

[4] A Slepian-Wolf source coding problem was considered, which is a problem dual to the two-user MAC. The solution is based on monotone chain rule expansions.

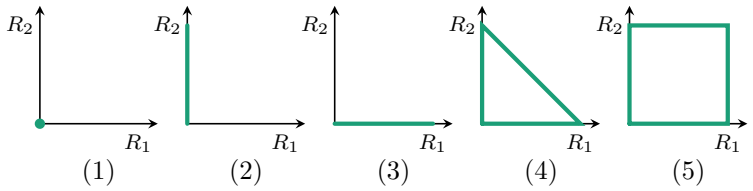

Figure 2.3: Five extreme channels for two-user MAC polarization [47]. The figure illustrates: (1) a completely useless channel, where the capacity region is a single point at $(0, 0)$; (2, 3) two channels that are useless for one of the two users; (4) a MAC with a TDMA-like capacity region; and (5) a channel where both users can transmit in parallel.

Theoretical results for two-user MAC polarization were further extended to the K-user MAC case in [50]. Moreover, the achievability of the entire uniform rate region[5] for the K-user MAC was proposed in [51].

At the same time, there are no efficient decoding or optimization methods for the FBL case.

Schemes based on LDPC codes

Finally, let us consider the capacity-approaching scheme based on LDPC codes presented in [52]. The authors employed a message passing algorithm (MPA) to iteratively decode both codes, and each LDPC code was optimized for degree distributions using density evolution (DE). Their results showed that the resulting decoding threshold is only 0.18 dB away from Shannon's limit.

Moreover, another capacity-approaching scheme is based on spatially coupled LDPC codes [53], whose authors validated their claim using DE.

2.3 MAC with partial user activity

Now, let us consider RA protocols. In this scenario, users generate messages at random time instances and attempt to transmit them over a shared channel.

In 1970, Abramson proposed the first random access protocol, which is also the simplest, known as ALOHA [54]. The core idea is straightforward: whenever a new packet arrives, the transmitter immediately sends it. If two or more transmissions occur simultaneously, the signals interfere, making them undecodable and resulting in a collision. This original protocol is known as pure ALOHA.

In this non-synchronized system, a collision occurs even when transmitted messages partially overlap. To mitigate this, time synchronization was later introduced [11], giving rise to what is now

[5]For a MAC, the uniform rate region is the achievable region when the input distributions are uniform.

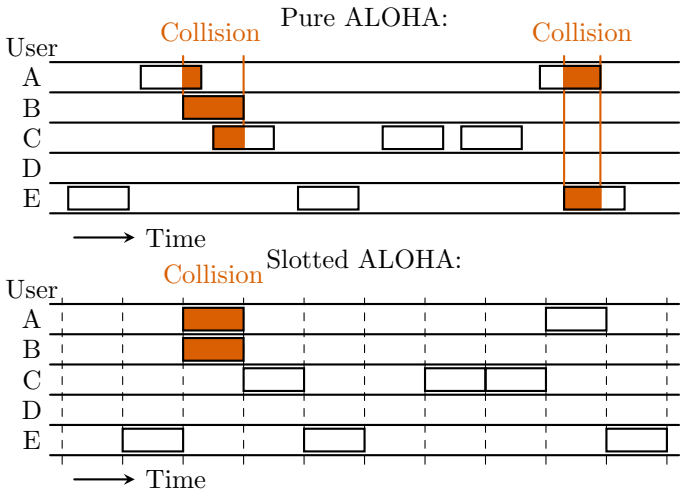

Figure 2.4: Pure (top) and slotted (bottom) ALOHA. Packets are generated at the same time instances, resulting in five packet collisions during transmission in the case of pure ALOHA. In SA, transmission is allowed only in the nearest slot after a packet appears. As a result, the number of collided packets is reduced to two. Slot boundaries are marked by dashed lines.

known as SA. An example illustrating how synchronization reduces the number of collisions is shown in Fig. 2.4. Thus, in the subsequent analysis, we consider only the SA system.

Below are some key assumptions of the SA model. Based on these assumptions, various throughput optimization techniques have been developed.

2.3.1 Slotted ALOHA model assumptions

As we demonstrate later, the use of collision resolution algorithms can significantly enhance the performance of SA. We now outline the assumptions for SA used in the study of collision resolution algorithms.

1. Slotted transmission system: Each message has a fixed duration and is transmitted within a unit slot length.
2. Collision: Occurs when two or more simultaneous transmissions happen within a slot, and successful reception is impossible.
3. Immediate feedback: The system provides feedback indicating whether the slot was empty (no transmission), a success (a single packet was transmitted), or a collision (two or more packets were transmitted).

To work with a mathematical model of ALOHA, we make the following additional assumptions:

1. An infinite number of transmitters.

2. The arrival rate of incoming messages follows a Poisson point process with intensity λ. This approximation is reasonable given a large number of independent sources.

Remark 2.2 (Worst-case performance estimation in terms of collisions)**.** *The assumption of an infinite number of transmitters combined with a finite message rate allows for estimating worst-case performance in terms of collisions. Indeed, if the number of transmitters were finite, a message arriving at a transmitting station would be queued rather than causing a collision.*

Remark 2.3 (Correlated transmissions)**.** *We should also note that correlated transmissions (which are typical in sensor networks [55]) may lead to significantly worse performance than a Poisson model assuming independent transmissions.*

Remark 2.4 (Collision model considerations)**.** *The ALOHA model does not account for physical-layer coding techniques. The previously mentioned SIC, which is a physical-layer coding technique, could significantly alter system performance. Nevertheless, the above assumptions provide a simplified setup for studying collision resolution algorithms.*

Remark 2.5 (Collisions and capture effect)**.** *In the case of real signal propagation models and additive white Gaussian noise (AWGN), a phenomenon known as the* capture effect *occurs [56]. When two messages collide and arrive with different received power levels, the TIN decoding process may allow for partial message recovery, where at least the user with the higher signal-to-interference-plus-noise ratio (SINR) value is successfully decoded.*

2.3.2 Throughput and stability of slotted ALOHA

Given a unit slot length, it is reasonable to define throughput as the average number of successfully delivered packets per slot. For an SA system without any collision resolution algorithms or retransmissions, the throughput depends on the input data rate, λ, as $\lambda e^{-\lambda}$, which reaches its maximum at $\lambda = 1$, yielding $1/e \approx 0.3679$ (following the Poisson traffic model proposed in Section 2.3.1). Thus, only about 36.8% of slots contain a non-collided transmission, while the remaining slots are either empty or experience collisions.

To define stability, let us introduce a retransmission mechanism. When a transmitter has channel feedback, collisions can be detected immediately. In this case, the packet becomes *backlogged* for future retransmission. Recall that the number of transmitters is infinite, and a backlogged packet is not affected by subsequent messages. Different retransmission algorithms may influence the system's throughput.

Let us first specify the simplest retransmission policy: any backlogged packet is transmitted with probability p in the subsequent slot, independently of past slots and other packets. The value of p can be adjusted based on collision statistics. Small values of p lead to high delays, whereas large values result in excessive collisions.

This system can be represented as a Markov chain, where the state is the number of backlogged messages, k. Given λ and p, the set of transition probabilities can be easily derived [7]. It turns out that the resulting Markov chain is non-ergodic [57].

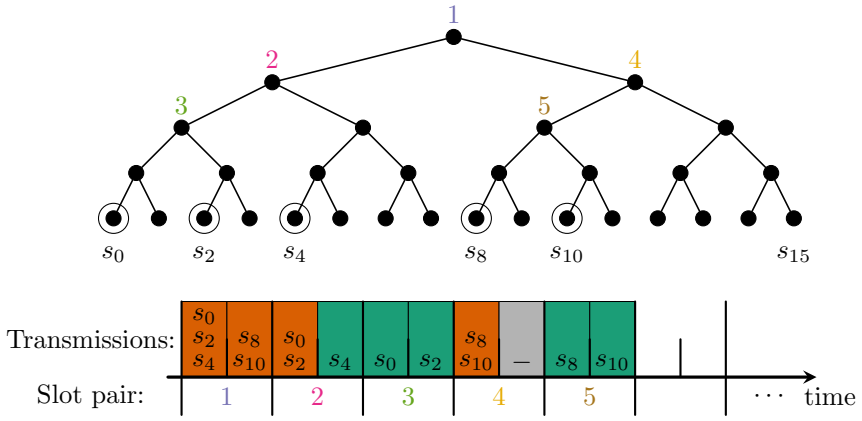

Figure 2.5: Example of the tree-split algorithm, reproduced from [12]. (Top) Alignment of all stations into a tree. Active stations s_0, s_2, s_4, s_8, and s_{10} are marked by circles. (Bottom) Sequence of transmissions during the collision resolution algorithm. Slots with collisions are marked by orange rectangles with active station indices, collision-free slots are represented by green rectangles, and empty slots are shown in grey. The slot pair index and the corresponding root node are indicated by colored numbers.

The *drift* of the system is defined as the expected increment $D_k(\lambda, p)$ in the number of backlogged packets for each state k during a slot duration. When $D_k > 0$, the number of packets scheduled for retransmission is expected to increase. By analyzing D_k for different values of k, we can identify regimes where the system becomes unstable, meaning the number of backlogged packets grows without bound. Consequently, the system's throughput approaches zero.

The problem of ALOHA stabilization was first studied in [58]. The authors examined a class of algorithms that control p based solely on collision feedback. The resulting control algorithm was stable for $\lambda < 1/e$. Note that for $\lambda > 1/e$, the drift value is always positive [7]. Hence, the throughput of the SA system is upper-limited by $1/e$, and collisions remain a crucial throughput-limiting factor. In the next section, we describe advanced collision resolution algorithms that may increase the throughput of the SA system.

2.3.3 Lower and upper bounds on throughput in slotted ALOHA

To achieve higher throughput and lower delay, the following modifications can be made to the retransmission strategy outlined above. First, the immediate transmission of a newly arrived packet can be delayed if previous collisions remain unresolved. Second, more advanced collision resolution policies can be employed. In this section, we briefly review various explored methods for throughput optimization.

We begin with the tree-split algorithm proposed in [12, 59, 60]. The idea of this algorithm is to arrange all stations in a binary tree, as shown in Fig. 2.5.[6] When a collision occurs, the system enters a collision resolution regime. In this regime, all subsequent slots are grouped into pairs, and

[6]For simplicity, we consider the case where the number of stations is a power of two.

a tree-traversing algorithm (e.g., depth-first search) is initiated. The goal of this algorithm is to mark all leaf nodes (stations) as visited (i.e., collisions resolved).

To achieve this, the algorithm starts at the root node. Stations from the left and right branches of the root transmit in the first and second slots, respectively. If a collision-free transmission or an empty slot occurs, all nodes connected to the corresponding branch are marked as visited (resolved). The tree-traversing algorithm continues splitting until all nodes are marked as visited.

Subsequent modifications of tree-based algorithms, such as more than two groups at the initial collision resolution step, resulted in a throughput of 0.46 [61], which remains stable for smaller input rates.

The second family of algorithms is based on the first-come, first-serve (FCFS) algorithms, where each arriving packet is marked with a timestamp, and packets whose timestamps are within a certain window are allowed to transmit [62]. The window edges move in accordance with channel feedback.

The state-of-the-art algorithm for the classical SA setup was proposed in [13], achieving a throughput of up to 0.4877. Finally, the state-of-the-art upper bound for classical SA throughput is given in [63].

$$0.4877 \overset{[13]}{\leq} R \overset{[63]}{\leq} 0.5683. \tag{2.10}$$

Note that by leveraging the SIC technique, a throughput of 0.693 can be achieved using the tree-based algorithm, as reported in [64]. However, the use of *coded* SA, as presented in the next section, may further improve the system's throughput.

2.3.4 Coded slotted ALOHA

Let us introduce time diversity into the ALOHA system: a single transmission is followed by the transmission of a *replica* of the original message [14], which is sent in a randomly selected slot within a *frame*. In this setup, the frame serves to separate different bursts and limit the distance between two replicas. This approach is known as contention resolution diversity slotted ALOHA (CRDSA). If one of the two replicas does not experience a collision, the position of the second replica can be determined from the successfully decoded message, allowing the second replica to be subtracted from another slot, potentially resolving additional collisions – a process known as the SIC step.

This idea was further improved in [15], where an arbitrary number of replicas was allowed and chosen randomly from a specific distribution. All replicas are transmitted in randomly selected slots within a frame. The parameters of this distribution were optimized, significantly improving the resulting throughput, which can reach approximately 0.97 for large frames and close to 0.8 in practical implementations – a significant improvement over ordinary SA, as shown in (2.10). This method is known as IRSA. An example of IRSA is shown in Fig. 5.1 and Fig. 5.2 and described in Section 5.1.

The transmission and decoding processes in the IRSA system can be described using a bipartite

graph, known as a Tanner graph. The vertex set of the graph consists of user nodes and slot nodes. An edge exists between a user node and a slot node if a transmission occurs from the corresponding user in the corresponding slot. The random family of these bipartite graphs can be characterized by a corresponding degree distribution. Hence, density evolution [65], similar to LDPC optimization, can be applied here.

2.3.5 Carrier sensing

Carrier-sense multiple-access (CSMA) was proposed in [66]. The main idea of carrier sensing is to quickly detect when the slot is idle and immediately proceed to the next slot. This idle-detection time depends on the practical implementation of the system and may include hardware switching time, propagation delays, etc. There are two differences compared to the SA system described previously. The first is that an idle slot has a duration of $\alpha \ll 1$. The second is that a packet arriving during another transmission is immediately backlogged, and its transmission starts with probability p after each subsequent idle slot. Packets arriving during an idle slot are transmitted in the next slot, as usual.

The analysis of the system can be performed via a Markov chain [7]. The drift value D_k can be expressed as follows:

$$D_k = \lambda\alpha + \lambda\left(1 - e^{-\lambda\alpha}(1-p)^k\right) - \left(\lambda\alpha + \frac{pk}{1-p}\right)e^{-\lambda\alpha}(1-p)^k .$$

For $\lambda(1+\alpha) < 1$, the value of D_k reaches its minimum at

$$p^\star = \frac{1-\lambda(1+\alpha)}{k-\lambda(1+\alpha)}.$$

D_k is negative for all values of k (with $p^\star$ substituted) as soon as

$$\lambda(1+\alpha) < e^{-1+\lambda}.$$

Applying a power series expansion over $1-\lambda$, the throughput approaches one for small α:

$$\lambda < 1 - \sqrt{2\alpha}.$$

Moreover, within CSMA, the difference between slotted and pure ALOHA disappears, with the maximum throughput of pure ALOHA with CSMA being equal to $1 - 2\sqrt{\alpha}$ [7].

2.4 Many-access channels

Finally, let us consider the information-theoretic approach, in which the number of users is allowed to grow with the blocklength. This line of research is the closest to the URA direction. Henceforth, we refer to this paradigm as *many-user information theory* and to the corresponding channels as *many-access channels*.

We present the many-access channel results in three steps, following their evolution in the literature. In the first step (see Section 2.4.1), the asymptotics over blocklength $n \to \infty$ were considered first, followed by the limit of the user count $K \to \infty$. Next, in Section 2.4.2, we consider the case where both K and n tend to infinity simultaneously, providing additional insights into the relationship between K and n. However, in this regime, each user expends infinite energy to send a message with a fixed number of information bits, leading to an infinite E_b/N_0. Finally, this issue is addressed in Section 2.4.3, where the main conclusion is that achieving zero probability of joint decoding error becomes infeasible in this case.

2.4.1 Noiseless and noisy many-access BAC

We begin with the paper [20], assuming that all K users in the system are active. The K messages emanating from the K sources are encoded independently using K binary block codes[7]

$$\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_K, \quad \mathcal{C}_i \subseteq \{0,1\}^n, \quad |\mathcal{C}_i| = M_i$$

of the same length n.

The goal is to maximize the sum rate

$$R_\Sigma(K) = \sum_{i=1}^{K} R_i, \quad R_i = \frac{\log_2 M_i}{n}, i \in [K].$$

The authors consider a noiseless BAC, where the number of users increases with the blocklength. The i-th user sends an integer $X_i \in \{0,1\}$, and the output symbol Y is the real sum of the K inputs, i.e., the channel output is

$$Y = \sum\nolimits_{i=1}^{K} X_i \in \{0,1,\ldots,K\}.$$

The first result is the capacity region for such a channel. Here, we provide only the sum rate inequality (dominant facet inequality):

$$R_\Sigma(K) \le C_\Sigma(K) = \max_{P_{X_1} \otimes \ldots \otimes P_{X_K}} I(X_1, \ldots, X_K; Y) = \sum_{i=0}^{K} \frac{\binom{K}{i}}{2^K} \log_2 \frac{2^K}{\binom{K}{i}},$$

where the maximization is performed over independent distributions P_{X_i}, and the maximum is achieved for $P_{X_i} = \mathrm{Bern}(1/2)$, $i \in [K]$.

The asymptotics can be summarized as follows.

Theorem 2.2. *The maximal achievable rate of a K-user uniquely decodable code for the binary K-user noiseless BAC is asymptotically equal to $\frac{1}{2}\log_2(\pi e K/2)$ with increasing K.*

Finally, the authors proposed uniquely decodable codes for the many-access BAC. The key ingredient of the construction is the difference matrix $\boldsymbol{D}$ of size $K \times n$ over $\{0,1,-1\}$, where the rows of $\boldsymbol{D}$ are linearly independent over $\{0,1,-1\}$. Given $\boldsymbol{D}$, it is possible to construct K user codes $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_K$ that are uniquely decodable[8]. Each user code consists of two codewords, i.e., $M_i = 2$,

[7] Here, we consider the case of different encoders or codebooks.

[8] Note that in this case, we have zero-error decoding.

$i \in [K]$.

The most significant result is that the sum rate of such codes,

$$R_\Sigma(K) = \frac{K}{n} = 1 + \frac{1}{2}\log_2 n$$

achieves $R_\Sigma(K)$ asymptotically, i.e.,

$$\lim_{K\to\infty} \frac{R_\Sigma(K)}{C_\Sigma(K)} = 1.$$

The authors also generalized the construction for noisy channels. In this case, the sum rate is given by

$$R_\Sigma(K) = \frac{K}{n} = 1 + \frac{1}{2}\log_2 \frac{n}{d_{\min}},$$

where $d_{\min}$ is the minimal L_1 distance of the K-user code, defined as

$$d_{\min} = \min \left\| \sum_{i=1}^{K} \boldsymbol{c}_i - \sum_{i=1}^{K} \boldsymbol{c}'_i \right\|_1,$$

where the minimum is taken over all pairs $(\boldsymbol{c}_1, \ldots, \boldsymbol{c}_K) \neq (\boldsymbol{c}'_1, \ldots, \boldsymbol{c}'_K)$ with $\boldsymbol{c}_i, \boldsymbol{c}'_i \in \mathcal{C}_i$, $i \in [K]$.

2.4.2 Many-access GMAC

Now, let us proceed with the inspirational papers of X. Chen, D. Guo, and their co-authors on many-access channels [67, 68, 69]. The authors proposed a new paradigm, referred to as many-user information theory, in which the number of users is allowed to grow with the blocklength. Notably, this is one of the first works to address both the partial activity case and the issue of a growing number of users as the blocklength increases.

The paper [69] highlights an important observation: even in works that consider a growing number of users, the blocklength is typically sent to infinity before the number of users. Indeed, [20] follows exactly this approach. We note that while the capacity computed in this manner can still be used in the converse bound, it does not necessarily apply to the achievability bound. The reason is as follows: the argument that joint typicality holds with probability 1 as the blocklength grows to infinity does not directly extend to models where the number of users also grows to infinity. Thus, the achievability bound derivation must be reworked.

Now, let us consider the model and the main result. Let K be the total number of users in the system. The users share the same message set $[M]$ but use different encoders $f_i : [M] \bigcup \{0\} \to \mathbb{R}^n$. Each user is active with probability $\alpha^{(n)}$, and both this probability and the message selection strategy are identical for all users ($i \in [K]$):

$$\Pr[W_i = W] = \begin{cases} 1 - \alpha^{(n)}, & W = 0, \\ \alpha^{(n)}/M, & W \in [M]. \end{cases}$$

Suppose each user accesses the channel independently with identical probability $\alpha^{(n)}$ during any given block. The messages are encoded and sent via the Gaussian many-access channel:

$$\mathbf{y} = \sum_{i=1}^{K} f_i(W_i) + \mathbf{z}, \tag{2.11}$$

where $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_n)$ is Gaussian noise. Note that $\|f_i(W_i)\|^2 \leq Pn$. If user i is inactive, it is assumed to transmit the all-zero codeword, i.e., $f_i(0) = \mathbf{0}$.

The authors focus on the *joint* probability of error:

$$P_J^{(n)} = \Pr\left[\mathrm{dec}(\mathbf{y}) \neq (W_1, \ldots, W_K)\right],$$

where the decoder must return zeros for users who were inactive.

The main result is stated in [69, Theorem 1]. This theorem considers different regimes; for the reader's convenience, we do not reproduce it in full but instead provide the result relevant to the regime of interest. We focus on the case where the average number of active users (K_a) grows linearly with the blocklength (n), i.e.,

$$K_a^{(n)} = \alpha^{(n)} K^{(n)} = \mu n,$$

where μ is some constant. At the same time, the total number of users can scale faster, e.g., as n^2.

For this case, the symmetric message-length capacity (in bits) is given by:

$$k^{(n)} \sim \left(\frac{1}{2\mu} \log_2(1 + \mu n P) - \frac{h(\alpha^{(n)})}{\alpha^{(n)}} \right)_+,$$

which means there exists a sequence of codebooks with message lengths (in bits) arbitrarily close to k_n such that as $n \to \infty$, the average probability of error $P_J^{(n)}$ vanishes. The capacity is achieved by i.i.d. Gaussian inputs.

Note that $f^{(n)} \sim g^{(n)}$ if and only if

$$\lim_{n\to\infty} \frac{f^{(n)}}{g^{(n)}} = 1.$$

We use equivalence rather than equality, since if $k^{(n)}$ is the message-length capacity, then $k^{(n)} + o(k^{(n)})$ is also considered capacity.

Overview of the results. The expression for k_n consists of two parts:

- The first term resembles the classical capacity expression and depends only on the average number of active users.
- The second term represents the loss due to the need to identify active users and depends on the ratio of the average number of active users to the total number of users.

Example 2.1. *Let us consider the example from [69]. Assume $P = 10$ dB and $K_a^{(n)} = n/4$. We consider three cases: (a) $K^{(n)} = n$, (b) $K^{(n)} = n^2$ and (c) $K^{(n)} = n^3$. The corresponding message-length capacities are:*

(a) $k^{(n)} \sim 2\log_2(1 + 5n/2) - 4h(1/4)$

(b) $k^{(n)} \sim 2\log_2(1 + 5n/2) - \log_2(4en)$

(c) $k^{(n)} = 0$.

Remark 2.6 (Infinite energy per bit)**.** *The authors of [69] assume a fixed power P. However, the energy required to send one information bit tends to infinity as n grows. Indeed, consider the case where all users are active:*

$$K_a^{(n)} = K^{(n)} = \mu n.$$

The energy per information bit is given by:

$$\frac{Pn}{\frac{1}{2\mu}\log_2(1 + \mu n P)} \to \infty.$$

Remark 2.7. *By* energy per bit*, we consider the ratio E_b/N_0, where E_b represents the energy required to transmit one information bit and N_0 denotes the noise power spectral density. Through appropriate scaling of both signal and noise, we may assume $N_0 = 1$ without loss of generality, making the quantities E_b and E_b/N_0 equivalent. Therefore, these terms may be used interchangeably throughout this monograph. Furthermore, given n channel uses and a transmit* power P*, the total transmitted* energy *equals Pn.*

2.4.3 Finite energy and joint probability of error

Recall that energy efficiency is of critical importance for massive MTC. Namely we are interested in *finite* energy per information bit. We have the following result [70].

Proposition 2.1. *Assume the case where all users are active, i.e., $K_a = K$, and each user aims to transmit $k = 1$ bit over the channel* (2.11) *with finite energy $\mathcal{E}$. Then we have*

$$1 - P_J \leq \frac{\mathcal{E}\log e + \log 2}{\log K}.$$

Thus, if $K \to \infty$, then the success probability $1 - P_J \to 0$, implying that the users cannot reliably transmit even a single bit.

Proof. First, let us show that the standard Fano-inequality-based converse does not work. We have (recalling that $M = 2$)

$$1 - P_J \leq \frac{\frac{n}{2}\log\left(1 + \frac{K}{n}\mathcal{E}\right) + \log 2}{K\log M} = \frac{1}{2\log 2}\frac{n}{K}\log\left(1 + \frac{K}{n}\mathcal{E}\right) + \frac{1}{K},$$

and we may obtain different relationships depending on the ratio K/n (e.g., in the case $K = \mu n$).

Now, let us introduce a *genie* that provides us with the vector

$$\boldsymbol{W}' = (W_1', \ldots, W_K') \in \{0,1\}^K$$

such that

$$d(\boldsymbol{W}, \boldsymbol{W}') = 1,$$

where $\boldsymbol{W} = (W_1, \ldots, W_K)$ represents the transmitted messages, and $d(\boldsymbol{a}, \boldsymbol{b})$ is the Hamming distance between vectors $\boldsymbol{a}$ and $\boldsymbol{b}$.

We note that the genie's help reduces the number of hypotheses. Initially, there are 2^K possible messages, but with the genie's assistance, this number is reduced to K, meaning that we only need to determine the error position. Thus,

$$1 - P_J \leq \Pr\left[U = \hat{U}\right],$$

where U is the index of the user whose message was received in error.

Next, we reformulate the problem in terms of the modified received signal:

$$Y' = Y - \sum_{i=1}^{K} f_i(W_i') = f_U(W_U) - f_U(W_U') + \boldsymbol{z}.$$

The goal is now to identify U.

Consider the new codebook:

$$\boldsymbol{C} = [\boldsymbol{c}_1, \ldots, \boldsymbol{c}_K],$$

of size $n \times K$, where

$$\boldsymbol{c}_i = f_i(W_i' \oplus 1) - f_i(W_i'),$$

and note that $\|\boldsymbol{c}_i\|^2 \leq 2\mathcal{E}$.

Applying Fano's inequality to this new problem, we obtain

$$\Pr\left[U = \hat{U}\right] \leq \frac{\frac{n}{2}\log\left(1 + \frac{2\mathcal{E}}{n}\right) + \log 2}{\log K} \leq \frac{\mathcal{E}\log e + \log 2}{\log K}.$$

□

Thus, the probability of joint decoding error cannot be made arbitrarily small under a finite energy constraint.

Remark 2.8. *Given these results, in the subsequent discussion, we switch from joint decoding to the PUPE criterion. The asymptotic results for finite energy are studied in Appendix F.*

2.5 Outcomes

In this monograph, we focus on uncoordinated multiple access, as this form of channel access is both natural and advantageous for massive MTC scenarios involving a large number of infrequently transmitting devices. Coordinated multiple access is a separate problem and falls beyond the scope of this monograph, although some tools from this line of research (e.g., decoding approaches) are employed within the URA framework.

In this chapter, we examined different approaches to the MAC problem, discussing both information-theoretic and random-access methods. The former focuses on physical layer effects but does not account for the randomness of user activity. Moreover, it assumes a fixed number of users while the blocklength tends to infinity. In contrast, the latter provides deep insights into collision resolution algorithms by considering the randomness of user activity, but most works largely ignore the impact of noise at the physical layer. Some papers do take noise into consideration (see, e.g., [15, Appendix]), but none of them aim to formulate the corresponding system requirements.

To bridge this gap, the URA paradigm was introduced in [5]. The URA framework not only addresses the computational challenges posed by a vast number of devices but also treats RA as a coding problem—incorporating both medium access protocols and physical layer effects. The URA paradigm extends previous work in several ways: it incorporates FBL performance, allows for an unbounded total number of users, and prioritizes energy per bit over the traditional notion of rate under vanishing error probability.

At the same time, we emphasize that URA is an information-theoretic framework that offers a new perspective on random access, providing a fundamental benchmark to assess the performance of various uncoordinated access schemes. URA does not represent a new method of accessing a shared medium.

The following chapters provide a detailed overview of the URA problem, covering fundamental limits, low-complexity schemes, and its connection to the CS problem.

Chapter 3

URA problem statement and compressed sensing interpretation

Recall that the main problem is to provide multiple access to a massive number of uncoordinated and infrequently communicating devices. In this chapter, we highlight the most important aspects of this problem and provide the corresponding system model and information-theoretic formulation, as proposed by Y. Polyanskiy in [5].

3.1 System model

We consider the scenario of partial user activity. Assume that there are $K_{\text{tot}} \gg 1$ users[1] in the system, but only $K_a \ll K_{\text{tot}}$ of them are active at any given time. Throughout this chapter, we assume that the number of active users (K_a) is fixed and known at the receiver. The papers [71, 72] extend these results to the scenario where the number of active users is random and unknown. This extension will be discussed briefly in Chapter 4. It is worth noting that this extension is a major aspect of any practical scheme, since the receiver lacks a priori knowledge of the number of active users. Instead, it must estimate this quantity from the received signal. All users utilize the same message set $[M]$ and aim to transmit $k = \log_2 M$ bits. Each user selects a message to transmit uniformly and independently of the other users. Communication occurs in a frame-synchronized manner, with each frame consisting of n channel uses.

The main features of this problem formulation are as follows:

1. *Large number of active users with short data packets.* Unlike the classical MAC problem formulation, where K_a is fixed while $n, k \to \infty$, we consider a scenario in which k is fixed while K_a and n are large.

2. *Users share the same encoder.* Given the vast number of devices, assigning different encoders

[1] In what follows, K_{tot} is of no importance, and one may assume $K_{\text{tot}} = \infty$.

to each user would result in prohibitive receiver complexity. The receiver would not know which decoder to use and, in the worst case, would need to try all of them. Therefore, a promising strategy is to employ a common encoder $f : [M] \to \mathbb{R}^n$ for all users, leading to a random-access scenario. A natural power constraint,

$$\|f(W)\|_2^2 \leq nP, \quad W \in [M],$$

is also imposed.

Since users are indistinguishable, the receiver is required only to recover the transmitted messages without identifying their senders. In other words, decoding is performed up to permutation.[2] Such schemes are referred to as unsourced random access (URA), since the source of the message is irrelevant.

Users may include their identity as part of the k-bit payload, but this is not mandatory. For example, a fire sensor only needs to transmit the coordinates of a fire without an explicit identifier.

3. *User-centric probability of error or PUPE.*[3] Let

$$\mathcal{T} = \{W_1, W_2, \ldots, W_{K_a}\}$$

denote the set of transmitted messages. The decoder outputs a set $\mathcal{R} \subseteq [M]$, and we measure the probability of error as follows:

$$P_e = \mathbb{E}\left[\frac{|\mathcal{T} \backslash \mathcal{R}|}{K_a}\right] = \frac{1}{K_a} \sum_{i=1}^{K_a} \Pr\left[W_i \notin \mathcal{R}\right]. \tag{3.1}$$

Tracking P_e is sufficient when the output list size is fixed (e.g., the usual assumption for bound derivations is $|\mathcal{R}| = K_a$). However, when $|\mathcal{R}|$ is a random variable (as in practical schemes), we also define the *false alarm rate (FAR)*:

$$P_f = \mathbb{E}\left[\frac{|\mathcal{R} \backslash \mathcal{T}|}{|\mathcal{R}|}\right]. \tag{3.2}$$

4. *Energy efficiency.* Since the devices are autonomous and battery-powered, the goal is to minimize the energy per bit while ensuring a maximum tolerable PUPE of $P_e \leq \varepsilon$:

$$\min_{\text{s. t. } P_e \leq \varepsilon} \frac{E_b}{N_0}, \quad \text{where} \quad \frac{E_b}{N_0} = \frac{Pn}{2k}. \tag{3.3}$$

In this monograph, we focus on the Gaussian channel, as it provides key insights into the URA problem without unnecessary complications. The fading model and its main implications will be discussed briefly in Chapter 6.

[2] Note that we only consider channels that are permutation-invariant, allowing this type of decoding.

[3] To justify the choice of PUPE, see Proposition 2.1, which shows that achieving a small joint probability of error is infeasible.

Let $\boldsymbol{x}^{(i)} = f(W_i)$, for $i \in [K_a]$. The output of the Gaussian channel, $\mathbf{y} \in \mathbb{R}^n$, is given by:

$$\mathbf{y} = \sum_{i=1}^{K_a} \boldsymbol{x}^{(i)} + \mathbf{z}, \tag{3.4}$$

where $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_n)$ is the AWGN.

Remark 3.1. *In URA analysis, we focus on the non-asymptotic regime, as users utilize the same codebook of fixed size, and we cannot increase the number of users indefinitely (since $|\mathcal{T}| \leq M$). We provide the asymptotic analysis of the bounds described in Chapter 4 in Appendix F. For this purpose, we shift to a different codebook scenario (see Section 2.4.3).*

Remark 3.2. *Note that the factor* 2 *in* (3.3) *corresponds to a real-valued AWGN channel. In the case of a complex-valued channel (Chapter 6), this factor should be omitted.*

Remark 3.3. *We note that URA does not introduce a new way of accessing the channel. For example, the ALOHA system presented in Section 2.3 is a good example of a URA scheme. Instead, URA provides a new information-theoretic model that incorporates both medium access protocols and physical layer effects, focusing on energy per bit rather than the more classical throughput and rate under vanishing error. This model applies to already existing schemes, such as ALOHA.*

Remark 3.4. *In this manuscript, we assume perfect synchronicity, as proposed in the original work [5]. We consider that synchronization is supported by additional mechanisms such as beaconing and proper frame structuring. In the case of perfect synchronicity, the analysis becomes clearer and more straightforward.*

At the same time, we note that keeping autonomous low-power devices synchronized is a challenging problem. Asynchronous URA has been addressed in various research papers [73, 74, 75, 76]; however, even the fundamental limits for this setup remain unknown. We mention this as an interesting open problem in Chapter 7.

3.2 URA as an approximate support recovery problem

In some situations, it is beneficial to represent sets[4] of messages as corresponding indicator vectors. Consider a multi-set:

$$\mathcal{T} = \{W_1, W_2, \ldots, W_{K_a}\}, \quad W_i \in [M], \quad i \in K_a.$$

We can represent this set as a vector:

$$\boldsymbol{u} = [u_1, u_2, \ldots, u_M]^T,$$

where u_i denotes the number of times message i was chosen or the multiplicity of i in $\mathcal{T}$.

[4] or multi-sets if the same message was chosen by multiple users.

Example 3.1. *Let $M = 5$, $K_a = 8$, and*

$$\mathcal{T} = \{1, 2, 2, 3, 3, 5, 5, 5\}.$$

The corresponding vector $\boldsymbol{u}$ is given by:

$$\boldsymbol{u} = [1, 2, 2, 0, 3]^T.$$

It is worth mentioning that the URA problem formulation allows for a standard CS interpretation [77, 78]. Recall that M is the number of messages and n is the blocklength. We can represent the channel output as

$$\mathbf{y} = \boldsymbol{X}\boldsymbol{u} + \mathbf{z}, \tag{3.5}$$

where:

- $\mathbf{y}$ is the channel output, representing a noisy mixture of users' codewords,
- $\mathbf{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M] = [x_{i,j}]$ is a codebook of size $n \times M$ with $M = 2^k$, where $\boldsymbol{x}_W = f(W)$ for $W \in [M]$,
- $\mathbf{u}$ represents the activity vector, and $\mathbf{z}$ is the AWGN vector. Note that $|\mathrm{supp}(\boldsymbol{u})| \leq K_a$ and $\sum_{i=1}^{M} u_i = K_a$.

The matrix $\boldsymbol{X}$ can be viewed as a sensing matrix, and $\boldsymbol{u}$ is a sparse vector (with sparsity defined by the number of active users) to be reconstructed. More precisely, our problem is the approximate support recovery (ASR) problem [79], as we only need to determine the support of $\boldsymbol{u}$ up to some distortion defined by the desired PUPE:

$$P_e = \mathbb{E}\left[\frac{|\mathrm{supp}(\boldsymbol{u}) \backslash \mathrm{supp}(\hat{\boldsymbol{u}})|}{K_a}\right] \leq \varepsilon,$$

where $\hat{\boldsymbol{u}}$ is the recovered activity vector or the indicator vector of the set $\mathcal{R}$.

Remark 3.5. *The number of unique messages $|\mathcal{T}|$ can be less than K_a. Assuming $W_i \sim \mathrm{Unif}([M])$, $i \in [K_a]$, we can estimate the probability of this event as follows:*

$$p' = \Pr\left[|\mathcal{T}| < K_a\right] = 1 - \prod_{i=0}^{K_a - 1}\left(1 - \frac{i}{M}\right) \leq \frac{\binom{K_a}{2}}{M}.$$

In this monograph, we consider $\log_2 M \approx 100$ bits and $1 \leq K_a \leq 500$. For these parameters, p' is negligible, allowing us to disregard the case where $|\mathcal{T}| < K_a$ and thus work with $\boldsymbol{u}, \hat{\boldsymbol{u}} \in \{0, 1\}^M$, where $|\mathrm{supp}(\boldsymbol{u})| = |\mathrm{supp}(\hat{\boldsymbol{u}})| = K_a$.

We establish a schematic correspondence between the URA and ASR problems in Fig. 3.1. The sensing matrix is represented by a codebook shared among all users. This matrix is multiplied by the activity vector $\boldsymbol{u}$ (with nonzero elements marked by orange squares). The resulting vector, after being corrupted by noise, produces the channel output.

Remark 3.6. *A key observation in related research is that the dimensionality of this problem is enormous—on the order of 2^{100} or greater. This characteristic makes the application of standard CS algorithms infeasible.*

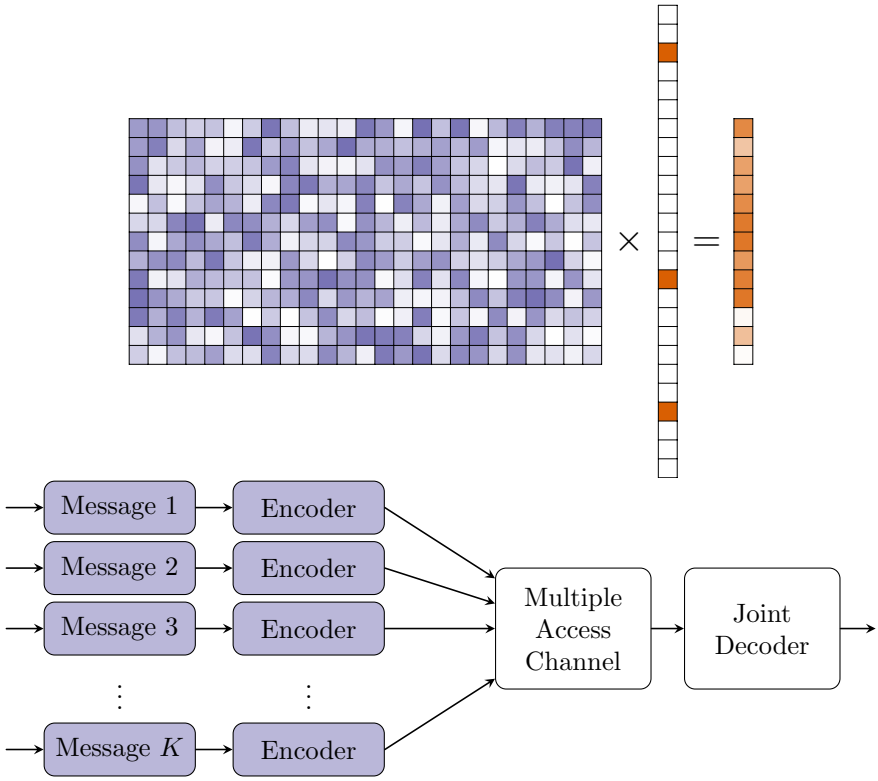

Figure 3.1: Equivalence of the CS problem (top) and the same-codebook multiple-access problem (bottom). The CS problem is given by (3.5). Orange squares represent the nonzero elements of the users' activity vector. This figure is reproduced from [80]

.

3.3 Converse bounds

In this section, we describe two converse bounds for the URA problem. The first is a single-user converse, with the only difference being that it accounts for the output list of messages. The second utilizes a capacity versus rate-distortion argument.

Theorem 3.1 (Single-user converse [80])**.** *Consider K_a users transmitting messages $W_i \in [M]$, $i \in [K_a]$, via a Gaussian channel within n channel uses, under a maximum tolerable PUPE ε. Given that the decoder's output list size is ℓ_0, then*

$$nP \geq \left(Q^{-1}(\ell_0/M) + Q^{-1}(\varepsilon)\right)^2, \quad Q(x) = \int_x^{+\infty} \frac{1}{\sqrt{2\pi}} e^{-y^2/2} dy, \tag{3.6}$$

where $Q^{-1}(\cdot)$ is the inverse of the Q-function defined above.

Proof. We provide a sketch of the proof:

- In [31], it was shown that any single-user channel code over a Gaussian channel with parameters (n, M, P) and error probability $P_e \triangleq \Pr[W \neq \hat{W}] \leq \varepsilon$ must satisfy

$$nP \geq \left(Q^{-1}(1/M) + Q^{-1}(\varepsilon)\right)^2.$$

- This argument extends naturally to show that list-ℓ_0 decodable codes (where $P_e \triangleq \Pr[W \notin \mathcal{R}] \leq \varepsilon$ and $|\mathcal{R}| = \ell_0$) must satisfy (3.6).

- *Symmetry argument*: It suffices to obtain a lower bound on the probability that a particular user's message is not in the decoded list.

- *Genie argument*: Assume the decoder has access to the interference from all other users. The equivalent channel is given by:

$$\mathbf{y}' = \boldsymbol{x}^{(1)} + \mathbf{z},$$

to which the bound above applies.

□

Theorem 3.2 (Multi-user converse [5])**.** *Consider K_a users transmitting messages $W_i \in [M]$, $i \in [K_a]$, via a Gaussian channel within n channel uses, under a maximum tolerable PUPE ε. Given that the decoder's output list size is ℓ_0, then*

$$\log_2 M \leq \frac{1}{(1-\varepsilon)}\left(\frac{n}{K_a}C(K_aP) + h(\varepsilon) + (1-\varepsilon)\log_2 \ell_0\right),$$

where $C(x) = 1/2 \log_2(1+x)$.

Proof. Consider the i-th user. Let $\varepsilon_i = \Pr[\mathsf{W}_i \notin \mathcal{R}]$. Using Fano's list inequality, we obtain:

$$H(\mathsf{W}_i|\mathbf{y}) \leq h(\varepsilon_i) + (1-\varepsilon_i)\log_2 \ell_0 + \varepsilon_i \log_2 (M - \ell_0). \tag{3.7}$$

Summing inequalities (3.7) over all $i \in [K_a]$ and noting that

$$H(\mathsf{W}_i|\mathbf{y}) = H(\mathsf{W}_i) - I(\mathsf{W}_i; \mathbf{y}),$$

we obtain:

$$\begin{aligned}\sum_{i=1}^{K_a} I(\mathsf{W}_i; \mathbf{y}) \geq & \sum_{i=1}^{K_a} H(\mathsf{W}_i) \\ & - \sum_{i=1}^{K_a} h(\varepsilon_i) - \left(1 - \sum_{i=1}^{K_a} \varepsilon_i\right)\log_2 \ell_0 \\ & - \sum_{i=1}^{K_a} \varepsilon_i \log_2 (M - \ell_0).\end{aligned}$$

Note that:

$$\begin{aligned} H(\mathsf{W}_i) &= \log_2 M, \quad \text{for } i \in [K_a], \\ \sum_{i=1}^{K_a} \varepsilon_i &= K_a \varepsilon, \\ \sum_{i=1}^{K_a} h\left(\varepsilon_i\right) &\geq K_a h(\varepsilon). \end{aligned}$$

Using the Markov chain:

$$(\mathsf{W}_1, \ldots, \mathsf{W}_{K_a}) \to (\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(K_a)}) \to \mathbf{y} \to \mathcal{R},$$

we derive:

$$\begin{aligned} \sum_{i=1}^{K_a} I\left(\mathsf{W}_i; \mathbf{y}\right) &\leq I(\mathsf{W}_1, \mathsf{W}_2, \ldots, \mathsf{W}_{K_a}; \mathbf{y}) && \text{(independent messages)} \\ &\leq I(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(K_a)}; \mathbf{y}) && \text{(data processing ineq.)} \\ &\leq \sum_{j=1}^{n} I(\mathsf{x}_j^{(1)}, \mathsf{x}_j^{(2)}, \ldots, \mathsf{x}_j^{(K_a)}; \mathsf{y}_j) && \text{(memoryless channel)} \\ &\leq nC(K_a P). \end{aligned}$$

□

Remark 3.7. *A very similar problem was considered in [79] and presented in [80]. The main difference is that [79, 80] consider $\mathcal{T}$ chosen uniformly from $\binom{M}{K_a}$ variants (or equivalently, the indicator vector $\boldsymbol{u} \in \{0,1\}^M$, where $|\mathrm{supp}(\boldsymbol{u})| = K_a$). The converse presented above remains valid even when messages are repeated. For the regime of interest, the two converses coincide.*

We refer the reader to Section 4.6, where different achievability bounds are compared, highlighting the gap between achievability and converse bounds presented in Theorems 3.1 and 3.2. The results are shown in Fig. 4.6. Note that the converse is depicted as a single line, representing the maximum of the bounds from Theorems 3.1 and 3.2.

3.4 Parameters of interest

We need to fix the parameters (n, k, K_a). Consider a typical LoRa network:

- Payload size: $k \approx 100$ bits.
- A message transmitted using LoRa with a spreading factor of 11 occupies approximately $k \cdot {}^{2^{11}}/_{11}$ complex channel uses, resulting in $n \approx 3 \times 10^4$ real channel uses.

Following [5], we use the following parameters:

- Frame length: $n = 3 \times 10^4$ (real channel uses),
- User payload: $k = 100$ bits,
- Number of active users: $K_a = 1, \ldots, 500$,
- Target PUPE: $\varepsilon \in \{0.001, 0.05, 0.1\}$.

The goal is to find the minimal E_b/N_0 in (3.3) such that $P_e \leq \varepsilon$.

We also note that users share a large number of channel uses compared to the number of information bits, leading to an individual coding rate of approximately 1/300.

Chapter 4

URA over Gaussian MAC: achievability bounds

In this chapter, we consider achievability bounds for the URA over Gaussian MAC (G-URA). Specifically, we are interested in the tradeoff between energy efficiency (E_b/N_0) and the number of active users (K_a). The term "achievability" implies that we do not account for decoding complexity and instead consider the potential capabilities under an infeasible decoding algorithm.

We begin the chapter by defining the ML decoding rule and establishing a general setup for evaluating achievability bounds, as presented in Section 4.1. Within this setup, we assume that the number of active users is known at the receiver.

In Section 4.2, we introduce Gallager's ρ-trick and Fano's trick – key components of the subsequent analysis. Next, in Section 4.3, we derive three achievability bounds. The first bound, presented by Y. Polyanskiy [5], is based on Gallager's ρ-trick (see Theorem 4.1). This bound has become a foundational result in URA research. The second bound is based on Fano's trick (see Theorem 4.2). We find that this bound provides a similar energy efficiency estimate to the first one. However, a straightforward modification of this bound enables us to estimate energy efficiency under binary codebooks (see Theorem 4.4). Notably, we observe that switching from Gaussian codebooks to binary does not lead to a loss in achievable energy efficiency estimates.

Next, we describe an achievability bound that combines both tricks. Originally designed to evaluate the performance of sparse regression codes (SPARCs), this bound is presented in Theorem 4.3. However, these results are also applicable to the URA problem.

In Section 4.5, we discuss additional research on achievable energy efficiency analysis. We present results for a random number of active users, where energy efficiency depends on the probability of error and the false alarm rate (FAR), as discussed in Section 4.5.1. Finally, in Section 4.5.2, we explore the application of Gordon's lemma to the URA problem. We conclude this chapter with a numerical analysis of the described bounds, as presented in Section 4.6.

4.1 Maximum likelihood decoding rule

Recall that $\mathcal{T} = \{W_1, W_2, \ldots, W_{K_a}\}$ is the set of transmitted messages, and $\mathcal{R} \subseteq [M]$, with $|\mathcal{R}| = K_a$, is the set of received messages. The decoder's task is to recover the set of transmitted messages up to permutation. Note that

$$\Pr\left[\bigcup_{i\neq j}\{W_i = W_j\}\right] \leq \frac{\binom{K_a}{2}}{M},$$

which is negligible for the system parameters of interest (see Remark 3.5). Thus, we do not consider multi-sets in this section.

Next, we introduce helpful notation. For $\mathcal{I} \subseteq [M]$, we define

$$\boldsymbol{s}_{\mathcal{I}} = \sum_{W\in\mathcal{I}} f(W) = \sum_{W\in\mathcal{I}} \boldsymbol{x}_W.$$

We now formulate the ML decoding rule:

$$\mathcal{R} = \underset{\mathcal{R}'\subseteq[M],|\mathcal{R}'|=K_a}{\arg\min} \left\|\mathbf{y} - \boldsymbol{s}_{\mathcal{R}'}\right\|^2. \tag{4.1}$$

Remark 4.1. *We note that the decoder mentioned above is optimal for the joint error probability rather than for PUPE. A PUPE-optimal decoder should compute*

$$\Pr[W \in \mathcal{T} \mid \boldsymbol{y}] \propto \sum_{\mathcal{T}':W\in\mathcal{T}'} p(\boldsymbol{y} \mid \boldsymbol{s}_{\mathcal{T}'}) \Pr[\mathcal{T}']$$

for all $W \in [M]$, and then output the top-K_a messages. The choice of the joint decoder was motivated by its suitability for analysis.

4.1.1 Probabilistic method, t-error events and P_e estimate

It is not possible to analyze P_e for a specific codebook. Therefore, we utilize the so-called probabilistic method [81] or random coding [82]. Specifically, we consider the ensemble $\mathcal{G}$ of codebooks $\mathbf{X}$ specified by the probability distribution $P_{\mathbf{X}}$[1] and compute

$$P_e = \mathbb{E}_{\mathbf{X}}[P_e(\mathbf{X})].$$

We can then assert that there exists a codebook $\boldsymbol{X}^* \in \mathcal{G}(M, n, P)$ such that $P_e(\boldsymbol{X}^*) \leq P_e$. Due to symmetry when averaging over codebooks, we assume w.l.o.g. that $\mathcal{T} = [K_a]$. Let $\mathcal{C} = \mathcal{T} \bigcap \mathcal{R}$, $\mathcal{M} = \mathcal{T}\backslash\mathcal{R}$ and $\mathcal{F} = \mathcal{R}\backslash\mathcal{T}$ be the sets of correctly received, missed, and falsely detected messages, respectively (see Fig. 4.1).

Following the approach of [5], we consider t-error events $\mathcal{E}_t = \{|\mathcal{M}| = t\}$ and use the following estimate:

$$P_e \leq \sum_{t=1}^{K_a} \frac{t}{K_a} \Pr[\mathcal{E}_t] + p_0, \tag{4.2}$$

[1] To utilize the symmetry property, we consider only ensembles with i.i.d. codewords.

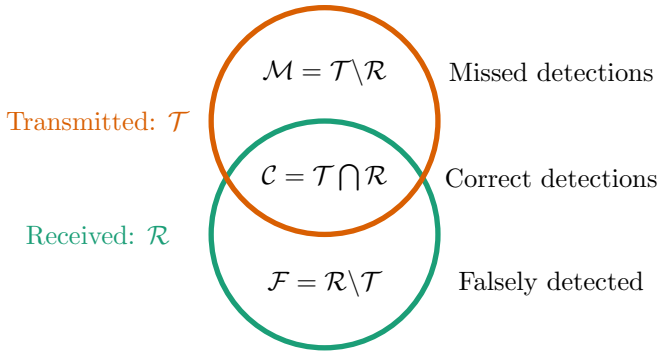

Figure 4.1: Correctly detected, missed, and falsely detected codewords. The set of transmitted messages has size $|\mathcal{T}| = K_a$. Throughout this chapter, we assume $|\mathcal{R}| = K_a$ for all bounds discussed, which implies that the receiver knows the number of transmitted messages. In this case, we assume that $|\mathcal{M}| = |\mathcal{F}|$.

where

$$p_0 = \Pr\left[\left[\bigcup_{i \neq j} \{W_i = W_j\}\right] \cup \left[\bigcup_{i \in [K_a]} \{\|f(W_i)\|^2 > Pn\}\right]\right] \tag{4.3}$$

is the probability that different users select the same codewords from the codebook or that the signal power exceeds Pn. Thus, in what follows, we do not consider colliding messages. For ensembles of Gaussian codebooks considered in Section 4.3, the probability of power violation, p_0, is bounded by (4.6). For binary codebooks considered in Section 4.4, we have $p_0 = 0$. Recall that the number of active users is assumed to be known at the receiver.

Let us examine the error event in more detail. In the case of ML decoding, a decoding error occurs if there exists a set of messages $\mathcal{R}$ such that the sum of the corresponding codewords is closer to the received sequence $\mathbf{y}$ than the sum of the codewords corresponding to the transmitted messages $\mathcal{T}$:

$$\|\mathbf{y} - \mathbf{s}_{\mathcal{T}}\|^2 \geq \|\mathbf{y} - \mathbf{s}_{\mathcal{R}}\|^2 .$$

Taking into account that $\mathbf{y} = \mathbf{s}_{\mathcal{T}} + \mathbf{z}$, $\mathbf{s}_{\mathcal{T}} = \mathbf{s}_{\mathcal{M}} + \mathbf{s}_{\mathcal{C}}$ and $\mathbf{s}_{\mathcal{R}} = \mathbf{s}_{\mathcal{F}} + \mathbf{s}_{\mathcal{C}}$, we obtain the following decoding error event:

$$\mathcal{E}(\mathcal{M}, \mathcal{F}) = \left\{\|\mathbf{z}\|^2 - \|\mathbf{s}_{\mathcal{M}} - \mathbf{s}_{\mathcal{F}} + \mathbf{z}\|^2 \geq 0\right\}. \tag{4.4}$$

Note that

$$\mathcal{E}_t = \bigcup_{\mathcal{M} \subseteq \mathcal{T}, \mathcal{F} \subseteq [M] \backslash [K_a]} \mathcal{E}(\mathcal{M}, \mathcal{F}).$$

Let us begin with the naive approach and apply the union bound in a straightforward manner. We have

$$\Pr[\mathcal{E}_t] \quad = \quad \Pr\left[\bigcup_{\mathcal{M},\mathcal{F}} \mathcal{E}(\mathcal{M}, \mathcal{F})\right] = \mathbb{E}_{\mathbf{X}}\left[\Pr\left[\bigcup_{\mathcal{M},\mathcal{F}} \mathcal{E}(\mathcal{M}, \mathcal{F}) \,\middle|\, \mathbf{X}\right]\right]$$

$$\leq \binom{K_a}{t}\binom{M-K_a}{t}\mathbb{E}_{\mathbf{X}}\left[\Pr\left[\mathcal{E}(\mathcal{M}',\mathcal{F}')\,|\,\mathbf{X}\right]\right],$$

where $\mathcal{M}'=[t]$, $\mathcal{F}'=[K_a+t]\backslash[K_a]$. The last step follows from the union bound and the symmetry property (as we are averaging over the codebook).

We note that the second binomial coefficient is extremely large (recall that M is on the order of 2^{100}). Thus, the naive approach significantly overestimates the desired probability. In what follows, we describe several approaches to tighten the union bound.

4.2 Useful tricks to tighten the union bound

In this section, we describe all the tools needed for the proofs. We begin with approaches to improve the union bound, as proposed by Gallager[2] [83] and Fano [84].

4.2.1 Gallager's trick

Gallager's ρ-trick is a method to improve the union bound, which can be formulated as follows. Let $\mathcal{E}_i$, for $i\in[m]$ be the events. Then, for any $0\leq\rho\leq 1$ we have

$$\Pr\left[\bigcup_{i=1}^{m}\mathcal{E}_i\right]\leq\left(\sum_{i=1}^{m}\Pr\left[\mathcal{E}_i\right]\right)^{\rho}.$$

It is reasonable to apply this method at intermediate steps in estimating conditional probabilities. Specifically, one needs to find a suitable random variable V such that

$$\Pr\left[\mathcal{E}_i\,|\,V\right]\leq\exp\left[-F(V)\right]$$

for some function $F(V)$.

Then the ρ-trick leads to the following estimate:

$$\Pr\left[\bigcup_{i=1}^{m}\mathcal{E}_i\right]\leq m^{\rho}\mathbb{E}_V\left[\exp\left[-\rho F(V)\right]\right].$$

4.2.2 Fano's trick

Let us explain the idea using a simple single-user case. Let $\mathcal{C}=\{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_M\}$ and

$$\mathbf{y}=\boldsymbol{x}_1+\mathbf{z},$$

where $\boldsymbol{x}_1\in\mathcal{C}$ and $\mathbf{z}$ is the noise vector.

[2] also known as Gallager's ρ-trick

Consider the ML or minimum Euclidean distance decoding. The error event is $\mathcal{E} = \bigcup_{m=2}^{M} \mathcal{E}_m$, where $\mathcal{E}_m = \{\|\mathbf{y} - \boldsymbol{x}_1\|^2 \geq \|\mathbf{y} - \boldsymbol{x}_m\|^2\}$.

Let us consider why the union bound overestimates the probability. This occurs when several $\boldsymbol{x}_m$ are closer to $\mathbf{y}$ than $\boldsymbol{x}_1$. Clearly, this situation becomes more likely when the norm of the noise is large.

Fano's method can be described as follows. Let us choose a so-called "good" region $\mathcal{B}$. We proceed as follows:

$$\Pr[\mathcal{E}] \leq \Pr\left[\mathcal{E} \bigcap \mathcal{B}\right] + \Pr[\mathcal{B}^c], \tag{4.5}$$

and then apply the union bound for the first term only.

In the single-user example, a reasonable choice is $\mathcal{B} = \{\|z\|^2 \leq \beta n\}$ for some $\beta > 0$. In the general case, the choice of $\mathcal{B}$ poses a significant challenge.

4.3 Achievability bounds

In this section, we present three achievability bounds. All of these bounds consider a Gaussian codebook and a paradigm of averaging over multiple codebooks generated from the *ensemble*. To proceed, let us begin with the definition of the ensemble.

Definition 4.1. *Let $\mathcal{G}(M, n, P)$ be the ensemble of Gaussian codebooks of size $n \times M$, where each element is sampled i.i.d. from $\mathcal{N}(0, P)$.*

The key difference between the approaches presented below lies in how the union bound is tightened. We utilize Gallager's ρ-trick [5], Fano's trick [85], and their combination presented in [86]. We note that the problem and approaches considered in this section share many similarities with the optimal decoder analysis of SPARCs [86, Chapter 2].

Fix $P' < P$ and consider $\mathcal{G}(M, n, P')$. The following estimate is valid for p_0 (see (4.3))

$$p_0 \leq \overline{p}_0 \triangleq \frac{\binom{K_a}{2}}{M} + K_a \Pr\left[\frac{1}{n}\sum_{i=1}^{n} \xi_i^2 > \frac{P}{P'}\right], \ \xi_i \overset{i.i.d}{\sim} \mathcal{N}(0,1). \tag{4.6}$$

4.3.1 The bound based on Gallager's ρ-trick

We start with the bound that utilizes the Gallager's ρ-trick and the chain rule:

$$\Pr_{\mathsf{x},\mathsf{y},\mathsf{z}}[\mathcal{E}(\mathsf{x},\mathsf{y},\mathsf{z})] = \mathbb{E}_{\mathsf{z}}\left[\mathbb{E}_{\mathsf{y}\,|\,\mathsf{z}}\left[\Pr_{\mathsf{x}\,|\,\mathsf{y},\mathsf{z}}\left[\mathcal{E}(\mathsf{x},\mathsf{y},\mathsf{z}) \,|\, \mathsf{y},\mathsf{z}\right]\right]\right].$$

In what follows, we work with independent random vectors, so there is no need to deal with expectations over conditional distributions.

Theorem 4.1 (Y. Polyanskiy, [5]). *Fix $P' < P$. There exists a codebook $\boldsymbol{X}^* \in \mathcal{G}(M, n, P')$ satisfying the power constraint P' and providing P_e using eq. (4.2), where p_0 is bounded by (4.6), and $\Pr[\mathcal{E}_t] \le p_t$, where*

$$p_t = \inf_{0 \le \rho_1, \rho_2 \le 1} \exp\left[-n\left(-\rho_1\rho_2 R_1 - \rho_1 R_2 + E_0(\rho_1, \rho_2)\right)\right],$$

$$R_1 = \frac{1}{n}\log\binom{M-K_a}{t}, \quad R_2 = \frac{1}{n}\log\binom{K_a}{t},$$

$$E_0(\rho_1, \rho_2) = \frac{1}{2}\left(\rho_1 a + \log(1 - 2b\rho_1)\right),$$

$$a = \rho_2 \log(1 + 2P't\lambda) + \log(1 + 2P't\mu), \quad b = \rho_2\lambda - \frac{\mu}{1 + 2P't\mu},$$

$$\mu = \frac{\rho_2\lambda}{1 + 2P't\lambda}, \quad \lambda = \frac{P't - 1 + \sqrt{D}}{4(1 + \rho_1\rho_2)P't}, \quad D = (P't-1)^2 + 4P't\frac{1+\rho_1\rho_2}{1+\rho_2}.$$

Proof. If we look at (4.2), it is clear that we need to deal with $\Pr[\mathcal{E}_t]$, where the randomness is induced by $\mathbf{z}$ and the random matrix $\mathbf{X}$. Our goal is to show that $\Pr[\mathcal{E}_t] \le p_t$, where p_t is given in the theorem statement.

Let

$$\mathcal{E}(\mathcal{M}) = \bigcup_{\mathcal{F} \subseteq [M]\setminus[K_a]} \mathcal{E}(\mathcal{M}, \mathcal{F}).$$

Let $0 \le \rho_1, \rho_2 \le 1$, and we can represent $\Pr[\mathcal{E}_t]$ as follows:

$$\begin{aligned}
\Pr[\mathcal{E}_t] &= \mathbb{E}_{\mathbf{z}}\left[\Pr[\mathcal{E}_t \mid \mathbf{z}]\right] \\
&= \mathbb{E}_{\mathbf{z}}\left[\Pr\left[\bigcup\nolimits_{\mathcal{M}\subseteq[K_a]} \mathcal{E}(\mathcal{M}) \,\middle|\, \boldsymbol{z}\right]\right] \\
&\le \mathbb{E}_{\mathbf{z}}\left[\left(\Pr\left[\bigcup\nolimits_{\mathcal{M}\subseteq[K_a]} \mathcal{E}(\mathcal{M}) \,\middle|\, \boldsymbol{z}\right]\right)^{\rho_1}\right] && (\rho\text{-trick}) \\
&\le \mathbb{E}_{\mathbf{z}}\left[\left(\sum\nolimits_{\mathcal{M}\subseteq[K_a]} \Pr\left[\mathcal{E}(\mathcal{M}) \mid \boldsymbol{z}\right]\right)^{\rho_1}\right] && (\text{union bound}) \\
&= \mathbb{E}_{\mathbf{z}}\left[\left(\sum\nolimits_{\mathcal{M}\subseteq[K_a]} \mathbb{E}_{\mathbf{s}_{\mathcal{M}}}\left[\Pr\left[\mathcal{E}(\mathcal{M}) \mid \boldsymbol{z}, \boldsymbol{s}_{\mathcal{M}}\right]\right]\right)^{\rho_1}\right] \\
&= \mathbb{E}_{\mathbf{z}}\left[\left(\binom{K_a}{t}\mathbb{E}_{\mathbf{s}_{\mathcal{M}'}}\left[\Pr\left[\mathcal{E}(\mathcal{M}') \mid \boldsymbol{z}, \boldsymbol{s}_{\mathcal{M}'}\right]\right]\right)^{\rho_1}\right], && (4.7)
\end{aligned}$$

where the last transition is valid due to symmetry. W.l.o.g., $\mathcal{M}' = [t]$ in the last expression and in what follows.

Now consider

$$\Pr\left[\mathcal{E}(\mathcal{M}') \,\middle|\, \boldsymbol{z}, \boldsymbol{s}_{\mathcal{M}'}\right] = \Pr\left[\bigcup_{\mathcal{F}\subseteq[M]\setminus[K_a]} \mathcal{E}(\mathcal{M}, \mathcal{F}) \,\middle|\, \boldsymbol{z}, \boldsymbol{s}_{\mathcal{M}'}\right]$$

$$\leq \left(\Pr \left[\bigcup_{\mathcal{F} \subseteq [M] \backslash [K_a]} \mathcal{E}(\mathcal{M}', \mathcal{F}) \,\middle|\, \boldsymbol{z}, \boldsymbol{s}_{\mathcal{M}'} \right] \right)^{\rho_2} \qquad (\rho\text{-trick})$$

$$\leq \left(\sum\nolimits_{\mathcal{F} \subseteq [M] \backslash [K_a]} \Pr \left[\mathcal{E}(\mathcal{M}', \mathcal{F}) \mid \boldsymbol{z}, \boldsymbol{s}_{\mathcal{M}'} \right] \right)^{\rho_2} \qquad (\text{union bound})$$

$$\leq \left(\binom{M - K_a}{t} \Pr \left[\mathcal{E}(\mathcal{M}', \mathcal{F}') \mid \boldsymbol{z}, \boldsymbol{s}_{\mathcal{M}'} \right] \right)^{\rho_2}, \qquad (4.8)$$

where the last transition is again due to symmetry. W.l.o.g., $\mathcal{F}' = [K_a + t] \backslash [K_a]$ in the last expression and in what follows.

We start with $\Pr \left[\mathcal{E}(\mathcal{M}', \mathcal{F}') \mid \boldsymbol{s}_{\mathcal{M}}, \boldsymbol{z} \right]$. In accordance with (4.4), we have

$$\Pr \left[\mathcal{E}(\mathcal{M}', \mathcal{F}') \mid \boldsymbol{z}, \boldsymbol{s}_{\mathcal{M}'} \right] = \Pr \left[\|\boldsymbol{z}\|^2 - \|\boldsymbol{s}_{\mathcal{M}'} - \mathbf{s}_{\mathcal{F}'} + \boldsymbol{z}\|^2 \geq 0 \,\middle|\, \boldsymbol{z}, \boldsymbol{s}_{\mathcal{M}'} \right].$$

Let $\lambda > 0$. Applying the Chernoff bound (Lemma G.1), we obtain

$$\Pr \left[\mathcal{E}(\mathcal{M}', \mathcal{F}') \mid \boldsymbol{z}, \boldsymbol{s}_{\mathcal{M}'} \right] \leq \mathbb{E}_{\mathbf{s}_{\mathcal{F}'}} \left[\exp \left[\lambda \left(\|\boldsymbol{z}\|^2 - \|\boldsymbol{s}_{\mathcal{M}'} - \mathbf{s}_{\mathcal{F}'} + \boldsymbol{z}\|^2 \right) \right] \right].$$

and the vectors $\boldsymbol{z}$ and $\boldsymbol{s}_{\mathcal{M}'}$ are fixed. Using Corollary G.1, we get

$$\Pr \left[\mathcal{E}(\mathcal{M}', \mathcal{F}') \mid \boldsymbol{z}, \boldsymbol{s}_{\mathcal{M}'} \right] \leq \exp \left[E_1(\boldsymbol{s}_{\mathcal{M}'}, \boldsymbol{z}) \right],$$

where

$$E_1(\boldsymbol{s}_{\mathcal{M}'}, \boldsymbol{z}) = \lambda \left(\|\boldsymbol{z}\|^2 - \frac{\|\boldsymbol{s}_{\mathcal{M}'} + \boldsymbol{z}\|^2}{1 + 2tP'\lambda} \right).$$

Substituting this result into (4.8), we obtain

$$\Pr \left[\mathcal{E}(\mathcal{M}') \mid \boldsymbol{z}, \boldsymbol{s}_{\mathcal{M}'} \right] \leq \binom{M - K_a}{t}^{\rho_2} \exp \left[\rho_2 E_1(\boldsymbol{s}_{\mathcal{M}'}, \boldsymbol{z}) \right].$$

Taking here expectation[3] over $\mathbf{s}_{\mathcal{M}'}$ (see (4.7)), we get

$$\Pr \left[\mathcal{E}(\mathcal{M}') \mid \boldsymbol{z} \right] \leq \binom{M - K_a}{t}^{\rho_2} \exp \left[b \|\boldsymbol{z}\|^2 - na \right].$$

Substituting this result into (4.7), we finally obtain

$$\Pr \left[\mathcal{E}_t \mid \boldsymbol{z} \right] \leq \binom{K_a}{t}^{\rho_1} \binom{M - K_a}{t}^{\rho_1 \rho_2} \exp \left[\rho_1 (b \|\boldsymbol{z}\|^2 - na) \right].$$

Taking expectation over $\mathbf{z}$, we obtain the theorem statement. □

Remark 4.2. *One may ask whether the provided sequence of expectations is optimal. In the chain rule, we can use any sequence, for example, by first computing the expectation over* $\mathbf{z}$*. Let us provide some intuition. There are* $\binom{M-K_a}{t}$ *possible choices for* $\mathcal{F}$*, and this binomial coefficient is extremely large. Therefore, it is reasonable to start with* $\mathbf{s}_{\mathcal{F}}$*, as done in the proof, since Gallager's trick will hopefully significantly reduce this large coefficient.*

[3] We again apply Corollary G.1.

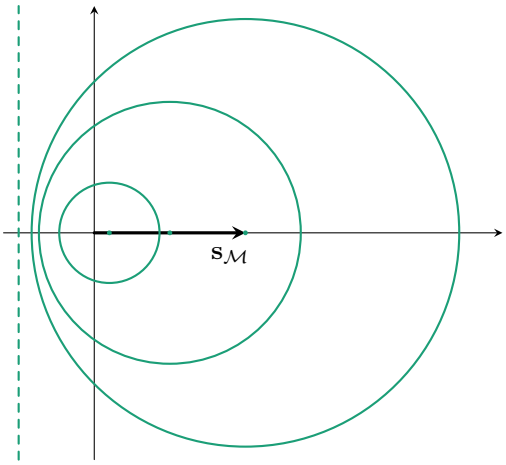

Figure 4.2: Illustration of the ball presented in (4.9) for $\beta = 0$ and different values of α such that $\alpha/1-\alpha \in \{0.1, 0.5, 1.0\}$. Each ball is represented by a green line, and a larger radius corresponds to a larger value of α. The dashed line corresponds to the extreme case $\alpha = 1$. For $\beta > 0$, the radius of each ball increases, and the dashed line shifts to the left. Note that the zero vector is always inside the ball for any $0 < \alpha \leq 1$ and $\beta \geq 0$. Hence, the intersection of these balls is never empty.

4.3.2 The bound based on Fano's trick

Let us fix $0 \leq \alpha < 1$, $\beta \geq 0$. We introduce the following "good" region:

$$\mathcal{B} = \bigcap_{\mathcal{M} \subseteq [K_a]} \mathcal{B}(\mathcal{M}), \quad \mathcal{B}(\mathcal{M}) = \left\{ \alpha \left\| \mathbf{s}_{\mathcal{M}} + \mathbf{z} \right\|^2 + \beta n > \left\| \mathbf{z} \right\|^2 \right\}. \tag{4.9}$$

Remark 4.3 (Visualization of the "good" region)**.** *Let us consider a geometric interpretation of the region* (4.9)*. Suppose the vector* $\mathbf{s}_{\mathcal{M}}$ *is fixed (since we consider a union over all* $\binom{K_a}{t}$ *missed codewords in the subsequent theorem proof). Thus, given* $\mathbf{s}_{\mathcal{M}}$*, the condition* (4.9) *for a single* $\mathcal{B}(\mathcal{M})$ *can be rewritten as follows (see Fig.* 4.2*).*

For $0 < \alpha < 1$*, we have:*

$$\left\{ \left\| \mathbf{z} - \mathbf{v}_0 \right\|^2 < r^2 \right\}, \quad \mathbf{v}_0 = \frac{\alpha}{1-\alpha} \mathbf{s}_{\mathcal{M}}, \quad r^2 = \frac{\alpha \left\| \mathbf{s}_{\mathcal{M}} \right\|^2 + \beta n (1-\alpha)}{(1-\alpha)^2},$$

which defines a ball.

Note that α *controls both the center* $\mathbf{v}_0$ *and the radius* r *of the sphere, while* β *affects only the radius. Additionally, each ball contains the zero point; hence, their intersection is non-empty.*

For $\alpha = 1$*, we have:*

$$\left\{ \mathbf{s}_{\mathcal{M}}^T \cdot \mathbf{z} > -\frac{\left\| \mathbf{s}_{\mathcal{M}} \right\|^2 + \beta n}{2} \right\}, \quad \alpha = 1,$$

which defines a hyperplane.

Remark 4.4. *Let us provide some comments on the choice of the "good" region:*

- *Adding $\mathbf{s}_{\mathcal{C}}$ to the region description does not lead to a performance improvement, since correctly received codewords do not appear in* (4.4)*. However, the use of $\mathbf{s}_{\mathcal{C}}$ may be beneficial in the case of fading, as shown in* (6.12)*.*
- *The region description does not include $\mathbf{s}_{\mathcal{F}}$ to avoid a large binomial coefficient $\binom{M-K_a}{t}$. Indeed, if $\mathbf{s}_{\mathcal{F}}$ were included in* (4.9)*, then calculating $\Pr[\mathcal{B}^c]$ in eq.* (4.5) *would involve taking the complement of an intersection of a large number of events for $\mathcal{F} \subseteq [M] \setminus [K_a]$. After applying a union bound, this large binomial coefficient would inevitably appear.*

Theorem 4.2 (Achievability based on Fano's trick [85])**.** *Fix $P' < P$. There exists a codebook $\boldsymbol{X}^* \in \mathcal{G}(M, n, P')$ satisfying the power constraint P' and providing P_e using eq.* (4.2)*, where p_0 is bounded by* (4.6)*, and $\Pr[\mathcal{E}_t] \leq p_t$, where*

$$p_t = \inf_{\alpha,\beta \geq 0} \left(p_{1,t} + p_{2,t}\right),$$

$$p_{1,t} = \inf_{u,v>0,\lambda_{\boldsymbol{A}}>0} \exp\left[-n\left(-R_1 - R_2 + \frac{1}{2}\log\det\left(\boldsymbol{I}_3 - 2\boldsymbol{A}\right) - v\beta\right)\right],$$

$$p_{2,t} = \inf_{\delta>0,\lambda_{\boldsymbol{B}}>0} \exp\left[-n\left(-R_2 + \frac{1}{2}\log\det\left(\boldsymbol{I}_2 - 2\delta\boldsymbol{B}\right) + \delta\beta\right)\right],$$

$$R_1 = \frac{1}{n}\log\binom{M-K_a}{t}, \quad R_2 = \frac{1}{n}\log\binom{K_a}{t},$$

$$\boldsymbol{A} = \begin{pmatrix} (\alpha-1)\,v & (\alpha v - u)\sqrt{P't} & u\sqrt{P't} \\ (\alpha v - u)\sqrt{P't} & (\alpha v - u)\,P't & uP't \\ u\sqrt{P't} & uP't & -uP't \end{pmatrix},$$

$$\boldsymbol{B} = \begin{pmatrix} 1-\alpha & -\alpha\sqrt{P't} \\ -\alpha\sqrt{P't} & -\alpha P't \end{pmatrix}.$$

By $\lambda_{\boldsymbol{A}}$ and $\lambda_{\boldsymbol{B}}$, we mean the minimum eigenvalues of $\boldsymbol{I}_3 - 2\boldsymbol{A}$ and $\boldsymbol{I}_2 - 2\delta\boldsymbol{B}$ accordingly.

Proof. We start by applying Fano's trick (4.5) for $\mathcal{E}_t$, where $\mathcal{B}$ is given by (4.9).

Let us start with $\Pr[\mathcal{E}_t \bigcap \mathcal{B}]$. We have

$$\begin{aligned}\Pr\left[\mathcal{E}_t \bigcap \mathcal{B}\right] &\leq \Pr\left[\bigcup_{\mathcal{M},\mathcal{F}} \mathcal{E}(\mathcal{M},\mathcal{F}) \bigcap \mathcal{B}(\mathcal{M})\right] \\ &\leq \binom{K_a}{t}\binom{M-K_a}{t} \Pr\left[\mathcal{E}(\mathcal{M}',\mathcal{F}') \bigcap \mathcal{B}(\mathcal{M}')\right],\end{aligned}$$

where $\mathcal{M}' = [t]$, $\mathcal{F}' = [K_a + t] \backslash [K_a]$. The last transition is due to the union bound and symmetry.

Consider $\Pr[\mathcal{E}(\mathcal{M}',\mathcal{F}') \bigcap \mathcal{B}(\mathcal{M}')]$. Recall that

$$\mathcal{E}(\mathcal{M}',\mathcal{F}') = \left\{\|\mathbf{z}\|^2 - \|\mathbf{s}_{\mathcal{M}'} - \mathbf{s}_{\mathcal{F}'} + \mathbf{z}\|^2 \geq 0\right\}$$

and

$$\mathcal{B}(\mathcal{M}') = \left\{ \alpha \left\| \mathbf{s}_{\mathcal{M}'} + \mathbf{z} \right\|^2 + \beta n > \left\| \mathbf{z} \right\|^2 \right\}.$$

Let us introduce the vector

$$\boldsymbol{\eta}^T = \left(\mathbf{z}^T,\, \mathbf{s}_{\mathcal{M}'}^T / \sqrt{P't},\, \mathbf{s}_{\mathcal{F}'}^T / \sqrt{P't} \right), \quad \boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_{3n}).$$

Using this notation, we can rewrite the events $\mathcal{E}(\mathcal{M}', \mathcal{F}')$ and $\mathcal{B}(\mathcal{M}')$ as follows:

$$\mathcal{E}(\mathcal{M}', \mathcal{F}') = \left\{ \boldsymbol{\eta}^T \boldsymbol{A}_e \boldsymbol{\eta} \geq 0 \right\}$$

and

$$\mathcal{B}(\mathcal{M}') = \left\{ \boldsymbol{\eta}^T \boldsymbol{A}_r \boldsymbol{\eta} + \beta n > 0 \right\},$$

where

$$\boldsymbol{A}_e = \begin{pmatrix} \mathbf{0} & -\sqrt{P't}\boldsymbol{I}_n & \sqrt{P't}\boldsymbol{I}_n \\ -\sqrt{P't}\boldsymbol{I}_n & -P't\boldsymbol{I}_n & P't\boldsymbol{I}_n \\ \sqrt{P't}\boldsymbol{I}_n & P't\boldsymbol{I}_n & -P't\boldsymbol{I}_n \end{pmatrix},$$

$$\boldsymbol{A}_r = \begin{pmatrix} (\alpha - 1)\,\boldsymbol{I}_n & \alpha\sqrt{P't}\boldsymbol{I}_n & \mathbf{0} \\ \alpha\sqrt{P't}\boldsymbol{I}_n & \alpha P't\boldsymbol{I}_n & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix}.$$

Now, we apply the Chernoff bound (Lemma G.1):

$$\Pr\left[\mathcal{E}(\mathcal{M}', \mathcal{F}') \bigcap \mathcal{B}(\mathcal{M}') \right] \leq \inf_{u,v>0} \mathbb{E}_{\boldsymbol{\eta}} \exp\left[\boldsymbol{\eta}^T \boldsymbol{A}_n \boldsymbol{\eta} + v\beta n \right],$$

where $\boldsymbol{A}_n = u\boldsymbol{A}_e + v\boldsymbol{A}_r$.

Using Lemma G.2 (with $\boldsymbol{b}$ equal to the zero vector), we obtain

$$\Pr\left[\mathcal{E}(\mathcal{M}', \mathcal{F}') \bigcap \mathcal{B}(\mathcal{M}') \right] \leq \inf_{u,v>0} \exp\left[-\frac{1}{2} \log\det\left(\boldsymbol{I}_{3n} - 2\boldsymbol{A}_n \right) + v\beta n \right],$$

when $\lambda_{\min}\left(\boldsymbol{I}_{3n} - 2\boldsymbol{A}_n \right) > 0$.

Let us note that the matrix $\boldsymbol{A}_n$ can be represented as $\boldsymbol{I}_n \otimes \boldsymbol{A}$ by reordering its columns and rows. So, we can represent $(\boldsymbol{I}_{3n} - 2\boldsymbol{A}_n)$ as $\boldsymbol{I}_n \otimes (\boldsymbol{I}_3 - 2\boldsymbol{A})$, and it is clear that

$$-1/2 \log\det\left(\boldsymbol{I}_{3n} - 2\boldsymbol{A}_n \right) = -n/2 \log\det\left(\boldsymbol{I}_3 - 2\boldsymbol{A} \right).$$

Thus, $\Pr\left[\mathcal{E}_t \bigcap \mathcal{B} \right] \leq p_{1,t}$.

Similarly, we deal with

$$\Pr\left[\mathcal{B}^c \right] \leq \binom{K_a}{t} \Pr\left[\mathcal{B}^c(\mathcal{M}') \right] \leq p_{2,t}.$$

This concludes the proof. □

4.3.3 Combination of the tricks

One may notice that Gallager's and Fano's tricks can be applied jointly. In this section, we present one possible variant, which is a straightforward modification of the bound for SPARCs provided in [86].

Theorem 4.3 (Achievability based on the combination of Fano's and Gallager's tricks [86])**.** *Fix $P' < P$. There exists a codebook $\boldsymbol{X}^* \in \mathcal{G}(M, n, P')$ satisfying the power constraint P' and providing P_e using eq. (4.2), where p_0 is bounded by (4.6), and $\Pr[\mathcal{E}_t] \le p_t$, where*

$$p_t = \inf_{\alpha,\tau \ge 0} (p_{1,t} + p_{2,t}),$$

$$p_{1,t} = \inf_{0 \le \gamma \le 1} \exp\left[-n\left(-\gamma R_1 - R_2 + E_1(\gamma)\right)\right],$$

$$p_{2,t} = \inf_{\delta > 0} \exp\left[-nE_2(\delta)\right],$$

$$R_1 = \frac{1}{n}\log\binom{M-K_a}{t}, \quad R_2 = \frac{1}{n}\log\binom{K_a}{t},$$

$$E_1(\gamma) = \frac{1}{2}\left(\gamma\log(1+P't) + \log\left(1-\gamma^2(1-\rho_e^2)\right) - \gamma\tau\right),$$

$$E_2(\delta) = \frac{1}{2}\left(\log\left(1-\delta^2(1-\rho_r^2)\right) + \tau\delta\right),$$

$$\rho_e^2 = \frac{(1+\alpha P't)^2}{(1+\alpha^2 K_a P')(1+tP')}, \quad \rho_r^2 = \frac{1}{1+\alpha^2 K_a P'}.$$

Proof. Let $\alpha, \beta > 0$. In what follows, we demonstrate how to choose these constants. We start by applying Fano's trick (4.5) for $\mathcal{E}_t$, where

$$\begin{aligned}\mathcal{B} &= \left\{\beta\|\mathbf{y} - (1-\alpha)\mathbf{s}_{\mathcal{T}}\|^2 - \|\mathbf{y} - \mathbf{s}_{\mathcal{T}}\|^2 + \tau n \ge 0\right\} \\ &= \left\{\beta\|\mathbf{z} + \alpha\mathbf{s}_{\mathcal{T}}\|^2 - \|\mathbf{z}\|^2 + \tau n \ge 0\right\},\end{aligned}$$

and

$$\beta = \beta'/(1+\alpha^2 K_a P).$$

Similar to the proofs of Theorems 4.2 and 4.1, we can write ($0 \le \gamma \le 1$)

$$\begin{aligned}&\Pr\left[\mathcal{E}_t \bigcap \mathcal{B}\right] \le \Pr\left[\bigcup_{\mathcal{M},\mathcal{F}} \mathcal{E}(\mathcal{M},\mathcal{F}) \bigcap \mathcal{B}(\mathcal{M})\right] \\ &\le \binom{K_a}{t}\mathbb{E}_{\mathbf{z},\mathbf{s}_{\mathcal{T}}}\left[\left(\binom{M-K_a}{t}\Pr\left[\mathcal{E}(\mathcal{M}',\mathcal{F}') \bigcap \mathcal{B} \,\middle|\, \boldsymbol{s}_{\mathcal{T}}, \boldsymbol{z}\right]\right)^{\gamma}\right]. \\ &= \binom{K_a}{t}\binom{M-K_a}{t}^{\gamma}\mathbb{E}_{\mathbf{z},\mathbf{s}_{\mathcal{T}}}\left[\left(\Pr\left[\mathcal{E}(\mathcal{M}',\mathcal{F}') \bigcap \mathcal{B} \,\middle|\, \boldsymbol{s}_{\mathcal{T}}, \boldsymbol{z}\right]\right)^{\gamma}\right].\end{aligned}$$

Recall that

$$\mathcal{E}(\mathcal{M}', \mathcal{F}') = \left\{ \|\mathbf{z}\|^2 - \|\mathbf{s}_{\mathcal{M}'} - \mathbf{s}_{\mathcal{F}'} + \mathbf{z}\|^2 \geq 0 \right\}.$$

Proceeding further,

$$\begin{aligned}
&\Pr\left[\mathcal{E}(\mathcal{M}', \mathcal{F}') \bigcap \mathcal{B} \;\middle|\; \boldsymbol{s}_{\mathcal{T}}, \boldsymbol{z}\right] \\
&\leq \Pr\left[\beta\|\boldsymbol{z} + \alpha \boldsymbol{s}_{\mathcal{T}}\|^2 - \|\boldsymbol{s}_{\mathcal{M}'} - \mathbf{s}_{\mathcal{F}'} + \boldsymbol{z}\|^2 + \tau n \geq 0 \;\middle|\; \boldsymbol{s}_{\mathcal{T}}, \boldsymbol{z}\right],
\end{aligned}$$

where the last transition follows from the fact that

$$\Pr[\xi_1 \geq 0, \xi_2 \geq 0] \leq \Pr[\xi_1 + \xi_2 \geq 0].$$

Applying the Chernoff bound with $\lambda = \frac{1}{2}$[4], we obtain

$$\begin{aligned}
&\Pr\left[\mathcal{E}(\mathcal{M}', \mathcal{F}') \bigcap \mathcal{B} \;\middle|\; \boldsymbol{s}_{\mathcal{T}}, \boldsymbol{z}\right] \\
&\leq \exp\left[\frac{1}{2}\left(\beta\|\boldsymbol{z} + \alpha \boldsymbol{s}_{\mathcal{T}}\|^2 - \frac{\|\boldsymbol{s}_{\mathcal{M}} + \boldsymbol{z}\|^2}{1 + P't}\right) + \frac{\tau n}{2} - \frac{n}{2}\log(1 + P't)\right].
\end{aligned}$$

Raising the estimate to the power γ and proceeding with the expectation over $\mathbf{z}$ and $\mathbf{s}_{\mathcal{T}}$, we introduce the notation

$$A = \exp\left[\frac{\gamma\tau n}{2} - \frac{\gamma n}{2}\log(1 + P't)\right],$$

and obtain

$$\begin{aligned}
&\mathbb{E}_{\mathbf{z}, \mathbf{s}_{\mathcal{T}}}\left[\Pr\left[\mathcal{E}(\mathcal{M}', \mathcal{F}') \bigcap \mathcal{B} \;\middle|\; \boldsymbol{s}_{\mathcal{T}}, \boldsymbol{z}\right]\right] \\
&\leq A\mathbb{E}_{\mathbf{z}, \mathbf{s}_{\mathcal{T}}}\left[\exp\left[\frac{\gamma}{2}\left(\beta\|\mathbf{z} + \alpha\mathbf{s}_{\mathcal{T}}\|^2 - \frac{\|\boldsymbol{s}_{\mathcal{M}} + \boldsymbol{z}\|^2}{1 + P't}\right)\right]\right] \\
&\leq A\mathbb{E}_{\boldsymbol{\eta}_1, \boldsymbol{\eta}_2}\left[\exp\left[\frac{\gamma\beta'}{2}\|\boldsymbol{\eta}_1\|^2 - \frac{\gamma}{2}\|\boldsymbol{\eta}_2\|^2\right]\right] \\
&= A\exp\left[-\frac{n}{2}\log\left(1 - \gamma\beta' + \gamma - \gamma^2\beta'(1 - \rho_e^2)\right)\right],
\end{aligned}$$

where the last transition follows from Corollary G.2, and ρ_e is defined as

$$\mathrm{Cov}(\boldsymbol{\eta}_1, \boldsymbol{\eta}_2) = \mathbb{E}[\boldsymbol{\eta}_1^T \boldsymbol{\eta}_2] = \rho_e \boldsymbol{I}_n,$$

where

$$\rho_e^2 = \frac{(1 + \alpha P't)^2}{(1 + \alpha^2 K_a P')(1 + tP')}. \tag{4.10}$$

Now consider

$$\Pr[\mathcal{B}^c] = \Pr\left[\beta\|\mathbf{z} + \alpha\mathbf{s}_{\mathcal{T}}\|^2 - \|\mathbf{z}\|^2 + \tau n < 0\right]$$

[4]One can optimize over λ to improve the bound, but we follow the original approach.

$$
\begin{aligned}
&\leq \mathbb{E}_{\mathbf{z},\mathbf{s}_\mathcal{T}}\left[\exp\left[\frac{\delta}{2}\left\|\mathbf{z}\right\|^2 - \frac{\delta}{2}\beta\|\mathbf{z}-\alpha\mathbf{s}_\mathcal{T}\|^2 - \frac{\delta\tau}{2}n\right]\right] \\
&= \exp\left[-\frac{\delta\tau n}{2}\right]\mathbb{E}_{\boldsymbol{\eta}_1,\boldsymbol{\eta}_2}\left[\exp\left[\frac{\delta}{2}\left\|\boldsymbol{\eta_1}\right\|^2 - \frac{\delta}{2}\beta'\|\boldsymbol{\eta}_2\|^2\right]\right] \\
&= \exp\left[-\frac{\delta\tau n}{2}\right]\exp\left[-\frac{n}{2}\log\left(1-\delta+\delta\beta'-\delta^2\beta'(1-\rho_r^2)\right)\right],
\end{aligned}
$$

as

$$
\rho_r^2 = \frac{1}{1+\alpha^2 K_a P'}.
$$

For simplicity, the authors of [86] chose $\beta' = 1$. We follow the same choice but note that optimizing this parameter may further improve the bound.

Thus, collecting all terms together and setting $\beta' = 1$, we obtain the theorem statement.

The final region is given by

$$
\begin{aligned}
\mathcal{B} &= \left\{\frac{\left\|\mathbf{y}-(1-\alpha)\mathbf{s}_\mathcal{T}\right\|^2}{1+\alpha^2 K_a P} - \left\|\mathbf{y}-\mathbf{s}_\mathcal{T}\right\|^2 + \tau n \geq 0\right\} \\
&= \left\{\frac{\left\|\mathbf{z}+\alpha\mathbf{s}_\mathcal{T}\right\|^2}{1+\alpha^2 K_a P} - \left\|\mathbf{z}\right\|^2 + \tau n \geq 0\right\}.
\end{aligned}
$$

□

Remark 4.5. *The authors of [86] show that $\alpha = t/K_a$ maximizes the expression* (4.10)*, resulting in*

$$
\rho_e^2 = \frac{1+\frac{t^2}{K_a}P'}{1+tP'}, \quad \rho_r^2 = \frac{1}{1+\frac{t^2}{K_a}P'},
$$

which are used in the original theorem. However, as numerical results (see Section 4.6) show, further optimization of α allows for better performance, particularly in the regime where the number of active users K_a is small.

Throughout Theorems 4.1, 4.2, and 4.3, we considered different approaches to estimate the achievable energy efficiency for the URA setup. As we further demonstrate through numerical analysis in Section 4.6, all these bounds yield similar energy efficiency estimates, with the gap being no greater than 0.25 dB. However, the results presented in Theorem 4.2 can similarly be generalized to a binary codebook, as we demonstrate later in Theorem 4.4.

Let us also recall the converse bounds previously defined in Theorems 3.1 and 3.2. As we further demonstrate, there is a gap between achievability and converse, with the difference being up to 1.2 dB. We refer the reader to Section 4.6 for a detailed discussion and comparison of the presented bounds.

4.4 The case of binary codebooks

We note that the bounds from the previous section assume the use of Gaussian signaling (or codebook), which is not ideal for practical applications. Indeed, a Gaussian codebook may be suboptimal, especially in the case of short blocks, where different codewords may have significantly different energy levels. To overcome this issue, codewords can be sampled from a *power shell* or from a uniform distribution on the multidimensional sphere (see Appendix B.2.2), ensuring equal power for all codewords. Moreover, Gaussian codebooks may have limited practical applications, as storing such a codebook and generating the transmitted signal can be computationally complex.

In this section, we address this problem and consider binary signaling (or BPSK modulation). We aim to answer a natural question: does binary signaling restrict energy efficiency? In other words, we seek to establish the fundamental limits for binary-input GMAC. This question was previously studied in [85], and we follow the description presented in that paper.

We start with the definition of the binary codebook ensemble.

Definition 4.2. *Let $\mathcal{B}(M,n,P)$ be the ensemble of binary codebooks of size $n \times M$, where each element is sampled i.i.d. from $\left\{-\sqrt{P},\sqrt{P}\right\}$ with probability $1/2$.*

Remark 4.6. *Note that there is no need to check the power constraint for this ensemble as it is fulfilled by design.*

Now we are ready to formulate and prove the main result.

Theorem 4.4 (Achievability for binary codebook based on Fano's trick [85])**.** *There exists a codebook $\boldsymbol{X}^* \in \mathcal{B}(M,n,P)$ providing P_e using eq. (4.2), where $p_0 = 0$, and $\Pr[\mathcal{E}_t] \le p_{1,t} + p_{2,t}$, where*

$$p_{1,t} = \binom{M-K_a}{t}\binom{K_a}{t} \inf_{u,v>0} \Bigg[\exp\left[-n\zeta(\alpha,\beta,v)\right] \times \\ \times \left(\sum_{m=-t}^{t} \sum_{f=-t}^{t} \rho_m \rho_f \exp\left[P\phi(\alpha,u,v,m,f)\right] \right)^n \Bigg],$$

where

$$\zeta(\cdot) = \frac{1}{2}\log\left(1-2v(\alpha-1)\right) - \beta v,$$

$$\phi(\cdot) = \left(2\frac{\left(\alpha v m - u(m-f)\right)^2}{1-2v(\alpha-1)} + \alpha v m^2 - u(m-f)^2 \right),$$

and

$$p_{2,t} = \binom{K_a}{t} \inf_{\delta>0} \left[\exp\left[-n\xi(\alpha,\beta,\delta)\right] \left(\sum_{m=-t}^{t} \rho_m \exp\left[P\psi(\alpha,\delta,m)\right] \right)^n \right],$$

where

$$\xi(\cdot) = \frac{1}{2}\log\left(1-2\delta(1-\alpha)\right) + \delta\beta, \quad \psi(\cdot) = \delta\alpha\frac{2\delta-1}{1-2\delta(1-\alpha)}m^2,$$

$$\rho_i = \begin{cases} \binom{t}{\frac{1}{2}(i+t)} 2^{-t}, & i \in \{2j-t, \forall j : 0 \le j \le t\} \\ 0, & otherwise \end{cases}$$

with the following conditions holding:

$$1 - 2v\left(\alpha - 1\right) > 0, \quad 1 - 2\delta\left(1 - \alpha\right) > 0.$$

Proof. We utilize the approach from Theorem 4.2 with exactly the same good region (4.9) in Fano's trick (4.5) for $\mathcal{E}_t$.

We now show how to modify the proof for the case of a binary codebook. Let us start with $\Pr\left[\mathcal{E}_t \bigcap \mathcal{B}\right]$. We have

$$\Pr\left[\mathcal{E}_t \bigcap \mathcal{B}\right] \le \binom{K_a}{t}\binom{M-K_a}{t} \Pr\left[\mathcal{E}(\mathcal{M}', \mathcal{F}') \bigcap \mathcal{B}(\mathcal{M}')\right],$$

where $\mathcal{M}' = [t]$, $\mathcal{F}' = [K_a + t] \backslash [K_a]$, and the events are defined as follows:

$$\begin{aligned} \mathcal{E}(\mathcal{M}', \mathcal{F}') &= \left\{ \|\mathbf{z}\|^2 - \|\mathbf{s}_{\mathcal{M}'} - \mathbf{s}_{\mathcal{F}'} + \mathbf{z}\|^2 \ge 0 \right\}, \\ \mathcal{B}(\mathcal{M}') &= \left\{ \alpha \|\mathbf{s}_{\mathcal{M}'} + \mathbf{z}\|^2 + \beta n > \|\mathbf{z}\|^2 \right\}. \end{aligned}$$

The major difference from the proof of Theorem 4.2 is that $\mathbf{s}_{\mathcal{M}'}$ and $\mathbf{s}_{\mathcal{F}'}$ are not Gaussian. Expanding the norms, we rewrite the events as

$$\begin{aligned} \mathcal{E}(\mathcal{M}', \mathcal{F}') &= \left\{ -2\left(\mathbf{s}_{\mathcal{M}'} - \mathbf{s}_{\mathcal{F}'}\right)^T \mathbf{z} - \|\mathbf{s}_{\mathcal{M}'} - \mathbf{s}_{\mathcal{F}'}\|^2 > 0 \right\}, \\ \mathcal{B}(\mathcal{M}') &= \left\{ \left(\alpha - 1\right) \|\mathbf{z}\|^2 + 2\alpha \mathbf{s}_{\mathcal{M}'}^T \mathbf{z} + \alpha \|\mathbf{s}_{\mathcal{M}'}\|^2 + \beta n > 0 \right\}. \end{aligned}$$

Thus, we can estimate the probability $\Pr\left[\mathcal{E}(\mathcal{M}', \mathcal{F}') \bigcap \mathcal{B}(\mathcal{M}')\right]$ using the Chernoff bound for the joint events in the following way:

$$\begin{aligned} \Pr\left[\mathcal{E}(\mathcal{M}', \mathcal{F}') \bigcap \mathcal{B}(\mathcal{M}')\right] \le \inf_{u,v>0} \mathbb{E}_{\mathbf{z}, \mathbf{s}_{\mathcal{M}'}, \mathbf{s}_{\mathcal{F}'}} \exp\Big[&\left(\alpha - 1\right) v \|\mathbf{z}\|^2 \\ &+ 2\left(\alpha v \mathbf{s}_{\mathcal{M}'}^T - u\left(\mathbf{s}_{\mathcal{M}'} - \mathbf{s}_{\mathcal{F}'}\right)^T\right)\mathbf{z} \\ &+ \alpha v \|\mathbf{s}_{\mathcal{M}'}\|^2 - u\|\mathbf{s}_{\mathcal{M}'} - \mathbf{s}_{\mathcal{F}'}\|^2 + \beta v n \Big]. \end{aligned}$$

First, let us take the expectation over $\mathbf{z}$ while treating $\mathbf{s}_{\mathcal{M}'}$ and $\mathbf{s}_{\mathcal{F}'}$ as fixed. To do this, we apply Lemma G.2 and obtain

$$\begin{aligned} \Pr\left[\mathcal{E}(\mathcal{M}', \mathcal{F}') \bigcap \mathcal{B}(\mathcal{M}')\right] \le \inf_{u,v>0} \mathbb{E}_{\mathbf{s}_{\mathcal{M}'}, \mathbf{s}_{\mathcal{F}'}} \exp\Big[&-\frac{n}{2}\log\left(1 - 2\left(\alpha - 1\right) v\right) \\ &+ 2\frac{\|\alpha v \mathbf{s}_{\mathcal{M}'} - u\left(\mathbf{s}_{\mathcal{M}'} - \mathbf{s}_{\mathcal{F}'}\right)\|^2}{1 - 2\left(\alpha - 1\right) v} \\ &+ \alpha v \|\mathbf{s}_{\mathcal{M}'}\|^2 - u\|\mathbf{s}_{\mathcal{M}'} - \mathbf{s}_{\mathcal{F}'}\|^2 + \beta v n \Big]. \end{aligned} \tag{4.11}$$

To take the expectation over $\mathbf{s}_{\mathcal{M}'}$ and $\mathbf{s}_{\mathcal{F}'}$, let us consider

$$\mathbf{s}_{\mathcal{M}'} = \left(s_{\mathcal{M}',1}, s_{\mathcal{M}',2}, \ldots, s_{\mathcal{M}',n}\right)$$

(for $\mathbf{s}_{\mathcal{F}'}$, the arguments are similar), which is the sum of t codewords, each satisfying the power constraint P for every coordinate. Then, we observe that

$$s_{\mathcal{M}',l} \in \left\{(2j-t)\sqrt{P}, \forall j : 0 \leq j \leq t\right\}, \quad \forall l \in [n].$$

Changing the sign of j terms modifies the sum by $2j\sqrt{P}$ in absolute value. For instance, if all t terms initially have the value $-\sqrt{P}$ and we randomly select j terms to change to $\sqrt{P}$, then the sum will be $(2j-t)\sqrt{P}$.

The number of ways to obtain the sum $s_{\mathcal{M}',l} = (2j-t)\sqrt{P}$, $\forall j : 0 \leq j \leq t$ is given by the binomial coefficient $\binom{t}{j}$. Since each of the t terms can independently take one of two values, the total number of possible combinations is 2^t. Thus, the probability distribution of $s_{\mathcal{M}',l}$ is:

$$\rho_i = \Pr\left[s_{\mathcal{M}',l} = i\sqrt{P}\right] = \frac{1}{2^t}\binom{t}{\frac{i+t}{2}}$$

for $i \in \{2j-t, \forall j : 0 \leq j \leq t\}$ and zero otherwise.

Now, we note that

$$\mathbb{E}\left[\exp\left[\|\mathbf{s}_{\mathcal{M}'}\|^2\right]\right] = \sum_{(i_1,i_2,\ldots,i_n)} \prod_{l=1}^{n} \rho_{i_l} \exp\left[i_l^2 P\right] = \left(\sum_{i=-t}^{t} \rho_i \exp\left[i^2 P\right]\right)^n.$$

Finally, applying this approach to (4.11), we obtain $\Pr\left[\mathcal{E}_t \bigcap \mathcal{B}\right] \leq p_{1,t}$. Similarly, we have $\Pr\left[\mathcal{B}^c\right] \leq p_{2,t}$. □

In Section 4.6, we provide a numerical comparison of this bound and find that it yields the same energy efficiency estimates as the bound for the Gaussian codebook presented in Theorem 4.2.

4.5 Further comments and discussion

4.5.1 Random number of active users

All the bounds considered in this chapter assume that K_a is fixed and known to the receiver. The authors of [72] pose a very important and timely question: How do the fundamental limits change when K_a is *random* and *unknown* to the receiver? For this case, they derive the random coding bound (RCB), which is based on Gallager's trick. Let us discuss the major differences in comparison to [5].

We start with the decoding rule. Recall the rule from [5]:

$$\mathcal{R} = \underset{\mathcal{R}' \subseteq [M], |\mathcal{R}'| = K_a}{\arg\min} \left\| \mathbf{y} - \boldsymbol{s}_{\mathcal{R}'} \right\|^2,$$

which is reasonable since the receiver knows K_a.

The most straightforward way to generalize this rule is to consider:

$$\begin{aligned} \mathcal{R} &= \underset{\mathcal{R}' \subseteq [M]}{\arg\max} \left\{ \exp\left[-\frac{\left\| \mathbf{y} - \boldsymbol{s}_{\mathcal{R}'} \right\|^2}{2} \right] \Pr[\mathcal{R}'] \right\} \\ &= \underset{\mathcal{R}' \subseteq [M]}{\arg\min} \left\{ \left\| \mathbf{y} - \boldsymbol{s}_{\mathcal{R}'} \right\|^2 - 2 \log \Pr[\mathcal{R}'] \right\}, \end{aligned}$$

where $\Pr[\mathcal{R}']$ is the prior distribution of K_a.

The rule above is difficult to analyze because it requires considering all possible $\mathcal{R}'$, which makes the union bound greatly overestimate the desired probability. To address this, the authors of [72] propose a two-stage approach. First, they estimate K_a using the following rule:

$$\hat{K}_a = \underset{K \in [K_l, K_u]}{\arg\max}\ m(\boldsymbol{y}, K),$$

where $m(\boldsymbol{y}, K)$ is a suitably chosen metric, and K_l and K_u are appropriately selected lower and upper bounds for K_a.

Then, given $\hat{K}_a$, the receiver outputs:

$$\mathcal{R} = \underset{\mathcal{R}' \subseteq [M], \underline{K} \leq |\mathcal{R}'| \leq \overline{K}}{\arg\min} \left\| \mathbf{y} - \boldsymbol{s}_{\mathcal{R}'} \right\|^2,$$

where $\underline{K} = \max\{K_l, \hat{K}_a - r\}$, $\overline{K} = \min\{K_u, \hat{K}_a + r\}$, and r is a nonnegative integer referred to as the *decoding radius*.

We do not present the final theorem here but note that the techniques are very similar to those in [5]. However, in this case, one must consider both P_e and P_f since the output list size is a random variable. We discuss the bound for random K_a when presenting the numerical results in Section 4.6.

4.5.2 Gordon's lemma and achievability improvements

Let us once again look at Fano's approach and the good region from Theorem 4.2. Recall that we decided not to use $\mathbf{s}_{\mathcal{F}}$ in the region description to avoid a large binomial coefficient $\binom{M-K_a}{t}$. Are there any other ways to calculate the probability $\Pr[\mathcal{B}^c]$ without using the union bound and the large binomial coefficient? In this section, we describe the approach from [87], which relies on Gordon's inequality [88] for the expectation of the minimum of the Gaussian process. As shown in [85], the techniques from [87] can be reformulated as an instance of Fano's method.

As an example, let us consider the following simplified region:

$$\mathcal{B} = \{\|\mathbf{s}_{\mathcal{F}}\| \geq \sqrt{\beta n}\}.$$

Recall that

$$\Pr\left[\mathcal{E}_t\right] \leq \Pr\left[\mathcal{E}_t \bigcap \mathcal{B}\right] + \Pr\left[\mathcal{B}^c\right],$$

and for the first term ($\Pr\left[\mathcal{E}_t \bigcap \mathcal{B}\right]$), we can proceed exactly as in the proof of Theorem 4.2. We omit this and only consider $\Pr\left[\mathcal{B}^c\right]$.

We have

$$\begin{aligned}\Pr\left[\mathcal{B}^c\right] &= \Pr\left[\bigcup_{\mathcal{F}\subset[M]\setminus[K_a]} \{\|\mathbf{s}_{\mathcal{F}}\| < \sqrt{\beta n}\}\right] \\ &= \Pr\left[\min_{\mathcal{F}\subset[M]\setminus[K_a]} \|\mathbf{s}_{\mathcal{F}}\| < \sqrt{\beta n}\right].\end{aligned}$$

The latter term can be upper-bounded using concentration properties and Gordon's inequality. Let us mention the main ingredients.

Definition 4.1 (Gaussian width)**.** *Given a closed set $\mathcal{S} \subseteq \mathbb{R}^d$, its Gaussian width $w(\mathcal{S})$ is defined as*

$$w(\mathcal{S}) = \mathbb{E}\left[\max_{\boldsymbol{x}\in\mathcal{S}}\{\mathbf{g}^T\boldsymbol{x}\}\right],$$

where $\mathbf{g} \sim \mathcal{N}(0, \boldsymbol{I}_d)$.

Theorem 4.5 (Gordon's theorem, [88])**.** *Let $\mathbf{G} \in \mathbb{R}^{k\times d}$ be a random matrix with i.i.d. entries $\mathbf{g}_{i,j} \sim \mathcal{N}(0,1), i \in [k], j \in [d]$, and $\mathcal{S} \subseteq \mathbb{S}^{d-1}$ be a closed subset of the unit sphere in d dimensions. Then*

$$\mathbb{E}\left[\max_{\boldsymbol{x}\in\mathcal{S}} \|\mathbf{G}\boldsymbol{x}\|\right] \leq a_k + w(\mathcal{S}),$$
$$\mathbb{E}\left[\min_{\boldsymbol{x}\in\mathcal{S}} \|\mathbf{G}\boldsymbol{x}\|\right] \geq a_k - w(\mathcal{S}),$$

where $a_k = \mathbb{E}[\|\mathbf{g}\|]$, $\mathbf{g} \sim \mathcal{N}(0, \boldsymbol{I}_k)$, and $w(\mathcal{S})$ is the Gaussian width of the set $\mathcal{S}$. Recall that $a_k = \sqrt{2}\Gamma((k+1)/2)/\Gamma(k/2)$ satisfies

$$\frac{k}{\sqrt{k+1}} \leq a_k \leq \sqrt{k}.$$

Lemma 4.1 (Gaussian concentration, see e.g. [89, Corollary 3.3])**.** *Let $f : \mathbb{R}^d \to \mathbb{R}$ be an ℓ-Lipschitz function and $\mathbf{g} \sim \mathcal{N}(0, \boldsymbol{I}_d)$. Then,*

$$\Pr\left[f(\mathbf{g}) \geq \mathbb{E}_{\mathbf{g}}[f] + t\right] \geq 1 - e^{-\frac{t^2}{2\ell^2}},$$
$$\Pr\left[f(\mathbf{g}) \leq \mathbb{E}_{\mathbf{g}}[f] - t\right] \leq e^{-\frac{t^2}{2\ell^2}}.$$

Lemma 4.2 (Size of the maximum)**.** *Let* $\mathbf{g} = [\mathsf{g}_1, \ldots, \mathsf{g}_d] \sim \mathcal{N}(0, \mathbf{\Sigma})$ *and* $\mathbb{E}[\|\mathsf{g}_i\|^2] = 1$ *for* $i \in [d]$*. Irrespective of the covariance structure, we have the following bound:*

$$\mathbb{E}\left[\max_{i\in[d]} \mathsf{g}_i\right] \le \sqrt{2\log d}.$$

Proof.

$$\begin{aligned}\mathbb{E}\left[\max_{i\in[d]} \mathsf{g}_i\right] &= \frac{1}{\beta}\mathbb{E}\left[\log\exp[\beta \max_i \mathsf{g}_i]\right] \\ &\le \frac{1}{\beta}\mathbb{E}\left[\log\sum_{i\in[d]}\exp[\beta\mathsf{g}_i]\right] \\ &\le \frac{1}{\beta}\log\sum_{i\in[d]}\mathbb{E}\left[\exp[\beta\mathsf{g}_i]\right] = \frac{\beta}{2} + \frac{\log n}{\beta}.\end{aligned}$$

Taking $\beta = \log n$ yields the lemma statement. □

Let us now return to our problem. Recall that we consider the case of a Gaussian codebook X. Let $\mathsf{G} = \mathsf{X}_{[M]\setminus[K_a]}$. The matrix G is of size $n \times (M - K_a)$, with $g_{i,j} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, P')$.

$$\begin{aligned}&\Pr\left[\min_{\mathcal{F}\subset[M]\setminus[K_a]} \|\mathbf{s}_{\mathcal{F}}\| < \sqrt{\beta n}\right] \\ &= \Pr\left[\min_{\boldsymbol{x}\in\{0,1\}^{M-K_a}, \|x\|_0 = t} \|\mathbf{G}\boldsymbol{x}\| < \sqrt{\beta n}\right] \\ &\le \exp\left(-\frac{c^2}{2\ell^2}\right),\end{aligned}$$

where $(x)_+ = \max\{x, 0\}$,

$$c = \left(\mu - \sqrt{\beta n}\right)_+,$$

and

$$\mu = \mathbb{E}\left[\min_{\boldsymbol{x}\in\{0,1\}^{M-K_a}, \|\boldsymbol{x}\|_0 = t} \|\mathbf{G}\boldsymbol{x}\|\right] \ge \frac{n\sqrt{P't}}{\sqrt{n+1}} - \sqrt{2P't\ln\binom{M-K_a}{t}}.$$

Note that the function $F(\boldsymbol{G}) = \min_{\boldsymbol{x}\in\{0,1\}^{M-K_a}, \|x\|_0 = t} \|\boldsymbol{G}\boldsymbol{x}\|$ is ℓ-Lipschitz with $\ell = \sqrt{t}$. Indeed,

$$\begin{aligned}|F(\mathbf{G}_1) - F(\mathbf{G}_2)| &\le \max_{\boldsymbol{x}\in\{0,1\}^{M-K_a}, \|\boldsymbol{x}\|_0 = t} |\|\mathbf{G}_1\boldsymbol{x}\| - \|\mathbf{G}_2\boldsymbol{x}\|| \\ &\le \max_{\boldsymbol{x}\in\{0,1\}^{M-K_a}, \|\boldsymbol{x}\|_0 = t} [\|(\mathbf{G}_1 - \mathbf{G}_2)\boldsymbol{x}\|] \\ &\le \sqrt{t}\,\|(\mathbf{G}_1 - \mathbf{G}_2)\|_{2,2} \le \sqrt{t}\,\|(\mathbf{G}_1 - \mathbf{G}_2)\|_F,\end{aligned}$$

where

$$\|\boldsymbol{A}\|_{\alpha,\beta} \triangleq \sup_{\boldsymbol{x}:\, \|\boldsymbol{x}\|_\alpha \le 1} \|\boldsymbol{A}\boldsymbol{x}\|_\beta.$$

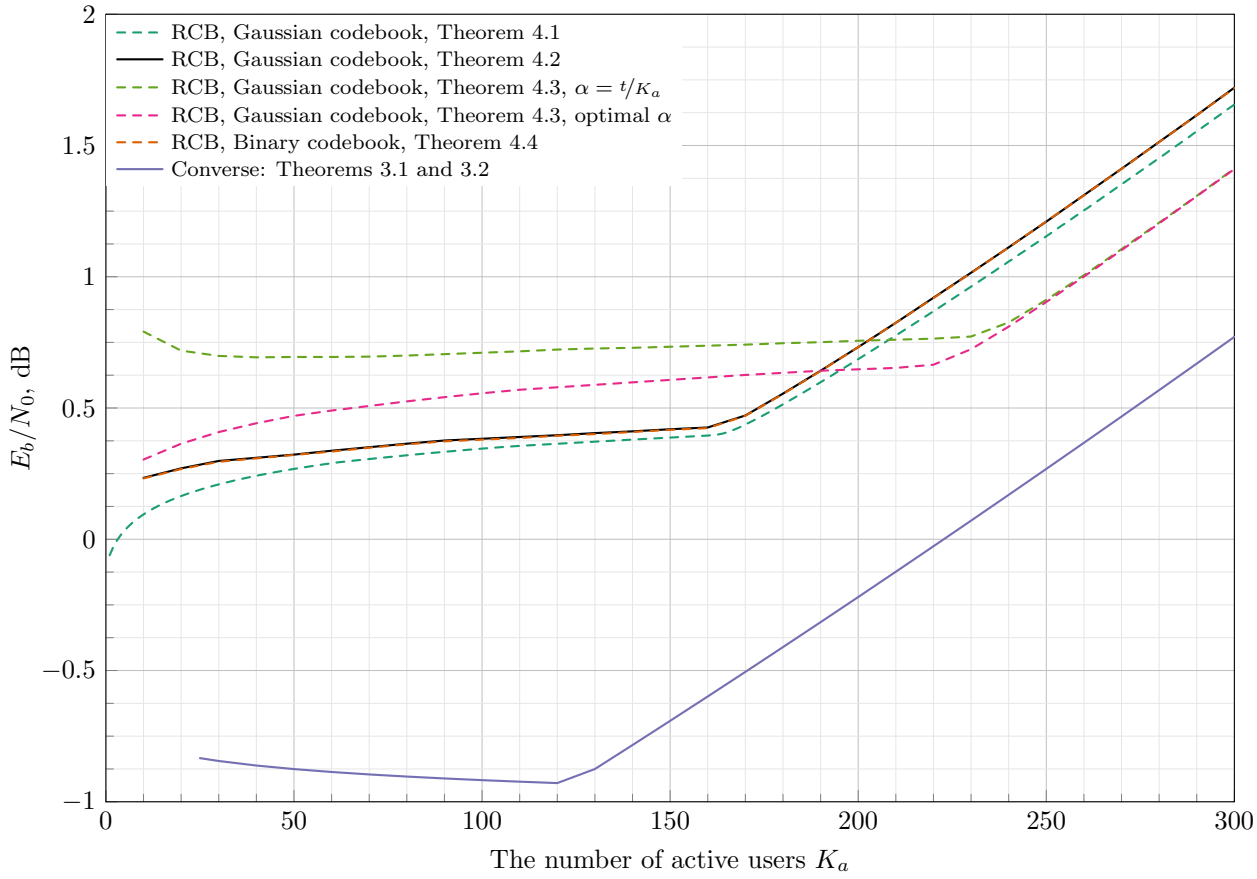

Figure 4.3: AWGN MAC achievability bounds are presented for $k = 100$ information bits, a frame length of $n = 30000$, and an error probability constraint of $P_e < 0.05$. The converse bound is depicted as a single line, representing the maximum of the results from Theorems 3.1 and 3.2.

We use the result of this lemma to analyze the asymptotic regime presented in Appendix F. This lemma proves to be highly efficient, and the resulting achievability bound coincides with the converse bound. However, we do not evaluate this lemma in the FBL regime, as it yields energy efficiency estimates several decibels higher than those presented below.

4.6 Numerical evaluation

Let us consider a setup with a frame length of $n = 3\times10^4$ real channel uses and $k = 100$ information bits transmitted by each active user. This setup was proposed in [5] and [90].

We begin with a known number of active users. We select $P_e < 0.05$ and analyze the minimum energy per bit (E_b/N_0) as a function of the number of active users, K_a. The RCB (Theorem 4.1, [5]) is represented in Fig. 4.3 by a dashed green line. A surprising observation is that the energy per bit remains nearly constant—almost as if only a single user were transmitting—until reaching a

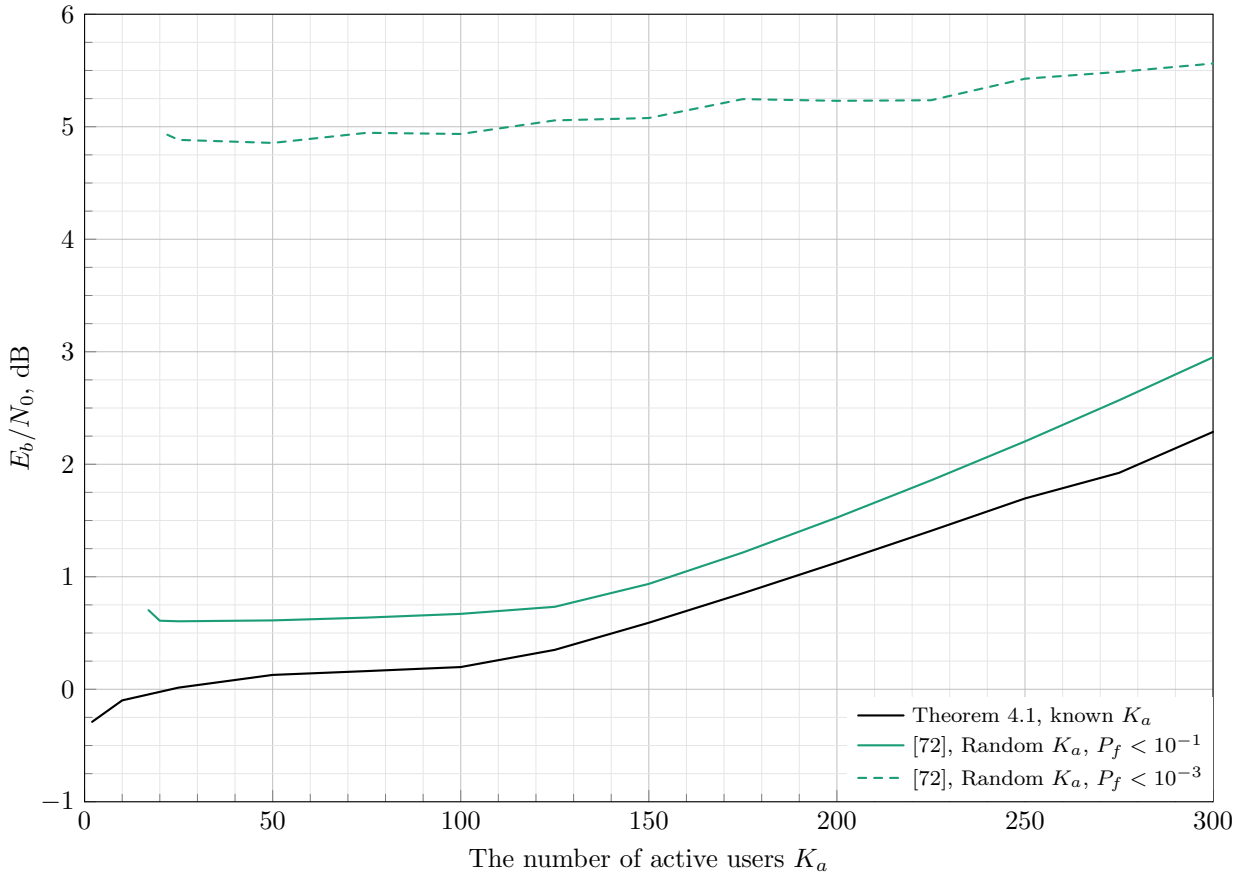

Figure 4.4: AWGN MAC achievability bounds are presented for $k = 128$ information bits, a frame length of $n = 19200$, an error probability constraint of $P_e < 0.1$, and a random number of active users. Results for different values of P_f are shown. A stricter FAR requirement leads to a higher required E_b/N_0.

critical value of K_a (for the parameters from Theorem 4.1, this value is approximately $K_a \approx 150$).

Thus, the performance is *noise-limited*. The key takeaway is that additional users can be accommodated without increasing energy requirements until a critical threshold is reached. The primary reason for this effect is that the dominant probability, $\Pr[\mathcal{E}_t]$, in (4.2) corresponds to $t = 1$ when K_a is small. However, as K_a increases, the last term, $t = K_a$, becomes dominant due to the large combinatorial factor $\binom{M-K_a}{t}$, which results in the *interference-limited* regime.

Next, let us consider the RCB based on Fano's trick (Theorem 4.2). The resulting energy efficiency is presented by a solid black line. As can be seen in Fig. 4.3, the results obtained for the Gaussian (Theorem 4.2) and binary (Theorem 4.4) codebooks are almost identical and very close to the original achievability bound from Theorem 4.1 (e.g., for 250 active users, the values are 1.210 dB, 1.211 dB, and 1.154 dB, respectively).

The results for the RCB, based on a combination of tricks (Theorem 4.3), are shown by dashed

green and magenta lines for $\alpha = t/K_a$ and the optimized α, respectively. We highlight both curves to improve the original bound from Theorem 4.1 for large values of K_a.

Next, let us consider a random number of active users [72]. For this bound, we evaluate the previously defined energy efficiency as a function of the average number of active users. We note that the number of active users follows a Poisson distribution.

The main challenge here is the presence of two probabilities: P_e (3.1) and P_f (3.2). The results are presented for $n = 19200$, $k = 100$, and $P_e < 0.1$, as shown in Fig. 4.4. When the condition $P_f < 0.1$ is imposed, the bound for a random K_a (depicted by a solid green line) demonstrates energy efficiency very close to that of the bound for a known K_a [5] (depicted by a black line). Surprisingly, imposing stricter limits on P_f (e.g., 10^{-3} instead of 10^{-1}) results in an additional gap of approximately 2 dB.

The achievability bound for $P_f < 10^{-3}$ is depicted by a dashed green line. Furthermore, the transition between noise-limited and interference-limited regimes (caused by the behavior of different terms in (4.2)) becomes indistinguishable.

Remark 4.7. *A stricter FAR requirement leads to a higher required E_b/N_0. In the extreme case where $P_f \to 1$, we obtain $E_b/N_0 \to 0$ (or $-\infty$ dB). Indeed, in this scenario, the receiver can take the entire codebook as the list of decoded messages, leading to $P_e = 0$ and $P_f = 1$.*

Chapter 5

URA over Gaussian MAC: low-complexity schemes

In this chapter, we address the problem of constructing low-complexity coding schemes for the URA over Gaussian MAC (G-URA). The main challenges we aim to overcome are as follows:

- Ultra-low rate: if we look at Section 3.4, we observe that the coding rate is $\approx 1/300$ bits per channel use for typical values of parameters of interest.
- Large *collision order*[1] in a frame: K_a can be as high as 500.
- Same codebook.

The existing MAC coding schemes are efficient in two opposite situations:

1. Large dimension, or equivalently, codebook size, and a small collision order, typically not exceeding 2–4.
2. Large collision order and a small codebook size, typically on the order of 2^{15}.

The latter case corresponds to the classical CS problem with standard decoding algorithms such as LASSO, non-negative least squares (NNLS), or OMP; see the description in Appendix B. Thus, the task of URA is to support a huge number of extremely low-bit-rate user streams, which appears to be much more complex than supporting a small number of users, each with a high bit rate. Finally, the same encoder assumption introduces additional challenges. Indeed, all low-complexity schemes described in Chapter 2 utilize single-user decoders (explicitly or implicitly). The same codebook and equal powers eliminate user diversity and thus do not allow relying on single-user decoders.

[1] In the G-URA setup, a *collision* is understood as multiple simultaneous transmissions in the frame or slot, depending on the context, and the *order of collision* equals the number of simultaneous transmissions.

As an example, we consider the case of a two-user MAC, where users utilize LDPC codes. When LDPC codes are different[2], the performance of the system is good and even close to the point on the dominant facet when $n \to \infty$ [52]. At the same time, when the same code is used, the system performs extremely poorly. We return to this case at the end of the chapter to provide some recent results and formulate open problems.

Before we proceed with the detailed description, we formulate two main ideas behind all existing low-complexity schemes for the URA channel: (a) sparsifying the collisions and (b) splitting the high-dimensional problem into a number of low-dimensional ones. If we return to the CS interpretation (see eq. (3.5)), the task can be formulated as follows: find the sensing matrix $\boldsymbol{X}$ such that low-complexity algorithms do exist. See the examples of sensing matrices of URA low-complexity schemes in Appendix C.

The question of low-complexity URA schemes for the GMAC is widely addressed in the literature. Below, we present the main techniques:

- Schemes based on the T-fold coded SA protocol or ALOHA with multi-packet reception (Section 5.1).
- Schemes based on IDMA techniques [91] were proposed in [92] (Section 5.2).
- CCS approach [93] and modifications that include t-error correcting list-recoverable codes [94], SPARCs combined with the AMP algorithm [95], iterative approaches [96], where the inner AMP decoder and the outer tree decoder are allowed to exchange soft information within a joint message-passing algorithm (Section 5.3).
- Random spreading and correlation-based energy detector with polar codes and equal power levels [97], polar codes and different power levels [98], LDPC codes and SIC within the decoding process [99] (Section 5.4).

The presentation of low-complexity schemes is followed by a discussion on the applicability of the same linear codes to the G-URA problem in Section 5.5. Finally, a numerical comparison of the presented schemes is provided in Section 5.6. The main results are summarized in Fig. 5.15.

5.1 T-fold coded slotted ALOHA

In this section, we focus on the IRSA protocol presented in Section 2.3.4 and analyze the achievable energy efficiency under PUPE constraints (3.3), instead of throughput. This protocol assumes that users send multiple replicas of the same packet (repetition), forming a subclass of coded SA protocols. A key aspect of the URA problem is the large number of devices, which leads to a high collision probability, making collision resolution necessary.

[2]Different codes can be created from one LDPC code by means of different interleavers, or one can consider two different cosets of the same LDPC code.

T-fold IRSA (or IRSA with multi-packet reception) enables the resolution of collisions up to order T. Recall that in the classical collision model described in Section 2.3.1, messages from a collided slot cannot be extracted. However, in an AWGN channel, the successful reception of multiple transmissions—up to T—may be possible with a suitable slot decoding algorithm. The details of this algorithm will be outlined throughout this section, depending on the particular scenario. The value of T is determined by computational complexity limitations at the receiver.

We begin with the IRSA protocol description in Section 5.1.1, followed by the DE analysis in Section 5.1.2. Next, we present a T-fold IRSA achievability bound formulated in Theorem 5.2. Some IRSA modifications adopted for practical schemes are introduced in Section 5.1.3, followed by an analysis of practical collision-resolution methods. These methods, based on LDPC codes [100] and polar codes [101], as well as the compute-and-forward strategy [102], which was applied to the URA setup in [90], are presented in Section 5.1.4.

5.1.1 Basic T-fold IRSA scheme

In this section, we describe the basic T-fold IRSA protocol used in [100]. The main difference from previous works [15, 103, 104] is that we do not add pointers to the locations of the replicas. Instead, we determine the slot count and slot indices based on the message to be transmitted.

Assume the frame is split into L slots of size $n' = n/L$ channel uses. We now describe the main features of the transmission process:

- The user chooses a message $W \sim \text{Unif}([M])$, encodes it, and obtains a codeword $f(W) \in \mathbb{R}^{n'}$.
- Users repeat their codewords in multiple slots. To choose the slots, all users share *the same* slot-selection function $g : [M] \to \{0,1\}^L$, i.e., the codeword $f(W)$ is transmitted in the slots with indices from $\text{supp}\left(g\left(W\right)\right)$.
- The function g also determines the number of replicas $D : [M] \to [L]$, where $D(W) = \text{wt}\left(g\left(W\right)\right)$.
- The uniform message distribution induces the replica count distribution, i.e., the distribution of the random variable $D(W)$, which is the same for all users.

Remark 5.1. *We note that the functions $g(\cdot)$ and $D(\cdot)$ are deterministic functions of the message. This fact is crucial because the receiver does not know the positions of multiple replicas until the message is decoded in at least one slot. Once the message is decoded, the receiver can determine from which slots the replicas of the decoded message should be removed and then perform the SIC step.*

Remark 5.2. *In our analysis (see Section 5.1.2), we assume that the slot-selection function $g\left(\cdot\right)$ is designed such that, for a given number of replicas $d = D\left(W\right)$, the slot indices are selected uniformly from the $\binom{L}{d}$ possible choices. In practical schemes, we construct $g\left(\cdot\right)$ to approximate this assumption as closely as possible. One possible approach is to use the message W as a seed for a pseudo-random number generator. Once the message is recovered, the receiver sets the same seed and determines the replica locations.*

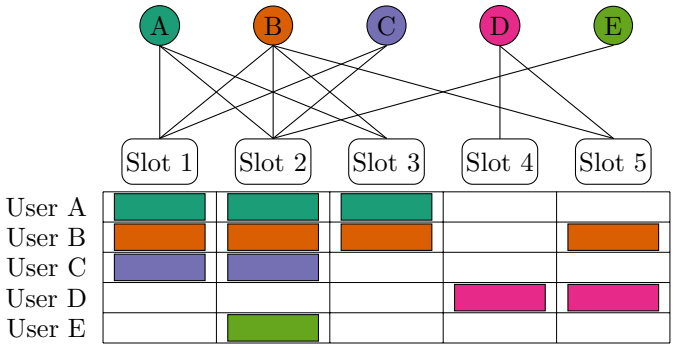

Figure 5.1: Example of a bipartite graph (top) corresponding to the case of 5 active users transmitting in 5 slots (bottom). User-nodes are labeled A–E and have different colors. Slot-nodes are marked by numbers. The messages transmitted in different slots are represented by colored rectangles, with each color matching a different user.

Bipartite graph representation. The transmission and decoding processes can be described using a bipartite graph (see Fig. 5.1), known as a Tanner graph [105]. The vertex set of this graph consists of the set of user-nodes $\mathcal{U} = \{u_1, u_2, \ldots, u_{K_a}\}$, which represents the users, and the set of slot-nodes $\mathcal{S} = \{s_1, s_2, \ldots, s_L\}$, which corresponds to the signals received in the slots. A user-node u_i and a slot-node s_j are connected by an edge if and only if the i-th user transmitted a packet in the j-th slot.

Following [65], we represent user-node degree distributions from both the node and edge perspectives using polynomials (or generating functions). Specifically, we introduce the polynomials:

$$\Lambda(x) = \sum_{i=1}^{L} \Lambda_i x^i, \quad \text{and} \quad \lambda(x) = \frac{\Lambda'(x)}{\Lambda'(1)} = \sum_{i=1}^{L} \lambda_i x^{i-1}, \tag{5.1}$$

where Λ_i denotes the fraction of user-nodes with degree i, and λ_i denotes the fraction of edges incident to user-nodes of degree i. Here, $\Lambda'(x)$ represents the derivative with respect to the formal variable x. Note that

$$\Lambda'(1) = \sum_{i=1}^{L} i\Lambda_i$$

is the average degree of user-nodes.

Example 5.1. *Consider the graph in Fig.* 5.1. *We have:*

$$\Lambda(x) = \frac{1}{5}x + \frac{2}{5}x^2 + \frac{1}{5}x^3 + \frac{1}{5}x^4,$$

and

$$\lambda(x) = \frac{1}{\frac{1}{5}+\frac{4}{5}+\frac{3}{5}+\frac{4}{5}}\left(\frac{1}{5} + \frac{4}{5}x + \frac{3}{5}x^2 + \frac{4}{5}x^3\right) = \frac{1}{12} + \frac{1}{3}x + \frac{1}{4}x^2 + \frac{1}{3}x^3.$$

Similar to (5.1), we define the slot-node degree distributions from the node perspective $R(x)$ and

the edge perspective $\rho(x)$ as follows:

$$R(x) = \sum_{i=0}^{K_a} R_i x^i, \quad \text{and} \quad \rho(x) = \sum_{i=1}^{K_a} \rho_i x^{i-1}.$$

Next, we compute the probability that a user selects any given slot (denoted by event $\mathcal{E}$). Since the process does not depend on specific user or slot indices, we omit them for simplicity. Let the user transmit d replicas. Following Remark 5.2 (uniform slot selection), we obtain:

$$\Pr[\mathcal{E}|D=d] = \frac{\binom{L-1}{d-1}}{\binom{L}{d}} = \frac{d}{L},$$

which implies

$$\Pr[\mathcal{E}] = \mathbb{E}_D\left[\Pr[\mathcal{E}|D]\right] = \frac{\Lambda'(1)}{L} = \frac{G\Lambda'(1)}{K_a},$$

where $G = K_a/L$ and $G\Lambda'(1)$ represents the average degree of slot-nodes.

Thus, the slot-node distribution (from the node perspective) follows a binomial distribution:

$$\text{Bino}\left(K_a, \frac{G\Lambda'(1)}{K_a}\right).$$

That is, defining $\gamma = G\Lambda'(1)$, we obtain:

$$\begin{aligned} R(x) &= \sum_{i=0}^{K_a} \binom{K_a}{i} \left(\frac{\gamma}{K_a}\right)^i \left(1 - \frac{\gamma}{K_a}\right)^{K_a - i} x^i \\ &= \left(1 - \frac{\gamma}{K_a}\right)^{K_a} \left(1 + \frac{\frac{\gamma}{K_a}x}{1 - \frac{\gamma}{K_a}}\right)^{K_a}. \end{aligned} \tag{5.2}$$

Decoding. Now, let us describe a general approach to the decoding algorithm, which is based on two main procedures:

1. The slot decoding algorithm, which outputs the list of decoded[3] messages.
2. The SIC step, which relies on the function $g(\cdot)$ and subtracts replicas (if present) of successfully decoded messages.

Clearly, after a single SIC step, the collision order will decrease in some slots. Hence, these steps may be iteratively repeated in different combinations with respect to different slots, depending on the particular decoding algorithm. We refer the reader to Section 5.1.4 for specific practical slot decoding algorithms.

The example below considers a slot decoding sequence for the messages presented in Fig. 5.1.

[3]Note that this is an estimate of the transmitted list, so some messages can be missing and some false messages may be added by the decoder. The issue of false messages should be addressed by the decoder to avoid error propagation within the SIC process. One possible solution is to add control information (e.g. add the CRC) to the packet.

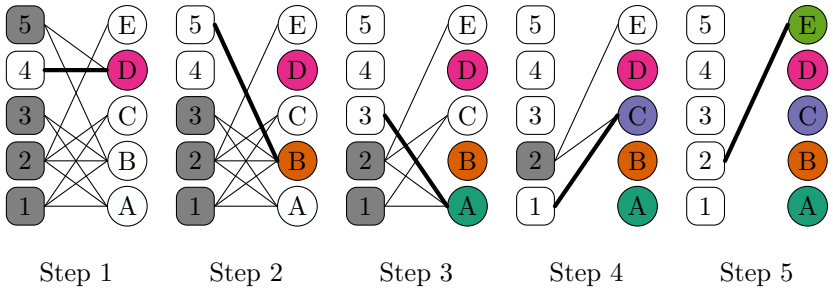

Figure 5.2: Sequence of slot decoding followed by SIC steps applied to the case presented in Fig. 5.1 for $T = 1$. The bold line corresponds to the edge incident to the slot that can be decoded at the corresponding step. At the subsequent SIC step, all edges incident to the same user-node as the bold one will be removed. Resolved slots are marked with a white background, unresolved slots are marked with a gray background. Resolved users are marked with a color that matches Fig. 5.1.

Example 5.2. *Consider the example with $K_a = 5$ active users and $L = 5$ slots, as presented in Fig.* 5.1*. Assume the first user (A) sends a codeword in slots 1, 2, and 3; the second user (B) in slots 1, 2, 3, and 5; the third user (C) in slots 1 and 2; the fourth user (D) in slots 4 and 5; and the fifth user (E) in slot 2.*

As a result, there are 3 simultaneous transmissions in slot 1, 4 transmissions in slot 2, collisions of order 2 in slots 3 and 5, and the only collision-free slot is slot 4.

Consider a decoder with $T = 1$ (see Fig. 5.2*). The decoder searches for a collision-free slot, then decodes the message transmitted in this slot and subtracts this message from all slots where it was replicated. There is a natural question: how does the decoder select the collision-free slot? The strategies may be different, but one of them is to try all slots in parallel with a decoder designed for $T = 1$. Another reasonable strategy is to select the slot with the minimal received energy.*

The SIC step begins after the successful decoding of slot 4 at step 1 (with all remaining slots marked gray, indicating slots with collisions) and decodes the message of user D. The corresponding edge in the bipartite graph is marked with a bold line. After the SIC step, all edges incident to the successfully decoded user (D, in this case) are removed, as all transmitted replicas are subtracted.

At the second decoding step, slot 5 becomes collision-free, resulting in the successful decoding of user B. The third step resolves slot 3, allowing user A to be decoded. After the SIC step, slot 1 becomes collision-free. In the final step, user E, who transmitted in slot 2, can be decoded.

For a decoder with $T = 2$, the decoding algorithm may behave differently. For instance, slots 3, 4, and 5 have a collision order not higher than two. Hence, these slots may be decoded first in any order. After the corresponding SIC steps, the collision order of slot 1 will be reduced to one (collision-free transmission), and slot 2 will have a collision order of two. Moreover, decoding slot 2 will be sufficient to extract all remaining messages in the frame.

5.1.2 Achievability bound for T-fold IRSA scheme

Slot decoding

Let us first note that, in order to obtain an achievability bound for the whole scheme, we do not restrict the complexity of slot decoding and instead use the random coding bounds described in Chapter 4.

Consider a particular slot—w.l.o.g., let it be the first slot. Let $\mathbf{y}_1$ be the received signal of n' channel uses. We assume that r users transmit in the first slot. Recall that k denotes the number of information bits sent by each user and that P is the transmit power. The random coding bound from Theorem 4.1 states that, in the random Gaussian ensemble $\mathcal{G}\left(2^k, n', P\right)$ (see Definition 4.1), there exists a codebook $\boldsymbol{X}^*$ such that

$$\begin{aligned}&\frac{1}{r}\sum_{i=1}^{r}\Pr\left(W_i \notin \mathcal{R}\left(\mathbf{y}_1\right) \mid \boldsymbol{X}^*\right) = \Pr\left(W_1 \notin \mathcal{R}\left(\mathbf{y}_1\right) \mid \boldsymbol{X}^*\right)\\ &\leq p\left(n', k, P, r\right) \triangleq \mathbb{E}_{\boldsymbol{X}}\left[\Pr\left(W_1 \notin \mathcal{R}\left(\mathbf{y}_1\right) \mid \boldsymbol{X}, r\right)\right].\end{aligned}$$

The first equality holds due to the symmetry of users (it is sufficient to consider the probability that a particular user's message is not in the decoded list). The last expectation is taken over the Gaussian ensemble.

Here, we emphasize *the main problem*—the bound assumes that the number of active users (r) is known. However, due to the randomness of T-fold IRSA, the number of users transmitting in a particular slot is a random variable.

Another issue is that the codebook $\boldsymbol{X}^*$ is constructed for a specific number of users, but we need a codebook that can resolve collisions for any order $r \in [T]$.

To address these issues, the authors of [106] propose shifting to blind decoding. Let

$$\mathcal{T} = \{W_1, \ldots, W_r\} \subset [M], \quad \text{where} \quad |\mathcal{T}| = r,$$

denote the set of transmitted messages. The blind ML decoding rule is given by:

$$\mathcal{R} = \underset{\mathcal{R}' \subseteq [M], |\mathcal{R}'| \leq T}{\arg\min} \left\|\mathbf{y}_1 - \boldsymbol{s}_{\mathcal{R}'}\right\|^2. \tag{5.3}$$

Recall that T is the maximum collision order that can be resolved.

Theorem 5.1. *Fix P' and T. Then, the average (over the random Gaussian ensemble) per-user error probabilities can be calculated as follows for $r \in [T]$:*

$$\mathbb{E}_{\boldsymbol{X}}\left[\Pr\left(W_1 \notin \mathcal{R}\left(\mathbf{y}_1\right) \mid \boldsymbol{X}, r\right)\right] \leq \sum_{t=1}^{r} \frac{t}{r} p_t\left(r, T\right),$$

where

$$p_t(r, T) = \sum_{\hat{t}=0}^{T-r+t} \exp\left[-nE\left(t, \hat{t}\right)\right], \tag{5.4}$$

$$E(t,\hat{t}) = \max_{0\le\rho,\rho_1\le 1,\lambda>0} \left[-\rho_1\rho_2 R_1 - \rho_1 R_2 + E_0\left(\rho_1,\rho_2\right)\right],$$

$$R_1 = \frac{1}{n}\log\binom{M-r}{\hat{t}}, \quad R_2 = \frac{1}{n}\log\binom{r}{t},$$

$$E_0\left(\rho_1,\rho_2\right) = \frac{1}{2}\left(\rho_1 a + \log(1-2b\rho_1)\right),$$

$$a = \rho_2\log\left(1+2P'\hat{t}\lambda\right) + \log\left(1+2P't\mu\right),$$

$$b = \rho_2\lambda - \frac{\mu}{1+2P't\mu}, \quad \mu = \frac{\rho_2\lambda}{1+2P'\hat{t}\lambda}.$$

Proof. The proof of this theorem follows the same structure as that of Theorem 4.1, with the main difference being in the summation limits of (5.4). □

We now need to choose a codebook that is effective for all collision orders up to T. Let us formally state this requirement.

Statement 5.1. *Let us choose positive values α_i, for $i \in [T]$, such that $\sum_{i=1}^{T}\alpha_i < 1$. In a random Gaussian ensemble, there exists a codebook $\tilde{\boldsymbol{X}}$ such that the following inequalities hold for all $r \in [T]$:*

$$\begin{aligned}\Pr\left(W_1 \notin \mathcal{R}(\mathbf{y}_1) \,\middle|\, \tilde{\boldsymbol{X}}, r\right) &\le \tilde{p}\left(n,k,P,T,r\right)\\ &\triangleq \frac{1}{\alpha_i}\mathbb{E}_{\boldsymbol{X}}\left[\Pr\left(W_1 \notin \mathcal{R}(\mathbf{y}_1) \mid \boldsymbol{X}, r\right)\right].\end{aligned}$$

Proof. We estimate the probability of a bad code—one for which the inequalities do not hold for at least one r. Applying Markov's inequality and a union bound, we see that this probability is upper-bounded by

$$\sum_{i=1}^{T}\alpha_i < 1.$$

□

Finally, we refer the reader to Section 4.5.1, where the problem of a random number of active users has been addressed. The results presented in Section 4.5.1 were developed later and can serve as an alternative approach to the T-fold IRSA achievability bound.

Performance analysis via density evolution

The iterative SIC procedure can be analyzed by means of the DE method proposed in [15, 107] for IRSA. This approach is, in fact, equivalent to the DE analysis for LDPC codes [65] over the erasure channel. We consider message passing decoding, where slot-nodes and user-nodes send outgoing

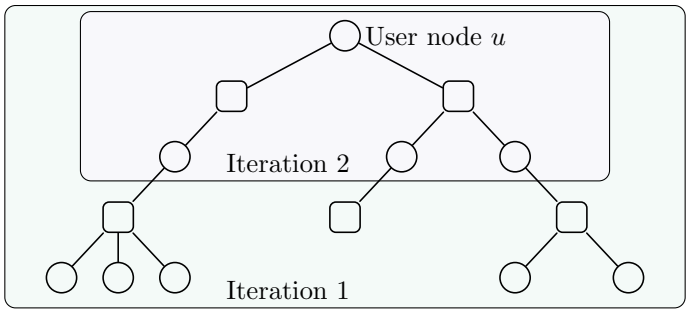

Figure 5.3: Computational graph: Neighborhood of user node u for 2 iterations.

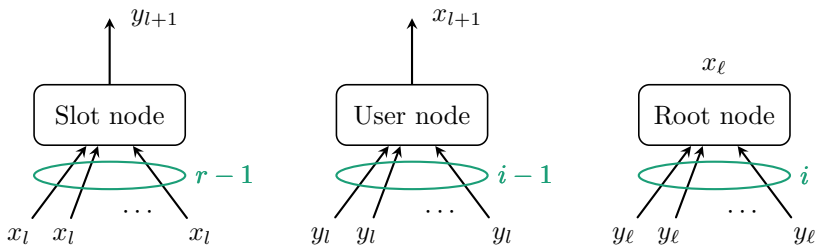

Figure 5.4: IRSA DE rules. Left: Messages to the slot node at intermediate iteration l, center: Messages to the user node at intermediate iteration l, right: Messages to the user node at the final iteration ℓ.

messages along each edge. Each message can take two possible values: either the recovered packet or an erasure[4] if the packet cannot be recovered.

We consider the ensemble of Tanner graphs $\mathcal{G}(K_a, L, \lambda(x), \rho(x))$ corresponding to the multiple-access scheme with K_a users, L slots, and the degree distributions $\lambda(x)$ and $\rho(x)$. We are interested in the decoding performance averaged over this ensemble in the limit as $K_a, L \to \infty$. Note that in the limit $K_a \to \infty$, the distribution (5.2) becomes a Poisson distribution, i.e.

$$R(x) = \rho(x) = e^{-\gamma(1-x)} = \sum_{r=1}^{\infty} \frac{\gamma^{r-1}}{(r-1)!} x^{r-1} = \sum_{r=1}^{\infty} \rho_r x^{r-1},$$

where $\gamma = G\Lambda'(1)$ should remain constant.

Let us choose a slot-node u. Message passing takes place on the local neighborhood of this node. If we unroll the Tanner graph with the root at user-node u, we obtain the computation graph depicted in Fig. 5.3. In what follows, we use the fact that any neighborhood of constant depth in $\mathcal{G}(K_a, L, \lambda(x), \rho(x))$ is tree-like with probability one. We refer the reader to [65] for a detailed explanation.

We follow the exposition from [106]. Before we proceed with the DE rules, we list the main assumptions:

[4]In what follows, we use the term "erased message".

1. A code $\tilde{\boldsymbol{X}}$ constructed in accordance with Statement 5.1 is used in the system.

2. If the collision order is greater than T, no message can be recovered from the slot. If the collision order is $t < T$, each message is recovered with probability $\tilde{p}\left(n', k, P, T, t\right)$.

3. The genie reveals information on false packets, i.e., such packets are excluded from the SIC process.

Remark 5.3. *Note that the genie assumption prevents the described bound from being a true achievability bound. This bound is only an optimistic estimate of the performance achievable within a T-fold IRSA scheme.*

Now let us write the DE rules. By x_l and y_l, we denote the probabilities that an outgoing message (see Fig. 5.4) from the user-node and the slot node, respectively, are erased during the l-th iteration.

Statement 5.2. *The DE rules are described in the following form:*

$$\begin{aligned} y_{l+1} &= 1 - \rho\left(1 - x_l\right) \sum_{t=0}^{T-1} \left(1 - \tilde{p}\left(n', k, P, T, t+1\right)\right) \times \frac{\left(GL'\left(1\right) x_l\right)^t}{t!}, \\ x_l &= \lambda\left(y_l\right), \ 1 \le l < \ell, \\ x_\ell &= \Lambda\left(y_\ell\right), \end{aligned}$$

where the function $\tilde{p}\left(n', k, P, T, r\right)$ is defined in Statement 5.1, and the initial condition is $x_0 = 1$, which means that the user messages are erased at the beginning, and we observe only the noisy signal sums in the slots.

Proof. Consider the l-th iteration. Let us start with the *slot node*. We want to calculate the erasure probability of the outgoing message y_{l+1} based on incoming messages that are erased with probabilities x_l. Let us start with a slot node of degree r. We can recover the outgoing message (or equivalently, the packet) if there are no more than $T - 1$[5] erased messages out of $r - 1$ incoming messages. Finally, the outgoing message is erased with probability

$$p_r = 1 - \sum_{t=0}^{\min(r,T)-1} \left(1 - \tilde{p}\left(n', k, P, T, t+1\right)\right) \binom{r-1}{t} x_l^t \left(1 - x_l\right)^{r-1-t}.$$

Note that ρ_r is the probability that the outgoing edge is connected to a slot node of degree r (see [65]), thus the total probability of the erased outgoing message can be calculated as

$$y_{l+1} = \sum_{r=1}^{\infty} \rho_r p_r.$$

By changing the summation order, we obtain the result from the proposed statement.

[5] In accordance with the DE rules, the outgoing message is assumed to be erased, and thus the total number of erased messages is no more than T.

Now proceed with the *user-node*. Clearly, the outgoing message is erased if all incoming messages are erased. Thus, for the node of degree i, the outgoing message is erased with probability

$$q_i = y_l^{i-1},$$

and the total probability of the erased outgoing message can be calculated as

$$x_l = \sum_{i=1}^{\infty} \lambda_i q_i = \lambda\left(y_l\right).$$

The root node should be considered separately. The only change is in the total probability calculation. Here, we need to use the probabilities for the user-node (rather than the edge) to be connected to a slot node of degree i. Thus,

$$x_\ell = \sum_{i=1}^{\infty} \Lambda_i q_i = \Lambda\left(y_\ell\right).$$

□

Theorem 5.2. *Let us fix P', n', $\Lambda(x)$, and ℓ. Then, the T-fold IRSA scheme with the code $\tilde{\boldsymbol{X}}$ achieves the PUPE (as $K_a, L \to \infty$)*

$$P_e \leq x_\ell + \Pr\left[\bigcup_{i \neq j} \{W_i = W_j\}\right].$$

Remark 5.4. *We note that, in the case of LDPC codes over the erasure channel DE, one wants to find the so-called threshold (for the erasure probability ε), i.e.,*

$$\varepsilon_{thr} = \sup\left\{\varepsilon \in [0,1] : \lim_{\ell \to \infty} x_\ell = 0\right\}.$$

In our case, the graph is infinite, but the slot is finite, and it is not possible to achieve $\lim_{\ell\to\infty} x_\ell = 0$ due to the presence of the $\tilde{p}\left(n', k, P, T, r\right)$ function.

Remark 5.5. *Clearly, the average transmitted energy can be calculated as $n'P'\Lambda'(1)$. Thus, the energy per information bit is given by the following expression:*

$$\frac{E_b}{N_0} = \frac{n'P'\Lambda'(1)}{2k}. \tag{5.5}$$

Our main goal is to minimize the required energy per bit E_b/N_0 given that $P_e \leq \varepsilon$. In Section 5.6.1, we utilize the described DE method to choose the system parameters: n' and $\Lambda(x)$.

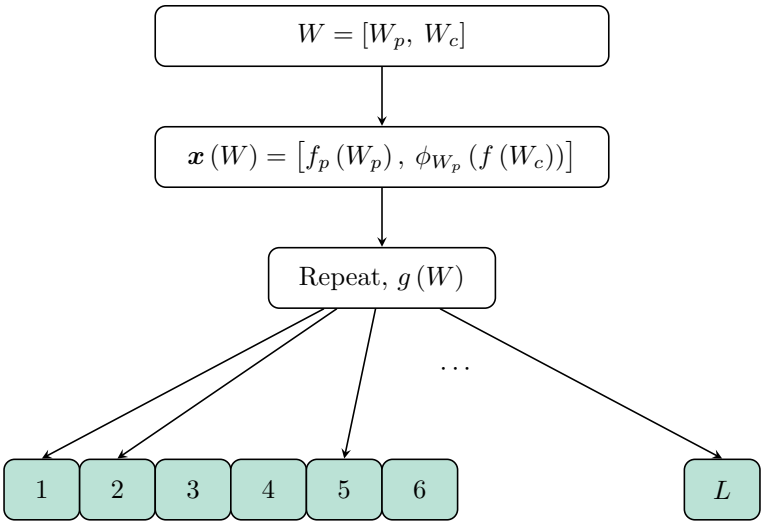

Figure 5.5: IRSA-S encoding process. The transmitted message W is split into two parts: W_p, which defines the preamble $f_p\,(W_p)$ and a transform $\phi_{W_p}\,(\cdot)$, and W_c, which is encoded using the same linear code with the encoding function $f\,(\cdot)$, followed by a transform $\phi_{W_p}\,(\cdot)$ (e.g., interleaving). The number of replicas and slot indices are determined by a function $g\,(W)$. Slots are represented by a row of green rectangles. Note that the entire message, including the preamble, is transmitted within each chosen slot. For clarity, we omit the user message index.

5.1.3 T-fold IRSA modifications for the G-URA

Recall the main problem: we need the same codebook scheme, while existing linear codes (such as LDPC and polar codes) perform extremely poorly in this regime. We refer the reader to Section 5.5 for a detailed discussion of this issue. Hence, we require a nonlinear same-codebook approach. A straightforward method is to use preambles (as illustrated in Fig. 5.5 and 5.6) that define a transformation over the codewords of a linear code. Preamble detection can be considered as a CS problem. Once a preamble is detected, the decoder can recover this transformation and decode the linear code.

In what follows, we refer to the IRSA presented in Section 5.1.1 as basic IRSA protocol (IRSA-B) and describe two strategies proposed in [100, 92], namely IRSA with per-slot preambles (IRSA-S) and IRSA with per-frame preambles (IRSA-F).

IRSA with per-slot preambles

In this variant, the base protocol remains unchanged, i.e., the replica count and slot selection are determined based on the message. The main modification lies in the word to be transmitted in the slot.

Consider the i-th user. We split the message into two parts: $W_i = [W_{i,p},\, W_{i,c}]$, where $W_{i,p} \in [M_p]$, $W_{i,c} \in [M_c]$, and $M = M_p M_c$. Analogously, the slot is divided into two parts of lengths n_p and n_c,

such that $n' = n_p + n_c$. Now, let us describe the word to be transmitted:

- Within the first n_p symbols, the user sends the preamble $f_p(W_{i,p})$, where $f_p : [M_p] \to \mathbb{R}^{n_p}$ is the encoder of a well-designed CS code. Note that the size of this code should not be large ($M_p \leq 2^{15}$), as we apply standard CS decoding algorithms such as LASSO, NNLS, or OMP. We discuss these codes and their decoding algorithms in Appendix B.
- Within the last n_c symbols, we transmit the user codeword $f(W_{i,c})$ transformed by $\phi_{W_{i,p}}$, where $f : [M_c] \to \mathbb{R}^{n_c}$ is the user code encoder[6], and the family of functions $\phi_W : \mathbb{R}^{n_c} \to \mathbb{R}^{n_c}$, with $W \in [M_p]$, is introduced to "create" different codes. In what follows, the function ϕ is implemented either as a permutation or as an addition with a shift vector, both determined by W_p.

Thus, the word transmitted in the slot is given by:

$$\boldsymbol{x}(W_i) = [f_p(W_{i,p}),\ \phi_{W_{i,p}}(f(W_{i,c}))].$$

Now, we describe the decoding process. Analogous to the IRSA-B, the graph is not known on the receiver side and is reconstructed during the decoding process. The slot decoding process then proceeds as follows:

1. Select a slot (preferably the one with the smallest estimated collision order, e.g., based on energy estimation).
2. Decode the preamble part using a CS algorithm. The output of this algorithm includes $\hat{r}$ – an estimate of the collision order – the set of preamble parts $\{\hat{W}_{i,p}\}$, and the set of functions $\{\phi_{\hat{W}_{i,p}}\}$, where $i \in [\hat{r}]$.
3. Decode the user messages and obtain $\{\hat{W}_{i,c}\}$. We discuss specific decoding algorithms for LDPC and polar codes in Section 5.1.4.
4. Perform the SIC step. As a result of steps 1–3, we obtain the set of message estimates $\{\hat{W}_i\}$, where $i \in [\hat{r}]$. Use the function $g(\cdot)$ to remove replicas.
5. Remove the slot from the slot list. If the list of slots is non-empty, return to step 1.

Remark 5.6. *Note that since we use a different codebook scenario in step 3, we do not need to solve the message assembling problem. For each $\hat{W}_p$, we either reconstruct $\hat{W}_c$ or encounter a decoding failure.*

IRSA with per-frame preambles

In the previous variant, slot-wise preambles were used, which simplify the requirements for the CS code and decoder. Indeed, one needs to solve the following CS problem:

$$\boldsymbol{y}_p = \boldsymbol{A}_p \boldsymbol{u}_p + \boldsymbol{z}_p, \tag{5.6}$$

[6] As previously, the encoder is the same for all users.

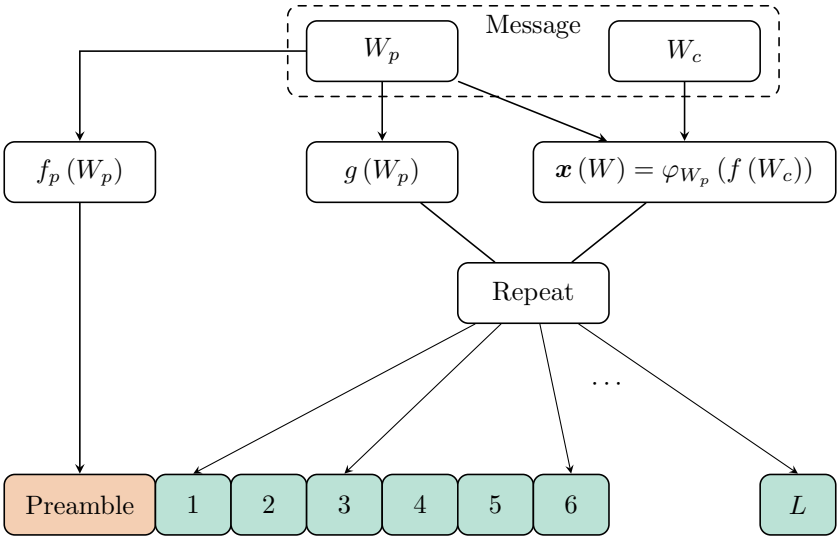

Figure 5.6: IRSA-F encoding process. The transmitted message W is split into two parts: W_p, which defines the preamble $f_p(W_p)$ and a transform $\phi_{W_p}(\cdot)$, and W_c, which is encoded using the same linear code with the encoding function $f(\cdot)$, followed by a transform $\phi_{W_p}(\cdot)$. The number of replicas and slot indices are determined by a function $g(W_p)$. All preambles are sent in a dedicated slot, depicted in orange, while $\varphi_{W_p}(f(W_c))$ is sent in the chosen slots, marked by green rectangles. For clarity, we omit the user message index.

where $\boldsymbol{u}_p = (u_{p,1}, \ldots, u_{p,M_p})$ and the sparsity $\mathrm{wt}(\boldsymbol{u}_p) = r \leq T$. Thus, the length n_p can be small. At the same time, we add preambles to all the slots, and thus the actual overhead is Ln_p channel uses. An illustration of the corresponding sensing matrix is given in Appendix C.

Another preamble placement was proposed in [92]. Again, consider the i-th user and split the message into two parts: $W_i = [W_{i,p}, W_{i,c}]$, where $W_{i,p} \in [M_p]$, $W_{i,c} \in [M_c]$, and $M = M_p M_c$. However, the main differences are as follows:

- The codeword placement and replica count depend only on $W_{i,p}$.
- The preamble of length n_p is placed before the frame, so the total length n is divided between the preamble (n_p) and L slots of length $(n - n_p)/L$.

Let us consider the decoding process.

1. Decode the preamble part using a CS decoder, i.e., solve problem (5.6), where
$$\boldsymbol{u}_p = (u_{p,1}, \ldots, u_{p,M_p})$$
and the sparsity $\mathrm{wt}(\boldsymbol{u}_p) = K_a$. This problem is more challenging compared to the slot-wise CS problem, and thus a stronger CS code is required.
2. Reconstruct the graph and $\{\phi_{\hat{W}_{i,p}}\}$ for $i \in [K_a]$. Note that this is the most significant difference compared to the IRSA-S case. If all preambles are recovered, the graph and code transformations are known.

3. Apply the IRSA-B decoder with the known graph. In this case, we can select slots with the minimal user population.

Remark 5.7. *We note that both protocols (IRSA-S and IRSA-F) assume an additional degree of freedom: namely, the preamble and codeword energies can be different. These energies should be chosen to optimize system performance. The only requirement is the total energy constraint:*

$$\textit{IRSA-S:}\quad (P_p n_p + P_c n_c) L'(1) \le Pn, \qquad (n_c = n' - n_p = n/L - n_p)$$
$$\textit{IRSA-F:}\quad P_p n_p + P_c n_c L'(1) \le Pn, \qquad (n_c = n' = (n - n_p)/L)$$

5.1.4 Collision resolution methods

In this section we consider collision resolution methods, i.e. we focus on one slot of length n' real channel uses and collision resolution capability $2 \le T \le 10$. In what follows we assume $r \le T$ users that transmitted the codewords $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \dots, \boldsymbol{x}^{(r)}$ have collided in the slot of interest

$$\boldsymbol{y} = \sum_{i=1}^{r} \boldsymbol{x}^{(i)} + \boldsymbol{z}, \quad \boldsymbol{z} \sim \mathcal{N}(0, \boldsymbol{I}_{n'}). \tag{5.7}$$

Compute and forward, IRSA-B

We begin with the first scheme proposed for G-URA in [90]. Although this scheme exhibits poor energy efficiency[7] compared to fundamental limits and other URA schemes, it incorporates inspiring ideas. The scheme relies on the compute-and-forward strategy [102], which aims to recover integer combinations of packets rather than the packets themselves.

To illustrate this idea, consider the following example. Suppose we have a two-user GMAC, where the users wish to transmit messages W_1 and W_2 by sending codewords $f(W_1)$ and $f(W_2)$. Recall that $f : [M] \to \mathbb{R}^{n'}$ and that the energy constraints are given by $\|f(W_1)\|^2 \le Pn'$ and $\|f(W_2)\|^2 \le Pn'$. The channel output is given by

$$\mathbf{y} = f(W_1) + f(W_2) + \mathbf{z}, \quad \mathbf{z} \sim \mathcal{N}(0, \boldsymbol{I}_{n'}).$$

The standard low-complexity decoding strategy utilizes a TIN step combined with an SIC step. This can be done in different ways, e.g., in a hard manner, in a soft manner (soft SIC), or through message passing between two TIN decoders. Nevertheless, the first TIN decoder deals with effective noise of energy $(P+1)n'$, which is complicated.

At the same time, if

$$f(W_1) + f(W_2) = f(W_1 \boxplus W_2), \tag{5.8}$$

then the receiver can decode $W_1 \boxplus W_2$ as in the single-user case, i.e., without interference. In general, the operation $\boxplus$ can be any arbitrary group operation, i.e., we only need $(\mathcal{M}, \boxplus)$ to be a group, where $\mathcal{M}$ is the closure of the set $\{W_i\}_{i=1}^{M}$ under the group operation $\boxplus$.

[7] We note that the authors of [90] consider pure T-fold ALOHA rather than T-fold IRSA, which may be the root cause of its poor performance.

In what follows, we use a particular $\boxplus$ operation. Namely, the messages are represented as binary vectors of length k, and the $\boxplus$ operation is an element-wise real addition of such vectors. Clearly, we have $\mathcal{M} = \mathbb{Z}^k$. Assume (5.8) holds, then the set $\Lambda = \{f(\boldsymbol{u})\}_{\boldsymbol{u}\in\mathbb{Z}^k}$ forms a group under real addition, and f is an isomorphism[8] between $(\mathbb{Z}^k, +)$ and $(\Lambda, +)$. Thus, we come to *lattices* and *lattice codes* [108].

Let us introduce necessary definitions.

Definition 5.1. *An n-dimensional lattice, Λ, is a discrete additive subgroup of $\mathbb{R}^n$. A lattice can always be described using a generator matrix $\boldsymbol{B} \in \mathbb{R}^{n\times n}$:*

$$\Lambda = \{\lambda = \boldsymbol{B}\boldsymbol{u},\ \ \boldsymbol{u} \in \mathbb{Z}^n\}.$$

Definition 5.2. *The Voronoi region $\mathcal{V}(\lambda)$ of a lattice point λ is the set of all points in $\mathbb{R}^n$ that are closer to λ than to any other lattice point.*

Definition 5.3. *Let $\mathcal{R} \subseteq \mathbb{R}^n$. In what follows, we refer to $\mathcal{R}$ as a shaping[9] region. A lattice code $\mathcal{C}$ is a finite subset of a lattice Λ:*

$$\mathcal{C} = \Lambda \bigcap \mathcal{R}.$$

The code consists of a finite subset of M points from the original lattice:

$$\mathcal{C} = \{\lambda_1, \lambda_2, \ldots, \lambda_M\},$$

where $\lambda_i \in \Lambda$, $i \in [M]$.

Usually, lattice codes are constructed using nested lattices and are called *nested lattice codes*.

Definition 5.4. *A lattice Λ_c is said to be nested in a lattice Λ_f if $\Lambda_c \subseteq \Lambda_f$. We will sometimes refer to Λ_c as the coarse lattice and Λ_f as the fine lattice.*

Definition 5.5. *Let Λ_f be a lattice such that $\Lambda_c \subseteq \Lambda_f$, and let $\mathcal{V}_c(0)$ be the 0-centered Voronoi region for Λ_c. Then,*

$$\mathcal{C} = \Lambda_f \bigcap \mathcal{V}_c(0)$$

is a nested lattice code.

Note that a nested lattice code $\mathcal{C}$ is isomorphic to the quotient group Λ_f/Λ_c, i.e., $\mathcal{C}$ forms a group under addition modulo Λ_c.

We note that the use of lattice codes for MACs is natural, as electromagnetic signals exhibit linear superposition. The illustration of the transmission scheme using nested lattice codes is presented in Fig. 5.7.

The aim of lattice coding is to denoise the sum of messages. Note that only a single-user decoder is required to accomplish this task, which is a clear advantage of this approach. Assume successful

[8] We need to extend the domain of f from $[M]$ to $\mathbb{Z}^k$ based on (5.8).

[9] We do not specify any particular requirements for $\mathcal{R}$ in this definition. Clearly, it is reasonable to choose $\mathcal{R}$ from convex bounded sets. From energy considerations, the closer $\mathcal{R}$ is to the ball centered at $\mathbf{0}$, the better.

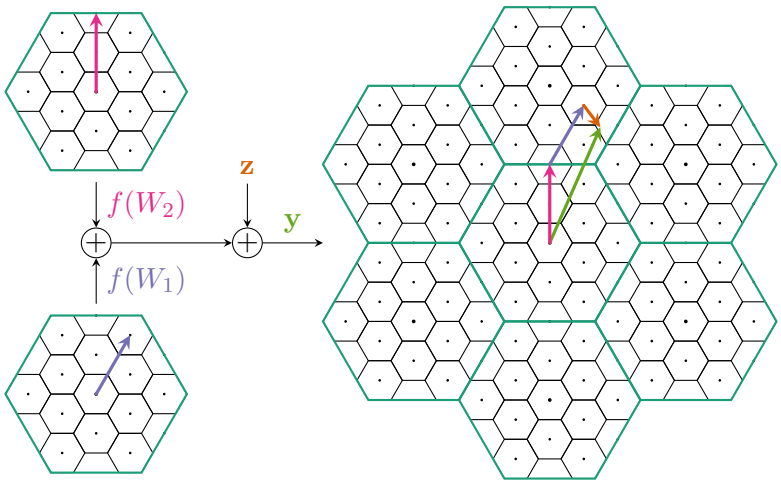

Figure 5.7: Illustration of nested lattice codes for GMAC.

denoising has been carried out. The second phase is to recover the transmitted messages from the denoised sum. For this purpose, one may use an outer code for the *noiseless* adder channel. The only requirement for such a code is unique decodability: all the sums of up to T codewords are different. See Section 2.1 for details.

Now, let us describe the scheme from [90].

Encoding. We start with the two-phase encoding process:

1. Encode the message W for the adder channel:
$$\boldsymbol{x}_{\text{ad}} = f_{\text{ad}}(W),$$
where $f_{\text{ad}} : [M] \to \mathbb{F}_p^{k'}$.
2. Encode $\boldsymbol{x}_{\text{ad}}$ with the lattice code:
$$\boldsymbol{x} = f_{\text{lc}}(\boldsymbol{x}_{\text{ad}}),$$
where $f_{\text{lc}} : \mathbb{F}_p^{k'} \to \mathbb{R}^{n'}$.

Let us describe the main components.

Outer code. The technique proposed in [25] is as follows. Let $\boldsymbol{H} = [\boldsymbol{h}_1, \ldots, \boldsymbol{h}_M]$ be the parity-check matrix of a T-error-correcting code over $\mathbb{F}_p$. Set $\mathcal{C}_{\text{ad}} = \{\boldsymbol{h}_1, \ldots, \boldsymbol{h}_M\}$. This construction ensures the desired property: all sums of up to T codewords (or columns) are distinct. For practical construction, one can use a BCH code over $\mathbb{F}_p$ with low-complexity decoding algorithms (e.g., the Berlekamp-Massey algorithm).

Inner code. For practical implementation, we need a high-dimensional lattice with a low-complexity decoding algorithm. The authors of [90] proposed utilizing the so-called Construction A [108], i.e.,
$$\Lambda = \mathcal{C}_{\text{lc}} + p\mathbb{Z}^n,$$

where p is a prime number and $\mathcal{C}_{\text{lc}}$ is a p-ary error-correcting code with a low-complexity decoding algorithm.

Remark 5.8. *Note that for Construction A, we have $\Lambda_f = \Lambda$ and $\Lambda_c = p\mathbb{Z}^n$. Hence, modulo-$\Lambda_c$ reduction is straightforward: it is just a component-wise modulo-p operation, with the result in the interval $[0, p)$.*

Now, consider the received signal (5.7) and recall that

$$\sum_{i=1}^{r} \boldsymbol{x}^{(i)} = \lambda \in \Lambda,$$

i.e., this sum is a lattice point.

Decoding. The paper [90] does not specify the lattice code explicitly. Instead, the authors suggest computing

$$\boldsymbol{y}' = \boldsymbol{y} \mod p = (\lambda + \boldsymbol{z}) \mod p = \boldsymbol{c} + (\boldsymbol{z} \mod p),$$

where $\boldsymbol{c} \in \mathcal{C}_{\text{lc}}$, and propose using the FBL achievability bound [31] to characterize the capabilities of $\mathcal{C}_{\text{lc}}$.

Here, we describe the algorithm for some low-complexity soft-input code $\mathcal{C}_{\text{lc}}$ over $\mathbb{F}_p$ (e.g., LDPC or a polar code can be used).

1. Let $\boldsymbol{y} = [y_1, \ldots, y_{n'}]$. The input of the $\mathcal{C}_{\text{lc}}$ decoder is the a priori distribution vector

   $$\boldsymbol{D} = [\boldsymbol{d}_1, \boldsymbol{d}_2, \ldots, \boldsymbol{d}_{n'}],$$

   where

   $$\boldsymbol{d}_i = [\Pr(c_i = 0|y_i), \ldots, \Pr(c_i = p-1|y_i)], \quad i \in [n'].$$

   First, we calculate $\boldsymbol{D}$ as follows:

   $$d_{i,m} = \Pr(c_i = m|y_i) = \frac{1}{\sqrt{2\pi}} \sum_{\substack{j=-\infty, \\ (j=m \mod p)}}^{+\infty} \exp[-(y_i - j)^2/2].$$

   Note that in practical implementation, it is reasonable to sum over $j \in [y_i - \Delta, y_i + \Delta]$ for some fixed Δ.

2. Decode $\mathcal{C}_{\text{lc}}$. We obtain

   $$\boldsymbol{y}_{\text{ac}} = \sum_{i=1}^{r} \boldsymbol{x}_{\text{ac}}^{(i)}.$$

3. Decode the outer code for the noiseless adder channel. Taking into account that the code consists of the columns of the BCH parity-check matrix, $\boldsymbol{y}_{\text{ac}}$ is nothing more than a syndrome. Indeed, we have

$$\boldsymbol{y}_{\text{ac}} = \boldsymbol{H}\boldsymbol{s},$$

where $\boldsymbol{s}$ is the indicator vector of transmitted messages, i.e., $\{W_1, \ldots, W_r\} = \text{supp}(\boldsymbol{s})$.

Thus, we arrive at the classical BCH code decoding problem. Note that $r \leq T$ and our code is a T-error-correcting code.

Remark 5.9. *The paper [90] focuses on the case $p = 2$, as binary codes are well-developed and significantly outperform non-binary ones in terms of soft decoding complexity. In what follows, we consider the case $p = 2$, in which case $\boldsymbol{D}$ can be replaced with the input log-likelihood ratio (LLR) vector.*

Remark 5.10. *One may argue that BCH decoding has high complexity since the parity-check matrix $\boldsymbol{H}$ has size $n' \times M$ (recall that $M = 2^k$ and $k = 100$ bits is of interest). Classical BCH decoding involves calculating the syndrome, which has linear complexity in M (thus, exponential complexity in k). However, there is no need to calculate the syndrome in our case, as this operation is performed in the channel, and we receive it directly. It can be shown (see [90]) that the complexity of BCH decoding is $O(kT^2 \log^2(T) \log\log(T))$ operations in $\mathbb{F}_{2^k}$.*

Remark 5.11. *In the scheme described above, users transmit symbols from the set $\{0, \sqrt{2P}\}$ (or equivalently, $\{0, 1\}$ under the power constraint P). In a practical implementation, it is more reasonable to use the set $\{-\sqrt{P}, \sqrt{P}\}$, which corresponds to a lattice coset. See [90] for further details.*

IRSA-S with LDPC codes

In this section, we consider the IRSA-S scheme (presented in Section 5.1.3), which is based on LDPC codes [100]. We employ a joint iterative decoding algorithm with low computational complexity [109, 52].

Recalling the IRSA-S scheme, the slot is divided into two parts, such that $n' = n'_p + n'_c$ channel uses. In the first part, we utilize BCH subcodes, as described in Appendix B.2.3, as a suitable candidate for the CS codebook. The preamble is recovered using the methods detailed in Appendix B.

Now, let us focus on the joint decoding of the LDPC codes positioned in the last n'_c channel uses. We define different codebooks, $\mathcal{C}_1, \ldots, \mathcal{C}_r$, which are derived from a common codebook $\mathcal{C}$ through random permutations based on ϕ_{W_p}.

The received signal in the slot is given by (5.7), where $\boldsymbol{x}^{(i)} = \tau\left(\boldsymbol{c}^{(i)}\right)$ is a BPSK-modulated codeword of each active user. To recover users' codewords, we employ a decoder based on the iterative belief propagation (BP) algorithm, which is a specific case of the broader class of MPAs. The decoding procedure can be represented as a graph (factor graph) [110], shown in Fig. 5.8.

There are three types of nodes in the graph. The LDPC codes of the users are represented by Tanner graphs with variable nodes (orange) and check nodes (blue). For the variable and check

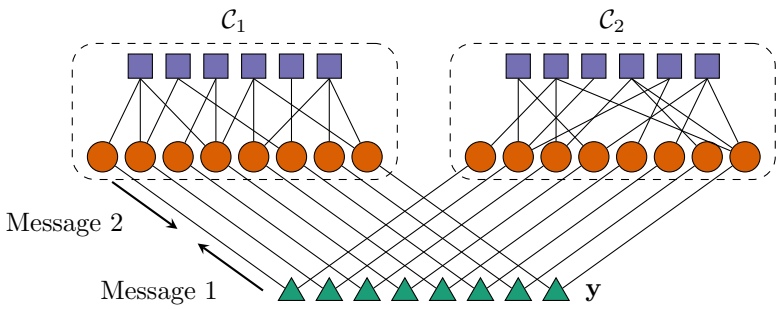

Figure 5.8: The joint decoding algorithm and factor graph for the case $r = 2$ are illustrated. Orange circles and blue squares represent the variable and check nodes of the LDPC codes $\mathcal{C}_1$ and $\mathcal{C}_\in$, respectively. The messages exchanged between these nodes follow the standard decoding procedure of LDPC codes. Additionally, there are functional nodes, depicted as green triangles, that represent the received signal $\boldsymbol{y}$.

nodes in the subgraphs of different users, the message exchange rule follows the same procedure as in the single-user case (see [65]).

Additionally, there is a third type of node in the graph – functional nodes (marked by green triangles). These nodes correspond to the elements of the received signal $\mathbf{y}$. To fully specify the decoding algorithm, we need to describe the messages exchanged between the variable nodes of each user's code and the functional nodes (Message type 1), as well as the messages from the functional nodes to the variable nodes of the users' codes (Message type 2).

Message type 1 (from functional nodes to LDPC codes). Following the principles of message-passing algorithms [65], the update rule to compute the message μ sent to the j-th variable node of the i-th user ($j \in [n_c']$, $i \in [r]$) from the functional node $\mathcal{F}_j$ is given as follows.

First, let us fix the index of the functional node j and omit it for clarity (as the j-th functional node may only be connected to the j-th variable node of each user's LDPC code; see Fig. 5.8). Let $y = \boldsymbol{y}_j$ be the signal received at the j-th position. Let $\boldsymbol{b} \in \{0,1\}^r$ represent a binary vector, where $\boldsymbol{b}_i$ corresponds to the bit value of the i-th user's codeword $\boldsymbol{c}^{(i)}$. Also, introduce a function:

$$\eta\left(\boldsymbol{b}\right) = p\left(y|\boldsymbol{b}\right) = \exp\left\{-\left(y - \sum_{i=1}^{r} \tau\left(\mathbf{b}_i\right)\right)^2\right\}, \tag{5.9}$$

which represents the p.d.f. of the received signal conditioned on $\boldsymbol{b}$, up to some constant multiplier that can be ignored when considering likelihood ratios. Note that $\boldsymbol{b}$ completely defines the noiseless received signal.

Let $\mu_{i\to\mathcal{F}}$ be the message sent from the i-th user's variable node to the functional node. Let l_i

represent the LLR value given by the i-th user's LDPC decoder. Then, define:

$$\mu_{i\to\mathcal{F}}(0) \triangleq \Pr[\boldsymbol{b}_i = 0] = \frac{e^{l_i}}{e^{l_i}+1}, \quad \mu_{i\to\mathcal{F}}(1) \triangleq 1 - \mu_{i\to\mathcal{F}}(0), \tag{5.10}$$

where l_i corresponds to message type 2. The values of l_i are initialized to zero and then updated by the iterative decoding algorithm.

Similarly, $\mu_{\mathcal{F}\to i}$ is the message sent from the functional node to the i-th user's variable node, and $\mu_{\mathcal{F}\to i}(0)$ is given by:

$$\mu_{\mathcal{F}\to i}(\beta) = \sum_{\boldsymbol{b}\in\{0,1\}^r, \boldsymbol{b}_i=\beta} \left[\eta(\boldsymbol{b}) \times \prod_{t\neq i} \mu_{t\to\mathcal{F}}(\boldsymbol{b}_t) \right], \quad \beta \in \{0,1\}, \tag{5.11}$$

where $\eta(\boldsymbol{b})$ is given by (5.9), and $\mu_{t\to\mathcal{F}}(\cdot)$ is given by (5.10). The summation above has 2^{r-1} terms, where r is the collision order. Consequently, the number of computations required to obtain the messages from the functional node $\mathcal{F}_i$ grows exponentially with the number of users.

Now, let us express the messages in terms of the LLRs $l_{\mathcal{F}\to i}$:

$$l_{\mathcal{F}\to i} = \log\left(\frac{\mu_{\mathcal{F}\to i}(0)}{\mu_{\mathcal{F}\to i}(1)}\right), \tag{5.12}$$

where $\mu_{\mathcal{F}\to i}(\beta)$ is given by (5.11). The value $l_{\mathcal{F}\to i}$ is then passed as the input LLR to the single-user LDPC decoder.

Note that an alternative approach to functional node processing is given by soft SIC (SoIC), as proposed in [111]. Soft interference represents the expected value of the codeword symbol $\mathbb{E}\boldsymbol{c}^{(i)}$, derived from (5.10) as follows (considering BPSK modulation):

$$\mathbb{E}\boldsymbol{x}^{(i)} = \sqrt{P}\left(\mu_{i\to\mathcal{F}}(0) - \mu_{i\to\mathcal{F}}(1)\right).$$

Then, the LLR specified in (5.12) is calculated as the LLR for BPSK modulation with SoIC:

$$l_{\mathcal{F}\to i} = 2\sqrt{P}\left(y - \sum_{t=1, t\neq i}^{r} \mathbb{E}\boldsymbol{x}^{(t)}\right),$$

where y is the received signal at the considered position, and $2\sqrt{P}y$ corresponds to the LLR of a BPSK-modulated signal with a power constraint P.

Message 2 (decoding the LDPC code). After decoding the functional nodes and computing the LLRs for the bits, this information is used to decode the LDPC code. The user applies the MPA algorithm to the LDPC code, which can be either the Sum-Product or the Min-Sum algorithm [65]. The output of this algorithm updates the values l_i presented in (5.10).

Finally, the message exchange schedule can be arbitrary. Functional node decoding (propagation of message type 1) may be followed by several decoding iterations of each single-user LDPC code.

To construct the LDPC codes, the modified protograph extrinsic information transfer (PEXIT) method from [112] is used. Here, only the significant differences from the single-user case are highlighted. The multi-user Gaussian channel is not symmetric; therefore, the PEXIT method cannot be applied directly to the all-zero codeword. To address this issue, we consider *channel adapters* from [113]. The resulting channel is symmetric, allowing us to use only the all-zero codeword for code construction. Note that the random adapters were used only for code construction.

IRSA-F with polar codes

Let us consider the IRSA-F scheme with polar codes employed in each slot [101]. The choice of polar codes is motivated by two reasons. First, polar codes [114] are a powerful tool for short code lengths under the successive cancellation list (SCL) decoding algorithm [115]. Second, as previously discussed in Section 2.2.3, polar codes exhibit capacity-achieving properties in MAC. In this section, we describe how polar codes can be used in GMAC in the FBL regime and present a joint successive cancellation (JSC) decoding algorithm.

Consider Arıkan's kernel and the corresponding *polar transform* of size $n' = 2^m$:

$$G_2 \triangleq \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}, \quad G_{n'} \triangleq G_2^{\otimes m},$$

where $\otimes$ denotes the Kronecker power.

To construct an (n', k) polar coset code, denote the set of frozen positions by $\mathcal{F}$, with $|\mathcal{F}| = n' - k$. Let $\boldsymbol{u}_{\mathcal{F}}$ denote the projection of the vector $\boldsymbol{u}$ onto the positions in $\mathcal{F}$. Now, we define a *polar coset code* $\mathcal{C}$ as follows:

$$\mathcal{C}\left(n', k, \mathcal{F}, \boldsymbol{f}\right) = \left\{ \boldsymbol{c} = \boldsymbol{u}^T G_{n'} \;\middle|\; \boldsymbol{u} \in \{0,1\}^{n'},\ \boldsymbol{u}_{\mathcal{F}} = \boldsymbol{f} \right\}.$$

The users utilize *different* polar coset codes $\mathcal{C}_i(n', k, \mathcal{F}_i, \mathbf{f}_i)$, $i = 1, \ldots, r$. Consider the i-th user. To send the information word $\boldsymbol{u}^{(i)}$, the user first encodes it using the code $\mathcal{C}_i(n', k, \mathcal{F}_i, \mathbf{f}_i)$, obtaining a codeword $\mathbf{c}^{(i)}$. This is then followed by BPSK modulation, resulting in $\boldsymbol{x}^{(i)} = \tau\left(\mathbf{c}^{(i)}\right)$. We refer the reader to [101], where the choice of information and frozen positions, along with frozen bit values, has been carefully optimized. In this section, we discuss how to decode a superposition of polar coset codes in a slot. In what follows, we assume a collision of order r.

To specify the decoding algorithm, let us first consider how the channel output is represented. As in the joint LDPC decoding algorithm, we consider a received signal position y, omitting the index j in $\boldsymbol{y}_j$. At this position, each of the r users may transmit either zero or one, resulting in 2^r possible transmitted signal values. These values can be indexed by a binary vector $\boldsymbol{b} \in \{0,1\}^r$, yielding 2^r points.

For the single-user case, the LLR represents a sufficient statistic for the received signal. With r simultaneous transmissions, we instead operate with a p.m.f. specified over *tuples* of bits of length r. The prior p.m.f. is determined by the received signal y at the given position as follows:

$$\mu\left(\boldsymbol{b}\right) \propto \eta\left(\boldsymbol{b}\right),$$

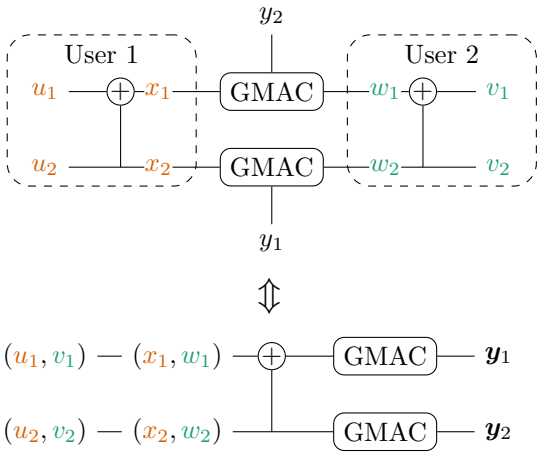

Figure 5.9: Representation of the GMAC as a polar code over $\mathbb{Z}_2^r$ for $r = 2$ and $n' = 2$.

where $\eta(\cdot)$ is defined in (5.9). The p.m.f. specified above can be obtained for each received position. Let us denote the channel output as a matrix $\boldsymbol{M} = [\mu_1, \ldots, \mu_{n'}]$ of size $2^r \times n'$, where the j-th column represents the p.m.f. values μ.

To describe the JSC decoding procedure, we begin with a simple yet instructive example for $n' = 2$ (see Fig. 5.9), illustrating the key idea behind the proposed method. We observe that, instead of working with bits from different users and multiple polar codes, we can treat the problem as working with a single polar code over $\mathbb{Z}_2^r$. In this example, we first decode a bit configuration (tuple) (u_1, v_1) – the first bits of the users – followed by a tuple (u_2, v_2) – the second bits of the users. Recall that the decoder operates with tuple distributions $\boldsymbol{M}$ rather than probabilities of individual bits.

To decode the tuple (u_1, v_1), we first need to calculate the distribution of the sum of two random variables over $\mathbb{Z}_2^r$. In what follows, we refer to this operation as the *check-node operation (CNOP)*. This can be done via convolution, i.e., $\hat{\mu}_1 = \mu_1 * \mu_2$. Since we work in the Abelian group $\mathbb{Z}_2^r$, there exists a Fourier transform (FT) Φ. In what follows, to perform a convolution, we use the fast Fourier transform (FFT)-based technique proposed in [116] for the case of LDPC codes over Abelian groups. Thus, the final rule is as follows:

$$\hat{\mu}_1(\boldsymbol{b}) \propto \Phi^{-1}(\Phi(\mu_1) \odot \Phi(\mu_2))(\boldsymbol{b}) \quad \forall \boldsymbol{b} \in \mathbb{Z}_2^r, \tag{5.13}$$

where $\odot$ denotes element-wise multiplication.

After calculating the p.m.f. $\hat{\mu}_1$, we make a hard decision $\hat{\boldsymbol{b}}_1$, taking into account the values of frozen bits at this position. Once $\hat{\boldsymbol{b}}_1$ is found, we proceed with the *variable-node operation (VNOP)*. The rule is given by:

$$\hat{\mu}_2(\boldsymbol{b}) \propto \mu_1\left(\boldsymbol{b} + \hat{\boldsymbol{b}}_1\right)\mu_2(\boldsymbol{b}) \quad \forall \boldsymbol{b} \in \mathbb{Z}_2^r. \tag{5.14}$$

Finally, the hard decision $\hat{\boldsymbol{b}}_2$ is made using $\hat{\mu}_2$.

Now let us proceed to the case $n' > 2$. Similar to the single-user case, we consider a recursive decoding algorithm that relies on a polar code representation based on the $(U, U + V)$ construction.

Algorithm 1 Joint successive cancellation (JSC) decoding algorithm

function JSC($\mathcal{C}\,(n, k, \mathcal{F}, \boldsymbol{f})$, $\boldsymbol{M}$)
  if $n = 1$ **then** ▷ Make a hard decision
    $\hat{\boldsymbol{b}} \leftarrow \text{hard}\,(\mathcal{F}, \boldsymbol{f}, \boldsymbol{M})$
    return $\hat{\boldsymbol{b}}$, $\boldsymbol{M}$
  else ▷ Proceed recursively
    $\mathcal{C}_{\mathcal{U}}$, $\mathcal{C}_V \leftarrow \mathcal{C}$ ▷ Perform a split using (5.15)
    $[\boldsymbol{M}_U, \boldsymbol{M}_V] \leftarrow \boldsymbol{M}$ ▷ Split into halves
    $\hat{\boldsymbol{M}}_U \leftarrow \text{CNOP}\,(\boldsymbol{M}_U, \boldsymbol{M}_V)$ ▷ Per-column CNOP (5.13)
    $\hat{\boldsymbol{B}}_U,\ \tilde{\boldsymbol{M}}_U \leftarrow \text{JSC}\left(\mathcal{C}_U, \hat{\boldsymbol{M}}_U\right)$ ▷ Decode $\mathcal{C}_U$ recursively
    $\hat{\boldsymbol{M}}_V \leftarrow \text{VNOP}\left(\boldsymbol{M}_U, \boldsymbol{M}_V, \hat{\boldsymbol{B}}_U\right)$ ▷ Per-column VNOP (5.14)
    $\boldsymbol{B}_V,\ \tilde{\boldsymbol{M}}_V \leftarrow \text{JSC}\left(\mathcal{C}_V, \hat{\boldsymbol{M}}_V\right)$ ▷ Decode $\mathcal{C}_V$ recursively
    $\hat{\boldsymbol{B}} \leftarrow \left[\hat{\boldsymbol{B}}_U \oplus \hat{\boldsymbol{B}}_V, \hat{\boldsymbol{B}}_V\right]$ ▷ Element-wise XOR
    return $\hat{\boldsymbol{B}}$, $\left[\tilde{\boldsymbol{M}}_U, \tilde{\boldsymbol{M}}_V\right]$ ▷ All tuples of length n
  end if
end function

Let us define a polar code split procedure. Given a polar code $\mathcal{C}$, we specify two subcodes $\mathcal{C}_U$ and $\mathcal{C}_V$ as follows:

$$\mathcal{C}\,(n, k, \mathcal{F}, \boldsymbol{f}) \overset{\text{split}}{\to} \mathcal{C}_U\,(n/2, k_U, \mathcal{F}_U, \boldsymbol{f}_U)\,, \quad \mathcal{C}_V\,(n/2, k_V, \mathcal{F}_V, \boldsymbol{f}_V)\,, \tag{5.15}$$

where $\mathcal{F}_U \subseteq [n/2]$, $\mathcal{F}_V \subseteq \{n/2 + 1, \ldots, n\}$, $\mathcal{F}_U \cup \mathcal{F}_V = \mathcal{F}$, $\boldsymbol{f} = (\boldsymbol{f}_U\ \boldsymbol{f}_V)$, and $k = k_U + k_V$.

Now, let us specify the recursive decoding algorithm. We denote a matrix $\boldsymbol{B} = [\boldsymbol{b}_1, \ldots, \boldsymbol{b}_{n'}]$ of size $r \times n'$ specifying the estimated bit tuples. The JSC is presented in Algorithm 1, where CNOP and VNOP operations are applied per column with respect to the input arguments and are denoted as functions $\text{CNOP}\,(\mu_1, \mu_2)$ and $\text{VNOP}\,(\mu_1, \mu_2, \boldsymbol{b})$, respectively. Finally, $\oplus$ denotes an element-wise modulo-two sum. Note that the structure of Algorithm 1 is exactly the same as that of the single-user polar code successive cancellation (SC) decoding algorithm, where single bits are replaced by bit tuples, and LLRs are replaced by probability mass functions (p.m.f.) μ.

Remark 5.12. *It is worth noting that we can easily improve the decoding procedure by using the SCL decoding method [115]. In other words, we take into account not only the most probable path (which, in our case, consists of tuples) but also ℓ different paths with the highest metric. To derive the path metric update, recall the hard decision step from Algorithm 1:*

$$\hat{\boldsymbol{b}} \leftarrow \textit{hard}\,(\mathcal{F}, \boldsymbol{f}, \boldsymbol{M})\,,$$

which corresponds to the case where $n = 1$. Hence, the matrix $\boldsymbol{M}$ represents a single p.m.f. μ. Thus, the path metric is updated by multiplying it with $\mu\left(\hat{\boldsymbol{b}}\right)$.

5.2 Sparse interleave division multiple-access

In this section, we describe the scheme proposed in [92]. This scheme is very similar to IRSA-S with LDPC codes, as presented above. Specifically, it implements the idea of sparsifying collisions, inherent in T-fold IRSA, while also utilizing users' LDPC codes. However, the major difference lies in the randomization of the locations of the LDPC codeword symbols. The scheme assumes random placement rather than slotted transmission.

Slotted transmission is well-suited for the basic IRSA protocol, as the receiver does not know the transmission graph and reconstructs it on the fly. Knowing the slot locations enables the receiver to decode. Now, consider the IRSA-F protocol. The CS part (or preambles) allows the receiver to first reconstruct the graph and then perform decoding. In this case, the main drawbacks of the scheme from [100] are as follows:

1. Slotted transmission depends on the existence of high-performance short-length codes. As the number of active users increases, the scheme ensures that the slot length decreases. Short LDPC codes are known to perform poorly, while polar codes are good but lack flexibility in length adaptation. Thus, designing effective multi-user codes for short block lengths remains an open problem.
2. Once the message is decoded, its copies are peeled from other slots using SIC. Consequently, the decoding and peeling operations are separate. Moreover, peeling is performed in a hard decision manner. However, since the transmission graph is known, a joint message-passing decoder can be applied.
3. Slotted transmission is merely a specific interleaving strategy. The class of all possible permutations is much broader and potentially contains more effective solutions.

The scheme from [92] addresses these issues. The authors demonstrate that a single sparse joint Tanner graph spanning all transmissions can significantly improve performance compared to the schemes in [100, 101].

Remark 5.13. *The scheme in [92] can be interpreted as a sparse version of IDMA [91], adapted to the uncoordinated and unsourced MAC by incorporating an additional CS component. The sparsity ensures acceptable computational complexity when decoding over the joint graph. We also note that the idea of using sparse access sequences appeared in [45], although that work focused on short repetition codes.*

Now, let us describe the transmission process. Consider the i-th user and split the message into two parts: $W_i = [W_{i,p}, W_{i,c}]$, where $W_{i,p} \in [M_p]$, $W_{i,c} \in [M_c]$, and $M = M_p M_c$. Similar to the IRSA-F protocol: (a) the replica count depends only on $W_{i,p}$, and (b) the preamble of length n'_p is placed before the frame, so the total length n is allocated to the preamble (n_p) and the remaining portion of length $n - n_p$ where users transmit their codewords.

In what follows, we consider the case $\log_2 M_p = k_p$ and $\log_2 M_c = k_c$, where $k_p, k_c \in \mathbb{N}$, with $k_p + k_c = k$.

1. The preamble is chosen using the preamble encoder $f_p : [M_p] \to \mathbb{R}^{n_p}$. Recall that $\|f_p(W_p)\|_2^2 \leq P_p n_p$ for all $W_p \in [M_p]$.

2. All users utilize the same binary $[n_c, k_c]$ LDPC code. The remaining k_c message bits are encoded using this code, and the i-th user obtains the binary codeword $\mathbf{c}_i$. The user then performs BPSK modulation, equivalently expressed as

$$\tilde{\mathbf{x}}^{(i)} = \tau(\mathbf{c}^{(i)}), \quad \tau(\mathbf{c}^{(i)}) = (\tau(c_1^{(i)}), \ldots, \tau(c_{n_c}^{(i)})).$$

3. Let $L = (n-n_p)/n_c$, which represents the maximum repetition count[10]. Applying the function $\ell : [M_p] \to [L]$, we determine the replica count $\ell(W_{i,p})$. Once again, we emphasize that the replica count is a deterministic function of $W_{i,p}$ and remains independent of $W_{i,c}$. At this step, we construct the word

$$\mathbf{x}^{(i)} = [\underbrace{\tilde{\mathbf{x}}^{(i)}, \tilde{\mathbf{x}}^{(i)}, \ldots, \tilde{\mathbf{x}}^{(i)}}_{\ell(W_{i,p})}, \mathbf{0}],$$

   where the codeword $\tilde{\mathbf{x}}^{(i)}$ is repeated $\ell(W_{i,p})$ times and then appended with $(n-n_p)-\ell(W_{i,p})n_c$ zeros. Thus, $\mathbf{x}^{(i)}$ has length $n - n_p$.

4. Finally, we select the permutation $\pi_{W_{i,p}}$ based on the $W_{i,p}$ message part and transmit the word

$$\left[f_p(W_{i,p}),\ \pi_{W_{i,p}}(\mathbf{x}^{(i)})\right]$$

   over the channel.

Now, consider the decoding process:

1. Decode the preamble part using a CS decoder by solving the problem (5.6), where $\mathbf{u}_p = (u_{p,1}, \ldots, u_{p,M_p})$ and the sparsity $\mathrm{wt}(\mathbf{u}_p) = K_a$.

2. Reconstruct the graph and extract $\{\pi_{\hat{W}_{i,p}}\}$ for $i \in [K_a]$. If all preambles are successfully recovered, the graph and code transformations are known.

3. Apply a joint message-passing decoder. The rules remain exactly the same as in Section 5.1.4, with the only difference being the graph structure.

Remark 5.14. *Similar to the IRSA-F protocol, we have*

$$P_p n_p + P_c n_c L'(1) \leq Pn.$$

Additionally, we highlight an approach to the URA problem that employs a tensor-based modulation method [117]. Given a large coherence block, this method is also applicable to the block fading model and scenarios where the receiver is equipped with multiple antennas.

[10]Recall that this scheme does not use slots.

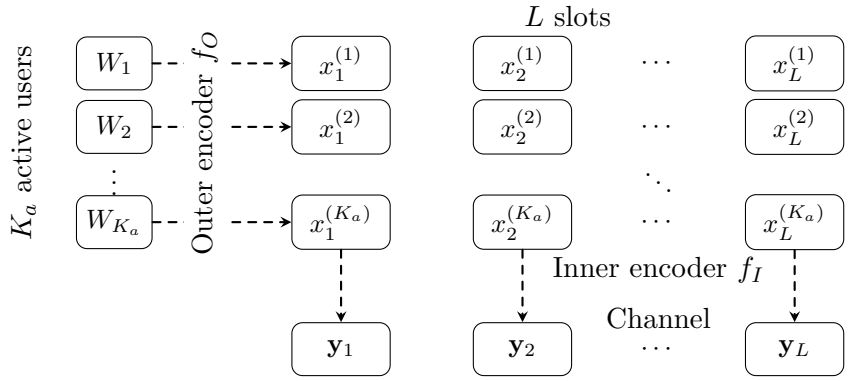

Figure 5.10: CCS scheme. First, the outer code of length L over the alphabet $[Q]$, with the encoder f_O, is applied. Then, the inner encoder f_I is applied to each symbol of the outer code. Finally, each inner codeword is transmitted through the Gaussian channel (5.16).

5.3 Coded compressed sensing

In this section, we describe an inspirational scheme from [118, 93]. A similar approach has already been used in CS and group testing literature [119, 120, 121, 122]. However, [93] presents the first application of this approach to the URA problem. Recall that the URA problem is a CS problem of immense dimension. The scheme from [93] utilizes the divide-and-conquer strategy, i.e., it splits the task into subtasks of smaller dimensions, solves the corresponding CS problems, and then assembles the results using an outer tree code. We note that a similar code construction, namely a convolutional code, was used in [123], but for a different single-user channel model (jamming channel or J-channel). The main drawback of the tree code-based scheme proposed in [93] is its inability to deal with errors, i.e., the codeword is not recovered if at least one of its fragments is lost. This drawback was addressed in [94]. Our description is based on the latter paper.

Let us briefly describe a CCS scheme from [93]. The transmission scheme is shown in Fig. 5.10. The idea is to apply a divide-and-conquer strategy implemented via concatenated coding. Recall that a frame of n real channel uses is divided into L slots of length n'. Consider the i-th user aiming to transmit a message $W_i \in [M]$. We use an outer code of length L over the alphabet $[Q]$ with the encoder $f_O : [M] \to [Q]^L$. First, we obtain a codeword $\boldsymbol{x}^{(i)} = (x_1^{(i)}, x_2^{(i)}, \dots, x_L^{(i)}) = f_O(W_i)$, where $\boldsymbol{x}^{(i)} \in [Q]^L$. Then, the inner encoder $f_I : [Q] \to \mathbb{C}^{n'}$ is applied to each symbol $x_l^{(i)}$, $l \in [L]$. The resulting codewords of the inner code are transmitted in the corresponding slots. Note that, in contrast to ALOHA protocol-based schemes, the user's transmission occupies the entire frame (L slots).

For the l-th slot, we have

$$\mathbf{y}_l = \sum_{i=1}^{K_a} f_I\left(x_l^{(i)}\right) + \mathbf{z}_l, \quad l \in [L], \tag{5.16}$$

where $\mathbf{z}_l \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_{n'})$.

To recover the transmitted codewords, we first solve a CS problem for each slot. Note that the

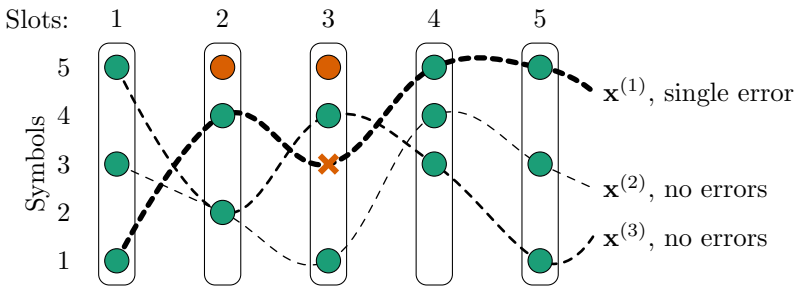

Figure 5.11: Outer code and the list-recovery problem (Example 5.3, case B). Rectangles represent slots. Green circles correspond to correctly detected symbols, orange circles correspond to falsely detected symbols, and orange crosses correspond to missed symbols. Dashed lines represent outer code codewords. For example, a codeword $\boldsymbol{x}^{(1)}$ is covered by the received lists in all but one position.

dimensions of these problems are much smaller compared to the dimension of the original problem (see (3.4)). Thus, one can use standard CS algorithms. For further details, see Section 5.6.3.

In the following example, we illustrate the transmission and the output of the inner decoder.

Example 5.3. *Let $K_a = 3$, $Q = L = 5$ and assume the codewords*

$$\begin{aligned} \mathbf{x}^{(1)} &= (1,\ 4,\ 3,\ 5,\ 5), \\ \mathbf{x}^{(2)} &= (3,\ 2,\ 1,\ 4,\ 3), \\ \mathbf{x}^{(3)} &= (5,\ 2,\ 4,\ 3,\ 1). \end{aligned}$$

are transmitted. The channel output for each slot is described by (5.16)*. For example, for the third slot, we have the following channel output realization:*

$$\boldsymbol{y}_3 = f_I(3) + f_I(1) + f_I(4) + \boldsymbol{z}_3.$$

Next, we solve a CS problem for each slot and obtain $\mathcal{Y}_l \subseteq [Q]$, $l \in [L]$ which represents the estimates of the sets of Q-ary symbols transmitted in each slot. Let us consider two cases:

***Case A*:** *There are no errors in the received lists. Thus,*

$$\begin{aligned} \mathcal{Y}_1 &= \{1, 3, 5\}, \\ \mathcal{Y}_2 &= \{2, 4\}, \\ \mathcal{Y}_3 &= \{1, 3, 4\}, \\ \mathcal{Y}_4 &= \{3, 4, 5\}, \\ \mathcal{Y}_5 &= \{1, 3, 5\}. \end{aligned}$$

We note that the decoder returns a set, and thus we have a list of size 2 for the second slot. There is no information about the symbols' multiplicities.

***Case B*:** *There are errors in the received lists. Let*

$$\mathcal{Y}_1 = \{1, 3, 5\},$$

$$\mathcal{Y}_2 = \{2, 4, 5\},$$
$$\mathcal{Y}_3 = \{1, 4, 5\},$$
$$\mathcal{Y}_4 = \{3, 4, 5\},$$
$$\mathcal{Y}_5 = \{1, 3, 5\}.$$

For the second slot, we observe a falsely detected symbol, 5*. For the third slot, the symbol* 3 *is lost, and the symbol* 5 *is falsely detected.*

The task of the outer code is to assemble the original codewords from the received lists. The lists may contain errors (missed and falsely detected symbols). This problem is known as the list-recovery problem [124]. We illustrate this process for Example 5.3, case B, in Fig. 5.11.

5.3.1 Channel for the outer code

Let us start with the case with no errors in the output lists. Clearly, the resulting channel $\mathfrak{W}_A$ is the channel without intensity information (A-channel) from [125], which is also called a hyperchannel in the literature (see [126]). Let the symbols $\mathsf{x}^{(1)}, \mathsf{x}^{(2)}, \ldots, \mathsf{x}^{(K_a)} \in [Q]$ be transmitted. Then, the channel output is

$$\mathcal{Y}^{(A)} = \bigcup_{i=1}^{K_a} \mathsf{x}^{(i)}.$$

The capacity of the A-channel is derived in [127]. If we consider indicator vectors of sets, this channel can be represented as a vector OR-channel.

Our channel $\mathfrak{W}$ is a concatenation of the A-channel $\mathfrak{W}_A$ with the channel $\mathfrak{W}_E$ that introduces errors. Let us consider the channel $\mathfrak{W}_E$ in more detail. The input of the channel is the set of transmitted symbols $\mathcal{Y}^{(A)} \subseteq [Q]$. Let $\mathcal{Y} \subseteq [Q]$ be the output set. We introduce the missed detection probability p_m as the probability that the transmitted symbol is not in the received list $\mathcal{Y}$. The false alarm probability p_f is the probability that some symbol that is not transmitted appears in the received list, i.e., for $x \in [Q]$,

$$p_m = \Pr\left[x \notin \mathcal{Y} | x \in \mathcal{Y}^{(A)}\right],$$
$$p_f = \Pr\left[x \in \mathcal{Y} | x \notin \mathcal{Y}^{(A)}\right],$$

where the channel operates independently on the elements.

Example 5.4. *Let us consider Example 5.3 and the second slot with* $\mathcal{Y}_2^{(A)} = \{2, 4\}$*. Each element* $\{2, 4\}$ *can be missed independently with probability* p_m*, and each element* $\{1, 3, 5\}$ *can appear in the received list* $\mathcal{Y}_2$ *independently with probability* p_f*.*

The capacity of this channel is derived in Appendix E.1.

5.3.2 Random coding bound

In this section, we present a RCB for the outer code. In what follows, we consider the *ensemble* of codes.

Definition 5.1. *Let $\mathcal{E}_1(M, L)$ be the ensemble of codebooks of size $M \times L$, where each element is sampled i.i.d. from* Unif $([Q])$.

Now, let us describe the decoding algorithm. Let

$$\boldsymbol{\mathcal{Y}} = (\mathcal{Y}_1, \dots, \mathcal{Y}_L), \quad \mathcal{Y}_l \subseteq [Q],$$

i.e., the received sequence of L sets, each set being an output of the channel $\mathfrak{W}$.

We require the decoder[11] to output all the messages W, such that

$$d(\boldsymbol{\mathcal{Y}}, \boldsymbol{x}) \leq t, \tag{5.17}$$

where $\boldsymbol{x} = f_O(W)$, $d(\boldsymbol{\mathcal{Y}}, \boldsymbol{x}) = |\{l : x_l \notin \mathcal{Y}_l\}|$.

Theorem 5.3 (Coded compressed sensing random coding bound [94])**.** *There exists a code $\mathcal{C} \in \mathcal{E}_1(M, L)$, such that*

$$P_e = \sum_{i=t+1}^{L} \binom{L}{i} p_m^i (1 - p_m)^{L-i}, \tag{5.18}$$

and

$$P_f \leq \sum_{r=1}^{K_a} \left[\nu_r (M - r) \sum_{i=0}^{t} \binom{L}{i} (1 - \mu_r)^i \mu_r^{L-i} \right], \tag{5.19}$$

where μ_r is given by (E.2) *and*

$$\nu_r = \frac{M!}{(M - r)! M^{K_a}} \left\{ {K_a \atop r} \right\},$$

where $\left\{ {K_a \atop r} \right\}$ is the Stirling number of the second kind.

Proof. Let us start with the FAR. Recall that $\mathcal{T}$ is the set of transmitted messages and $\mathcal{R} \subseteq [M]$ is a received list. Let us introduce the events

$$E_r = \{|\mathcal{T}| = r\}, \ r \in [K_a].$$

Note that the probability $\Pr[E_r]$ of obtaining r unique original elements when K_a items are taken with replacement from a sample of size M is equal to ν_r (see, e.g., [128]).

We have

$$P_f = \sum_{r=1}^{K_a} \nu_r \Pr[\mathcal{R} \backslash \mathcal{T} \neq \emptyset \mid E_r].$$

[11] Here and in what follows, we consider the outer code only and thus omit the word "outer".

Let us proceed with $\Pr[\mathcal{R}\backslash\mathcal{T} \neq \emptyset \mid E_r]$. W.l.o.g., let us assume that $\mathcal{T} = [r]$ and calculate the probability that some another message satisfies condition (5.17). Let us fix the message $\hat{W} \in [M] \setminus [r]$, and let $\hat{\boldsymbol{x}} = f_O(\hat{W})$. Recall (see Section 5.3.1),

$$\Pr[\hat{x}_l \in \mathcal{Y}_l \mid E_r] = \mu_r,$$

thus the probability of accepting the message $\hat{W}$ is equal to

$$\Pr\left[\sum_{l=1}^{L} \xi_l \leq t\right],$$

where $\xi_l \overset{i.i.d.}{\sim} \mathrm{Bern}(1-\mu_r)$. Applying the union bound, we obtain P_f from the theorem statement.

Finally, note that as we utilize a single-user receiver, P_e is just the probability that more than t errors in the transmitted codeword have occurred. □

In what follows, we are interested in codebooks of size $M \approx 2^{100}$ and utilize the following upper bound for P_f.

Corollary 5.1. *The following inequality holds:*

$$P_f \leq (M - K_a) \sum_{i=0}^{t} \binom{L}{i} (1-\mu_{K_a})^i \mu_{K_a}^{L-i} + p',$$

where

$$p' = \Pr[|\mathcal{S}| < K_a] = 1 - \prod_{i=0}^{K_a-1} \left(1 - \frac{i}{M}\right) \leq \frac{\binom{K_a}{2}}{M}.$$

5.3.3 t-tree code-based practical scheme

Let us describe and analyze a practical code construction, which is a modification of the tree code. The code structure is the same as in [93], with a crucial difference: a decoder capable of correcting up to t errors.

Code construction. Let us represent the user message W as a binary k-bit vector $\boldsymbol{u}$ and split it into chunks $\boldsymbol{u}^T = (\boldsymbol{u}_1^T, \ldots, \boldsymbol{u}_L^T)$, such that $\boldsymbol{u}_l$ is of length b_l bits, $l \in [L]$, and $\sum\limits_{l=1}^{L} b_l = k$.

Recall that $\boldsymbol{u}_{[l]} = (\boldsymbol{u}_1, \ldots, \boldsymbol{u}_l)$, and let $B_l = \sum_{l'=1}^{l} b_{l'}$. To construct the outer code, we choose the following encoding function f_O.

$$x_l = f_{O,l}(\boldsymbol{u}_{[l]}),\ \ l \in [L], \tag{5.20}$$

where $f_{O,l} : \{0,1\}^{B_l} \to [Q]$. The main idea of the proposed code construction is that the symbol x_l for the l-th slot depends only on the message chunks $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_l$. This property simplifies the decoding process (see Section 5.3.3).

Linear codes are preferred for practical schemes. Thus, we construct the functions $f_{O,l}(\cdot)$, $l \in [L]$ as follows. Let $c \in \mathbb{N}$, $Q = 2^c$. Let us fix a bijective mapping $\phi : \{0,1\}^c \to [Q]$. The major part of our construction is a binary linear code with a block upper-triangular generator matrix.

$$\boldsymbol{G} = \begin{pmatrix} \boldsymbol{G}_{1,1} & \boldsymbol{G}_{1,2} & \boldsymbol{G}_{1,3} & \dots & \boldsymbol{G}_{1,L} \\ \boldsymbol{0} & \boldsymbol{G}_{2,2} & \boldsymbol{G}_{2,3} & \dots & \boldsymbol{G}_{2,L} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{G}_{3,3} & \dots & \boldsymbol{G}_{3,L} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \dots & \boldsymbol{G}_{L,L} \end{pmatrix}, \tag{5.21}$$

where $\boldsymbol{G}_{l',l}$, $l', l \in [L]$, is a binary matrix of size $b_{l'} \times c$.

The codeword $\boldsymbol{x} = (x_1, \dots, x_L) \in [Q]^L$ is obtained as follows. We start with the binary vector $\boldsymbol{c} = \boldsymbol{u}\boldsymbol{G}$ and then obtain a codeword $\boldsymbol{x}$ by splitting $\boldsymbol{c}$ into chunks of length c and applying the mapping ϕ, i.e.,

$$x_l = \phi\left(\sum_{l'=1}^{l} \boldsymbol{u}_{l'}^T \boldsymbol{G}_{l',l}\right), \ l \in [L].$$

Decoding. Recall that the goal of the decoder is to recover all the messages $\boldsymbol{u}$, such that

$$d(\boldsymbol{\mathcal{Y}}, \boldsymbol{x}) \le t,$$

where $\boldsymbol{x} = f_O(\boldsymbol{u})$.

We note that $\boldsymbol{u}_{[l]}$ uniquely defines $\boldsymbol{x}_{[l]}$ for each $l \in [L]$. In what follows, we write $\boldsymbol{x}_{[l]} = f_O(\boldsymbol{u}_{[l]})$. This fact allows us to utilize a low-complexity decoding algorithm, which decodes the blocks $\boldsymbol{u}_l$ sequentially.

Let

$$\mathcal{V}_l = \left\{\boldsymbol{v}_l \in \{0,1\}^{B_l} : d(\boldsymbol{\mathcal{Y}}_{[l]}, f_O(\boldsymbol{v}_l)) \le t\right\}, \ l \in [L],$$

which represents the list of messages at each decoding step.

Let us consider the l-th decoding step. For each of the elements $\boldsymbol{v}_{l-1} \in \mathcal{V}_{l-1}$, we consider all possible continuations $\boldsymbol{u}_l \in \{0,1\}^{b_l}$. We include the path $\boldsymbol{v}_l = (\boldsymbol{v}_{l-1}, \boldsymbol{u}_l)$ into the set $\mathcal{V}_l$ if $d(\boldsymbol{\mathcal{Y}}_{[l]}, f_O(\boldsymbol{v}_l)) \le t$. See Algorithm 2 for the full description.

Remark 5.15. *As we see, Algorithm 2 is guaranteed to recover the transmitted message in case no more than t errors have occurred.*

Remark 5.16. *Note that the decoding complexity depends on $|\mathcal{V}_l|$, $l \in [L]$. Indeed, at the l-th decoding step we perform $|\mathcal{V}_{l-1}|2^{b_l}$ encodings. Taking into account that the encoding requires $B_l \times c$ additions in the binary field, we obtain the total decoding complexity of $c \times \left(\sum_{l=1}^{L} |\mathcal{V}_{l-1}|2^{b_l} B_l\right)$ binary additions.*

See Appendix E.2 for a detailed analysis of the average number of decoding paths.

Algorithm 2 t-tree linear code decoding algorithm

Input: $\boldsymbol{\mathcal{Y}}$
Output: $\mathcal{V}_L$ ▷ Decoded messages

$\mathcal{V}_0 \leftarrow \emptyset$
for $l \in [L]$ **do**
  $\mathcal{V}_l \leftarrow \emptyset$
  for $\boldsymbol{v}_{l-1} \in \mathcal{V}_{l-1}$ **do** ▷ For each element from the list
    for $\boldsymbol{u}_l \in \{0,1\}^{b_l}$ **do** ▷ For each next block
      $\boldsymbol{v}_l \leftarrow (\boldsymbol{v}_{l-1}, \boldsymbol{u}_l)$
      if $d(\boldsymbol{\mathcal{Y}}_{[l]}, f_O(\boldsymbol{v}_l)) \leq t$ **then**
        $\mathcal{V}_l \leftarrow \mathcal{V}_l \bigcup \boldsymbol{v}_l$
      end if
    end for
  end for
end for
return $\mathcal{V}_L$

5.3.4 Reed-Solomon code-based practical scheme

RS codes combined with Guruswami–Sudan decoding algorithm are known to solve the list-recovery problem. In this section, we describe an RS-code-based practical scheme from [94]. We start with the necessary definitions.

Recall that $\mathbb{F}_Q$ is the field with Q elements. Let $\mathbb{F}_Q[X]$ and $\mathbb{F}_Q[X,Y]$ denote the rings of univariate and bivariate polynomials over $\mathbb{F}_Q$. Let $\beta_1, \beta_2, \ldots, \beta_L \in \mathbb{F}_Q$, with $\beta_i \neq \beta_j$ for $i \neq j$. We define an $[n_O = L, k_O]$ RS code $\mathcal{C}$ as follows:

$$\mathcal{C}_{RS} \triangleq \{(f(\beta_1), f(\beta_2), \ldots, f(\beta_L)) : f(x) \in \mathbb{F}_Q[X], \deg f(x) < k_O\}.$$

Let us consider the bivariate polynomial

$$g(x,y) = \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} g_{i,j} x^i y^j \in \mathbb{F}_Q[X,Y].$$

We define the (w_1, w_2)-weighted degree of $g(x,y)$ as follows.

$$\deg_{w_1,w_2} g(x,y) = \max_{g_{i,j} \neq 0} \deg_{w_1,w_2}(x^i y^j),$$

where the (w_1, w_2)-weighted degree of the monomial $x^i y^j$ equals to $w_1 i + w_2 j$. Note that the $(1,1)$-weighted degree is simply the degree of a bivariate polynomial.

In what follows, we search for polynomials that pass through some point with multiplicity m. The bivariate polynomial $g(x,y)$ passes through the point $(\beta, \alpha) \in \mathbb{F}_Q^2$ if $g(\beta, \alpha) = 0$. Let us introduce the concept of root multiplicity. In the real field, the root multiplicity can be defined using partial derivatives, but for the finite field it is more challenging due to finite characteristic. Following [129],

we say that the polynomial $g(x, y)$ passes through the point (β, α) with multiplicity m if the shifted polynomial $g(x+\beta, y+\alpha)$ contains a monomial of degree[12] m and does not contain a monomial of smaller degree.

Naive approach. Let us apply the Guruswami–Sudan list recovery algorithm to our problem in a straightforward manner. We briefly explain the idea and refer the reader to [130] for details. Let us enumerate the elements of the field $\mathbb{F}_Q$ in some order, i.e., $\mathbb{F}_Q = \{\alpha_1, \alpha_2, \ldots, \alpha_Q\}$.

Encoding. Let $k_O = k/\log_2 Q$. Consider the user message W as a vector $\boldsymbol{f} = (f_0, \ldots, f_{k_O-1})$, where $f_i \in \mathbb{F}_Q$, and introduce the information polynomial $f(x) = \sum_{i=0}^{k_O-1} f_i x^i$. The user codeword is

$$\boldsymbol{x} = (f(\beta_1), f(\beta_2), \ldots, f(\beta_L)) \in \mathcal{C}_{RS}.$$

Decoding. Let us recall that the channel output is $\boldsymbol{\mathcal{Y}} = (\mathcal{Y}_1, \ldots, \mathcal{Y}_L)$, where $\mathcal{Y}_l \subseteq \mathbb{F}_Q$. Let us introduce the following set of points:

$$\mathcal{P} = \{(\beta_l, \alpha_i) : \alpha_i \in \mathcal{Y}_l, l \in [L], i \in [Q]\}.$$

We assign a multiplicity $m_{i,l} \in \mathbb{N}$ to each point $(\beta_l, \alpha_i) \in \mathcal{P}$. We note that multiplicities affect the decoding algorithm only. The larger $m_{i,l}$ are, the better the decoding performance is. At the same time, the decoding complexity grows with $m_{i,l}$. We provide all the details below.

To perform the decoding, we apply the Guruswami–Sudan algorithm. The input to this algorithm is the set of points $\mathcal{P}$ with the corresponding multiplicities $m_{i,l}$. The full description is given by Algorithm 3.

Algorithm 3 Guruswami–Sudan algorithm

Input: L, k_O, set of points $\mathcal{P}$ with multiplicities $m_{i,l}$

Output: List of polynomials $f(x)$

1: *Interpolation.* Find a bivariate polynomial $g(x, y)$ of minimal $(1, k_O - 1)$-weighted degree that passes through each point (β_l, α_i), $i \in [Q]$, $l \in [L]$, with multiplicity $m_{i,l}$.

2: *Factorization.* Find all the factors of $g(x, y)$ of the form $y - f(x)$, where $\deg f(x) < k_O$.

Analysis. It is convenient to represent the set $\mathcal{P}$ with corresponding multiplicities as a matrix $\boldsymbol{M} = \left(m'_{i,l}\right)$ of size $Q \times L$ as follows: $m'_{i,l} = m_{i,l}$ if $(\alpha_i, \beta_l) \in \mathcal{P}$ and $m'_{i,l} = 0$ otherwise.

Let us define the complexity of matrix $\boldsymbol{M}$ as

$$C(\mathbf{M}) = \frac{1}{2} \sum_{i=1}^{Q} \sum_{l=1}^{L} m'_{i,l}(m'_{i,l} + 1).$$

Algorithm 3 is known (see [129, 131]) to include $f(x)$ in the output list if

$$(\mathbf{M}, \mathbf{X}) \geq \sqrt{2(k_O - 1)C(\mathbf{M})}, \tag{5.22}$$

[12] Here, we refer to the $(1, 1)$-weighted degree.

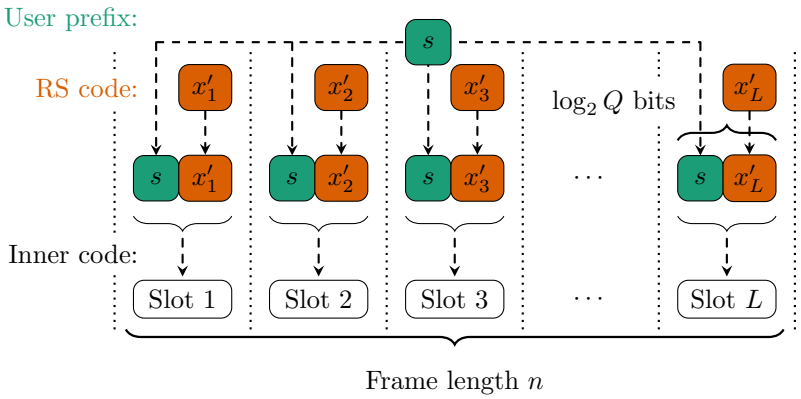

Figure 5.12: Frame structure with slots consisting of RS code symbols x'_l, where $l \in [L]$, and the prefix s.

where $(\cdot, \cdot)$ denotes the dot product of matrices, and the matrix $\mathbf{X} = (x_{i,l})$ corresponds to the codeword $\mathbf{x} = (f(\beta_1), \ldots, f(\beta_L))$, i.e., $x_{i,l} = 1$ if $\alpha_i = f(\beta_l)$. This matrix has only one unit in each column.

The following upper bound holds for the list size:

$$L(\mathbf{M}) \leq \sqrt{\frac{2C(\mathbf{M})}{k_O - 1}}.$$

Let us investigate the recovery condition (5.22) in more detail for the case when all the points from $\mathcal{P}$ have the same multiplicity equal to m. Assume also that we have lists of size K_a in each position. Then we have

$$m(L - d(\boldsymbol{\mathcal{Y}}, \boldsymbol{x})) = (\mathbf{M}, \mathbf{X}) \geq \sqrt{(k_O - 1)m(m+1)K_a L},$$

and thus we can recover the codeword if the number of errors t satisfies the inequality

$$t = d(\boldsymbol{\mathcal{Y}}, \boldsymbol{x}) \leq L\left(1 - \sqrt{\frac{(k_O - 1)}{L} K_a \frac{(m+1)}{m}}\right).$$

As we can see, to have error-correcting capabilities, we must require the outer code rate $R_O \triangleq k_O/L \leq 1/K_a$, which is infeasible even for a moderate number of users. Thus, our next goal is to reduce the order of collisions.

Modified scheme. Let $Q = q_p \times q$ and $b_p = \log_2(q_p)$. We propose to use b_p bit prefixes s_i, $i \in [K_a]$ to group K_a users in q_p groups, ensuring an acceptable level of collisions within each group.

Encoding. Consider the user aiming to transmit a k bit message W. First b_C CRC bits are calculated and added to the message to suppress the FAR below the required threshold. Each user

encodes $k+b_C$ bits by using an $[L, k_O]$ RS code $\mathcal{C}'_{RS}$ over the alphabet $\mathbb{F}_q$ and obtains the codeword $\boldsymbol{x}' \in \mathcal{C}'_{RS}$. The user then generates the prefix $s \sim \mathrm{Unif}([q_p])$. Consider the bijective mapping $\phi' : [q_p] \times [q] \to [Q]$, which is common to all users. The resulting codeword $\boldsymbol{x} \in [Q]^L$ is calculated as follows:

$$x_l = \phi'(s, x'_l), \ \ l \in [L].$$

The encoding process is illustrated in Fig. 5.12.

Decoding. Prefixes split the set $[Q]$ into non-intersecting ranges $\mathcal{I}_i$, where $|\mathcal{I}_i| = q$, $i \in [q_p]$. We handle the ranges $\mathcal{I}_i$ separately. Note that by applying ϕ'^{-1} and removing the prefix, we reduce the problem to the one described in Section 5.3.4, but with alphabet $\mathbb{F}_q$ and a smaller collision order.

5.3.5 Further modifications

The paper [95] continues this line of work and utilizes SPARCs combined with the AMP algorithm. A further modification of the CCS approach is presented in [96], where the inner AMP decoder and the outer tree decoder are allowed to exchange soft information within a joint MPA. In [132], the authors reduce complexity by utilizing the same sensing matrix as in [93]. The main difference lies in the use of a joint MPA (as in [96]) and the assumption that inner codewords are coupled via an outer code. Finally, the paper [133] proposes a coded demixing approach to enable the joint recovery of very high-dimensional signals that are sparse with respect to separate bases.

We also note the papers [134, 135], which further investigate coding for the A-channel. In particular, the paper [134] proposes an interesting list-pruning algorithm that may be extremely helpful for tree-code decoding.

The state-of-the-art CS-based solution proposed in [136] suggests eliminating the outer code entirely. The main idea is to use data from previous slots to customize the sensing matrix for subsequent slots. This scheme demonstrates the best energy efficiency among all CS-based approaches.

5.4 Random spreading

Previously, we discussed IRSA and sparse IDMA schemes. Simply put, such schemes rely on a TDMA strategy to sparsify collisions. From the MAC literature, we know of one more technique—direct-sequence spread spectrum (DSSS), which serves as the basis for the CDMA method. This section is devoted to applying this strategy to the URA scenario. In what follows, we describe in detail the scheme from [97] and briefly discuss the improvements introduced in [98, 99].

Let us start with an informal description of the scheme. The payload corresponding to each active user is split into two parts. The first component acts as a preamble, selecting a spreading or signature sequence from a codebook of sequences with good correlation properties. The remaining bits are encoded using the user's code. This codeword is then spread using the signature sequence chosen by the first part of the message.

The decoding procedure is iterative, with each iteration consisting of four main steps:

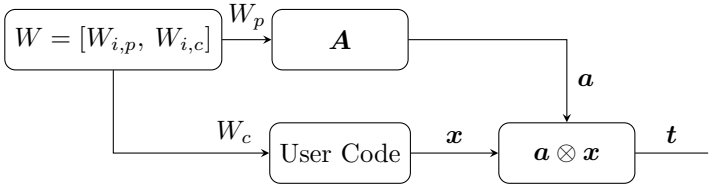

Figure 5.13: The message is split into two parts, W_p and W_c. The component W_p is responsible for selecting a spreading sequence, while W_c represents the message encoded by a user code. Spreading sequences are chosen from the matrix $\boldsymbol{A}$, and the encoded vector $\boldsymbol{t}$ is given by (5.23).

(a) an energy detector that utilizes the correlation properties of the signature sequences to identify the list of spreading sequences used,

(b) a MMSE estimator that produces LLR estimates based on the spreading sequences,

(c) a single-user decoding operation, and

(d) a SIC step, resulting in a residual channel output.

We note that the basic scheme relies on single-user decoding, which drastically reduces its complexity.

The codebook of spreading sequences plays a crucial role in determining the scheme's error performance. These sequences should exhibit good correlation properties or, equivalently, form a good CS code (see Appendix B for details on how to construct such codes). When the size of this codebook is large, active users are less likely to experience collisions in the space of signature sequences, a scenario that favors the use of a single-user decoder. However, the mutual coherence of such a codebook would be very high, adversely affecting the performance of the polar decoder. Conversely, smaller codebooks possess better correlation properties but result in a higher probability of collisions between the signature sequences chosen by active users. This, in turn, renders single-user decoding ineffective for users whose signature sequences collide.

Now, we proceed with a detailed description, starting with the encoding process.

5.4.1 Encoder

Consider the i-th user. Similar to previous schemes (see Section 5.1.3), we split the message into two parts:

$$W_i = [W_{i,p},\, W_{i,c}],$$

where $W_{i,p} \in [M_p]$, $W_{i,c} \in [M_c]$, and $M = M_p M_c$. Recall that the message parts $W_{i,p}$ and $W_{i,c}$ consist of k_p and k_c bits, respectively, i.e., $M_p = 2^{k_p}$ and $M_c = 2^{k_c}$.

The message part $W_{i,p}$ encodes the spreading sequence $f_p(W_{i,p})$, where $f_p : [M_p] \to \mathbb{R}^{n_p}$ is the encoder of some good CS code. Let us introduce the codebook (or sensing matrix) of CS code

$$\boldsymbol{A} = \left[\boldsymbol{a}_1, \ldots, \boldsymbol{a}_{M_p}\right],$$

where $\boldsymbol{a}_W = f_p(W)$ for $W \in [M_p]$. The encoding process consists of selecting the column indexed by $W_{i,p}$ from the matrix $\boldsymbol{A}$. In other words, the message part $W_{i,p}$ determines the choice of the spreading sequence. Throughout this work, we denote the spreading sequence chosen by the i-th user as

$$\boldsymbol{a}^{(i)} \triangleq \boldsymbol{a}_{W_{i,p}} = f_p\left(W_{i,p}\right).$$

The message part $W_{i,c}$ is encoded into the user's codeword $\boldsymbol{x}^{(i)} = f_c(W_{i,c})$, where $f_c : [M_c] \to \mathbb{R}^{n_c}$. In the numerical analysis presented in Section 5.6.4, we consider a BPSK-modulated binary linear $[n_c, k_c]$-code $\mathcal{C}$, i.e.,

$$\boldsymbol{x}^{(i)} = \tau\left(f_c\left(\boldsymbol{u}_c\right)\right),$$

where $\boldsymbol{u}_c \in \{0,1\}^{k_c}$ is the information vector corresponding to W_c. Throughout this discussion, $f_c\left(\boldsymbol{u}_c\right)$ and $f_c\left(W_c\right)$ are used interchangeably.

Finally, the modulated codeword $\boldsymbol{x}^{(i)}$ is spread using the chosen spreading sequence $\boldsymbol{a}^{(i)}$ to produce the transmitted signal

$$\boldsymbol{t}^{(i)} = \boldsymbol{x}^{(i)} \otimes \boldsymbol{a}^{(i)}, \tag{5.23}$$

where $\otimes$ denotes the Kronecker product operation. A graphical representation of the encoding process is shown in Fig. 5.13.

Remark 5.17 (Signal representation in matrix form)**.** *Let us transition to the matrix representation. We have*

$$\boldsymbol{T}^{(i)} = \boldsymbol{a}^{(i)} \left(\boldsymbol{x}^{(i)}\right)^T, \tag{5.24}$$

where $\boldsymbol{t}^{(i)}$ can be obtained by reshaping this matrix, i.e., by reading it column by column.

Taking into account that K_a users are transmitting simultaneously, we observe the following noisy signal mixture at the channel output

$$\mathbf{Y} = \sum_{i=1}^{K_a} \boldsymbol{T}^{(i)} + \mathbf{Z}, \tag{5.25}$$

where $\mathbf{Z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_{n_p \times n_c})$.

Remark 5.18. *Let $\mathcal{P} = \{W_{1,p}, \ldots, W_{K_a,p}\}$. Consider the general case where some preambles may coincide, i.e., $|\mathcal{P}| = r \leq K_a$ and $\mathcal{P} = \{j_1, j_2, \ldots, j_r\}$, where $j_1 < j_2 < \ldots < j_r$ are unique elements. Then, we have*

$$\mathbf{Y} = \boldsymbol{A}_{\mathcal{P}} \tilde{\boldsymbol{X}} + \mathbf{Z}, \tag{5.26}$$

where $\boldsymbol{A}_{\mathcal{P}}$ is formed from $\boldsymbol{A}$ by selecting columns indexed by $\mathcal{P}$, and

$$\tilde{\boldsymbol{X}} = \begin{bmatrix} \tilde{\boldsymbol{x}}_1^T \\ \tilde{\boldsymbol{x}}_2^T \\ \vdots \\ \tilde{\boldsymbol{x}}_r^T \end{bmatrix},$$

and $\tilde{\boldsymbol{x}}_l$ is the sum of the codewords of users that chose the preamble j_l, i.e.,

$$\tilde{\boldsymbol{x}}_l = \sum_{i: W_{i,p} = j_l} \boldsymbol{x}^{(i)}, \quad l \in [r].$$

5.4.2 Decoder

The iterative decoder consists of several components. First, an energy detector decodes the preamble parts of the transmitted messages, which are then mapped to columns of $\boldsymbol{A}$ to identify the set of spreading sequences selected by active users. Since signature sequences are chosen solely based on preambles, two or more users select the same sequence whenever they share a common preamble.

Based on the energy corresponding to a detected sequence, the energy detector also provides an estimate of the number of active users sharing that sequence. The MMSE estimator is then employed to produce soft estimates of the signals corresponding to the detected sequences. These soft estimates are passed to the users' decoders, and the recovered codewords are subtracted from the received signal in the spirit of SIC. The residual signal is then fed back into the energy detector for the next decoding iteration.

This iterative decoding process continues until all transmitted messages are recovered or the number of decoded messages no longer improves between consecutive iterations. Let us examine this process in more detail.

Detection of spreading sequences

At this stage, we need to determine the set of spreading sequences that were transmitted. Recall that we are utilizing a good CS codebook; thus, the natural approach is to examine the correlations between the channel output and all possible spreading sequences (i.e., the columns of the sensing matrix $\boldsymbol{A}$).

One challenge in implementing this algorithm arises from (5.23), as $\boldsymbol{x}^{(i)}$ is unknown at this stage of the decoding process. The optimal strategy, in terms of performance, would be to consider all possible codewords $\boldsymbol{x}^{(i)}$, but this approach is computationally infeasible.

To address this, the authors of [97] proposed a suboptimal strategy. Assume we are considering a spreading sequence $\boldsymbol{a}$[13]. We split the channel output into $n' = n_c/g$ groups of length $n_p g$ as follows:

$$\boldsymbol{y}^T = \left[\boldsymbol{y}_1^T, \boldsymbol{y}_2^T, \ldots, \boldsymbol{y}_{n'}^T\right]^T.$$

Now, to calculate the *score* of $\boldsymbol{a}$ we proceed as follows:

$$S(\boldsymbol{a}) = \sum_{i=1}^{n'} S_i(\boldsymbol{a}), \quad \text{where} \quad S_i(\boldsymbol{a}) = \max_{\boldsymbol{b}\in\{\pm 1\}^g} (\boldsymbol{b}\otimes\boldsymbol{a})^T \boldsymbol{y}_i.$$

The group size is carefully chosen to balance error performance and computational complexity. The columns of $\boldsymbol{A}$ are then sorted in descending order of their scores, and the detector outputs the top $K_a + \Delta$ sequences, where Δ is a fixed small non-negative integer. We denote by $\mathcal{D}$ the set of indices of these sequences. Notably, $\mathcal{D}$ serves as an estimate of the set of message parts $\mathcal{P}$.

[13] $\boldsymbol{a}$ is an arbitrary column of the matrix $\boldsymbol{A}$; we omit the subscript as nothing depends on it.

Demodulation and decoding

In this section, we describe the steps involved in decoding the messages corresponding to the detected sequences.

Recall eq. (5.26) and note that we know $\mathcal{D}$, the estimate of the set of spreading sequences $\mathcal{P}$. We have

$$\mathbf{Y} = \boldsymbol{A}_{\mathcal{D}}\tilde{\boldsymbol{X}} + \mathbf{Z}', \tag{5.27}$$

where $\mathbf{Z}'$ consists of the additive Gaussian noise $\mathbf{Z}$ and the interference from users with undetected spreading sequences. In what follows, we work under the assumption that $\mathcal{P} \subseteq \mathcal{D}$, implying that $\mathbf{Z}' = \mathbf{Z}$.

First, we apply the linear MMSE (LMMSE) estimator (see details in Appendix A) to obtain

$$\hat{\boldsymbol{X}} = \boldsymbol{A}_{\mathcal{D}}^T \boldsymbol{R}^{-1}\mathbf{Y}, \quad \boldsymbol{R} = (\boldsymbol{A}_{\mathcal{D}}\boldsymbol{A}_{\mathcal{D}}^T + \boldsymbol{I}_{n_p}).$$

The mean squared error (MSE) of this estimator is well approximated by

$$\boldsymbol{\Sigma} = \boldsymbol{I} - \mathbf{A}_{\mathcal{D}}^T \mathbf{R}^{-1}\mathbf{A}_{\mathcal{D}}, \tag{5.28}$$

where

$$\boldsymbol{\Sigma} \approx \operatorname{diag}\left(\sigma_{\text{mse}}^2(1), \sigma_{\text{mse}}^2(2), \ldots, \sigma_{\text{mse}}^2(\mathcal{D})\right).$$

In this context, $\sigma_{\text{mse}}^2(j)$ is the MSE in the estimate of the signal corresponding to sequence a_j for $j \in \mathcal{D}$.

Assume the case when there are no repeating sequences, i.e. $|\mathcal{P}| = K_a$. The next step is to calculate input LLR sequences for the users' decoders. Assuming the MSE is Gaussian, we proceed as follows:

$$\boldsymbol{l}_i = \frac{2}{\sigma_i^2}\hat{\boldsymbol{x}}_i, \quad i \in [K_a],$$

where $\hat{\boldsymbol{x}}_i$ is the i-th row of the matrix $\hat{\boldsymbol{X}}$.

Then, the calculated LLRs are passed to users' codes. In [97], the authors utilize single-user polar codes. The decoders can converge to a correct codeword, output a failure, or even produce a wrong decision. For the latter case, we assume the presence of a technique to discard such decisions (e.g., CRC). We denote by $\mathcal{D}^* \subseteq \mathcal{D}$ the collection of indices such that users' codes are decoded successfully.

The SIC removes the contributions from all the successfully decoded codewords $\hat{\boldsymbol{X}}_{\mathcal{D}^*}$ from the received signal to compute the residual

$$\boldsymbol{Y} - \boldsymbol{A}_{\mathcal{D}^*}\hat{\boldsymbol{X}}_{\mathcal{D}^*}.$$

This residual is passed for the next decoding iteration. This process continues until all the transmitted messages are recovered successfully or there is no improvement between two consecutive rounds of the iterative process.

5.4.3 Discussion and further improvements

Let us describe possible improvements suggested in the literature.

Joint decoding. The scheme in [97] relies on single-user codes. Thus, if several users choose the same spreading sequence, the corresponding messages will be missed with a relatively high probability. A straightforward solution is to enlarge the set of spreading sequences, but in this case, we lose the good correlation properties. A better solution is to adopt joint decoding (in the spirit of Section 5.1.4 with LDPC or polar codes) to avoid the need for large-sized codebooks for the spreading sequences. The idea is as follows: for those active users whose signature sequences collide, we leverage a joint decoder that does not treat the heavy interference from colliding users as noise. We emphasize that, although the envisioned scheme utilizes joint decoding, it is computationally less burdensome than [101] because joint decoding is performed only for colliding users.

Power diversity. It is well-known that power diversity improves the SIC process. The idea proposed in [98] is to divide the columns of $\boldsymbol{A}$ into m groups and assign different power levels to each. There are l_k columns with power level P_k in the k-th group, with $k \in [m]$. The power levels for each group must be chosen in such a way that the average power constraint is satisfied. The proposed approach divides the active users into different groups. The encoding procedure in this scheme remains unchanged, while the decoding differs in the preamble detection part: the energy detector is replaced with a covariance-based detector. Namely, we form

$$\Sigma = \boldsymbol{A}_N^T \boldsymbol{Y} \boldsymbol{Y}^T \boldsymbol{A}_N,$$

where $\boldsymbol{A}_N$ is obtained by scaling the columns of the codebook to 1 (after removing signatures of correctly decoded users) to prevent a higher detection probability for the sequences with greater power levels. The covariance-based detector outputs $K_a + \Delta$ signatures corresponding to the largest diagonal elements of Σ.

Simulations show that the proposed approach outperforms the existing methods, especially when the number of active users is large.

SoIC. The authors of [99] noted that to apply the SIC procedure, we should decode the user code. At the same time, the information from non-converged decoders is not utilized, even though it may help the overall iterative decoding process. The paper [99] proposes replacing polar codes with LDPC codes (since LDPC codes are much better in terms of soft output) and utilizing SoIC [43, 137, 138]. We also mention the paper [139], which describes DE analysis of iterative SoIC multi-user detection/decoding.

Assume the user code provides us with the extrinsic information vector $\boldsymbol{\gamma} = [\gamma_1, \gamma_2, \ldots, \gamma_{n_c}]$. We calculate the expected values of the modulated symbols and remove them from the channel output, i.e.

$$\overline{\boldsymbol{x}} = [\mathbb{E}[\mathsf{x}_1|\gamma_1], \ldots, \mathbb{E}[\mathsf{x}_{n_c}|\gamma_{n_c}]] = [\tanh(\gamma_1/2), \ldots, \tanh(\gamma_{n_c}/2)],$$

and then subtract them:

$$\mathbf{Y} - \boldsymbol{A}_{\mathcal{D}}\overline{\boldsymbol{X}}_{\mathcal{D}}.$$

This scheme is the state-of-the-art scheme for the G-URA. Notably, the proposed scheme outperforms the RCB derived in Theorem 4.1 when the active population is less than 75 for parameters taken from [90]. Furthermore, the proposed scheme outperforms the schemes with joint decoding [97] and power diversity [98] when the active population exceeds 200 users. However, in a private discussion, the authors of this approach informed us that they were unable to validate the resulting energy efficiency curve (which will be discussed in Section 5.6).

Hadamard-based spreading with convolutional codes. Finally, let us highlight an elegant scheme that considers spreading differently. All previously discussed spreading techniques are based on a Kronecker product (5.23). The authors of [140] propose a scheme based on the *Hadamard* product, where the encoded codeword spread by the signature will not increase in length. The authors also use a slotted system, where users choose the transmission slot at random. To detect different signatures, dedicated pilot bits are inserted. The choice of the signature is also based on splitting the message into two parts, with a convolutional code considered as the user code. The final decision makes it straightforward to apply the Viterbi algorithm to a joint decoding problem. Despite its simplicity, the proposed solution offers quite good performance.

5.5 Comments on same linear codes

As we know (Chapter 4), there exist codes that can operate in the G-URA. At the same time, the same BPSK-modulated linear codes (e.g., LDPC or polar codes) perform poorly in the G-URA. However, these codes could still be strong candidates for practical schemes, as binary linear codes are well-studied and have low-complexity soft decoding algorithms.

Thus, as explained throughout this chapter, all low-complexity schemes rely on the following approach:

a) generate different codes from a single linear code (e.g., through permutations or shifts);

b) append the transmitted codewords with a preamble that helps the receiver identify the correct permutation or shift.

This strategy is indeed effective, but the use of a preamble introduces additional overhead. This naturally leads to the question: *Are there linear codes that can operate in the G-URA?*

To better understand the problem, let us examine Fig. 5.14, which is also presented in [141, Figure 1]. This figure presents the results for the two-user G-URA, namely,

$$\mathbf{y} = \boldsymbol{x}^{(1)} + \boldsymbol{x}^{(2)} + \mathbf{z},$$

where $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)} \in \pm\sqrt{P}^n$ and $\mathbf{z} \sim \mathcal{N}(0, I_n)$.

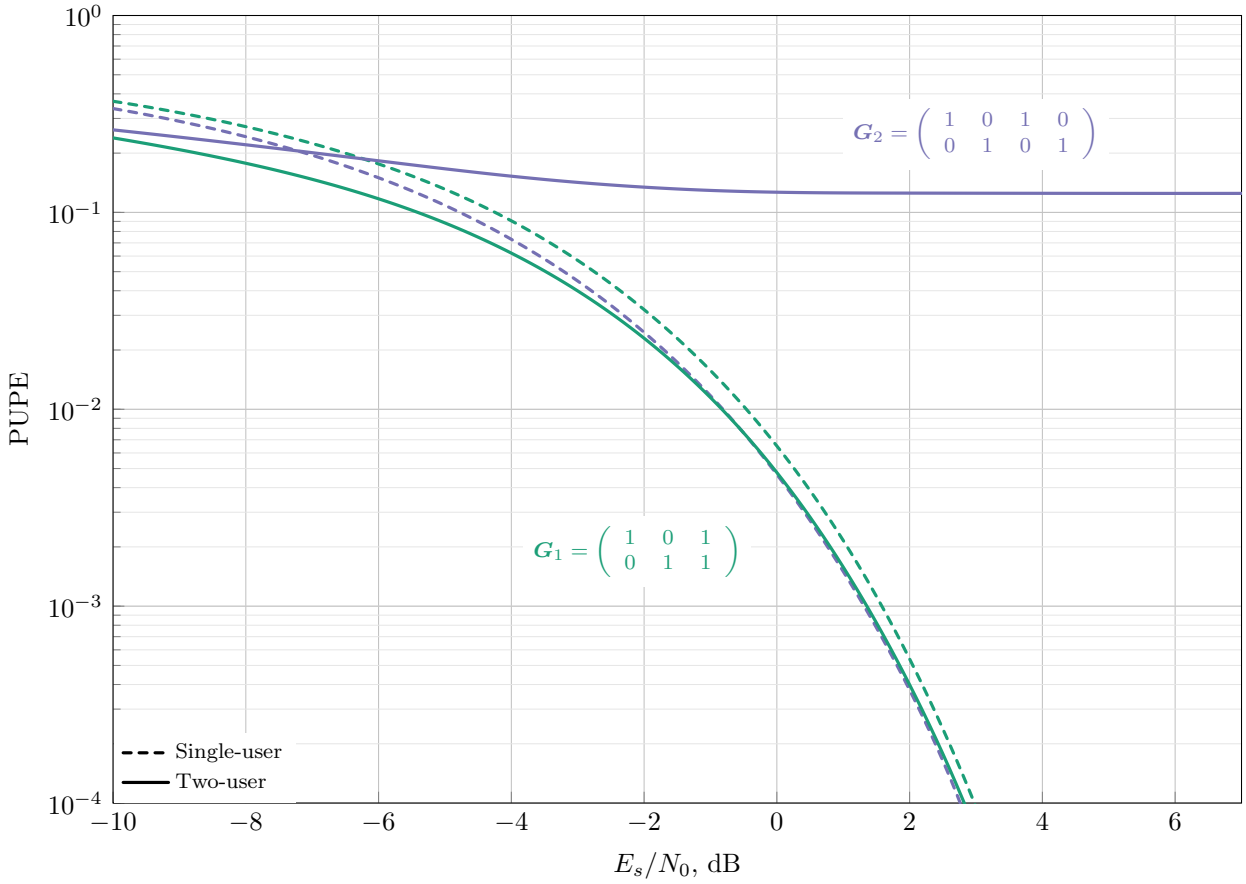

Figure 5.14: PUPE vs. SNR for two binary linear block codes over a BAC with AWGN. A toy example demonstrates two different linear codes with generator matrices $\boldsymbol{G}_1$ and $\boldsymbol{G}_2$, followed by BPSK modulation. These codes exhibit similar performance in a single-user regime but significantly different performance in a multi-user regime. This result is presented in [141, Figure 1].

Two linear codes are compared: $\mathcal{C}_1$, with generator matrix $\boldsymbol{G}_1$, which is a $[3, 2]$ single-parity-check (SPC) code, and $\mathcal{C}_2$, a $[4, 2]$ code with generator matrix $\boldsymbol{G}_2$, which is the direct sum of two repetition codes of length 2. The block error probability of these codes in the absence of interference (i.e., single-user transmission) is depicted as a reference with dashed lines.

For the multi-user case, the decoder outputs the unordered set of codewords $\hat{\boldsymbol{x}}^{(1)}, \hat{\boldsymbol{x}}^{(2)}$ that maximizes $p(\boldsymbol{y}|\boldsymbol{x}^{(1)} + \boldsymbol{x}^{(2)})$. We observe that $\mathcal{C}_2$ performs better in the single-user setting but exhibits a high error floor in the multi-user scenario. Hence, the root cause of the error floor is multi-user interference rather than AWGN.

Thus, let us focus on a simplified scenario of URA in a two-user noiseless BAC. Despite its simplicity, this setting proves to be particularly rich from a coding theory perspective.

Here, we briefly discuss the results presented in [141], starting with the decoding algorithm. Since we are not considering noise, there is no need to work with the alphabet $\{\pm\sqrt{P}\}$; instead, we use

the binary alphabet $\{0,1\}$. We now note that ML decoding over the BAC reduces to an erasure decoding problem. Assume we receive $\boldsymbol{y} \in \{0,1,2\}^n$. Considering the j-th element of $\boldsymbol{y}$, we have three possible cases:

1. $y_j = 0 \implies x_j^{(1)} = 0$ and $x_j^{(2)} = 0$;
2. $y_j = 2 \implies x_j^{(1)} = 1$ and $x_j^{(2)} = 1$;
3. $y_j = 1 \implies \left(x_j^{(1)}, x_j^{(2)}\right) \in \{(0,1),\ (1,0)\}$;

To handle the decoding process, we introduce an auxiliary vector $\tilde{\boldsymbol{y}}$, defined as follows:

1. $\tilde{y}_j = 0$ if $y_j = 0$,
2. $\tilde{y}_j = 1$ if $y_j = 2$,
3. $\tilde{y}_j = *$ if $y_j = 1$.

Notably, uncertainty arises only in case (3). Consequently, the decoder begins by constructing the list $\mathcal{L}$ of all codewords $\boldsymbol{x}$ that are compatible with the channel output. Specifically, defining the erasure positions as $\mathcal{E} = \{j : y_j = 1\}$ and the non-erasure positions as $\mathcal{E}^c = [n] \backslash \mathcal{E}$, we obtain

$$\mathcal{L}' = \{\boldsymbol{x} : \boldsymbol{x}_{\mathcal{E}^c} = \tilde{\boldsymbol{y}}_{\mathcal{E}^c}\}.$$

The final step involves searching for all unordered codeword pairs $\{\boldsymbol{x}_1, \boldsymbol{x}_2\} \in \mathcal{L}' \times \mathcal{L}'$ whose integer sum matches $\boldsymbol{y}$. If a unique solution exists, the decoder outputs this pair; otherwise, if multiple solutions exist, the decoder selects one at random.

For the case of linear codes, the first step is straightforward. Let $\boldsymbol{H}$ be the parity-check matrix of the users' code. Then, $\mathcal{L}'$ is the set of solutions to the following system of linear equations:

$$\boldsymbol{H}_{\mathcal{E}} \boldsymbol{x}_{\mathcal{E}}^T = \boldsymbol{H}_{\mathcal{E}^c} \boldsymbol{x}_{\mathcal{E}^c}^T. \tag{5.29}$$

We note that this system always has at least two solutions, implying that $\operatorname{rank}(\boldsymbol{H}_{\mathcal{E}}) < |\mathcal{E}|$. One possible approach to solving this system is the *pivoting* strategy: setting an arbitrary element (pivot) of $\boldsymbol{x}_{\mathcal{E}}$ to a value in $\{0,1\}$ and then solving the resulting system. If $\operatorname{rank}(\boldsymbol{H}_{\mathcal{E}}) = |\mathcal{E}| - s$, then there are 2^{s-1} solutions. The authors of [141] make the following observations:

1. Any solution of (5.29) participates in exactly one valid pair.
2. If $\boldsymbol{x}^{(1)} \notin \mathcal{L}$, then $\boldsymbol{x}^{(2)}$ is also not in $\mathcal{L}$.
3. The probability of decoding error is given by

$$P_e = 1 - \sum_{s=1}^{|\mathcal{E}|} 2^{-(s-1)} \Pr[\operatorname{rank}(\boldsymbol{H}_{\mathcal{E}}) = |\mathcal{E}| - s].$$

4. The number of valid codeword pairs computed by the decoder depends on the transmitted pair only through the support set of their difference.

Let us formulate an auxiliary lemma from [141]:

Lemma 5.1 (All-zero codeword)**.** *Consider two users transmitting over the BAC using an $[n, k]$ binary linear block code $\mathcal{C}$, and denote the transmitted codewords as $\boldsymbol{x}^{(1)}$ and $\boldsymbol{x}^{(2)}$. Then,*

$$\Pr\left[\mathbf{x}^{(1)} \notin \mathcal{L} | \mathbf{x}^{(1)} = \boldsymbol{c}\right] = \Pr\left[\mathbf{x}^{(1)} \notin \mathcal{L} | \mathbf{x}^{(1)} = \mathbf{0}\right], \quad \forall \boldsymbol{c} \in \mathcal{C}.$$

Note that the statement remains valid if we exchange $\boldsymbol{x}^{(1)}$ and $\boldsymbol{x}^{(2)}$. Lemma 5.1 implies that

$$P_e = \Pr\left[\mathbf{x}^{(1)} \notin \mathcal{L} | \mathbf{x}^{(1)} = \mathbf{0}\right],$$

i.e., we can evaluate the PUPE of a binary linear block code under ML decoding by the special case where one of the two users transmits the all-zero codeword.

Now, consider the case where $\mathbf{x}^{(1)} = \mathbf{0}$. In this case, we have $\mathcal{E} = \mathrm{supp}(\boldsymbol{x}^{(2)})$, and (5.29) has exactly two solutions if and only if $\boldsymbol{x}^{(2)}$ is *minimal*.

Definition 5.6. *The codeword $\boldsymbol{c} \in \mathcal{C}$ is called* minimal *if it does not cover any other codeword, i.e., there does not exist $\boldsymbol{c}' \in \mathcal{C}$ such that* $\mathrm{supp}(\boldsymbol{c}') \subseteq \mathrm{supp}(\boldsymbol{c})$.

Let $\mathcal{M}(\mathcal{C})$ denote the set of minimal codewords in $\mathcal{C}$. We can formulate the following statement:

$$\Pr[\mathrm{rank}(\boldsymbol{H}_{\mathcal{E}}) < |\mathcal{E}| - 1] = \Pr[\boldsymbol{x}^{(2)} \notin \mathcal{M}(\mathcal{C})].$$

As a result, minimal codes achieve zero P_e over a two-user BAC with the same-codebook constraint.

Returning to Fig. 5.14, we observe that the $[3, 2]$ single-parity-check (SPC) code is a minimal code, meaning it consists only of minimal codewords. At the same time, the $[4, 2]$ code includes a non-minimal all-one codeword, which is the root cause of the error floor.

Theorem 5.4 (Converse for two-user BAC)**.** *Consider two users transmitting over the BAC with a $[n, k]$ binary linear block code. Then, we have*

$$P_e \geq \frac{1}{2}\left(1 - \frac{n}{2k-2}\right).$$

Thus, the maximal achievable symmetric rate is $R \leq 1/2$.

Proof. Let $\boldsymbol{x}^{(1)} = \mathbf{0}$. We have

$$\begin{aligned} P_e &\geq \frac{1}{2}\Pr\left[\boldsymbol{x}^{(2)} \notin \mathcal{M}(\mathcal{C})\right] \\ &\geq \frac{1}{2}\Pr\left[\mathrm{wt}_H\left\{\boldsymbol{x}^{(2)}\right\} > n - k + 1\right] \end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{2}\left(1 - \Pr\left[n - \mathrm{wt}_H\left\{\boldsymbol{x}^{(2)}\right\} \geq k-1\right]\right) \\
&\geq \frac{1}{2}\left(1 - \frac{\mathbb{E}\left[n - \mathrm{wt}_H\left\{\boldsymbol{x}^{(2)}\right\}\right]}{k-1}\right) \quad \text{(Markov's inequality)} \\
&= \frac{1}{2}\left(1 - \frac{n}{2k-2}\right).
\end{aligned}
$$

□

Remark 5.19. *The following results show that transmission over the two-user BAC with linear block codes is fundamentally limited to a symmetric rate no larger than 1/2, in contrast to the maximum symmetric rate of 3/4 achievable by nonlinear block codes over the two-user BAC.*

Remark 5.20. *Note that the result stated in Theorem 5.4 extends to cosets of the linear code.*

The papers [141, 142] also address the question of whether it is possible to achieve $R = 1/2$ via iterative decoding. It appears that for LDPC codes, the choice of the pivot[14] is of critical importance. If the pivot is selected randomly, then any irregular LDPC code ensemble has a non-vanishing error probability. If the pivot is selected optimally, then it is possible to attain vanishing PUPE. The paper [142] continues the investigation. In particular, the authors analyze the expected fraction of good pivots, compute optimized degree distributions, and provide a simple multi-edge type LDPC code construction that can provably achieve the two-user unsourced BAC limit for linear codes.

In this section, we address the question of constructing the same codebook codes for the URA. All the schemes described in this chapter utilize preambles chosen from a small CS codebook. The preambles can be used in two different ways:

- Same codes that are constructed as a concatenation of preambles and different linear codes (the interleaver or shift is chosen based on the preamble).
- Same codes that are simply a concatenation of preambles.

The main disadvantages are as follows: (a) preambles introduce additional overhead; (b) we cannot significantly increase the number of preambles (2^{15} is an upper bound due to complexity considerations).

Thus, an important question arises: what are alternative approaches to constructing low-complexity same codes? This section considers a relatively simple scenario of a two-user BAC, but even in this case, it becomes evident that users cannot effectively utilize well-developed linear codes and their cosets. Non-linear codes are necessary to achieve the capacity ($C = 3/4$) of the two-user unsourced BAC.

The design of such codes – where the main challenge is developing low-complexity decoders – remains an open and challenging problem. We can reformulate the problem as follows: we need a large CS codebook with a polynomial-time recovery algorithm. Some progress in this direction has been made in [143].

[14] A good pivot is a variable node associated with a received "erasure" symbol for which revealing its value allows recovering both transmitted messages, up to a vanishingly small fraction of residual erasures, by simple BCH decoding.

5.6 Numerical comparisons

In this section, we present the results of numerical experiments that estimate the energy efficiency of the solutions described above. All previously introduced schemes assume a frame-synchronous transmission system. Energy efficiency is defined as the minimum energy spent per transmitted information bit, E_b/N_0, under the maximum tolerable PUPE, as defined in (3.3).

Starting from [5, 90], a commonly used scenario for AWGN URA assumes a frame length of $n = 3 \times 10^4$ real channel uses, $k = 100$ information bits, and a maximum tolerable PUPE of $\varepsilon = 0.05$.

A numerical comparison is presented in Fig. 5.15 alongside state-of-the-art results. This figure illustrates the energy efficiency E_b/N_0 of different schemes proposed in the literature as a function of the number of active users, K_a. The latter term represents the number of messages sent within a single frame. This figure omits the first low-complexity scheme proposed in [90] because it demonstrated relatively low energy efficiency.

We focus on four previously mentioned groups of algorithms. The first group is based on the T-fold IRSA (see Section 5.6.1). The second group follows a sparse IDMA approach (Section 5.6.2). The third group is based on a CCS approach (Section 5.6.3). Finally, we consider a family of random spreading algorithms presented in Section 5.6.4. Different families of algorithms are represented by different colors. As a reference, we have also included Polyanskiy's achievability bound (presented in Theorem 4.1) and the IRSA achievability bound, formulated in Theorem 5.2. Theoretical bounds are marked by dashed lines.

5.6.1 T-fold irregular repetition slotted ALOHA-based schemes

T-fold irregular repetition slotted ALOHA achievability bound

Let us start with the achievability bound for IRSA and numerically evaluate Theorem 5.2. The main objective is to select the polynomial Λ (see (5.1)) and determine the number of slots, or equivalently, the slot length n':

$$\left\{\Lambda\left(x\right), n'\right\} = \underset{\Lambda(x), n'}{\arg\min} \left(\frac{E_b}{N_0} : P_e \leq \varepsilon \right),$$

where E_b/N_0 is defined in (5.5). The optimization procedure was performed separately for each value of K_a. The resulting energy efficiency is represented by a green dashed line in Fig. 5.15 and corresponds to $T = 4$. The number of slots varies with K_a, as shown in Appendix D for $T = 1, 2, 4$.

Note that transitioning from a random coding approach to the T-fold IRSA results in a performance loss. However, all schemes based on the T-fold ALOHA with the same value of T can be considered as a reference.

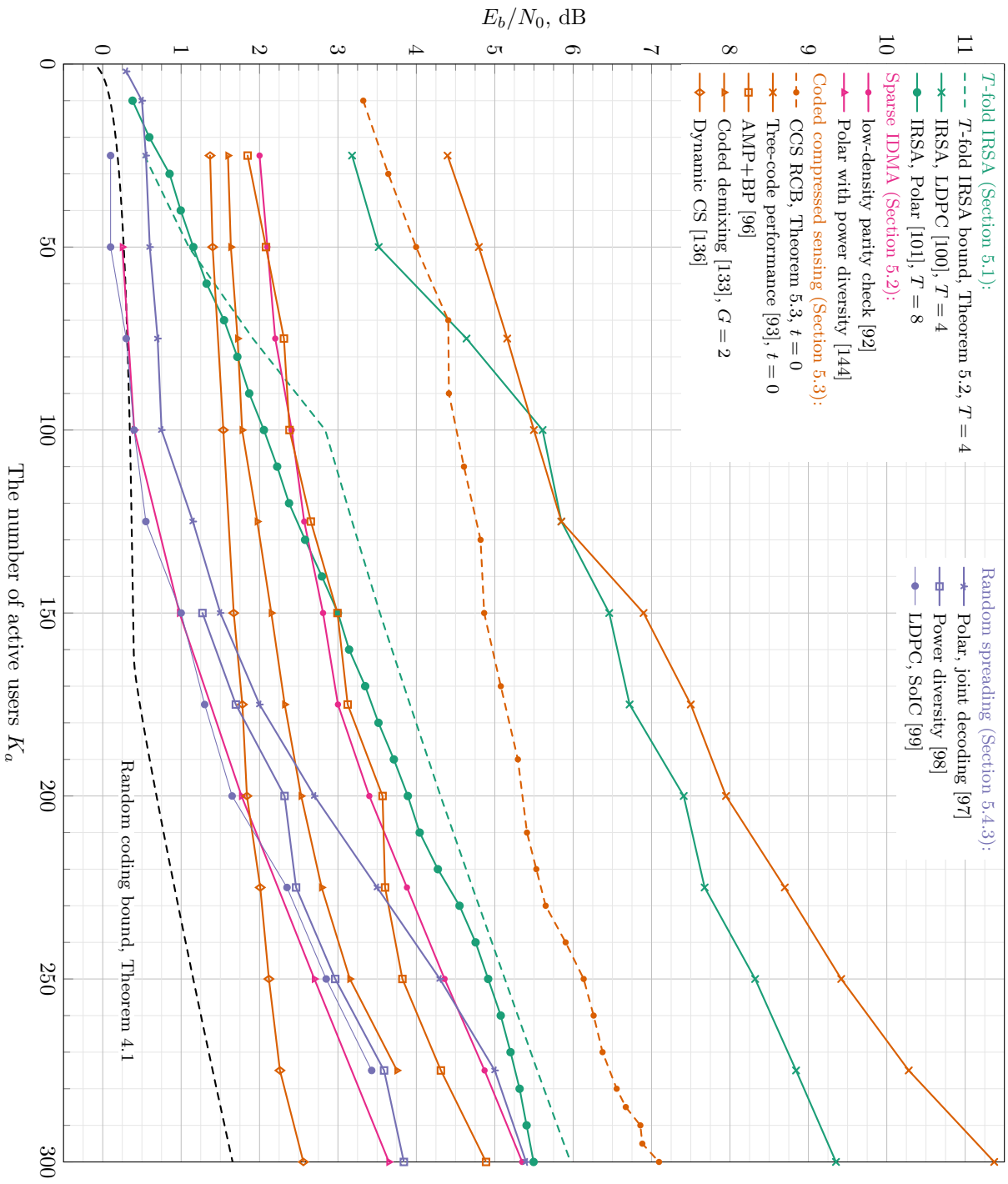

Figure 5.15: The minimum E_b/N_0 required to achieve an error rate below 5% as a function of the active user count. Theoretical bounds are represented by dashed lines, while practical schemes are shown with solid lines. Different colors correspond to various families of algorithms described in Sections 5.1, 5.2, 5.3, and 5.4. In a private discussion, the authors of [99] informed us that they could not validate the resulting curve. Hence, we mark it with a thin line.

T-fold IRSA with LDPC codes and a joint decoding algorithm

Consider the joint decoding of LDPC codes, as presented in Section 5.1.4. The authors of this scheme [100] performed an optimization search for system parameters as follows. Recall that this scheme relies on the IRSA-S approach (see Fig. 5.5), where k_p out of k information bits are allocated to the preamble (required for proper permutation to enhance joint decoding), while the remaining $k_c = k - k_p$ information bits are encoded using an LDPC code for each sequence transmitted within a slot.

The authors found that a sensing matrix based on a BCH code exhibited better decoding performance compared to randomly generated sensing matrices. Additionally, they optimized the LDPC code length by applying the RCB from Theorem 4.1 to estimate the slot-level decoding error. Given the optimal slot length, a regular $(3, 6)$ LDPC code was selected. The resulting parameters of the scheme are: $k_p = 9$, $k_c = 91$, a preamble length of $n_p = 63$, and an LDPC code length of $n_c = 394$, followed by BPSK modulation.

The search for the polynomial $\Lambda(x)$ in (5.1) was restricted to a single parameter β, with $\Lambda(x) = \beta x + (1 - \beta)x^2$. The authors also limited the maximum number of replicas to two. The reasoning for this is provided in Appendix D: higher values of T result in optimal polynomials in (5.1) having fewer transmission attempts (i.e., lower-degree polynomials). Recall that the authors considered $T = 4$, and the polynomials presented in Table D.3 are of degree two.

The energy efficiency of this scheme is depicted in Fig. 5.15 by a green line with $\times$-shaped markers and corresponds to a maximum of $T = 4$ simultaneous transmissions per slot. Note that there is a gap between this scheme and the achievability bound presented in Theorem 5.2. Nevertheless, it shows a significant improvement over [90], exceeding 10 dB for $K_a = 300$.

T-fold IRSA with polar codes and a joint decoding algorithm

The next improvement was made by replacing LDPC codes with polar codes [101], as presented in Section 5.1.4. Moreover, switching to polar codes resulted in higher values of T that a joint decoder can successfully resolve. Inspired by [92], the IRSA-F (see Fig. 5.6) was utilized. This preamble has a length of $n_p = 2000$ and carries $k_p = 15$ information bits, which are dedicated to recovering the transmission graph and frozen-bit patterns for each polar code. The users employ $(512, 85)$ polar coset codes.

The slot length is fixed for all K_a values, as it is challenging to assign an arbitrary length to the polar code. To choose the frozen positions, we utilized the proposed design procedure. We selected a common set of frozen tuples for all users, and the frozen bit values were chosen at random for each user, since using identical frozen values leads to poor performance. The list size is $\ell = 64$. At the same time, $(512, 85)$ polar coset codes with the JSC decoder can recover collisions of order up to $T = 8$.

The resulting energy efficiency is presented in Fig. 5.15 by a green line with circular markers. This scheme outperforms the LDPC-based scheme presented above and is very close to the IRSA achievability bound [106] ($T = 4$). Moreover, this scheme closely matches the sophisticated scheme

from Section 5.2.

5.6.2 Sparse interleave division multiple-access-based schemes

The energy efficiency of the sparse IDMA scheme described in Section 5.2 is presented in Fig. 5.15 by a magenta line with circular markers. The authors chose $k_p = 15$ bits for the preamble in the IRSA-F scheme, and the remaining $k_c = 85$ bits were encoded using the LDPC code. The structure of the LDPC code was carefully optimized for each number of active users K_a. The choice of the coding rate is:

- 0.125 for $K_a = 25, \ldots, 125$,
- 0.25 for $K_a = 150, \ldots, 200$,
- 0.4 for $K_a = 225, \ldots, 300$.

Each codeword appeared in the channel twice.

The length of the preamble is $n_p = 2000$. The CS design matrix uses $n_p/2$ randomly chosen rows from the discrete Fourier transform (DFT) matrix. Then, the imaginary and real parts of this matrix are stacked together to form a real-valued sensing matrix. The resulting matrix is a good choice for CS-based preamble detection because it satisfies the restricted isometry property (RIP) with high probability [145].

The resulting energy efficiency shows better performance compared to the IRSA-based scheme. The plot presented in Fig. 5.15 corresponds to the case without SIC. The use of SIC provides an improvement of about 0.5 dB for small values of $K_a \lesssim 100$ and approximately 0.2 dB for larger K_a values.

This scheme was further improved in [144] and is known as on-off division multiple access (ODMA). Compared to the previously described scheme, there are three main differences. The first is that the IRSA-like scheme has been replaced by a single transmission without replicas. The second is the replacement of LDPC codes with polar codes. The third difference is power diversity: instead of transmitting with a fixed energy, a random energy is chosen by each user such that the average power constraint is satisfied.

The positions where BPSK-modulated symbols of the polar-encoded message are transmitted are derived from the preamble. The choice of the preamble is given by the first k_p bits of the information message. Preamble detection is solved by a CS decoding algorithm. The authors have chosen the following parameters:

- Preamble codebook size 2^{k_p}, where $k_p = 12$ for $K_a = 50$, $k_p = 13$ for $K_a \leq 150$, and $k_p = 14$ otherwise.
- Polar codes of length $n_c = 512$ bits for $K_a \leq 200$ and $n_c = 256$ otherwise.

The authors considered the CRC of size 16 bits and the SCL decoding algorithm with a list size of 128. The performance of this scheme is represented by a magenta line with triangular markers.

5.6.3 Coded compressed sensing-based schemes

Numerical evaluation of Theorem 5.3

Let us start the numerical analysis of the CCS-based schemes with the RCB from Theorem 5.3 for $t = 0$, presented by a dashed orange line. The number of slots in a frame was optimized for each number of active users, and an OMP [146] decoding algorithm was utilized. The inner codebook is Gaussian and has a size of 2^{15} for each slot. The resulting energy efficiency is presented in Fig. 5.15 by a dashed orange line.

t-tree code numerical evaluation

Next, let us consider the performance of the tree-code scheme presented in Section 5.3.3 with $t = 0$. Recall that the case $t = 0$ was first considered in [93], and we follow these results in our presentation. The inner codebook size is 2^{15} for $K_a \geq 150$, and 2^{14} for $K_a < 150$. The inner codebook is a subcode of the $(2047, 23)$ BCH code. The slot length equals 2047. The inner code was decoded by the NNLS algorithm. The number of slots is fixed and equals 11. Hence, the frame length equals 22517. The authors suggested transmitting $k = 75$ information bits, which results in a similar code rate of 100/30000 for fair comparison.

The main problem was the allocation of information bits to each slot. When decoding each subsequent slot, all codeword candidates must be kept, and this number may dramatically grow [94]. Moreover, the final performance of the system may drastically degrade in the case of inappropriate information bit allocation. The information bit allocation has been carefully optimized, and the resulting performance is shown in Fig. 5.15 by an orange line with $\times$-shaped markers. The distance between the RCB and the scheme is about 2 dB at $K_a = 150$, and it gradually increases.

Further improvements

More sophisticated schemes that include interaction between inner and outer codes show significantly better energy efficiency results. The results from [96] are shown by an orange line with square markers, and the results from [133] are shown by triangle markers.

It is worth mentioning the last scheme, which does not use an outer code [136]. The outer code requires additional redundancy. The proposed solution eliminates this redundancy and makes the scheme the best-performing one when $K_a > 200$.

Table 5.1: This is a summary of the encoding parameters used in [97] as functions of the number of active users.

K_a	k_p	$k_c = k - k_p$	n_p	n_c	List size	CRC size
$25 - 75$	9	91	29	1024	32	16
$100 - 175$	8	92	29	1024	32	16
$200 - 225$	9	91	59	512	32	12
$250 - 300$	10	90	117	256	32	10

5.6.4 Random spreading-based solutions

In this section, we provide the numerical results of the practical schemes described in Section 5.4.3, namely joint decoding of polar codes, power diversity, and SoIC.

Joint decoding of polar codes

Let us start with polar codes and spreading [97]. As in the case of IRSA-F, the authors propose to select the spreading sequence based on the first k_p information bits. The length of the spreading sequence n_s generates a trade-off between the interference mitigation capability and the single-user code rate, because the frame length $n = n_p \times n_c$, and n_c is the code length. The optimal parameters for each number of active users are presented in Table 5.1. As soon as the CRC-aided polar codes have been selected, the CRC size is also specified. The resulting energy efficiency is presented in Fig. 5.15 by a blue line with star-shaped markers.

For a fixed value of K_a, the quantities n_c and n_p are optimized empirically to minimize the energy per bit required to achieve a target probability of error. The spreading sequences were generated from a Gaussian codebook.

Power diversity

The next scheme [98] considers power diversity for spreading sequences that improves single preamble detection and subsequent SIC. The authors of [98] did not change the spreading sequence and code lengths ($n_c = 256$, $n_p = 117$), but varied the number of bits used to derive the spreading sequence: $k_p = 14$ for $150 \leq K_a < 200$, $k_p = 15$ for $200 \leq K_a < 250$, and $k_p = 16$ for $K_a \geq 250$. This choice of parameters results in a small probability of preamble collision. The decoding list size was substantially larger (512) compared to the previous solution (32). The spreading sequences were also generated from a Gaussian codebook, but all spreading sequences were split into several groups having different powers. The number of groups and power levels were optimized to improve the signal-to-interference-plus-noise ratio at each decoding step. The resulting energy efficiency is presented in Fig. 5.15 by a blue line with square markers. The curve starts from $K_a = 150$ because smaller K_a values have only a single power splitting group.

Soft SIC

Finally, let us consider random spreading with a SoIC [99]. The spreading sequences are generated using $k_p = 12$ bits from the information message, and the length of the spreading sequence is $n_p = 86$. The LDPC code rate is 1/4. The structure of LDPC codes was optimized using a DE on protographs [112]. The performance of this scheme is presented by a blue line with circular markers in Fig. 5.15.

Chapter 6

Fading channels

More realistic channel models, such as the Rayleigh fading channel, have also been considered in the literature. While the AWGN channel may be a suitable model for satellite communications when multipath propagation is limited, the Rayleigh fading channel is more appropriate for cellular communications, as it assumes a rich scattering environment.

To operate in a fading channel, channel state information (CSI) must be acquired. However, estimating the channel for a large number of devices transmitting short packets is prohibitively difficult. Indeed, the best way to estimate the channel is to include a preamble or reference signals in the transmitted message. However, in the case of uncoordinated transmission, we must ensure that these reference signals remain nearly orthogonal. Unfortunately, orthogonality can be compromised by both short transmitted messages and a large number of simultaneously active users. Consequently, the URA model assumes the absence of the CSI at both the transmitters and the receiver. The works [147, 148, 149, 94] describe fundamental limits and practical schemes for single-antenna quasi-static Rayleigh fading channels. The most notable observation is that fading increases the required energy per bit by approximately 10 dB under the same order of PUPE requirements (see Fig. 5.15 and Fig. 6.2).

A. Fengler and G. Caire were the first to demonstrate that employing multiple antennas at the BS can significantly improve energy efficiency without increasing transmitter complexity [150]. Recall that the transmitters are autonomous, battery-powered devices and should be as simple as possible. The fundamental limits for URA with a MIMO receiver were established in [151, 152, 153]. Of particular importance is the scaling law derived in [154] and refined in [151]: with n channel uses and a sufficiently large number of BS antennas ($K_a/L = o(1)$), up to $K_a = O(n^2)$ active users can be served among K_{tot} potential users.

Practical receiver schemes have been extensively studied in the literature. The design of these schemes typically involves pilot-based channel estimation followed by decoding and SIC [154, 155], or a CCS approach combined with non-Bayesian activity detection [150], SIC [156], or the exploitation of slot-wise correlations [157]. The scheme proposed in [158] integrates these approaches and delivers state-of-the-art performance for the MIMO channel.

This chapter is organized as follows. In Section 6.1.1, we specify the channel model – a quasi-static Rayleigh fading MAC with a BS equipped with either a single or multiple receive antennas. To define the achievability bounds, we propose two decoding rules in Section 6.1.2: ML decoding and projection-based decoding. Furthermore, we show that projection-based decoding achieves the ε-capacity. Next, in Section 6.2, we consider single-antenna bounds, and in Section 6.3, we describe known results for the multiple-antenna case. Finally, we conclude this chapter by discussing low-complexity schemes presented in Sections 6.4.1 and 6.4.2 for single-antenna and multiple-antenna scenarios, respectively.

6.1 Channel model

6.1.1 Rayleigh fading model

The main cause of fading is multipath propagation. The receiver experiences interference of multiple transmitted signal copies. Each copy of the signal can experience random attenuation due to reflections from obstacles and random delays due to different propagation path length. Finally, a moving receiver may experience a Doppler spread, but this is not the case for the massive machine-type communications, where most transmitting devices are stationary or moving relatively slowly.

The Rayleigh fading model corresponds to a so-called rich scattering environment. When the number of scatterers becomes large and each reflection coefficient is random, the central limit theorem can be applied. Moreover, in such a rich scattering environment, there is no line-of-sight component, and the received power distribution becomes uniform over the angle-of-arrival (AoA).

As a result, for a single-antenna receiver, the channel coefficient follows a circularly symmetric complex normal distribution, and the magnitude of this channel coefficient becomes Rayleigh-distributed. For a MIMO receiver, assuming a uniform power distribution over the AoA and a regular antenna array, the resulting channel coefficients at each receiving antenna follow the same distribution [34].

When the transmitting devices are stationary and operate in a slowly-varying environment, the resulting channel model becomes quasi-static. Furthermore, since each device transmits infrequently, the channel coefficient remains constant during a transmission but changes from block to block, with channel realizations being independent across different blocks. Moreover, the channel coefficients (under the aforementioned central limit theorem) follow a circularly symmetric complex normal distribution, meaning that the phase of the signal is uniformly distributed. Additionally, small synchronization errors can also be described by this phenomenon.

The time-domain delay results in a phase shift of the signal in the frequency domain. On the other hand, orthogonal frequency-division multiplexing (OFDM) provides an elegant solution for frequency-domain channel equalization, as long as the length of the cyclic prefix exceeds the maximum relative delay value [159]. However, OFDM signals have a large peak-to-average power ratio, which may lead to high energy consumption under linear amplification, potentially impacting battery life. Therefore, we plan to focus on simpler (and constant-envelope) modulations at low rates. In this context, synchronization errors can partially be modeled as phase multipliers in the channel

coefficients.

Now, let us formulate the URA problem for the fading channel scenario with a receiver equipped with L antennas. We assume that each transmitting device has a single transmit antenna. Let $\mathcal{T} = W_1, W_2, \ldots, W_{K_a}$ denote the set of transmitted messages (using the same codebook with blocklength n). The channel model can be described as follows.

$$\mathbf{Y} = \boldsymbol{X}\boldsymbol{\Phi}(\mathcal{T})\mathbf{H} + \mathbf{Z}, \tag{6.1}$$

where $\mathbf{Y} = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_L] \in \mathbb{C}^{n\times L}$ is the channel output matrix, $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M] \in \mathbb{C}^{n\times M}$ with $\boldsymbol{x}_W = f(W), W \in [M]$, is the common codebook. Here, $\boldsymbol{\Phi}(\mathcal{T}) \in \{0,1\}^{M\times K_a}$ represents the activity matrix, where each column contains a single non-zero element at the position corresponding to the user's message. The matrix $\mathbf{Z} \in \mathbb{C}^{n\times L}$ denotes the AWGN matrix, with each element sampled i.i.d. from $\mathcal{CN}(0,1)$. Finally, $\mathbf{H} \in \mathbb{C}^{K_a\times L}$ is the matrix of fading coefficients, where each element is sampled i.i.d. from $\mathcal{CN}(0,1)$. Note that the fading coefficients are independent of the codewords and the noise matrix $\mathbf{Z}$.

In what follows, we assume that $\mathbf{H}$ is unknown to both the transmitters and the receiver (the unknown CSI scenario). As a result, the receiver observes L linear combinations of the transmitted messages, where the coefficients of these linear combinations are determined by the rows of the matrix $\mathbf{H}$.

6.1.2 Decoding rules

In the case of the AWGN channel model, we considered the set of codewords closest to the received signal, as described in (4.1). However, since the new channel model involves unknown CSI, the decoding rule must be modified. The first idea was proposed in [160]. In the absence of noise, the signal received by each antenna lies in the subspace spanned by the transmitted codewords, regardless of the fading coefficients. Thus, the most likely subset of transmitted codewords is the one that maximizes the projection of the received signal onto the subspace spanned by these codewords.

As we will see later, the projection method ignores the prior distribution of fading coefficients. Additionally, the projection-based decoder has a limitation: it restricts the maximum number of active users to the blocklength, i.e., $K_a < n$. For the case of multiple antennas, as shown in [151], the maximum number of users that can be decoded may exceed the blocklength. To address this issue, an ML-based decoding rule was proposed in [151].

ML decoding rules [151, 152] estimate the transmitted codeword set by maximizing the conditional probability

$$\Pr\left[\mathbf{Y}|\boldsymbol{X}_{\mathcal{R}'}\right] = \mathbb{E}_{\mathbf{H}}\left\{\Pr\left[\mathbf{Y}|\boldsymbol{X}_{\mathcal{R}'}, \mathbf{H}\right]\right\},$$

which is the expectation over the channel statistics. Note that the receiver does not know the channel realizations but may have access to the channel statistics in many scenarios. Let us denote the matrix composed of the codewords from $\mathcal{R}'$ as $\boldsymbol{X}_{\mathcal{R}'}$. The ML decoding rule is given by

$$\mathcal{R} = \underset{\mathcal{R}'\subset[M],|\mathcal{R}'|=K_a}{\arg\max} \mathbb{E}_{\mathbf{H}}\left\{\Pr\left[\boldsymbol{Y}|\boldsymbol{X}_{\mathcal{R}'}, \mathbf{H}\right]\right\}. \tag{6.2}$$

The idea of the projection decoder is to take not the expectation, but the maximum of the corresponding conditional probability, as described in [148]. The projection-based decoding rule is given by

$$\mathcal{R} = \underset{\mathcal{R}' \subseteq [M], |\mathcal{R}'| = K_a}{\arg\max} \max_{\boldsymbol{H}} \left\{ \Pr\left[\boldsymbol{Y} | \boldsymbol{X}_{\mathcal{R}'}, \boldsymbol{H} \right] \right\}. \tag{6.3}$$

Note that the probability $\Pr\left[\boldsymbol{Y} | \boldsymbol{X}_{\mathcal{R}'}, \boldsymbol{H} \right]$ is governed by Gaussian noise, as the conditioning results in a known linear combination of a candidate set $\boldsymbol{X}_{\mathcal{R}'}$ of codewords. Assuming Gaussian noise, we can rewrite (6.3) as

$$\begin{aligned} \mathcal{R} &= \underset{\mathcal{R}' \subseteq [M], |\mathcal{R}'| = K_a}{\arg\min} \min_{\boldsymbol{H}} \| \boldsymbol{Y} - \boldsymbol{X}_{\mathcal{R}'} \boldsymbol{H} \|_F^2 \\ &= \underset{\mathcal{R}' \subseteq [M], |\mathcal{R}'| = K_a}{\arg\min} \left(\| \boldsymbol{Y} \|_F^2 - \| \boldsymbol{P}_{\mathcal{R}'} \boldsymbol{Y} \|_F^2 \right) \\ &= \underset{\mathcal{R}' \subseteq [M], |\mathcal{R}'| = K_a}{\arg\max} \| \boldsymbol{P}_{\mathcal{R}'} \boldsymbol{Y} \|_F^2, \end{aligned} \tag{6.4}$$

where the second equality follows from the fact that the received signal $\boldsymbol{Y}$ is fixed. Here $\| \cdot \|_F^2$ denotes the Frobenius norm, and $\boldsymbol{P}_{\mathcal{R}'}$ is the orthogonal projection onto the subspace spanned by the codewords from $\mathcal{R}'$.

A comparison of (6.2) and (6.3) clarifies the previous statement that a projection-based decoding rule does not take channel statistics into account.

6.2 Single-antenna quasi-static Rayleigh fading MAC

In this section, we present existing bounds for the single-antenna quasi-static Rayleigh fading MAC. First, we describe the ε-capacity bound, which serves as a suitable capacity measure for quasi-static channels. Next, we describe the Shamai-Bettesh asymptotic bound, which provides insight into the corresponding bounds in the finite blocklength regime. Finally, we consider the RCB for the FBL scenario (Theorem 6.2) and the converse bound (Theorem 6.3).

6.2.1 Capacity region, all-active users

In the presence of fading, the capacity of the channel should be defined differently compared to Shannon's formula for the AWGN channel. Consider a slow fading channel model [34], or a channel with a large coherence block. Our quasi-static model fits well with the definition of the slow fading model. For any prospective rate of reliable communication, there is a probability that a channel realization will be of such a small magnitude that the user will be unable to communicate reliably over this channel at the selected rate. Thus, in the slow-fading regime, an ε-capacity should be introduced instead.

Similarly, let $C_{\varepsilon,J}$ denote the ε-capacity region, which can be defined for the case of all-active K_a [147] under *joint* decoding. A rate tuple $(R_1, \ldots, R_{K_a})$ is said to be ε-achievable [161] if

there is a sequence of codes whose rates are asymptotically at least R_i such that the joint error is asymptotically smaller than ε.

$$C_{\varepsilon,J} = \{R = (R_1, \ldots, R_{K_a}) : \forall i, R_i \geq 0 \,\text{and}\, P_0(R) \leq \varepsilon\},$$

where

$$P_0(R) = \Pr\left[\bigcup_{S \subset [K_a]} \left\{\log\left(1 + P\sum_{i \in S} |H_i|^2\right) \leq \sum_{i \in S} R_i\right\}\right].$$

Theorem 6.1 (Projection decoding achieves $C_{\varepsilon,J}$ [147])**.** *Let $R \in C_{\varepsilon,J}$. Then R is ε-achievable through a sequence of codes with the decoder being the projection decoder* (6.3)*.*

Nevertheless, in the FBL regime and under the PUPE metric, the projection decoder may be suboptimal. However, this type of decoder allows the achievability bound to be addressed, as presented in Theorem 6.5.

6.2.2 Shamai-Bettesh capacity bound

There is another asymptotic bound ($n \to \infty$) for the PUPE in the case of symmetric rates and large K_a, the Shamai-Bettesh capacity bound from [162]. The idea is as follows: the joint decoder knows the realization of the fading coefficients, and users are ranked according to the strength of their fading coefficients. It first tries to decode all users. If it fails (i.e., if the rate vector is not inside the instantaneous full capacity region), it drops the user with the smallest fading coefficient and tries to decode the remaining $K_a - 1$ users. The dropped user then contributes to the noise.

This process continues iteratively, and the fraction of users that were not decoded corresponds to the PUPE. Since the case under discussion involves large K_a, the order statistics of the absolute value of the fading coefficients crystallize (i.e., become almost non-random), and hence analytical expressions can be derived for the outage in terms of spectral efficiency (kK_a/n) and total power.

Note that the Shamai-Bettesh bound is only an achievable bound (i.e., it is not guaranteed to be tight) for the capacity under PUPE. It does not apply to our setting for two reasons. First, it assumes different codebooks for different users, and second, it assumes asymptotically large blocklength.

Note also that the asymptotic regime considered in the Shamai-Bettesh bound is as follows: first, n is taken to ∞ (under a fixed K_a), and second, K_a is also taken to infinity. However, based on studies of the non-fading channels [87], the correct asymptotic regime is to take both K_a and n to infinity at a fixed ratio [147].

6.2.3 FBL regime, partial activity

Let us formulate the achievability bound for a single-antenna URA under a quasi-static Rayleigh fading channel. We provide a simplified formulation and outline the main steps of the proof,

referring the reader to [148] for the complete proofs.

As in the case of AWGN, we consider the event $\Pr[\mathcal{E}_t]$ (4.2), which represents exactly t errors occurring. We assume a circularly symmetric Gaussian codebook, similar to Definition 4.1.

Definition 6.1. *Let $\mathcal{G}_{\mathcal{CN}}(M, n, P)$ be the ensemble of Gaussian codebooks of size $n \times M$, where each element is sampled i.i.d. from $\mathcal{CN}(0, P)$.*

To estimate the PUPE, we must evaluate $\Pr[\mathcal{E}_t]$ for $t \in [K_a]$. Similar to the AWGN case, we denote the probability of power violation and message collision as p_0.

Theorem 6.2 (Achievability bound for single-antenna case [148])**.** *Fix $P' < P$. There exists a codebook $\boldsymbol{X}^* \in \mathcal{G}_{\mathcal{CN}}(M, n, P')$ satisfying the power constraint P' and providing P_e using eq.* (4.2)*, where p_0 is bounded by* (4.6)*, and $\Pr[\mathcal{E}_t] \leq p_t$, where*

$$p_t \leq \inf_{\delta > 0} \left(\binom{K_a}{t} e^{-(n-K_a)\delta} + p_{1,t} \right), \tag{6.5}$$

where

$$p_{1,t} = \Pr\left[\bigcup_{\substack{\mathcal{M} \subset \mathcal{T} \\ |\mathcal{M}| = t}} \{ G_{\boldsymbol{Y},\mathcal{T},\mathcal{C},t} \geq V_{n,t} \} \right], \tag{6.6}$$

$$G_{\boldsymbol{Y},\mathcal{T},\mathcal{C},t} = \frac{\|\boldsymbol{Y}\|^2 - \|\boldsymbol{P}_{\mathcal{T}}\boldsymbol{Y}\|^2}{\|\boldsymbol{Y}\|^2 - \|\boldsymbol{P}_{\mathcal{C}}\boldsymbol{Y}\|^2}. \tag{6.7}$$

The Frobenius norm in (6.7) *becomes a simple Euclidean norm for the single-antenna case, and*

$$V_{n,t} = e^{-(\delta + R_1 + s_t)}, \quad s_t = \frac{\ln \binom{n-K_a+t-1}{t-1}}{n - K_a}, \quad R_1 = \frac{\ln \binom{M-K_a}{t}}{n - K_a}.$$

The original formulation [148] of the theorem above assumed that the number of active users may differ from the intended list size at the receiver. This approach was inspired by the Shamai-Bettesh asymptotic bound described above (see Section 6.2.2). This approach may be beneficial in the case of large K_a: it may be better not to consider the weakest users but instead decode the remaining ones with a smaller probability of error. In the formulation above, we omit this assumption for simplicity and consider the case when the decoder tries to extract all active users.

This theorem introduces a projection ratio (6.7), which depends on the choice of the correctly received codewords, and a union probability is taken in (6.6). This union over all choices can be replaced by simply selecting the *t-weakest*[1] codewords as missed, significantly reducing the computational complexity. To evaluate the achievability bound numerically, the tail probability of the projection ratio can be computed either by sampling or analytically (see the complete proof of this theorem in [148]).

[1] Or codewords that have the smallest received energy

The main step of this theorem's proof avoids a large combinatorial term $\binom{M-K_a}{t}$ described in Section 4.1.1. This is done by observing that a projection of the fixed vector onto a random subspace generated by Gaussian vectors arises. The norm of this projection follows a beta distribution [147].

Let us rewrite the condition that exactly t errors occurred conditioned on the channel coefficients, noise, and a set of transmitted codewords. By applying the union bound, we have

$$\Pr\left[\mathcal{E}_t \middle| \mathcal{T}, \mathbf{H}, \mathbf{Z}\right] \leq \Pr\left[\bigcup_{\mathcal{M}} \bigcup_{\mathcal{F}} F_{\mathcal{F},\mathcal{C},t} \middle| \mathcal{T}, \boldsymbol{H}, \boldsymbol{Z}\right], \tag{6.8}$$

where

$$F_{\mathcal{F},\mathcal{C},t} = \left\{\|\boldsymbol{P}_{\mathcal{C}\cup\mathcal{F}}\boldsymbol{Y}\|^2 > \|\boldsymbol{P}_{\mathcal{C}\cup\mathcal{M}}\boldsymbol{Y}\|^2\right\}.$$

Note that the union $\bigcup_{\mathcal{M}}$ is taken over $\binom{M-K_a}{t}$ variants. Hence, the expectation over $\mathcal{M}$ should be performed first. To perform this step, let us fix the set of correctly received codewords $\mathcal{C}$ and show that the projection norm $\|\boldsymbol{P}_{\mathcal{C}\cup\mathcal{F}}\boldsymbol{Y}\|^2$ is distributed as

$$\|\boldsymbol{P}_{\mathcal{C}\cup\mathcal{F}}\boldsymbol{Y}\|^2 \sim \|\boldsymbol{P}_{\mathcal{C}}\boldsymbol{Y}\|^2 + \left\|\boldsymbol{P}_{\mathcal{C}}^{\perp}\boldsymbol{Y}\right\|^2 \beta\left(n-K_a,t\right).$$

Following the notation from [153], $\boldsymbol{P}_{\mathcal{C}\bigcup\mathcal{F}} = \boldsymbol{P}_{\mathcal{C}} + \boldsymbol{P}_{\mathcal{C}^{\perp},\mathcal{F}}$ by a Gram-Schmidt procedure, where $\boldsymbol{P}_{\mathcal{C}^{\perp},\mathcal{F}}$ is a projection onto span $\left\{\boldsymbol{P}_{\mathcal{C}}^{\perp}\boldsymbol{X}_{\mathcal{F}}\right\}$, and $\boldsymbol{P}_{\mathcal{C}}^{\perp} = \boldsymbol{I} - \boldsymbol{P}_{\mathcal{C}}$.

Finally, under the fixed $\mathcal{C}$, $\|\boldsymbol{P}_{\mathcal{C}\cup\mathcal{F}}\boldsymbol{Y}\|^2$ is a random variable that depends only on falsely detected codewords. Moreover,

$$\frac{\left\|\boldsymbol{P}_{\mathcal{C}^{\perp},\mathcal{F}}\boldsymbol{Y}\right\|^2}{\left\|\boldsymbol{P}_{\mathcal{C}}^{\perp}\boldsymbol{Y}\right\|^2} \sim \beta\left(n-K_a,t\right) \tag{6.9}$$

is beta-distributed [163] as a projection of a fixed vector onto a random subspace spanned by span $\left\{\boldsymbol{P}_{\mathcal{C}}^{\perp}\boldsymbol{X}_{\mathcal{F}}\right\}$ in the dimension $n-K_a+t$ – the dimensionality of a subspace orthogonal to span $\{\boldsymbol{X}_{\mathcal{C}}\}$.

The final building block of the theorem proof is that the cumulative density function of $1-\beta$ (where β is beta-distributed) random variable is bounded by

$$F_{1-\beta}\left(x;n-K_a,t\right) \leq \binom{n-K_a+t-1}{t-1} x^{n-Ka}, \tag{6.10}$$

and the resulting probability of t errors occurring will be upper-bounded by

$$\Pr\left[\mathcal{E}_t \middle| \mathcal{T}, \mathbf{H}, \mathbf{Z}\right] \leq \mathbb{E}_{\mathcal{T},\mathbf{H},\mathbf{Z}}\left[\min\left\{1, \sum_{\substack{\mathcal{C}\subset\mathcal{T} \\ |\mathcal{C}|=K_a-t}} e^{(n-K_a)(s_t+R_1)} G^{n-K_a}_{\boldsymbol{Y},\mathcal{T},\mathcal{C},t}\right\}\right].$$

The parameter δ in the theorem appears after applying Fano's trick (see Section 4.2.2) on the formula above.

To perform numerical evaluation of this bound, the following steps must be performed:

- Sample $\mathbf{H}$, codewords from $\mathcal{T}$, and $\mathbf{Z}$,
- Evaluate the cumulative distribution function of (6.7). A simplification by choosing t *weakest* users rather than checking $\binom{K_a}{t}$ variants can be considered,
- Perform optimization over δ.

Now let us describe a simple converse bound based on results from [164] and the meta-converse from [165].

Theorem 6.3 (Multi-user, single-antenna converse [148])**.** *Let*

$$L_n = n\log(1+PG) + \sum_{i=1}^{n}\left(1 - |\sqrt{PG}Z_i - \sqrt{1+PG}|^2\right)$$

and

$$S_n = n\log(1+PG) + \sum_{i=1}^{n}\left(1 - \frac{|\sqrt{PG}Z_i - 1|^2}{1+PG}\right),$$

where $G = |H|^2$ *and* $Z_i \overset{i.i.d.}{\sim} \mathcal{CN}(0,1)$*. Then, for every* n *and* $0 < \epsilon < 1$*, any* $(M, n-1, \epsilon)$ *code for the quasi-static* K_a *MAC satisfies*

$$\log(M) \leq \log(K_a) + \log\frac{1}{\Pr\left[L_n \geq n\gamma_n\right]}$$

where γ_n *is the solution of*

$$\Pr\left[S_n \leq n\gamma_n\right] = \epsilon.$$

6.3 Multi-antenna quasi-static Rayleigh fading MAC

In this section, we consider the case where a BS is equipped with multiple antennas – the so-called MIMO scenario. As discussed previously, the projection-based decoder achieves the ε-capacity in the single-antenna case. However, as shown in Section 6.3.1, the number of successfully detected users can exceed the frame length, i.e., K_a can be greater than the frame length n in the MIMO case. Consequently, when $K_a > n$, the projection-based decoding (6.3) will no longer work, as the subspace spanned by the transmitted codewords coincides with the full signal space.

To address this problem, the ML-based bound in (6.2) was proposed in [151]. Previously, we considered the single-antenna case, where the approach relied on eliminating the large combinatorial term $\binom{M-K_a}{t}$. However, in the proposed ML-based bound, this simplification does not lead to an efficient solution. Evaluating this bound requires sampling codewords from all possible sets: correctly detected codewords $\mathcal{C}$, falsely detected codewords $\mathcal{F}$, and missed codewords $\mathcal{M}$ (see Fig. 4.1).

On the other hand, a similar approach based on projection decoding (6.3) offers some improvements when the number of active users K_a is less than n. Nonetheless, deriving a projection-based achievability bound in a manner similar to the single-antenna case still does not yield an efficient solution. This is because the projection ratios (6.9) have an intractable distribution, requiring the handling of angles between random subspaces.

6.3.1 Scaling laws

The benefits of multiple antennas were analyzed by Fengler et al. [150], where the authors examined the scenario of *activity detection*. A set of active users sends their K_a randomly chosen preambles (selected from a total of K_{tot} preambles) in a dedicated slot. Since detecting a single transmission in a massive MIMO regime requires accurate channel estimation, activity detection plays a crucial role in practical implementations. The goal of the receiver is to detect the subset of preambles selected for transmission and then proceed to the user's signal detection step, where the CSI is obtained from the preambles.

This problem is closely related to the URA problem. Given multiple antennas ($L > 1$), the corresponding URA problem (6.1) can be formulated as a multiple measurement vector (MMV) problem in the context of CS [166] (see also Section 3.2).

The scaling law proposed in [150] states that given

$$\frac{K_a}{L} \approx o\left(1\right),$$

the number of detectable users is

$$\mathcal{O}\left(\frac{n^2}{\log^2\left(\frac{K_{\text{tot}}}{K_a}\right)}\right).$$

Notably, this number can be significantly higher compared to the projection-based decoding algorithm, where $K_a < n$.

6.3.2 Achievability bounds for MIMO receiver

In this section, we describe the achievability bound for the MIMO case based on random coding, as introduced in Section 4.1.1. Given the scaling law described above, there is a motivation to shift away from the projection-based decoding algorithm, which is limited by $K_a < n$. As the number of receiver antennas increases, an ML-based decoding algorithm is chosen for the bound [151].

We consider a Gaussian codebook ensemble (see Definition 6.1) and recall the channel model (6.1). Suppose the *channel statistics*, i.e., the distribution of channel coefficients – the elements of $\mathbf{H}$ – are known. Then, conditioned on $\boldsymbol{X}\boldsymbol{\Phi}(\mathcal{T})$, the columns of $\mathbf{Y}$, the signal vector received by each antenna, are independent and normally distributed:

$$\mathbf{y}_i \sim \mathcal{CN}\left(\mathbf{0}, \boldsymbol{I}_n + \boldsymbol{X}\boldsymbol{\Gamma}\boldsymbol{X}^H\right), \quad \boldsymbol{\Gamma} = \boldsymbol{\Phi}(\mathcal{T})\boldsymbol{\Phi}^H(\mathcal{T}),$$

where $(\cdot)^H$ denotes the Hermitian transpose. Letting $\boldsymbol{\Sigma} = \boldsymbol{I}_n + \boldsymbol{X}\boldsymbol{\Gamma}\boldsymbol{X}^H$, we obtain the following p.d.f.:

$$p\left(\mathbf{Y}|\boldsymbol{X}\boldsymbol{\Phi}\left(\mathcal{T}\right)\right) = \pi^{-Ln}\left|\boldsymbol{\Sigma}\right|^{-L}\exp\left(-\text{tr}\left(\boldsymbol{\Sigma}^{-1}\mathbf{Y}\mathbf{Y}^H\right)\right).$$

The corresponding decoding function to be optimized is:

$$g\left(\mathbf{Y}, \boldsymbol{X}\boldsymbol{\Gamma}\boldsymbol{X}^H\right) = L\log\left|\boldsymbol{\Sigma}\right| + \text{tr}\left(\boldsymbol{\Sigma}^{-1}\mathbf{Y}\mathbf{Y}^H\right). \tag{6.11}$$

The decoder aims to find the set of active users that minimizes (6.11). In the subsequent analysis, we define the matrix $\mathbf{\Gamma}$ formed by the message set $\mathcal{S}$ as $\mathbf{\Gamma}_{\mathcal{S}}$ and denote (6.11) as $g\left(\mathbf{\Gamma}_{\mathcal{S}}\right)$. As before, we estimate the probability of the event $\mathcal{E}_t$, in which exactly t errors occur:

$$\Pr\left[\mathcal{E}_t\right] = \Pr\left[\mathcal{E}_{t,\mathcal{M},\mathcal{F}}\right], \quad \mathcal{E}_{t,\mathcal{M},\mathcal{F}} = \bigcup_{\mathcal{F}} \bigcup_{\mathcal{M}} \left\{g\left(\mathbf{\Gamma}_{\mathcal{C}\cup\mathcal{F}}\right) \leq g\left(\mathbf{\Gamma}_{\mathcal{C}\cup\mathcal{M}}\right)\right\}.$$

Clearly, we can use the union bound to upper-bound the right-hand side. However, the union bound is known to overestimate the resulting probability. To tighten the bound, we use Fano's trick, i.e., we introduce the following region:

$$\mathcal{B}_{\mathcal{M}} = \{\mathbf{Y} : g\left(\mathbf{\Gamma}_{\mathcal{T}}\right) \leq \alpha g\left(\mathbf{\Gamma}_{\mathcal{C}}\right) + \beta nL\}, \quad \mathcal{B} = \bigcap_{\mathcal{M}} \mathcal{B}_{\mathcal{M}}. \tag{6.12}$$

Using Fano's trick, we upper-bound $\Pr\left[\mathcal{E}_t\right]$ as follows:

$$\Pr\left[\mathcal{E}_t\right] = \Pr\left[\mathcal{E}_{t,\mathcal{M},\mathcal{F}}\right] \leq \Pr\left[\mathcal{E}_{t,\mathcal{M},\mathcal{F}} \bigcap \mathcal{B}\right] + \Pr\left[\mathcal{B}^c\right]. \tag{6.13}$$

The main idea is as follows. We apply the union bound only when $\mathbf{Y}$ is within an allowed region, while we assume that $\Pr[E_t] = 1$ otherwise. This approach prevents overestimating the probability by avoiding the use of the union bound for $\mathbf{Y}$ outside the good region, where the union bound is effective.

Theorem 6.4 (ML-based achievability bound [151] for the same codebook [152])**.** *Fix $P' < P$ and consider a receiver equipped with L antennas. There exists a codebook $\boldsymbol{X}^* \in \mathcal{G}_{\mathcal{CN}}\left(M, n, P'\right)$ satisfying the power constraint P' and providing P_e using eq.* (4.2)*, where p_0 is bounded by* (4.6)*, and $\Pr\left[\mathcal{E}_t\right] \leq p_t$, where*

$$p_t \leq \inf_{0\leq\alpha\leq 1, 0\leq\beta} \left\{q_{1,t}\left(\alpha,\beta\right) + q_{2,t}\left(\alpha,\beta\right)\right\}, \tag{6.14}$$

where

$$q_{1,t}\left(\alpha,\beta\right) = \binom{K_a}{t}\binom{M-K_a}{t}\mathbb{E}_{\mathsf{X}}\left[\inf_{\substack{u\geq 0, r\geq 0,\\ \lambda_{min}\left(\mathsf{B}\right)>0}} \left(e^{Lrn\beta}\cdot e^{L\mu}\right)\right], \tag{6.15}$$

$$\mu = \left(u-r\right)\log\left|\mathsf{F}_{\mathcal{T}}\right| - u\log\left|\mathsf{F}_{\mathcal{R}}\right| + r\alpha\log\left|\mathsf{F}_{\mathcal{C}}\right| - \log\left|\mathsf{B}\right|,$$

$$q_{2,t} = \binom{K_a}{t}\inf_{\delta\geq 0}\mathbb{E}_{\mathsf{X}}\left[\frac{\gamma\left(Lm, c_\delta\right)}{\Gamma\left(Lm\right)} + 1 - \frac{\gamma\left(nL, \left(1+\delta\right)nL\right)}{\Gamma\left(Ln\right)}\right], \tag{6.16}$$

$$c_\delta = L\frac{n\left(1+\delta\right)\left(1-\alpha\right) - \alpha\log\left|\mathsf{F}_{\mathcal{C}}\right| + \log\left|\mathsf{F}_{\mathcal{T}}\right| - n\beta}{\alpha\prod_{i=1}^{m}\lambda_i^{1/m}},$$

$$\mathsf{B} = \left(1-u+r\right)\boldsymbol{I}_n + u\mathsf{F}_{\mathcal{R}}^{-1}\mathsf{F}_{\mathcal{T}} - r\alpha\mathsf{F}_{\mathcal{C}}^{-1}\mathsf{F}_{\mathcal{T}},$$

where $m = \min\left\{n, t\right\}$. The matrix $\mathsf{F}_{\mathcal{S}}$ is the covariance matrix corresponding to the codeword set $\mathcal{S}$, given by

$$\mathsf{F}_{\mathcal{S}} = \boldsymbol{I}_n + \mathsf{X}\mathbf{\Gamma}_{\mathcal{S}}\mathsf{X}^H.$$

By the notation above,

$$\mathbf{F}_{\mathcal{S}} = \boldsymbol{I}_n + \mathbf{X}_{\mathcal{S}}\mathbf{X}_{\mathcal{S}}^H.$$

Finally, $\lambda_1, \ldots, \lambda_m$ are the eigenvalues of the matrix $\mathbf{F}_{\mathcal{C}}^{-1}\mathbf{X}\Gamma_{\mathcal{M}}\mathbf{X}^H$ of rank m, arranged in decreasing order.

In this theorem, the expectation is taken over a codebook ensemble. Thus, w.l.o.g., let us assume that the correctly received, missed, and falsely detected codewords (see Section 3.1 and Fig. 4.1) correspond to the first $K_a + t$ codewords of the codebook $\mathbf{X}_{[K_a+t]}$. Hence, we define the sets as follows:

$$\begin{aligned}
\text{Transmimtted codewords } \mathcal{T} =& [K_a], \\
\text{Received codewords } \mathcal{R} =& [K_a + t] \backslash [t], \\
\text{Correctly received codewords } \mathcal{C} =& [K_a] \backslash [t], \\
\text{Missed codewords } \mathcal{M} =& [t], \\
\text{Falsely detected codewords } \mathcal{F} =& [K_a + t] \backslash [K_a].
\end{aligned}$$

The idea is to first apply Fano's trick (6.13). Then, for the first probability term, $\Pr\left[\mathcal{E}_{t,\mathcal{M},\mathcal{F}} \bigcap \mathcal{B}\right]$, a Chernoff bound can be applied (Lemma G.1).

The proof ultimately results in an expectation over a codebook ensemble, which can be evaluated numerically. However, the minimum required number of samples remains unknown. Moreover, sampling falsely detected codewords may be challenging due to the large number of possible combinations, given by $\binom{M-K_a}{t}$.

Theorem 6.5 (Projection-based achievability bound [153] for the same codebook)**.** *Fix $P' < P$ and consider a receiver equipped with L antennas. There exists a codebook $\boldsymbol{X}^* \in \mathcal{G}_{\mathcal{CN}}(M, n, P')$ satisfying the power constraint P' and providing P_e using eq.* (4.2)*, where p_0 is bounded by* (4.6)*, and $\Pr[\mathcal{E}_t] \leq p_t$, where*

$$\Pr[\mathcal{E}_t] \leq \inf_{0 \leq \alpha \leq 1} (p_{1,t} + p_{2,t}), \tag{6.17}$$

where

$$p_{1,t} = \binom{K_a}{t}\binom{M-K_a}{t}\mathbb{E}_{\mathbf{X}}\left[\inf_{u,v>0,\lambda_{\mathcal{D}}>0} |\boldsymbol{I} - \mathbf{D}\boldsymbol{\Sigma}|^{-L}\right], \tag{6.18}$$

$$p_{2,t} = \binom{K_a}{t}\mathbb{E}_{\mathbf{X}}\left[\inf_{\delta>0,\lambda_{\mathcal{B}}>0} |\boldsymbol{I} - \mathbf{B}\boldsymbol{\Sigma}|^{-L}\right], \tag{6.19}$$

where

$$\begin{aligned}
\boldsymbol{\Sigma} &= \boldsymbol{I}_n + \mathbf{X}_{\mathcal{T}}\mathbf{X}_{\mathcal{T}}^H, \\
\mathbf{D} &= u\mathbf{P}_{\mathcal{R}} - u\mathbf{P}_{\mathcal{T}} + \alpha v\mathbf{P}_{\mathcal{C}}^{\perp} - v\mathbf{P}_{\mathcal{T}}^{\perp}, \\
\mathbf{B} &= -\alpha\delta\mathbf{P}_{\mathcal{C}}^{\perp} + \delta\mathbf{P}_{\mathcal{T}}^{\perp}.
\end{aligned}$$

Here, $\lambda_{\mathcal{D}}$ and $\lambda_{\mathcal{B}}$ denote the minimum eigenvalues of $\boldsymbol{I} - \mathbf{D}\boldsymbol{\Sigma}$ and $\boldsymbol{I} - \mathbf{B}\boldsymbol{\Sigma}$, respectively. The expectations are taken over $\mathbf{X}_{\mathcal{T}\cup\mathcal{R}}$. The sets $\mathcal{T}$, $\mathcal{R}$, and $\mathcal{C}$ are defined above.

The idea behind this theorem is similar to that of the previous one, but the decoding function (6.11) is replaced with the projection rule (6.3). Similar to the ML-based achievability, Fano's trick is applied, followed by a Chernoff bound. Finally, the expectations over codewords must be evaluated.

6.3.3 Converse bound for MIMO receiver

The converse bound described below is presented in [151, Theorem 9].

Theorem 6.6 (No-CSI case converse [151])**.** *Assume that there are K_a active users among K potential users, each equipped with a single antenna, and that the number of BS antennas is L. Each user has an individual codebook of size M and length n. For massive random access in MIMO quasi-static Rayleigh fading channels with no CSI and known K_a, the minimum energy per bit required to satisfy the PUPE requirement in* (3.1) *can be lower-bounded as*

$$E^*_{b,\text{no-CSI},K_a}(n, M, \epsilon) \geq \inf \frac{nP}{\log_2 M}. \tag{6.20}$$

The infimum is taken over all $P > 0$ satisfying the following condition:

$$(1-\epsilon)\log_2 M - h_2(\varepsilon) \leq \frac{LC}{K_a} - \frac{L}{K_a}\mathbb{E}_{\mathbf{X}}\left[\log_2\left|\boldsymbol{I}_{K_a} + \mathbf{X}_{\mathcal{T}}^H\mathbf{X}_{\mathcal{T}}\right|\right], \tag{6.21}$$

$$C = \min\left\{n\log_2(1+K_aP), K_aM\log_2\left(1+\frac{nP}{M}\right)\right\}, \tag{6.22}$$

where codewords $\mathbf{X}_{\mathcal{T}}$ are drawn from $\mathcal{G}_{\mathcal{CN}}(M, n, P)$. The right-hand side of (6.21) *can be further loosened to*

$$\frac{LC}{K_a} - \frac{L}{K_a}\sum_{i=0}^{\tilde{n}-1}\left(\psi(\tilde{n}-i)\log_2 e + \log_2\left(P + \frac{1}{\tilde{n}-i}\right)\right),$$

where $\tilde{n} = \min(K_a, n)$, and $\psi(\cdot)$ denotes Euler's digamma function.

The idea behind this theorem is based on Fano's inequality, followed by a mutual information analysis inspired by [79].

6.3.4 Numerical comparison of the proposed bounds

In this section, we perform a numerical analysis of the proposed bounds for the scenario taken from [152]. The results are presented in Fig. 6.1. The ML bound shows better performance for $K_a < 400$, but beyond this point, the projection-based bound demonstrates better E_b/N_0, with a gap of approximately 1 dB. However, this gap vanishes as $K_a \to n$.

This phenomenon is driven by the following: Given a similar setup, the PUPE estimate is determined by individual terms corresponding to exactly t errors, as shown in equations (6.14) and (6.17). In both equations, there is a large combinatorial term, $\binom{M-K_a}{t}$, which leads to increased energy efficiency at high K_a values when the last term for $t = K_a$ dominates in (4.2). In our case, this term starts dominating earlier in the ML bound. However, as K_a approaches the blocklength n, the ML bound once again exhibits better energy efficiency, as the projection-based bound is not applicable for $K_a > n$.

We also found that the projection-based bound provides a tighter estimation of the "region" probability compared to the ML-based bound. Specifically, with the same parameters α in (6.17)

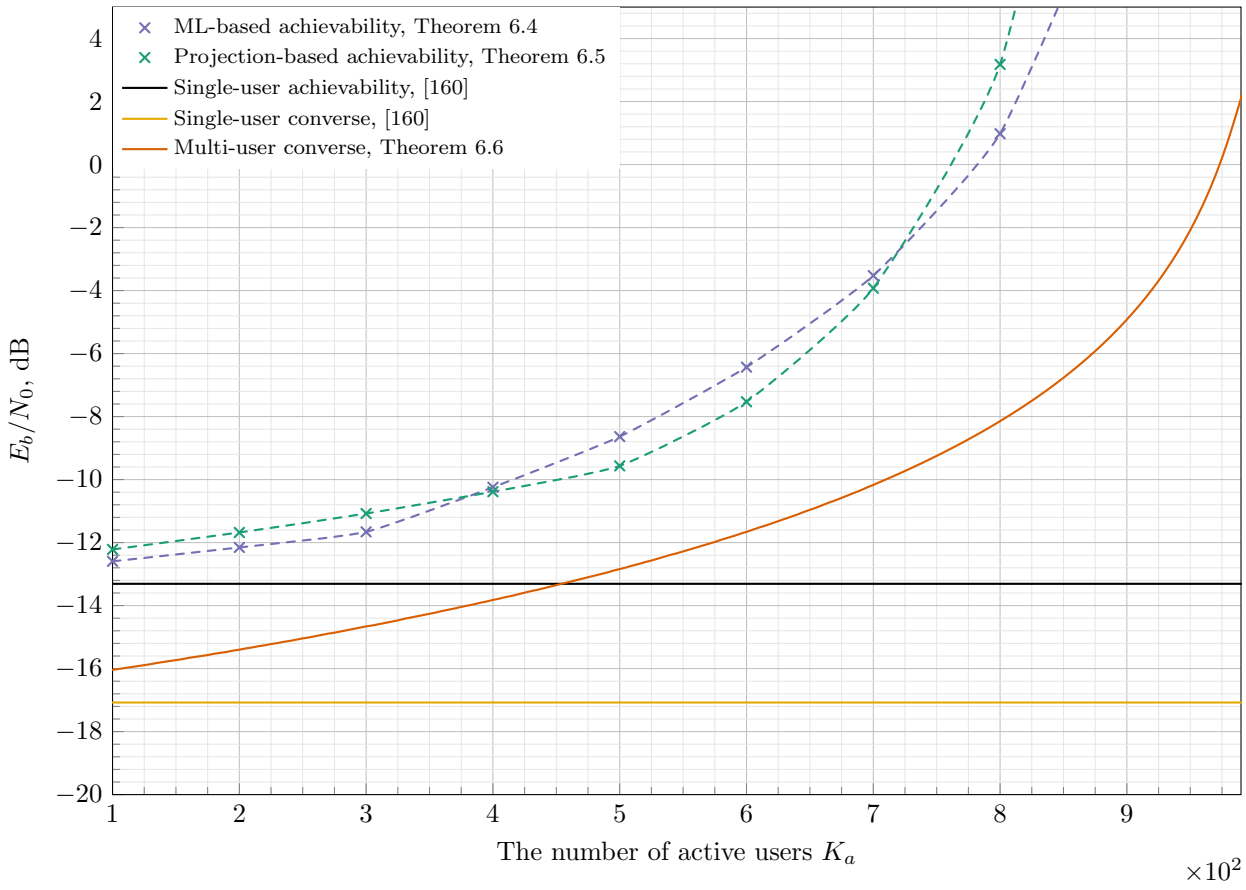

Figure 6.1: URA achievability bounds for the no-CSI setting with a frame length of $n = 1000$ channel uses and $k = 100$ information bits. The BS is equipped with $L = 64$ antennas, and the target PUPE is $P_e \leq 10^{-3}$. ML-based and projection-based achievability bounds are marked by dashed blue and green lines, respectively. As a reference, we consider a single-user converse (marked by a yellow solid line) and a multi-user converse (marked by an orange line).

and (6.14), and the same set of samples, the probability (6.15) for the ML-based bound is always smaller than its projection-based counterpart (6.18), whereas the probability (6.16) for the ML-based bound may be higher than the corresponding projection-based term (6.19).

Finally, let us once again highlight a key drawback of both bounds – the lack of a rigorous analysis of the required number of samples.

6.4 Low-complexity receiver architectures

Compared to the AWGN channel, the Rayleigh block-fading channel model assumes that channel coefficients are random (6.1). Nevertheless, this randomness can be leveraged as follows.

For the single-antenna case, there is a natural diversity in received power. There always exists a user with the maximum received power and, likely, the highest SINR. Hence, a straightforward approach is to utilize TIN-SIC. Considering the quasi-static fading model, we assume that the channel coherence time is smaller than the frame length, which naturally leads to a slotted system, or SA. We assume that the fading coefficient remains constant within a slot but changes randomly and independently from slot to slot.

In Section 5.1, we considered IRSA, where the main idea was to utilize time diversity by transmitting multiple copies of the same message in different slots. After successful decoding, the message can be subtracted from all other slots. Unfortunately, IRSA cannot be used in a fading scenario because subtraction requires channel estimation, which may have very low precision in the presence of other messages with higher received power. Hence, in the subsequent analysis (see Section 6.4.1), we consider a T-fold SA instead of IRSA. The analysis of SA is followed by an evaluation of CS-based schemes and their results.

In the multiple-antenna case, the BS has $L > 1$ receive antennas (6.1). In this case, random channel vectors (of length L) become almost orthogonal, especially when L is large. As a result, the number of simultaneously active users may increase significantly. However, channel estimation becomes more challenging. To obtain accurate channel estimates, preambles are typically embedded within each slot or coherence block. Low-complexity schemes are presented in Section 6.4.2.

6.4.1 Single-antenna case

In this section, we consider two main approaches: a T-fold SA and a CS-based approach. In the subsequent numerical analysis for the single-antenna case, we assume $k = 100$ information bits, a frame length of $n = 3 \times 10^4$ complex channel uses, and a maximum tolerable PUPE value of $\varepsilon = 0.1$.

T-fold slotted ALOHA

The main results are presented in Fig. 6.2. We begin with a TIN analysis, represented by an orange line. Similar to the AWGN case discussed in Chapter 1, this strategy is highly inefficient.

Next, a joint decoder was introduced in [159, 148, 159]. The key idea of this approach is to jointly perform decoding and channel estimation to recover a linear combination of LDPC-encoded and BPSK-modulated messages. The resulting energy efficiency of this scheme for $T = 4$ is represented by a blue line. This energy efficiency was achieved by using two slot lengths, depending on the value of K_a. For small K_a, a longer slot of length $n' = 400$ was more efficient, whereas for large K_a, a shorter slot of length $n' = 200$ was preferable, as it reduces the maximum number of simultaneous transmissions in a slot.

Having the slot length n' and the number of simultaneously active users in a slot, r, we can evaluate the PUPE within each slot for a given T. Let us denote this PUPE value as $p_{r,k,n'}$, where k is the number of information bits. Note that $p_{r,k,n'}$ can represent either a theoretical bound or the slot-level performance of a practical scheme. The final goal is to evaluate the PUPE for the entire frame.

Remark 6.1 (Evaluating per-frame PUPE given per-slot PUPE)**.** *Given $p_{r,k,n'}$, the per-frame PUPE p_e can be evaluated as follows. Assuming that each user selects a transmission slot randomly, the PUPE is given by:*

$$p_e = 1 - \sum_{r=1}^{T} \left(1 - p_{r,k,n'}\right) \binom{K_a - 1}{r - 1} \left(\frac{1}{L}\right)^{r-1} \left(1 - \frac{1}{L}\right)^{K_a - r}. \tag{6.23}$$

To evaluate the achievable energy efficiency of this scheme, we applied the achievability bound from Theorem 6.2 to the slot, followed by an assessment of the overall frame performance using eq. (6.23). The result is depicted by a dashed green line, representing the minimum E_b/N_0 over all possible slot lengths for each value of K_a.

During numerical analysis, we observed that the first codeword successfully decoded by the joint decoder typically has the highest received energy. Furthermore, given a known channel coefficient, this codeword can also be decoded using a TIN algorithm. This insight led to a modification of the scheme, replacing the joint decoder with a TIN-SIC decoder [168], where LDPC codes were substituted with polar codes due to their superior error-correcting performance for short blocks.

We found that the polar code-based scheme exhibits better energy efficiency (black line). We employed a CRC-aided list decoding algorithm and did not use preambles for CSI estimation. Instead, assuming the transmitted signal is BPSK-modulated, we leveraged the Gaussian mixture structure of the received signal to retrieve CSI via a clustering algorithm. Another challenge was the presence of falsely detected codewords. To mitigate this, we increased the CRC size to 21 bits. The SIC step was implemented using the OMP algorithm (see Appendix B.1.4).

Table 6.1: Optimal slot count L (and outer code rate $R_O = \frac{k}{L \log_2 Q}$) for RCB

K_a	$t=0$	$t=1$	$t=2$	$t=3$	$t=4$	$t=5$
50	12	14	15	16	18	19
100	14	15	17	18	19	20
150	15	16	18	19	20	21
200	16	17	19	20	21	22
250	16	18	19	21	22	23
300	17	18	20	21	22	24

Coded compressed sensing

In this section, we consider the CCS scheme presented in Section 5.3, now evaluated for a Rayleigh fading channel. Given the channel coherence time assumption, CCS also operates under a slotted transmission model. For each slot, we consider an inner codebook of size $Q = 2^{15}$. Larger inner codebook sizes improve energy efficiency but require significant computational resources for numerical evaluation.

As previously discussed, we consider a setup with $k = 100$ information bits, a frame length of $n = 3 \times 10^4$, and a slot length of $n' = n/L$, where L is the number of slots. Codewords of the inner code are generated with an i.i.d. uniform distribution over the (complex) power shell. We decode the inner code using OMP[146] and its MMSE-based extension[169]. Since OMP is a sequential algorithm, the number of output codewords equals the number of decoding steps, denoted as K_0.

Numerical results are presented in Fig. 6.3, where we plot energy efficiency as a function of the active user count K_a for different schemes. Energy efficiency is defined as the minimum E_b/N_0 over K_0, L, and other scheme-specific parameters/constraints, such that the PUPE satisfies $P_e \leq 0.1$ and the FAR satisfies $P_f \leq 10^{-3}$. These additional parameters/constraints include the maximum average number of decoding paths for the t-tree code (6.24) and the user prefix size for the RS case.

Let us start with the CCS-RCB analysis as presented in Theorem 5.3. The orange lines in the figure correspond to the CCS-RCB for $t = 0, \ldots, 5$. Recall that the parameter t represents the number of erasures in the outer-code codewords (see eq.(5.17) and the illustrative example presented inFig. 5.11). For $t = 5$, there is a significant improvement in E_b/N_0 (more than 10 dB for $K_a = 50$) compared to the $t = 0$ case. When the number of active users is small, the scheme with $t = 5$ demonstrates better energy efficiency than a T-fold SA with polar codes from [168], which is known as the best practical solution for the quasi-static fading channel with a single antenna at the receiver.

We also observe that higher values of t allow for better energy efficiency but support a smaller maximum number of simultaneously active users. The main reason for this behavior is the FAR constraint. Indeed, as the number of active users grows, the FAR also increases (5.19). Moreover, the higher the t values, the faster the FAR grows. Suppressing the FAR requires more slots, which necessitates a more robust (smaller p_m) inner code to guarantee low P_e (5.18). Ensuring more robust inner code performance under shorter slot lengths leads to inner code failure at some $\tilde{K}_a$, and this $\tilde{K}_a$ is smaller for higher values of t. For $t = 5$, $\tilde{K}_a \approx 410$, and for $t = 0$, $\tilde{K}_a \approx 540$. The optimal slot count values are presented in Table 6.1 for different values of t and K_a.

Table 6.2: Optimal slot count L (and outer code rate $R_O = \frac{k}{L \log_2 Q}$) achievability bound for the code from Corollary E.1. The maximum average number of paths $\mathbb{E}[|V_l|] \leq v^\star = 2^{10}$.

K_a:	50	100	150	200
$t = 0$	13 (0.513)	14 (0.476)	15 (0.444)	16 (0.417)
$t = 1$	27 (0.247)	34 (0.196)	37 (0.180)	–
$t = 2$	46 (0.145)	59 (0.145)	–	–
$t = 3$	66 (0.101)	–	–	–
$t = 4$	85 (0.078)	–	–	–
$t = 5$	100 (0.067)	–	–	–

Table 6.3: Greedy bit allocation results for $v^\star = 2^{10}$.

K_a	Information bits pattern
	$t = 0$
50	$\mathbf{b} = [15\ 11\ 9\ 8\ 9\ 8\ 9\ 8\ 9\ 8\ 6\ 0\ 0]$
100	$\mathbf{b} = [15\ 10\ 8\ 8\ 7\ 8\ 8\ 8\ 8\ 8\ 8\ 4\ 0\ 0]$
150	$\mathbf{b} = [15\ 9\ 7\ 7\ 8\ 7\ 7\ 8\ 7\ 7\ 8\ 7\ 3\ 0\ 0]$
200	$\mathbf{b} = [15\ 8\ 7\ 7\ 7\ 6\ 7\ 7\ 7\ 7\ 7\ 7\ 6\ 2\ 0\ 0]$
	$t = 1$
50	$\mathbf{b} = [10\ \underbrace{4 \ldots 4}_{22}\ 2\ 0\ 0\ 0]$
100	$\mathbf{b} = [10\ \underbrace{3 \ldots 3}_{30}\ 0\ 0\ 0]$
150	$\mathbf{b} = [10\ 2\ 2\ 2\ 2\ 2\ 3\ 2\ 3\ 2\ \underbrace{3 \ldots 3}_{7}\ 2\ \underbrace{3 \ldots 3}_{15}\ 2\ 0\ 0\ 0]$

Next, we evaluated the converse bound derived in Theorem E.1. We determined the minimum E_b/N_0 (over K_0 and L) such that (E.3) holds for different values of K_a. Additionally, we included the converse bound from Theorem 6.3 as a reference. The CCS converse (green line in Fig. 6.3) starts below the bound from Theorem6.3 (magenta line in Fig. 6.3) and intersects it at $K_a \approx 300$. We consider the overall converse bound to be the region above both bounds.

To evaluate the achievability bound for the t-tree code from Corollary E.1, the minimum E_b/N_0 search procedure must take the maximum average number of decoding paths ($v^\star$) into account as follows: minimize E_b/N_0, subject to

$$\mathbb{E}[|V_l|] \leq v^\star,\ l \in [L],\ \sum_{l=1}^{L} b_l = k,\ P_e < 0.1,\ P_f < 10^{-3}, \tag{6.24}$$

where $\mathbb{E}[|V_l|]$ is the sum of (E.5) and (E.6) in accordance with (E.4). To solve this problem, an optimization over K_0 and n' was performed jointly with a greedy information bits allocation, assigning the maximum number of information bits at each subsequent slot l while keeping the constraint $\mathbb{E}[|V_l|] \leq v^\star$. If the total number of allocated bits $\sum_{l=1}^{L} b_l < k$, we assume $P_e = 1$.

The resulting energy efficiency is presented in Fig. 6.3 by blue lines for $t = 0, \ldots, 5$ and $v^\star = 2^{10}$.

Table 6.4: System parameters for the RS-based solution.

K_a	K_0	L	b_p	R_{RS}	b_C
50	53	53	9	0.3208	19
100	109	52	9	0.3269	19
150	167	45	9	0.3778	19
200	229	41	9	0.4146	19

For $t = 0$, the result is very close to the CCS-RCB, but the difference becomes dramatic for $t = 5$. The constraint $\mathbb{E}[|V_l|] \leq v^\star = 2^{10}$ requires significantly more slots for $t > 0$ compared to the CCS-RCB. The outer coding rates are presented in Table 6.2, and the optimal bits allocation is shown in Table 6.3. For $t = 5$, one needs $L = 100$ slots for $K_a = 50$, which is much higher compared to the CCS-RCB ($L = 19$ for $K_a = 50$). A larger slot count makes it impossible to construct the t-tree code with $t = 5$ for $K_a > 60$.

Finally, we evaluated the RS-based practical solution. For the RS scheme, we found the minimum E_b/N_0 over K_0, L and the prefix size b_p (bits). We also adjusted the number of bits b_C to suppress falsely detected messages, but during the simulations, we simply evaluated b_C and corrected the E_b/N_0. The resulting energy efficiency of the RS-based scheme is presented by green crosses in Fig. 6.3. The RS code parameters are presented in Table 6.4, where R_{RS} does not take the CRC into account (the total coding rate is $R = R_{\mathrm{RS}} \cdot R_{\mathrm{CRC}}$, where $R_{\mathrm{CRC}} = k/(k + b_C)$). Increasing K_a requires both a longer prefix and a larger RS code length. These two requirements are contradictory because the RS code is constructed over the field of size $q = Q/2^{b_p}$, and $q > L$. On the other hand, increasing the prefix length decreases q, making it impossible to increase the number of slots while fitting $k = 100$ bits into the frame. The slot count increase also weakens the inner-code performance. We note that the practical RS-based scheme operates over a larger outer code length compared to the CCS-RCB.

6.4.2 Multiple-antenna case

For the multiple-antenna receiver, we consider a scenario with a frame length of $n = 3200$. The channel coherence time is assumed to be at least as long as the proposed frame length. The BS is equipped with $L = 50$ antennas. To maintain consistency with results presented in the literature, we also consider an error rate constraint of $P_e \leq 0.025$.

In this setup, employing multiple antennas at the receiver provides two key benefits. First, with L antennas, receive beamforming can be used to enhance the received signal energy. Second, a large number of antennas introduces a proportionally larger number of degrees of freedom in the communication system [34], allowing for a significantly higher number of simultaneously active users.

The most challenging aspect of the receiver is channel estimation. Since each user has L fading coefficients, the previously described clustering-based approach becomes inefficient. A straightforward method is to allocate a portion of the available resources for channel learning and then use the resulting channel estimate to perform a TIN step, followed by SIC. This approach has been

implemented in [154]. Pilot detection is closely related to the CS problem. In the case of multiple receiver antennas, this problem is known as the MMV problem. The energy efficiency of the pilot-based scheme is shown in Fig. 6.4 as a light green line. As a reference, we have re-evaluated the achievability bounds presented in Fig. 6.1. The current state-of-the-art solution [158] integrates activity detection, single-user coding, pilot- and temporally decision-aided iterative channel estimation, decoding, MMSE estimation, and SIC. The energy efficiency of this scheme is given by a magenta line.

Notably, there is a drastic improvement in energy efficiency compared to the single-antenna case. The resulting energy efficiency improves by approximately $10 \times \log_{10} L$ dB compared to the single-antenna case, due to receive beamforming. Additionally, the number of simultaneously active users increases: up to $K_a \approx 800$ at a frame length of $n = 3200$, compared to $K_a \approx 600$ at $n = 3 \times 10^4$.

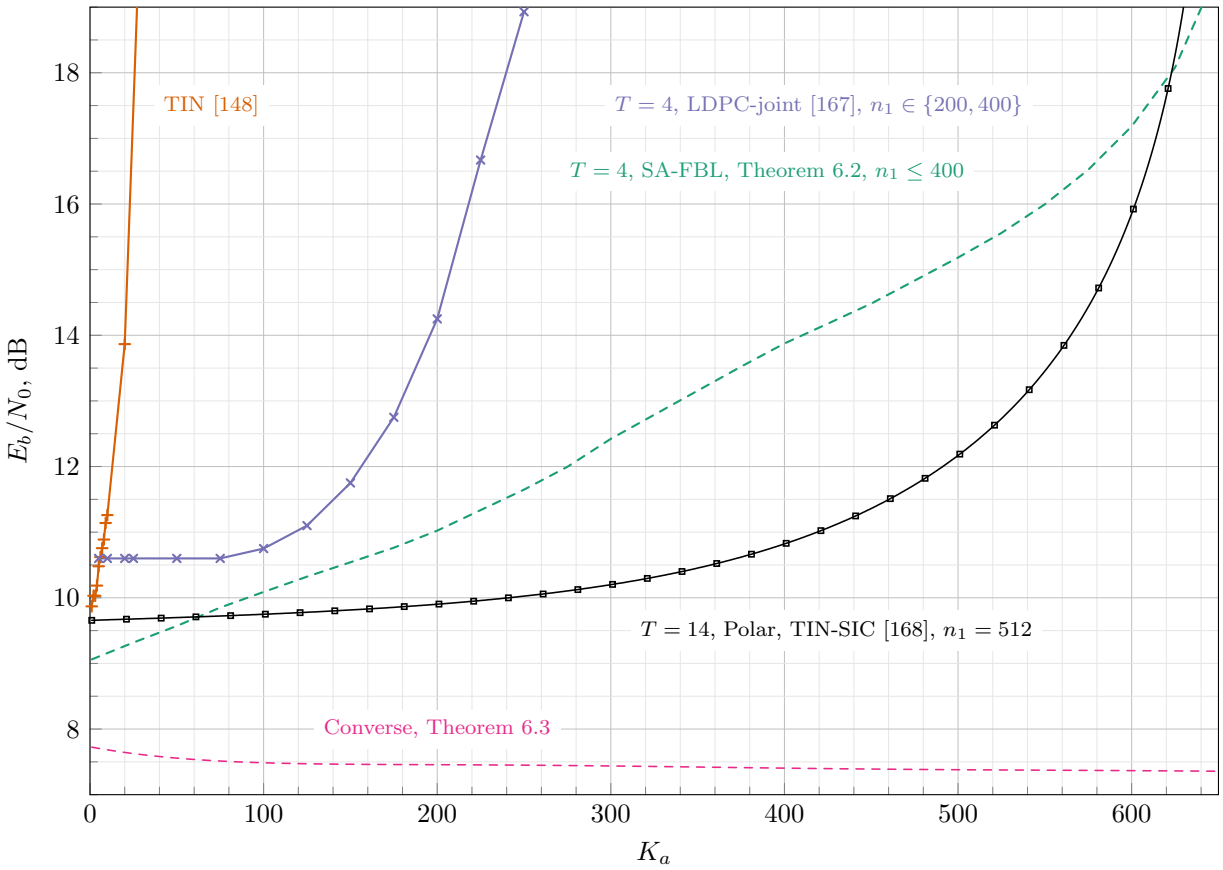

Figure 6.2: Minimum E_b/N_0 required to achieve the PUPE below 10% ($\varepsilon = 0.1$) as a function of the active user count. Theoretical bounds are represented by dashed lines, while practical schemes are shown with solid lines. The green dashed line corresponds to the achievability bound from Theorem 6.2, where the slot length n' is chosen optimally using (6.23) for $T = 4$ and $n' \leq 400$. The dashed magenta line corresponds to the converse bound from Theorem 6.3. For practical schemes, we consider TIN, $T = 4$-fold SA with a joint decoder [148] ($n' = 200$ and 400 depending on K_a), and the best-performing TIN-SIC scheme based on polar codes [168].

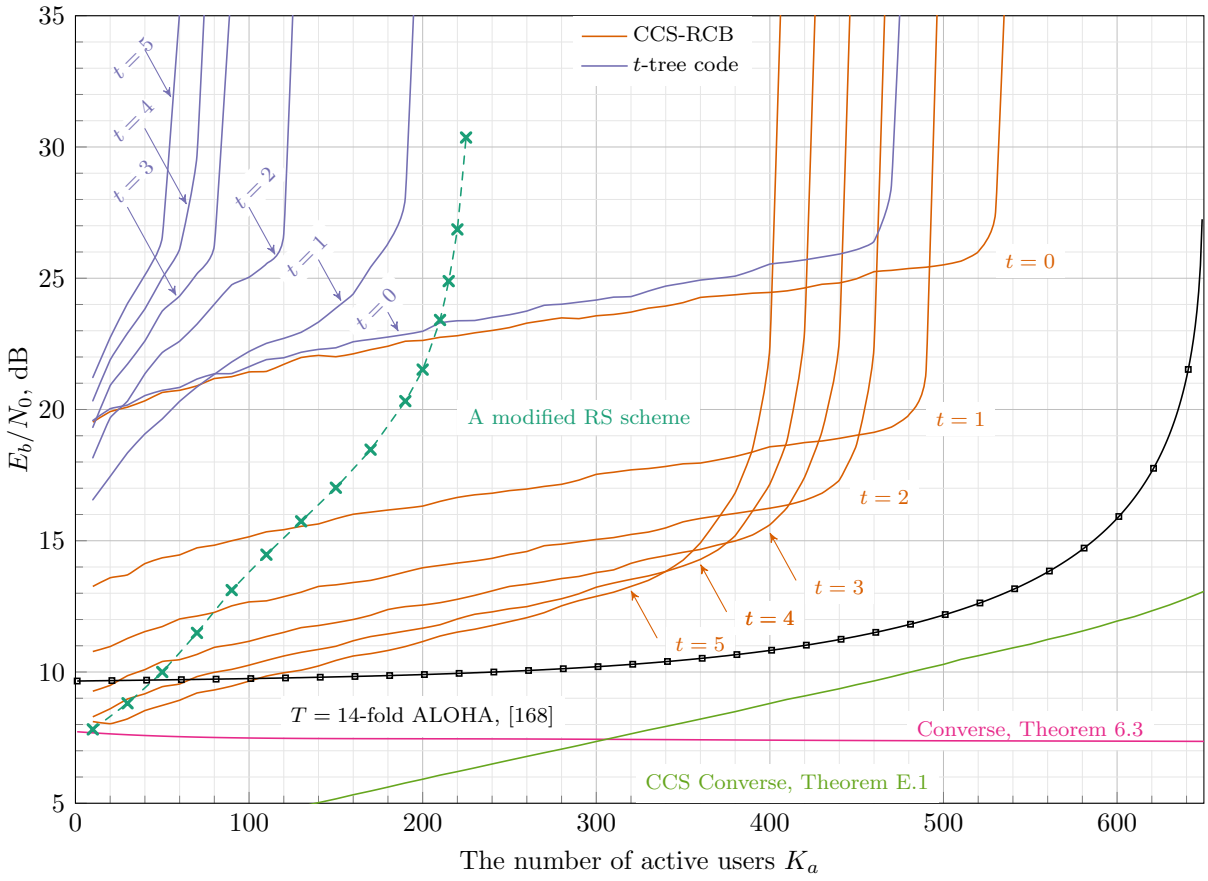

Figure 6.3: Numerical results for the single-antenna quasi-static Rayleigh fading channel. The maximum tolerable PUPE is $P_e \leq 10^{-1}$, and FAR is $P_f \leq 10^{-3}$. Parameters K_0 and n' are chosen to minimize the required E_b/N_0. The following curves are presented: CCS-RCB (see Theorem 5.3); t-tree code bounds for $t = 0, \ldots, 5$ (see Corollary E.1), with the maximum average number of decoding paths equal to $\mathbb{E}[|V_l|] \leq v^\star = 2^{10}$; a modified RS scheme from Section 5.3.4 (parameters taken from Table 6.4). A 14-fold SA practical scheme from [168] is added as a reference. Additionally, two converse bounds are provided: one from Theorem 6.3 and another for the CCS scheme, based on Fano's inequality (Theorem E.1).

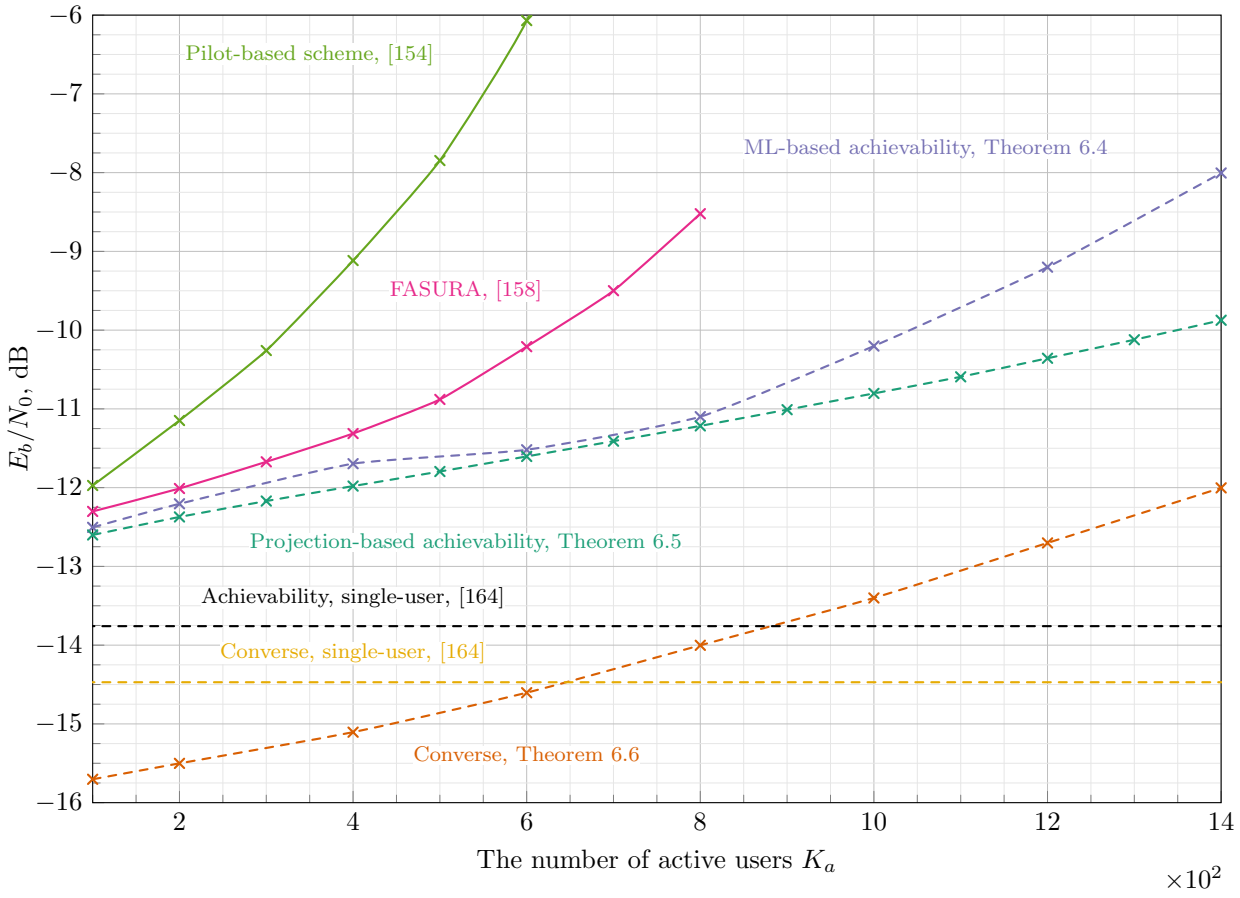

Figure 6.4: Minimum E_b/N_0 required to achieve an error rate of $P_e \leq 0.025$ as a function of the active user count. Theoretical bounds are represented by dashed lines, while practical schemes are shown with solid lines. As a reference, single-user achievability and converse bounds are included, along with ML-based and projection-based multi-user bounds from Theorems 6.4 and 6.5. The frame length is $n = 3200$, with $k = 100$ information bits. The BS is equipped with $L = 50$ antennas.

Chapter 7

Open problems

In this monograph, we have presented a comprehensive overview of the URA problem in noisy channels. The URA model provides an information-theoretic framework that accounts for FBL effects and energy efficiency. This framework enables the derivation of fundamental limits, facilitates the comparison and optimization of existing uncoordinated access schemes, and supports the development of new ones (e.g. coded CS schemes). Despite the substantial progress made and the numerous results obtained, several open problems remain:

- *Are there any better "good" regions which improve the achievability bound based on the Fano's trick?* Gordon's inequality seems to be useful as it improves the asymptotic bound [87]. Can this method improve finite-length results?
- *Is it possible to close the gap between achievability and converse bounds in the asymptotic regime (see Appendix F)?* It seems counterintuitive that for high user density per channel use, the current achievability bounds coincide with the converse bound, whereas for low user density, we observe a gap of approximately 0.8 dB.
- *Is there any polynomial complexity coding scheme with non-linear user codes that outperforms preamble-based approaches?* We note preamble-based approach is one of the methods to create non-linear codes. Let us briefly recall that the same codebook is constructed as follows: there are L linear codes $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_L$. The user randomly chooses the codebook and sends the codebook index and the codeword to the channel. The clear drawback of this scheme is the dependence on the preamble length on the number of active users and the required P_e.
- *Are there better codes for the A-channel and A-channel with errors?* The solutions from the literature are based either on the tree-coding or RS codes – the classes of codes suitable for the list recovery problem. In particular, it is interesting whether it is possible to modify classical LDPC or polar codes to work in this setup. Note the SCL decoder seems to be very close to the tree-code decoder, namely it recovers next symbol based on the results for the previous symbols. But there are no results on the use of polar codes for the A-channel.
- *Are there any other principles to construct sensing matrices with polynomial decoding algorithms?* Standard CS decoding algorithms check all correlations which is impossible in high-

dimensions. One of the ideas behind this question in the CHIRRUP approach from [143], which utilizes binary chirps which are simply codewords in the second order Reed-Muller (RM) code. Can we go further while preserving polynomial complexity?

- *Is it possible to obtain closed-form expression of the bounds for the fading MACs with MIMO receiver (and avoid the expectation over the codebook)?* The expected values need to be calculated in a precise manner, which is computationally expensive and probably not feasible.

- *Are there good low-complexity schemes for the MIMO URA setup which do not perform channel estimation and do not use pilots?*

- *General fading channels.* One more open problem is the design of URA schemes for more general fading channels. Most schemes are designed for the quasi-static Rayleigh fading channel and rely on the quasi-static property. However, quasi-static fading is rarely observed in practice, especially over the relatively long blocklengths considered in URA.

- *Asynchronous URA.* All the schemes and bounds presented in this monograph assume perfect per-frame synchronization. At the same time this condition is extremely difficult to fulfill as we consider low-power autonomous devices. The question of asynchronous MAC was considered in the literature [170], it is important to propose low-complexity schemes and bounds for this case. We mention several papers that may serve as useful starting points [171, 172, 173].

Appendix A

Minimum mean squared error estimation

Consider the following Markov chain

$$\mathbf{x} \to \mathbf{y} \to \hat{\mathbf{x}},$$

where $\hat{\mathbf{x}} = \hat{\mathbf{x}}(\boldsymbol{y})$ is the estimate of $\mathbf{x}$.

The goal is to minimize the MSE

$$\mathsf{MSE} = \mathbb{E}[(\hat{\mathbf{x}} - \mathbf{x})^2].$$

The MMSE estimator is defined as the estimator that achieves the minimal MSE:

$$\hat{\boldsymbol{x}}_*(\boldsymbol{y}) = \arg\min_{\boldsymbol{x}} \mathsf{MSE}.$$

It is known that, for a given $\boldsymbol{y}$,

$$\hat{\boldsymbol{x}}_*(\boldsymbol{y}) = \mathbb{E}[\mathbf{x}|\boldsymbol{y}].$$

In many cases, finding an analytical expression for the MMSE estimator is not feasible. Additionally, the numerical evaluation of the conditional expectation is computationally expensive. Thus, it is reasonable to consider a specific class of estimators – linear estimators, i.e.,

$$\hat{\boldsymbol{x}}^{(l)}(\boldsymbol{y}) = \boldsymbol{W}\boldsymbol{y} + \boldsymbol{b}.$$

The linear estimator is optimal if it minimizes the MSE within the class of linear estimators, i.e., solving

$$(\boldsymbol{R}_*, \boldsymbol{b}_*) = \arg\min_{\boldsymbol{W},\boldsymbol{b}} \mathsf{MSE},$$

where

$$\hat{\boldsymbol{x}}^{(l)}(\boldsymbol{y}) = \boldsymbol{W}\boldsymbol{y} + \boldsymbol{b}.$$

The optimal pair is given by

$$\boldsymbol{W}_* = \mathrm{Cov}\left[\mathbf{x}, \mathbf{y}\right] \left(\mathrm{Cov}\left[\mathbf{y}, \mathbf{y}\right]\right)^{-1}, \tag{A.1}$$

$$\boldsymbol{b}_* = \mathbb{E}\left[\mathbf{x}\right] - \boldsymbol{W}_* \mathbb{E}\left[\mathbf{y}\right], \tag{A.2}$$

where $\mathrm{Cov}\left[\mathbf{a}, \mathbf{b}\right] = \mathbb{E}\left[\left(\mathbf{a} - \mathbb{E}\left[\mathbf{a}\right]\right)\left(\mathbf{b} - \mathbb{E}\left[\mathbf{b}\right]\right)^T\right]$.

In what follows, we refer to the optimal linear estimator as the LMMSE estimator, given by

$$\hat{\boldsymbol{x}}_*^{(l)}(\boldsymbol{y}) = \boldsymbol{W}_* \boldsymbol{y} + \boldsymbol{b}_*.$$

The MSE value achievable by the LMMSE estimator is given by

$$\mathsf{MSE}_*^{(l)} = \mathrm{tr}\boldsymbol{\Sigma},$$

where

$$\boldsymbol{\Sigma} = \mathrm{Cov}\left[\mathbf{x}, \mathbf{x}\right] - \mathrm{Cov}\left[\mathbf{x}, \mathbf{y}\right] \left(\mathrm{Cov}\left[\mathbf{y}, \mathbf{y}\right]\right)^{-1} \mathrm{Cov}\left[\mathbf{y}, \mathbf{x}\right].$$

Now consider a linear observation process

$$\mathbf{y} = \boldsymbol{A}\mathbf{x} + \mathbf{z},$$

where $\mathbf{x}$ and $\mathbf{z}$ are independent, $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_n)$, $\mathbb{E}[\mathbf{x}] = \boldsymbol{\mu}$ and $\mathrm{Cov}\left[\mathbf{x}, \mathbf{x}\right] = \boldsymbol{\Gamma}$. We have

$$\mathrm{Cov}\left[\mathbf{x}, \mathbf{y}\right] = \Gamma A^T,$$
$$\mathrm{Cov}\left[\mathbf{y}, \mathbf{y}\right] = (\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^T + \boldsymbol{I}_n),$$

and thus,

$$\hat{\boldsymbol{x}}_*^{(l)}(\boldsymbol{y}) = \boldsymbol{\Gamma} A^T (\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^T + \boldsymbol{I}_n)^{-1} (\boldsymbol{y} - \boldsymbol{A}\boldsymbol{\mu}) + \boldsymbol{\mu}.$$

Remark A.1. *If $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Gamma})$, then the MMSE estimator coincides with the LMMSE estimator presented above.*

Appendix B

Standard CS codebooks and decoders

Let us start with a noisy CS problem, also known as a sparse linear regression problem. The main task is to recover the sparse vector $\boldsymbol{u}$ given n noisy linear measurements of the form $\boldsymbol{a}_i^T \boldsymbol{u} + \mathsf{z}_i$, $i \in [n]$. That is, we have

$$\mathbf{y} = \boldsymbol{A}\boldsymbol{u} + \mathbf{z}, \quad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_n), \tag{B.1}$$

where $\boldsymbol{u} \in \mathbb{R}^M$, and $\|\boldsymbol{u}\|_0 = t$. Recall that $\boldsymbol{A} \in \mathbb{R}^{n \times M}$ is referred to as the *sensing matrix*.

In most cases throughout this monograph, we are primarily interested in recovering only the support of $\boldsymbol{u}$, defined as

$$\operatorname{supp}(\boldsymbol{u}) = \{i : u_i \neq 0\}.$$

Such problems are known as *support recovery problems*.

In this monograph, we encounter two instances of problem (B.1), namely:

- Case A (GMAC): $\boldsymbol{u} \in \{0,1\}^M$, $\|\boldsymbol{u}\|_0 = t$;
- Case B (Fading[1] MAC): $\boldsymbol{u} \in \mathbb{R}^M$, $\|\boldsymbol{u}\|_0 = t$;

Remark B.1. *The support recovery problem is not necessarily simpler than the general CS problem. The reasoning is as follows: suppose we know the support $\mathcal{I}$, then we can apply MMSE or linear MMSE estimators (see Appendix A) to obtain an estimate $\hat{\boldsymbol{u}}_{\mathcal{I}}$ that minimizes the mean squared error, leading to a good overall estimate $\hat{\boldsymbol{u}}$.*

In this chapter, we describe standard CS approaches, focusing on decoding algorithms and codebook design strategies for cases where M is small. In particular, we consider $M \leq 2^{15}$. The elements of these codebooks serve as preambles in practical schemes, as discussed in Section 5.

[1]Strictly speaking, we should write $\boldsymbol{u} \in \mathbb{C}^M$, but we focus on the real case in this chapter. Note that the approaches remain largely the same.

B.1 Decoding algorithms

B.1.1 Non-negative least squares

Since we are dealing with additive white Gaussian noise (AWGN) $\mathbf{z}$, one possible approach to solving problem (B.1) is to find a vector $\boldsymbol{u}$ that minimizes the residual norm:

$$\hat{\boldsymbol{u}} = \arg\min_{\boldsymbol{u}} \|\mathbf{y} - \boldsymbol{A}\boldsymbol{u}\|^2 . \tag{B.2}$$

The problem above is convex and can be solved using standard algorithms, either analytically or via gradient-based methods. However, the solution (B.2) may contain negative elements in $\hat{\boldsymbol{u}}$, which can be undesirable in some scenarios.

To enforce non-negativity, the solution can be obtained through a non-negative least squares (NNLS) approach, which imposes an additional constraint:

$$\hat{\boldsymbol{u}} = \arg\min_{\boldsymbol{u}\geq\mathbf{0}} \|\mathbf{y} - \boldsymbol{A}\boldsymbol{u}\|^2 ,$$

where $\boldsymbol{u} \geq \mathbf{0}$ means that $u_i \geq 0$ for all $i \in [M]$. Since $\|\boldsymbol{x}\|^2 = \boldsymbol{x}^T\boldsymbol{x}$, the problem above can be reformulated equivalently as

$$\hat{\boldsymbol{u}} = \arg\min_{\boldsymbol{u}\geq\mathbf{0}} \left(\frac{1}{2}\boldsymbol{u}^T\boldsymbol{Q}\boldsymbol{u} + \boldsymbol{c}^T\boldsymbol{u}\right), \quad \boldsymbol{Q} = \boldsymbol{A}^T\boldsymbol{A}, \quad \boldsymbol{c} = -\boldsymbol{A}^T\boldsymbol{y},$$

which is a convex optimization problem (see [174]).

B.1.2 LASSO: Sparse linear regression

CS can be viewed as a linear regression problem. To enforce sparsity in the solution, ℓ_1-regularization can be applied:

$$\hat{\boldsymbol{u}} = \arg\min_{\boldsymbol{u}} \|\mathbf{y} - \boldsymbol{A}\boldsymbol{u}\|^2 + \lambda \|\boldsymbol{u}\|_1 .$$

This regularized linear regression approach is known as LASSO [175]. The parameter λ controls the sparsity of the solution.

B.1.3 Approximate message passing

AMP [176] is a family of low-complexity iterative algorithms used to solve a wide range of problems, including(B.1).

In this section, we provide a simplified description from [136], listed in Algorithm 4. In the algorithm description, subscripts indicate iterations, $\langle\cdot\rangle$ denotes the average of a vector, $\eta(\cdot)$ is a nonlinear denoising function (see, e.g., [136]), and $\eta'(\cdot)$ represents the component-wise derivative.

Algorithm 4 Approximate message passing (AMP) for (B.1) problem.

Input: $\boldsymbol{A}$, $\boldsymbol{y}$ ▷ Design matrix, observed vector
$\tilde{\boldsymbol{y}}_0 = \boldsymbol{y}$ ▷ Soft residual signal
$\hat{\boldsymbol{u}}_0 = 0$ ▷ Soft activity pattern estimate
for $t = 1, \ldots, T$ **do** ▷ Run T iterations
    $\hat{\boldsymbol{u}}_{t+1} = \eta_t \left(\hat{\boldsymbol{u}}_t + \boldsymbol{A}^T \tilde{\boldsymbol{y}}_t \right)$
    $\tilde{\boldsymbol{y}}_{t+1} = \boldsymbol{y} - \boldsymbol{A}\hat{\mathbf{u}}_{t+1} + \frac{M}{n} \tilde{\boldsymbol{y}} \left\langle \eta' \left(\hat{\boldsymbol{u}}_t + \boldsymbol{A}^T \tilde{\boldsymbol{y}}_t \right) \right\rangle$
end for
return $\hat{\boldsymbol{u}}$

The main idea of the algorithm is as follows. Consider the prior distribution of the elements of $\boldsymbol{u}$ as

$$p(\boldsymbol{u}) = \prod_{i=1}^{M} p(\boldsymbol{u}_i) = \prod_{i=1}^{M} \exp\{-\beta |\boldsymbol{u}_i|\}$$

Given the Gaussian channel, let us consider the posterior distribution μ:

$$\mu(\mathbf{u}) = \frac{1}{Z} \prod_{i=1}^{n} \exp\left\{ -\frac{\beta}{2} \left(\boldsymbol{y}_i - (\boldsymbol{A}\mathbf{u})_i \right)^2 \right\} \prod_{i=1}^{M} \exp\{-\beta |\boldsymbol{u}_i|\},$$

where Z is a normalizing constant.

To compute the posterior probabilities, an iterative sum-product algorithm on a dense graph can be used [176]. The idea of AMP is to apply the large system limit ($M \to \infty$) and sharpen the prior probability distribution as $\beta \to \infty$ [176].

Note that [176, Section V] also describes the iterative algorithm that solves a LASSO problem.

The idea of AMP for Case B of the problem (B.1) is similar and is described in [177].

B.1.4 Orthogonal matching pursuit

The idea of OMP is based on a SIC approach and the near-orthogonal property of the columns of random sensing matrices. The observed vector $\boldsymbol{y}$ in (B.1) (case B) is a linear combination of the columns of the sensing matrix $\boldsymbol{A}$. Thus, if these columns are nearly orthogonal, the concept of an energy detector from CDMA can be applied here. OMP is an iterative algorithm. At each iteration, it identifies the column of the design matrix that is most correlated with the received signal, adds the corresponding column to the set of *active* columns, and then performs cancellation of the already detected columns. The algorithm is specified in [146] and listed in Algorithm 5.

The weight coefficients estimate should be calculated jointly for the entire set of detected columns $\hat{\mathcal{T}}$, as the columns of the design matrix $\boldsymbol{A}$ may not be orthogonal. When calculating the estimate of weights $\hat{\boldsymbol{h}}$, we treat the set of design matrix columns as another matrix.

Note that the OMP is also suitable for case A in (B.1), where $\hat{\boldsymbol{u}}_i = 1$ for $i \in \mathcal{D}$, without the need to calculate a pseudoinverse.

Algorithm 5 Orthogonal matching pursuit for (B.1) problem.

Input: $\boldsymbol{A}$, $\boldsymbol{y}$, t ▷ Sensing matrix, observed vector, output list size
$\tilde{\boldsymbol{y}} = \boldsymbol{y}$ ▷ Residual signal
$\mathcal{D} = \varnothing$ ▷ Estimate of $\operatorname{supp}(\boldsymbol{u})$
while $|\mathcal{D}| < t$ **do** ▷ Run until t candidates detected
  $i = \arg\max_{j \in [M] \setminus \mathcal{D}} \left| \boldsymbol{y}^T \boldsymbol{a}_i \right|$ ▷ Find the highest correlation coefficient
  $\mathcal{D} = \mathcal{D} \bigcup \{i\}$ ▷ Update the detected columns set
  $\hat{\boldsymbol{u}}_{\mathcal{D}} = \boldsymbol{A}_{\mathcal{D}}^{\dagger} \boldsymbol{y}$ ▷ Get weights estimate by pseudoinverse
  $\tilde{\boldsymbol{y}} = \boldsymbol{y} - \boldsymbol{A}_{\mathcal{D}} \hat{\boldsymbol{u}}_{\mathcal{D}}$ ▷ update the residual signal
end while
return $\hat{\mathcal{D}}$, $\hat{\boldsymbol{u}}$

B.2 Codebook design

All the algorithms described above calculate correlations and thus rely on codebooks with good correlation properties. In this section, we describe several approaches to constructing such codebooks.

Our goal is to design a sensing matrix $\boldsymbol{A}$ such that $|\boldsymbol{a}_i^T \boldsymbol{a}_j|$, for $i, j \in [M]$ and $i \neq j$, is small. Recall also the energy constraint $\|\boldsymbol{a}_i\|^2 \leq Pn$, for $i \in [M]$. In this section, we focus on $P = 1$, i.e., the columns of $\boldsymbol{A}$ have unit norms. The case of arbitrary P is obtained by multiplying the codebooks above by $\sqrt{P}$.

B.2.1 Gaussian codebook

The most popular approach is to utilize the Gaussian codebook, i.e., each element $a_{i,j}$ of $\boldsymbol{A}$ is sampled i.i.d. from $\mathcal{N}(0, 1-\varepsilon)$ for some $\varepsilon > 0$. We need ε to satisfy the power constraint. Such codebooks have been proven to be effective for large n. Indeed, for any $i, j \in [M]$, $i \neq j$, we have

$$\frac{1}{n} \boldsymbol{a}_i^T \boldsymbol{a}_j \xrightarrow{P} 0.$$

To be precise, we can use the following equality:

$$\boldsymbol{a}_i^T \boldsymbol{a}_j = \left\| \frac{\boldsymbol{a}_i + \boldsymbol{a}_j}{2} \right\|^2 - \left\| \frac{\boldsymbol{a}_i - \boldsymbol{a}_j}{2} \right\|^2,$$

where the vectors $\boldsymbol{\eta}_1 = (\boldsymbol{a}_i + \boldsymbol{a}_j)/2$ and $\boldsymbol{\eta}_2 = (\boldsymbol{a}_i - \boldsymbol{a}_j)/2$ are independent. We note that $\boldsymbol{\eta}_1, \boldsymbol{\eta}_2 \sim \mathcal{N}(\boldsymbol{0}, (1-\varepsilon)\boldsymbol{I}_n)$.

Let us consider the variance:

$$\operatorname{Var}\left[\frac{1}{n} \boldsymbol{a}_i^T \boldsymbol{a}_j \right] = \frac{1}{n^2} \left(\operatorname{Var}\left[\|\boldsymbol{\eta}_1\|^2 \right] - \operatorname{Var}\left[\|\boldsymbol{\eta}_2\|^2 \right] \right) = \frac{4(1-\varepsilon)^2}{n},$$

where the last equality follows from the fact that the variance of a chi-squared distribution with n degrees of freedom is $2n$.

Thus, the random variable $\frac{1}{n}\boldsymbol{a}_i^T\boldsymbol{a}_j$ is well-concentrated around 0.

B.2.2 Spherical codebook

To avoid power constraint issues, it is reasonable to use a spherical codebook. Let $\mathbb{S}^{n-1}$ be the unit sphere in n dimensions. The codewords are uniformly and independently sampled from $\mathbb{S}^{n-1}$, i.e.,

$$\boldsymbol{a}_i \overset{i.i.d}{\sim} \mathrm{Unif}(\mathbb{S}^{n-1}), \quad i \in [M].$$

Remark B.2. *The standard way to generate such a column is to sample $\tilde{\boldsymbol{a}}_i \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_n)$ and then normalize the result as $\boldsymbol{a}_i = \tilde{\boldsymbol{a}}_i / \|\tilde{\boldsymbol{a}}_i\|$. It is easy to show that the resulting vector is distributed uniformly on $\mathbb{S}^{n-1}$.*

Note that for the conventional GMAC[2] [178], spherical codebooks have been shown to achieve a better second-order (dispersion) term in the asymptotic expansion of the achievable rate region as $n \to \infty$ compared to Gaussian codebooks. However, it is not clear if this result is valid for the G-URA setup. For the URA setup in the case of the MIMO quasi-static Rayleigh fading channel, the authors of [179] demonstrated that the use of spherical codebooks is beneficial in the FBL regime.

B.2.3 BCH subcodes

The codebooks described in the previous sections are easy to generate and work well for large n. However, when considering the preamble design task, we are required to make the preambles as short as possible to avoid additional overhead. In this regime, Gaussian and spherical codebooks may not perform as well.

To overcome this issue, we describe how to construct a codebook with good correlation properties for existing error-correction codes, particularly from BCH codes.

Let us start with the $[n, k]$-BCH code $\mathcal{C}$ that corrects t errors. The columns of $\boldsymbol{A}$ are to be chosen from the set $\mathcal{X} = \boldsymbol{x} = \tau(\boldsymbol{c}) : \boldsymbol{c} \in \mathcal{C}$, where $\tau(\cdot)$ is a BPSK modulation. Let us consider the correlation properties, specifically calculating $(i, j \in [M], i \neq j)$:

$$\boldsymbol{x}_i^T\boldsymbol{x}_j = \frac{1}{2}(\|\boldsymbol{x}_i\|^2 + \|\boldsymbol{x}_j\|^2 - \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2) = n - 2d_H(\boldsymbol{x}_i, \boldsymbol{x}_j).$$

Thus,

$$n - 2\mathrm{wt}_{\max}(\mathcal{C}) \leq \boldsymbol{x}_i^T\boldsymbol{x}_j \leq n - 2\mathrm{wt}_{\min}(\mathcal{C}) = n - 2(2t+1).$$

An interesting observation is that we need to control not only the minimal weight (code distance) but also the maximal weight of the code. The ideal case is to use an equidistant code.

[2]Different codebook case, all active users, fixed K_a

Example B.1. *For the* $[n = 63, k = 10, d = 27]$*-BCH code, we have*

$$-63 \leq \boldsymbol{x}_i^T \boldsymbol{x}_j \leq 9$$

since the code contains the all-one codeword of weight 63*.*

It seems to be easy to solve our problem for the BCH code. We consider only a subcode $\mathcal{C}_0 \subset \mathcal{C}$*, where* $\mathbf{1} \notin \mathcal{C}_0$*. Let us consider the* $[n = 63, k = 9, d = 27]$*-BCH subcode. We have*

$$-9 \leq \boldsymbol{x}_i^T \boldsymbol{x}_j \leq 9$$

since the maximal weight is equal to 36*.*

Let $\mathcal{C}_0$ be the desired subcode with $\dim(\mathcal{C}_0) = k_p$. We construct the sensing matrix $\boldsymbol{A}$ with $\boldsymbol{a}_i = \tau(\boldsymbol{c}_i)$, where $\boldsymbol{c}_i \in \mathcal{C}_0$. The size of $\boldsymbol{A}$ is $n \times 2^{k_p}$.

B.2.4 Codebook from the parity-check matrix of a t-error-correcting code

To obtain a good codebook, we need all the sums $\sum_{i \in \mathcal{I}} \boldsymbol{a}_i$, where $\mathcal{I} \subset [M]$ and $|\mathcal{I}| = t$, to be distinct and well-separated (to account for noise). Such codes are called *superimposed codes*. In what follows, we present one of the possible approaches to constructing such codes.

Let us utilize the technique proposed in [25]. Let $\boldsymbol{H} = [\boldsymbol{h}_1, \ldots, \boldsymbol{h}_n]$ be the parity-check matrix of a t-error-correcting code over $\mathbb{F}_2$. Set

$$\boldsymbol{a}_i = \tau(\boldsymbol{h}_i), \quad i \in [M].$$

We have the desired property: all sums of up to t columns of $\boldsymbol{A}$ are distinct.

Example B.2. *Consider a BCH code of length* n *that corrects* t *errors. In this case, we can construct a codebook of size approximately* $t \log n \times n$ *(since BCH codes are known to be quasi-perfect).*

B.2.5 Binary chirps

In this section, we describe a codebook based on RM codes. Let us introduce the main definitions.

Definition B.1 (Algebraic power)**.**

$$x^\sigma = \begin{cases} x, & \sigma = 1 \\ 1, & \sigma = 0 \end{cases}$$

Definition B.2 (Algebraic normal form or Zhegalkin polynomial)**.**

$$f(\boldsymbol{x}) = \bigoplus_{\boldsymbol{\sigma} \in \{0,1\}^m} b_{\boldsymbol{\sigma}} \boldsymbol{x}^{\boldsymbol{\sigma}}, \quad \boldsymbol{x}^{\boldsymbol{\sigma}} = x_1^{\sigma_1} \cdot \ldots \cdot x_m^{\sigma_m}, \quad \deg(\boldsymbol{x}^{\boldsymbol{\sigma}}) = \|\boldsymbol{\sigma}\|_0 .$$

Note that each Boolean function f has a unique algebraic normal form representation.

Definition B.3. *The degree of the function $f(\boldsymbol{x})$ is defined as*

$$\deg(f) = \max_{\boldsymbol{\sigma}: a_{\boldsymbol{\sigma}}=1} \deg(\boldsymbol{x}^{\boldsymbol{\sigma}}).$$

Definition B.4 (Reed-Muller (RM) code)**.** *Let $f : \{0,1\}^m \to \{0,1\}$ be the Boolean function with m arguments, and let*

$$\mathrm{Eval}(f) = [f(\boldsymbol{\sigma}_1), \ldots, f(\boldsymbol{\sigma}_{2^m})],$$

where $\boldsymbol{\sigma}_i \in \{0,1\}^m$, $i \in [2^m]$, then the RM code of order r is defined as follows:

$$RM(m,r) = \{\mathrm{Eval}(f) : \deg f \leq r\}.$$

The code is linear over $\mathbb{F}_2$ has length $n = 2^m$, and dimension $k = \sum_{i=0}^{r} \binom{m}{i}$.

Now, let us consider the codebook proposed in [143]. Let us start with the first-order RM code (or the case when $r = 1$). Consider all f such that $\deg(f) = 1$[3]. These functions can be expressed as

$$f(\boldsymbol{x}) = \boldsymbol{b}^T \boldsymbol{x} \qquad \mod 2.$$

Now, consider the modulated codeword. It can be represented as $\mathrm{Eval}\left((-1)^f\right)$, which is equivalent to $\mathrm{Eval}\left((-1)^{\boldsymbol{b}^T\boldsymbol{x}}\right)$[4]. The functions

$$\phi_{\boldsymbol{b}}(\boldsymbol{x}) = (-1)^{\boldsymbol{b}^T\boldsymbol{x}}$$

are called the Walsh functions, $\phi_{\boldsymbol{b}} : \{0,1\}^m \to \mathbb{R}$. These functions are orthogonal. However, we can construct only 2^m such functions.

In [143], it is suggested to use the second-order RM codes. Again, the constant function is removed, and we consider

$$\phi_{\boldsymbol{b},\boldsymbol{P}}(\boldsymbol{x}) = (-1)^{\boldsymbol{b}^T\boldsymbol{x} + \frac{1}{2}\boldsymbol{x}^T\boldsymbol{P}\boldsymbol{x}},$$

where $\boldsymbol{b} \in \{0,1\}^m$ and the matrix $\boldsymbol{P} \in \{0,1\}^{m\times m}$ is symmetric and has zeroes on the main diagonal[5].

We set

$$\boldsymbol{a}_{i(\boldsymbol{b},\boldsymbol{P})} = \mathrm{Eval}(\phi_{\boldsymbol{b},\boldsymbol{P}}),$$

where $i(\boldsymbol{b},\boldsymbol{P})$ is the function that returns the column index given $\boldsymbol{b}$ and $\boldsymbol{P}$. The size of $\boldsymbol{A}$ is $2^m \times 2^{m+\binom{m}{2}}$.

Finally, let us mention the main benefit of this codebook. The standard decoders (e.g., the OMP algorithm) calculate the correlation between the current measurement residual and every column,

[3]The reader may ask why the function of degree 0 was removed. The reason is exactly the same as in Section B.2.3 – the all-one codeword is removed to improve the correlation properties.

[4]Note that $\mod 2$ may be omitted.

[5]We only consider the real codebook here, but note that the authors of [143] also suggest the complex codebook $\phi_{\boldsymbol{b},\boldsymbol{P}}(\boldsymbol{x}) = i^{2\boldsymbol{b}^T\boldsymbol{x} + \boldsymbol{x}^T\boldsymbol{P}\boldsymbol{x}}$, where $\boldsymbol{P}$ is symmetric, but diagonal elements are allowed to be non-zero.

selecting the one with the largest absolute value. This is an $O(|\mathcal{C}|)$ procedure and, therefore, computationally intractable for large codebook sizes. In contrast, the reconstruction algorithm proposed in [143] performs a small number of Walsh-Hadamard transforms to identify the matrix $\boldsymbol{P}$ row by row, and subsequently identifies the vector $\boldsymbol{b}$. The overall complexity of the procedure for finding a codeword with maximum correlation is $O(m^2 2^m)$, which is sub-linear in the size of the codebook.

Appendix C

CS sensing matrices

As mentioned in Chapter 3, the URA problem can be considered a CS problem. In this section, we illustrate the CS sensing matrices corresponding to the IRSA protocol presented in Section 5.1.3 and the CCS protocol presented in Section 5.3.

Let us start with an IRSA-F protocol presented in Fig. 5.6 and consider the corresponding CS problem (5.6). Each user transmits $k = \log_2 M$ information bits. The user message $W \in [M]$ is split into W_p and W_c. Then, W_p determines a preamble, while the remaining part, W_c, is encoded using a code of length $n_c = (n - n_p)/L$. The encoded message is further replicated across multiple slots. The replica count and slot indices are determined by both W_p and W_c. The resulting sensing matrix is illustrated in Fig. C.1.

In the case of CCS, the sensing matrix construction procedure is presented in Fig. C.2. Within each slot, transmission is performed using an inner code with a codebook matrix $\boldsymbol{A}$ of size $n' \times Q$. Given that there are L slots in the frame, the sensing matrix construction begins by applying the direct product L times to the matrix $\boldsymbol{A}$, resulting in $\boldsymbol{A}^{\otimes L}$ of size $n'L \times Q^L$. Finally, only M rows of the resulting matrix are selected – forming a set of valid codewords for the outer code.

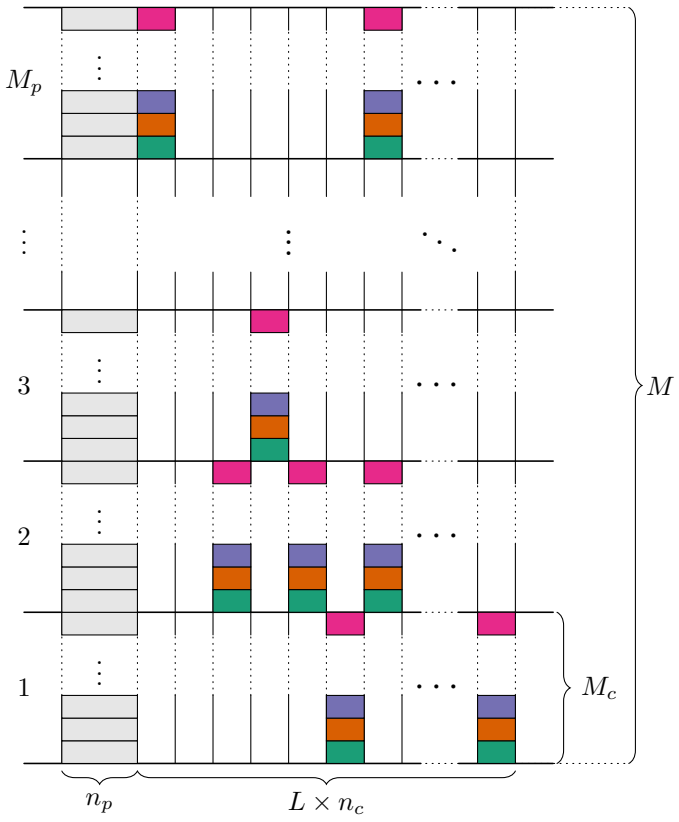

Figure C.1: IRSA-F Sensing Matrix Example. The preamble length is n_p, and the slot length is n_c. Different colors represent different codewords of user messages. Each user message appears M_p times in the resulting sensing matrix, with varying replica counts and slot indices.

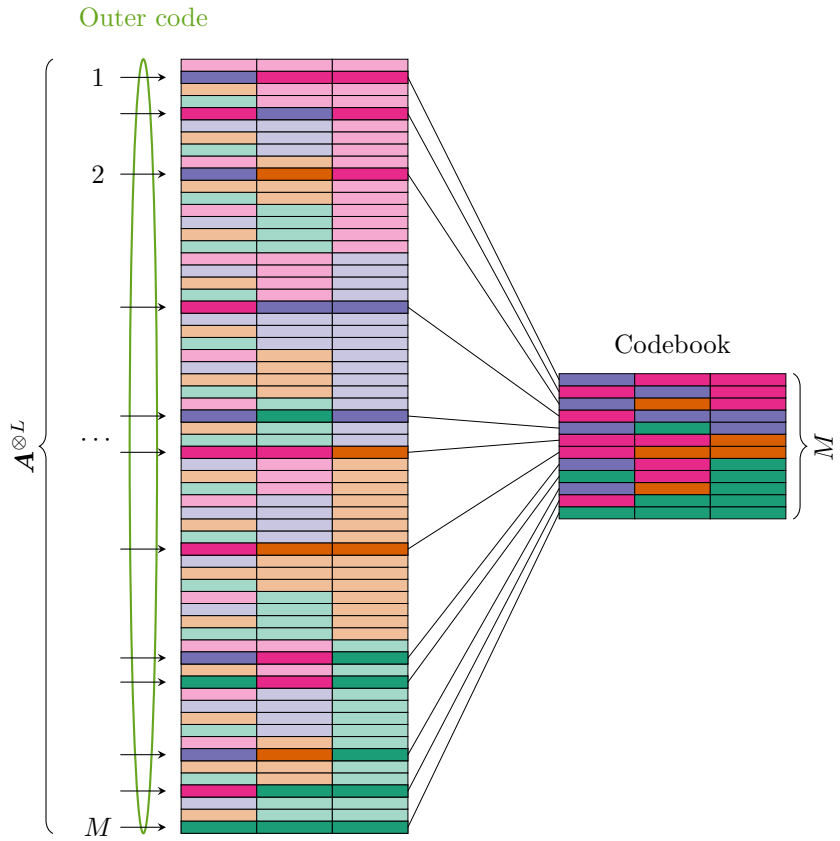

Figure C.2: CCS: sensing matrix. An example for $L = 3$ slots and a given outer code. Different colors represent different rows of the inner codebook matrix $\boldsymbol{A}$ of size $n' \times Q$. In this illustrative case, $Q = 4$. The matrix construction begins by combining all possible inner-code symbol combinations across L slots, forming the large matrix $\boldsymbol{A}^{\otimes L}$ of size $n'L \times Q^L$ (left). Then, the outer code selects M valid outer codewords from the Q^L variants. These valid codewords are highlighted with saturated colors on the left side of the figure, while the resulting CS matrix is shown on the right side.

Appendix D

IRSA: replica count distribution

Replica count distributions optimized using the density evolution procedure (over 10 iterations) are shown in Tables D.1, D.2, and D.3 for $T = 1$, 2, and 4, respectively. Recall that the replica count distribution is given by the polynomial $\Lambda(x)$ (5.1), where the coefficient of the i-th degree term corresponds to the probability that i replicas will be chosen.

Notably, smaller values of T require greater time diversity, which is reflected in the higher-degree terms of the replica count polynomial. On the other hand, the decoder for $T = 4$ can successfully decode slots with high collision orders, making additional replicas unnecessary in this case. Moreover, fewer replicas reduce the overall system load and improve the energy efficiency of the transmission scheme.

Table D.1: Optimal node degree distribution and E_b/N_0 (dB) for $T = 1$, 10 iterations

K_a	$\Lambda(x)$:				E_b/N_0, dB
25	$0.0928x$	$+0.9072x^2$			3.71
50		$+1.0000x^2$			4.48
100		$1.0000x^2$			5.89
150		$0.6211x^2$	$+0.3789x^3$		7.17
200		$0.4781x^2$	$+0.5219x^3$		8.31
250	$0.0706x$	$+0.2011x^2$	$+0.7283x^3$		9.42
300	$0.1297x$		$+0.8703x^3$		10.52
350	$0.1234x$		$+0.8766x^3$		11.62
400	$0.1184x$		$+0.8816x^3$		12.76
450	$0.1247x$		$+0.7991x^3$	$+0.0763x^4$	13.90
500	$0.1396x$		$+0.6716x^3$	$+0.1889x^4$	15.05
550	$0.1474x$		$+0.5906x^3$	$+0.2620x^4$	16.20
600	$0.1549x$		$+0.5239x^3$	$+0.3212x^4$	17.37

Table D.2: Optimal node degree distribution and E_b/N_0 (dB) for $T = 2$, 10 iterations

K_a	$\Lambda(x)$	E_b/N_0, dB
25	$1.0000x$	1.63
50	$0.4443x + 0.5557x^2$	3.09
100	$0.2099x + 0.7901x^2$	4.00
150	$0.1680x + 0.8320x^2$	4.92
200	$0.1382x + 0.8618x^2$	5.88
250	$0.1205x + 0.8795x^2$	6.86
300	$0.1008x + 0.8992x^2$	7.86
350	$0.0895x + 0.9105x^2$	8.87
400	$0.1206x + 0.7609x^2 + 0.1185x^3$	9.90
450	$0.1609x + 0.5953x^2 + 0.2438x^3$	10.92
500	$0.1867x + 0.4916x^2 + 0.3217x^3$	11.94
550	$0.2061x + 0.4160x^2 + 0.3779x^3$	12.97
600	$0.2184x + 0.3639x^2 + 0.4177x^3$	14.00

Table D.3: Optimal node degree distribution and E_b/N_0 (dB) for $T = 4$, 10 iterations

K	$\Lambda(x)$	E_b/N_0, dB
25	$1.0000x$	0.53
50	$1.0000x$	1.10
100	$0.5781x + 0.4219x^2$	2.84
150	$0.3987x + 0.6013x^2$	3.54
200	$0.3764x + 0.6236x^2$	4.31
250	$0.3620x + 0.6380x^2$	5.13
300	$0.3538x + 0.6462x^2$	5.97
350	$0.3479x + 0.6521x^2$	6.83
400	$0.3416x + 0.6584x^2$	7.71
450	$0.3349x + 0.6651x^2$	8.60
500	$0.3302x + 0.6698x^2$	9.52
550	$0.3269x + 0.6731x^2$	10.45
600	$0.3225x + 0.6775x^2$	11.40

Appendix E

Coded CS: further results

E.1 Capacity of the channel for the outer code

In this section, we estimate the capacity of the channel $\mathfrak{W}$ presented in Section 5.3.1. Recall Fig. 5.11 for an illustrative example. For simplicity, we will not perform optimization over all independent distributions of $\mathsf{x}^{(i)}$, $i \in [K_a]$. Instead, we consider only the uniform distribution. Thus, we obtain an expression for the achievable rate rather than the true capacity. We have

$$I_u = I(\mathsf{x}^{(1)}, \ldots, \mathsf{x}^{(K_a)}; \mathcal{Y}) = H(\mathcal{Y}) - H(\mathcal{Y}|\mathsf{x}^{(1)}, \ldots, \mathsf{x}^{(K_a)}), \tag{E.1}$$

where $\mathsf{x}^{(i)} \overset{i.i.d.}{\sim} \mathrm{Unif}([Q])$.

Let $\Omega = |\mathcal{Y}^{(A)}|$, then we have

$$\begin{aligned} H(\mathcal{Y}|\mathsf{x}^{(1)}, \ldots, \mathsf{x}^{(K_a)}) &= \mathbb{E}_\Omega[H(\mathcal{Y}|\Omega)] \\ &= \mathbb{E}_\Omega\left[\Omega h(p_m) + (Q-\Omega)h(p_f)\right] \\ &= \mathbb{E}\left[\Omega\right] h(p_m) + \left(Q - \mathbb{E}\left[\Omega\right]\right) h(p_f) \end{aligned}$$

$$= Q\left(1 - \left(\frac{Q-1}{Q}\right)^{K_a}\right) h(p_m) + Q\left(\frac{Q-1}{Q}\right)^{K_a} h(p_f),$$

where the second equality follows from the following facts: the channel $\mathfrak{W}_E$ operates on elements independently, $H\left(1_{\{x \in \mathcal{Y} | x \in \mathcal{Y}^{(A)}\}}\right) = h(p_m)$ and $H\left(1_{\{x \in \mathcal{Y} | x \notin \mathcal{Y}^{(A)}\}}\right) = h(p_f)$. The fourth equality follows from

$$\begin{aligned} \mathbb{E}\left[\Omega\right] &= \sum_{x \in [Q]} \mathbb{E}\left[1_{\{x \in \mathcal{Y}^{(A)}\}}\right] = Q \Pr\left[1 \in \mathcal{Y}^{(A)}\right] \\ &= Q\left(1 - \left(\frac{Q-1}{Q}\right)^{K_a}\right). \end{aligned}$$

The exact calculation of $H(\mathcal{Y})$ is more complicated. Let us introduce the notation

$$\mu_r = \left(1 - \left(\frac{Q-1}{Q}\right)^r\right)(1-p_m) + \left(\frac{Q-1}{Q}\right)^r p_f \tag{E.2}$$

and use the following estimate:

$$H(\mathcal{Y}) \leq \sum_{x\in[Q]} H\left(1_{\{x\in\mathcal{Y}\}}\right) = Qh\left(\Pr\left[1\in\mathcal{Y}\right]\right) = Qh\left(\mu_{K_a}\right).$$

We note that the estimates above are quite simple and can be found in [95]. These results are useful for formulating the following theorem.

Theorem E.1 (Outer channel converse, [94])**.** *Consider K_a users transmitting messages $W_i \in [M]$, $i \in [K_a]$, over the channel $\mathfrak{W}$ within L channel uses, under a maximum tolerable PUPE P_e. Under the following conditions: (a) the users employ a uniform input distribution, and (b) the decoder output list size is equal to ℓ_0, the following inequality holds:*

$$\log_2 M \leq \frac{1}{(1-P_e)}\left(\frac{LI_u}{K_a} + h\left(P_e\right) + (1-P_e)\log_2\ell_0\right), \tag{E.3}$$

where I_u is given by (E.1)*.*

Proof. Consider the i-th user. We have the following Markov chain:

$$\mathsf{W}_i \to \mathbf{x}^{(i)} \to \boldsymbol{\mathcal{Y}} \to \mathcal{R},$$

where $\mathbf{x}^{(i)} \in [Q]^L$, $\boldsymbol{\mathcal{Y}} = (\mathcal{Y}_1, \ldots, \mathcal{Y}_L)$, $\mathcal{Y}_l \subseteq [Q]$, $l \in [L]$, and $|\mathcal{R}| = \ell_0$. Let $\varepsilon_i = \Pr\left[\mathsf{W}_i \notin \mathcal{R}\right]$. Using Fano's list inequality, we have (3.7), repeated below

$$H\left(\mathsf{W}_i|\boldsymbol{\mathcal{Y}}\right) \leq h(\varepsilon_i) + (1-\varepsilon_i)\log_2\ell_0 + \varepsilon_i\log_2\left(M-\ell_0\right).$$

Summing the inequalities (3.7) for $i \in [K_a]$ and using the fact that

$$H\left(\mathsf{W}_i|\boldsymbol{\mathcal{Y}}\right) = H(\mathsf{W}_i) - I(\mathsf{W}_i;\boldsymbol{\mathcal{Y}}),$$

along with the data processing inequality

$$I\left(\mathbf{x}^{(i)};\boldsymbol{\mathcal{Y}}\right) \geq I\left(\mathsf{W}_i;\boldsymbol{\mathcal{Y}}\right),$$

we obtain

$$\begin{aligned}\sum_{i=1}^{K_a} I\left(\mathbf{x}^{(i)};\boldsymbol{\mathcal{Y}}\right) \geq &\sum_{i=1}^{K_a} H(\mathsf{W}_i) \\ &- \sum_{i=1}^{K_a} h\left(\varepsilon_i\right) - \left(1 - \sum_{i=1}^{K_a}\varepsilon_i\right)\log_2\ell_0\end{aligned}$$

$$- \sum_{i=1}^{K_a} \varepsilon_i \log_2 (M - \ell_0).$$

Finally, since

$$H(\mathsf{W}_i) = \log_2 M, \text{ for } i \in [K_a],$$

$$\sum_{i=1}^{K_a} I\left(\mathbf{x}^{(i)}; \boldsymbol{\mathcal{Y}}\right) \leq LI_u, \quad \sum_{i=1}^{K_a} \varepsilon_i = K_a P_e,$$

and

$$\sum_{i=1}^{K_a} h(\varepsilon_i) \geq K_a h(P_e),$$

we obtain the result stated in the theorem.

□

E.2 t-tree code: the average number of paths

Recall the practical scheme based on linear codes, as presented in Section 5.3.3. In this section, we analyze the FAR and the average decoding complexity. Note that at each decoding step $l \in [L]$, the list of messages $\mathcal{V}_l$ can be expressed as the following disjoint union:

$$\mathcal{V}_l = \mathcal{V}_l^{(c)} \bigsqcup \mathcal{V}_l^{(f)}, \ l \in [L], \tag{E.4}$$

where $\mathcal{V}_l^{(c)}$ represents the beginnings of transmitted messages, and $\mathcal{V}_l^{(f)}$ corresponds to falsely alarmed messages.

We begin by calculating $\mathbb{E}[|\mathcal{V}_l^{(c)}|]$ and $\mathbb{E}[|\mathcal{V}_l^{(f)}|]$ for $l \in [L]$. To do so, we apply random coding to the following ensemble.

Definition E.1. *The elements of the ensemble $\mathcal{E}_2(b_1, \ldots, b_L, L)$ are obtained by randomly selecting the generator matrix $\boldsymbol{G}$ with the structure defined by* (5.21)*, i.e., the elements of each matrix $\boldsymbol{G}_{l',l}$, where $1 \leq l' \leq l \leq L$, are sampled i.i.d. from the* Bern(1/2) *distribution.*

Theorem E.2. *Consider the ensemble $\mathcal{E}_2(b_1, \ldots, b_L, L)$. Let $M_l = 2^{B_l}$. The following bounds hold for $l \in [L]$:*

$$\mathbb{E}[|\mathcal{V}_l^{(c)}|] = v_l^{(c)} \triangleq M_l \rho_l \lambda_l, \tag{E.5}$$

$$\mathbb{E}[|\mathcal{V}_l^{(f)}|] \leq v_l^{(f)} \triangleq M_l \sum_{j=0}^{l-1} \rho_j \lambda_j, \tag{E.6}$$

where

$$\lambda_j = \left(1 - \frac{1}{M_{j+1}}\right)^{K_a} - \left(1 - \frac{1}{M_j}\right)^{K_a}, \ j \in [l-1],$$
$$\lambda_0 = \left(1 - \frac{1}{M_1}\right)^{K_a},$$
$$\lambda_l = 1 - \left(1 - \frac{1}{M_l}\right)^{K_a}, \tag{E.7}$$

and

$$\rho_j = \sum_{\substack{0 \le x \le j \\ 0 \le y \le l-j \\ x+y \le t}} \binom{j}{x}\binom{l-j}{y} p_m^x (1-p_m)^{j-x} \gamma_1^y \gamma_2^{l-j-y},$$

where

$$\gamma_1 = \left(\frac{K_a}{Q}\right) p_m + \left(1 - \frac{1}{Q}\right)(1 - p_f),$$

and

$$\gamma_2 = \left(\frac{K_a}{Q}\right)(1 - p_m) + \left(1 - \frac{1}{Q}\right) p_f.$$

Proof. Assume that the messages

$$\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \ldots, \mathbf{u}^{(K_a)} \tag{E.8}$$

were transmitted. Let $\hat{\boldsymbol{u}}$ be an information word and define $\hat{\boldsymbol{x}}_{[l]} = f_O(\hat{\boldsymbol{u}}_{[l]})$. In what follows, we assume $\hat{\boldsymbol{u}}$ to be fixed, while (E.8) are chosen uniformly at random.

Introduce the r.v.

$$\mathsf{D} = d(\boldsymbol{\mathcal{Y}}_{[l]}, \hat{\boldsymbol{x}}_{[l]})$$

and the events

$$E_j = \bigcap_{i=1}^{K_a} E_{i,j}^c, \quad j \in [l],$$

where $E_{i,j} = \{\hat{\boldsymbol{u}}_{[j]} = \mathbf{u}_{[j]}^{(i)}\}$ for $i \in [K_a]$. Note that the sequence of events satisfies $E_1 \subseteq E_2 \subseteq \ldots \subseteq E_l$.

Observe that $\hat{\boldsymbol{u}}_{[l]}$ is included in $\mathcal{V}_l^{(c)}$ if it was chosen by at least one user (event E_l^c) and $\mathsf{D} \le t$, and is included in $\mathcal{V}_l^{(f)}$ if it was not chosen (event E_l) and $\mathsf{D} \le t$. Thus, we have

$$\begin{aligned} \mathbb{E}[|\mathcal{V}_l^{(c)}|] &= M_l \Pr\left[\mathsf{D} \le t, E_l^c\right], \\ \mathbb{E}[|\mathcal{V}_l^{(f)}|] &= M_l \Pr\left[\mathsf{D} \le t, E_l\right]. \end{aligned}$$

Calculation of $\mathbb{E}[|\mathcal{V}_l^{(c)}|]$. Note that $\Pr[E_l^c] = \lambda_l$ since there are M_l possibilities for each $\mathbf{u}_{[l]}^{(i)}$, $i \in [K_a]$, and

$$\Pr\left[\mathsf{D} \le t \mid E_l^c\right] = \rho_l = \sum_{i=1}^{t} \binom{l}{i} p_m^i (1-p_m)^{l-i}.$$

Multiplying by M_l, we obtain (E.5).

Estimation of $\mathbb{E}[|\mathcal{V}_l^{(f)}|]$. The main difference compared to the proof of Theorem 5.3 is as follows. The beginning of the information word $\hat{\boldsymbol{u}}_{[l]}$ may coincide with the beginning of one of the transmitted information words. In this case, the beginnings of the codewords will also coincide, which must be accounted for in the analysis.

Introduce the events E'_j indicating that the longest match length is equal to j, i.e.,

$$E'_j = E_{j+1} \setminus E_j, \quad j = 0, \ldots, l-1, \quad \text{where} \quad E_0 \triangleq \emptyset. \tag{E.9}$$

We note that $E_l = \bigsqcup_{j=0}^{l-1} E'_j$, and thus

$$\Pr\left[\mathsf{D} \le t, E_l\right] = \sum_{j=0}^{l-1} \Pr\left[\mathsf{D} \le t | E'_j\right] \cdot \Pr\left[E'_j\right].$$

According to (E.9), the probability $\Pr\left[E'_j\right] = \lambda_j$, where λ_j is given by (E.7).

Now consider $\Pr\left[\mathsf{D} \le t \mid E'_j\right]$. Clearly,

$$\Pr\left[\hat{x}_{l'} \in \mathcal{Y}_{l'}^{(A)} \mid E'_j\right] = 1, \quad \text{for} \quad l' \in [j].$$

Thus, an error can only occur due to missed detection, and the result is determined by i.i.d. r.v.s $\xi_{l'} \sim \mathrm{Bern}(p_m)$.

Consider a slot $l' \in [l] \setminus [j]$. Unlike in Theorem 5.3, the symbols $\mathbf{x}_{l'}^{(i)}$ are not independent due to the linear mapping $f_{O,l'}$. Consequently, we apply the following bounds:

$$\frac{1}{Q} \le \Pr\left[\hat{x}_{l'} \in \mathcal{Y}_{l'}^{(A)} | E'_j\right] \le \frac{K_a}{Q}, \quad l' \in [l] \setminus [j].$$

Hence,

$$\Pr\left[\hat{x}_{l'} \notin \mathcal{Y}_{l'} | E'_j\right] \le \gamma_1, \quad \Pr\left[\hat{x}_{l'} \in \mathcal{Y}_{l'} | E'_j\right] \le \gamma_2,$$

for $l' \in [l] \setminus [j]$.

Under the condition E'_j, the r.v. D can be expressed as follows:

$$\mathsf{D} = \sum_{l'=1}^{j} \xi_{l'} + \sum_{l'=j+1}^{l} \psi_{l'},$$

where $\xi_{l'} \sim \mathrm{Bern}(p_m)$ and $\psi_{l'} \sim \mathrm{Bern}(\tilde{p})$. The following bounds hold: $\tilde{p} \le \gamma_1$ and $1 - \tilde{p} \le \gamma_2$.

Finally, we note that $\Pr\left[\mathsf{D} \le t \mid E'_j\right] = \rho_j$. Combining the obtained results, we obtain the theorem statement. □

Corollary E.1. *There exists a code* $\mathcal{C} \in \mathcal{E}_2(b_1, \ldots, b_L, L)$*, such that*

$$P_e = \sum_{i=t+1}^{L} \binom{L}{i} p_m^i (1-p_m)^{L-i} \quad \textit{and} \quad P_f \leq v_L^{(f)},$$

where $v_L^{(f)}$ *is given by* (E.6).

Proof. The proof for P_e is exactly the same as in Theorem 5.3. To estimate P_f, we apply Markov's inequality and use Theorem E.2. We have

$$P_f = \Pr\left[|\mathcal{R} \backslash \mathcal{T}| \geq 1\right] \leq \mathbb{E}[|\mathcal{R} \backslash \mathcal{T}|] = \mathbb{E}[|\mathcal{V}_L^{(f)}|] \leq v_L^{(f)}.$$

□

Appendix F

Asymptotics

Let us consider asymptotic regime as described in Section 2.4.3, where $n \to \infty$ and $K_a \to \infty$ such, that $K_a = \mu n$, where μ represents density of users per channel use, and the number of bits transmitted by each user remains fixed at $k = \log_2 M$. As shown, in such a setup, it makes sense to consider only the case where each user has a different codebook of size M.

In what follows, we will be interested in the fundamental trade-off between the user density μ and minimal required energy per bit:

$$\left(\frac{E_b}{N_0}\right)_{\min} = \inf\left\{\frac{nP}{2\log_2 M}\right\},$$

where the infimum is taken over all (n, M, P) for which there exist $K_a = \mu n$ codebooks, each of size M, that are decodable with PUPE $\leq \epsilon$.

To formulate the asymptotic bounds, it is convenient to introduce the following notation of the total energy $P_{tot} = K_a P$, so that

$$P_{tot} = 2\frac{E_b}{N_0}\mu \log_2 M.$$

Thus, P_{tot} determines $\frac{E_b}{N_0}$ and is fixed. Additionally, let $t = \theta K_a$, then $Pt = PK_a\theta = \theta P_{tot}$.

F.1 Converse bound

Let us consider the converse bound from [5]. In this case, we have the best of the following two bounds.

The first one is given by Fano's inequality:

$$(1-\epsilon)\,\mu \log_2 M \leq \frac{1}{2}\log_2\left(1 + P_{tot}\right) + \mu h\left(\epsilon\right), \tag{F.1}$$

where $h\left(\epsilon\right) = -\epsilon \log_2\left(\epsilon\right) - \left(1-\epsilon\right)\log_2\left(1-\epsilon\right)$ is the binary entropy function.

The second bound follows from the fact that each user transmits only finitely many bits, $\log_2 M$ [31, Theorem 2]

$$\log_2 M \leq -\log_2 Q\left(\sqrt{\frac{P_{tot}}{\mu}} + Q^{-1}(1-\epsilon)\right), \tag{F.2}$$

where $Q^{-1}(\cdot)$ denotes the inverse of the standard Gaussian tail function $Q(\cdot)$.

For both bounds, we need to determine such minimal $\frac{E_b}{N_0}$ that corresponding inequality holds and then select the maximal among them.

F.2 Achievability bounds

Now, let us consider the achievability bounds for the asymptotic regime. Recall the PUPE definition from (4.2)

$$P_e \leq \sum_{t=1}^{K_a} \frac{t}{K_a} \Pr[\mathcal{E}_t] + p_0.$$

It is clear that in this case, $p_0 = 0$ due to concentration theorem and different codebooks.

Let us rewrite the notation of PUPE in the following form for the fixed $\epsilon > 0$ (omitting p_0)

$$\tilde{P}_e \leq \sum_{t=\lfloor \epsilon K_a \rfloor}^{K_a} \frac{t}{K_a} \exp[-nE_t].$$

and consider the following exponent

$$\begin{aligned}
E &= \lim_{n\to\infty} \frac{-\ln \tilde{P}_e}{n} \\
&\geq \lim_{n\to\infty} -\frac{1}{n} \ln \sum_{t=\lfloor \epsilon K_a \rfloor}^{K_a} \frac{t}{K_a} \exp[-nE_t] \\
&\geq \lim_{n\to\infty} -\frac{1}{n} \ln \sum_{t=\lfloor \epsilon K_a \rfloor}^{K_a} \exp[-nE_t] \\
&\geq \lim_{n\to\infty} -\frac{1}{n} \ln \left(K_a \max_{\lfloor \epsilon K_a \rfloor \leq t \leq K_a} \{\exp[-nE_t]\} \right) \\
&= E_{min} = \min_{\epsilon \leq \theta \leq 1} E_\theta
\end{aligned}$$

where the interval $\epsilon \leq \theta \leq 1$ is justified by the fact that ensuring $E_{min} > 0$ in the asymptotic regime guarantees that PUPE $\leq \epsilon$.

Thus, for the fixed $\epsilon > 0$, the fundamental tradeoff of $\frac{E_b}{N_0}$ vs. μ is defined by the following condition:

$$E_{min} = \min_{\epsilon \leq \theta \leq 1} E_\theta > 0.$$

In other words, for the fixed ϵ and μ, we need to find minimal $\left(\frac{E_b}{N_0}\right)_{min}$, such that this condition holds.

To formulate asymptotic achievability bounds, we introduce some helpful notations. Consider the binomial coefficients R_1 and R_2, used in the bounds formulation of the Section 4.3, in the asymptotic regime. Since we consider different codebooks, $R_1 = \frac{1}{n}\log\binom{M-K_a}{t}$ is replaced by $\frac{1}{n}\log(M-1)^t$. Thus,

$$\tilde{R}_1 = \lim_{n\to\infty}\frac{1}{n}\ln(M-1)^t = \lim_{n\to\infty}\frac{t}{n}\ln(M-1) = \theta\mu\ln(M-1)$$

and

$$\tilde{R}_2 = \lim_{n\to\infty} R_2 = \lim_{n\to\infty}\frac{1}{n}\ln\binom{K_a}{t} = \lim_{n\to\infty}\frac{1}{n}\ln\binom{K_a}{\theta K_a} = \mu H(\theta),$$

where $H(\theta) = -\theta\ln(\theta) - (1-\theta)\ln(1-\theta)$ is entropy function. Moreover,

$$\lim_{n\to\infty}\frac{1}{-n}\ln\left(\exp[-nE_1] + \exp[-nE_2]\right) = \min\{E_1, E_2\}.$$

Now, we are ready to formulate the achievability bounds obtained in Section 4.3 in the asymptotic regime.

For Theorem 4.1, we have

$$E_\theta = \max_{0\le\rho_1,\rho_2\le 1}\left[-\rho_1\rho_2\tilde{R}_1 - \rho_1\tilde{R}_2 + E_0(\rho_1,\rho_2)\right], \tag{F.3}$$

where

$$E_0(\rho_1,\rho_2) = \frac{1}{2}\left(\rho_1 a + \log(1-2b\rho_1)\right),$$

$$a = \rho_2\log(1+2\lambda\theta P_{tot}) + \log(1+2\mu\theta P_{tot}), \quad b = \rho_2\lambda - \frac{\mu}{1+2\mu\theta P_{tot}},$$

$$\mu = \frac{\rho_2\lambda}{1+2\lambda\theta P_{tot}}, \quad \lambda = \frac{\theta P_{tot} - 1 + \sqrt{D}}{4(1+\rho_1\rho_2)\theta P_{tot}},$$

$$D = (\theta P_{tot}-1)^2 + 4\theta P_{tot}\frac{1+\rho_1\rho_2}{1+\rho_2}.$$

For Theorem 4.2, we can write

$$E_\theta = \max_{\alpha,\beta\ge 0}\min\left(\tilde{E}_1, \tilde{E}_2\right), \tag{F.4}$$

where

$$\tilde{E}_1 = \max_{u,v>0,\lambda_{\tilde{\boldsymbol{A}}}>0}\left\{-\tilde{R}_1 - \tilde{R}_2 + \frac{1}{2}\log\det\left(\mathbf{I}_3 - 2\tilde{\boldsymbol{A}}\right) - v\beta\right\},$$

$$\tilde{E}_2 = \max_{\delta>0,\lambda_{\tilde{\boldsymbol{B}}}>0}\left\{-\tilde{R}_2 + \frac{1}{2}\log\det\left(\mathbf{I}_2 - 2\delta\tilde{\boldsymbol{B}}\right) + \delta\beta\right\},$$

$$\tilde{\boldsymbol{A}} = \begin{pmatrix} (\alpha-1)v & (\alpha v - u)\sqrt{\theta P_{tot}} & u\sqrt{\theta P_{tot}} \\ (\alpha v - u)\sqrt{\theta P_{tot}} & (\alpha v - u)\theta P_{tot} & u\theta P_{tot} \\ u\sqrt{\theta P_{tot}} & u\theta P_{tot} & -u\theta P_{tot} \end{pmatrix},$$

$$\tilde{\boldsymbol{B}} = \begin{pmatrix} 1-\alpha & -\alpha\sqrt{\theta P_{tot}} \\ -\alpha\sqrt{\theta P_{tot}} & -\alpha\theta P_{tot} \end{pmatrix},$$

and $\lambda_{\tilde{\boldsymbol{A}}}$ and $\lambda_{\tilde{\boldsymbol{B}}}$ are the minimum eigenvalues of $\boldsymbol{I}_3 - 2\tilde{\boldsymbol{A}}$ and $\boldsymbol{I}_2 - 2\delta\tilde{\boldsymbol{B}}$ as previous.

Finally, for Theorem 4.3, we obtain

$$E_\theta = \max_{\alpha,\tau>0} \min\left\{\tilde{E}_1, \tilde{E}_2'\right\}, \tag{F.5}$$

$$\tilde{E}_1 = \max_{0\le\gamma\le 1}\left\{-\gamma\tilde{R}_1 - \tilde{R}_2 + E_1(\gamma)\right\},$$

$$\tilde{E}_2 = \max_{\delta>0}\left\{E_2(\delta)\right\},$$

$$E_1(\gamma) = \frac{1}{2}\left(\gamma\log(1+\theta P_{tot}) + \log\left(1-\gamma^2(1-\rho_e^2)\right) - \gamma\tau\right),$$

$$E_2(\delta) = \frac{1}{2}\left(\log\left(1-\delta^2(1-\rho_r^2)\right) + \tau\delta\right),$$

$$\rho_e^2 = \frac{(1+\alpha\theta P_{tot})^2}{(1+\alpha^2 P_{tot})(1+\theta P_{tot})}, \quad \rho_r^2 = \frac{1}{1+\alpha^2 P_{tot}}.$$

In [87], the best asymptotic achievability bound was obtained by applying Gordon's lemma (see Section 4.5.2):

$$E_\theta = -\tilde{R}_1 - \tilde{R}_2 + \frac{1}{2}\ln(1+2\lambda\theta P_{tot}) + \lambda\frac{\psi(\theta,\mu)}{1+2\lambda\theta P_{tot}} - \lambda, \tag{F.6}$$

where

$$\lambda = \frac{\theta P_{tot} + \sqrt{(\theta P_{tot})^2 + 4\psi(\theta,\mu)^2 - 2}}{4\theta P_{tot}},$$

$$\psi(\theta,\mu) = \sqrt{1+\theta P_{tot}} - \gamma(\theta)\sqrt{\mu P_{tot}},$$

$$\gamma(\theta) = \frac{1}{\sqrt{2\pi}}\exp\left[-\frac{1}{2}\left(Q^{-1}(\theta)\right)^2\right].$$

F.3 Numerical results

Let us consider a setup with a PUPE constraint of $\epsilon = 10^{-3}$ and $k = 100$ information bits transmitted by each user. In Fig. F.1, we analyze the minimum energy per bit (E_b/N_0) as a function of the user density per channel use, μ.

Surprisingly, the energy per bit remains nearly constant until reaching a critical value of μ (for the converse bound, this value is approximately $\mu \approx 0.006$). The behavior of Theorems 4.1, 4.2, and 4.3 is almost the same in both the finite-length and asymptotic regimes. Specifically, Theorems 4.1 and 4.2 outperform Theorem 4.3 for small values of μ, but the latter performs better for larger values of μ.

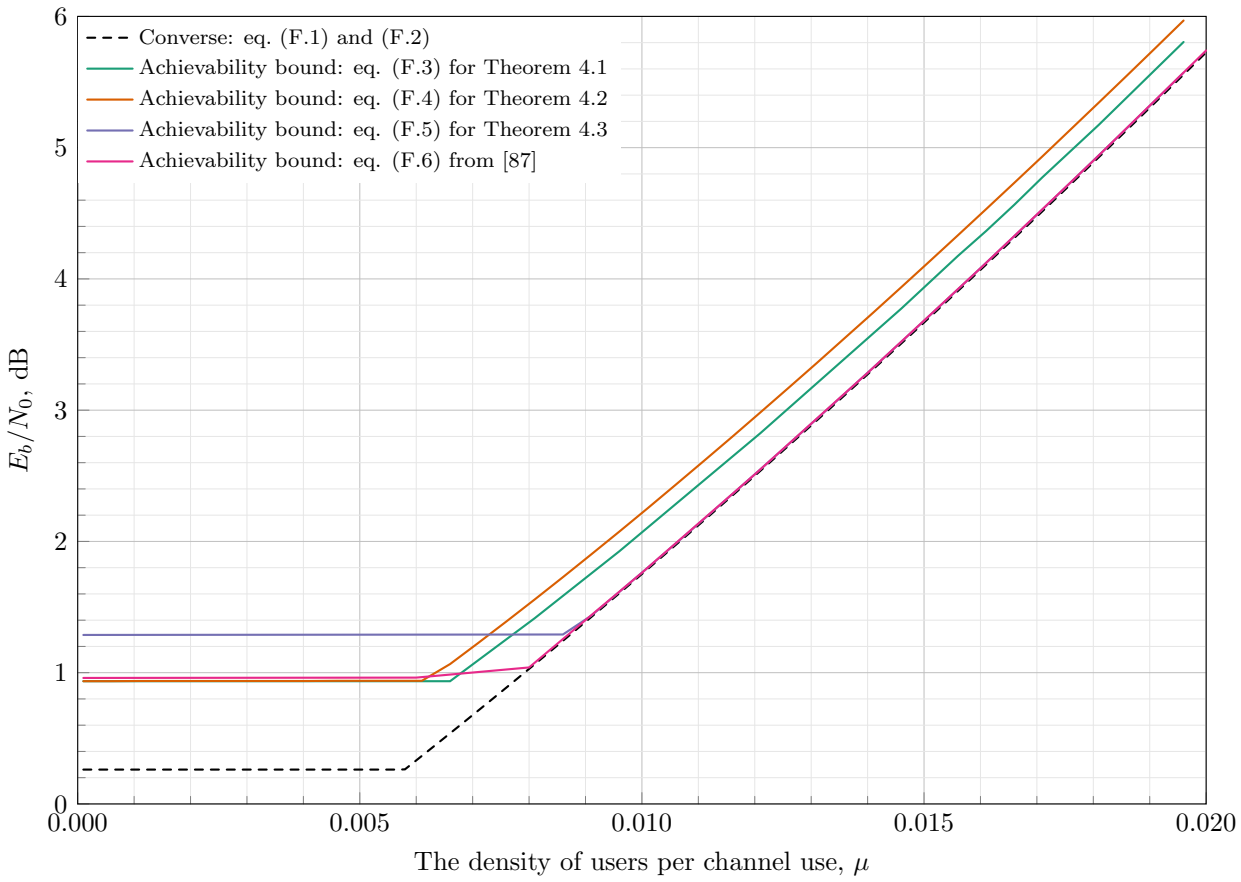

Figure F.1: Asymptotic bounds for $\epsilon = 10^{-3}$ and $k = 100$.

Interestingly, Theorem 4.2 loses to Theorem 4.1 for large values of μ in the asymptotic regime, although they almost coincide in the finite-length regime. It is also noteworthy that Theorem 4.3 achieves the converse bound for large values of μ. The best achievability bound is the one from [87], although it slightly underperforms compared to Theorems 4.1 and 4.2 at small values of μ.

The most interesting observation is the gap of approximately 0.8 dB between the converse bound and the achievability bounds for small values of user density, although, the bounds coincide for larger values of user density.

Appendix G

Some useful lemmas

In this chapter, we provide an overview of some useful lemmas that are used throughout this monograph. We present the Chernoff bound (Lemma G.1) and a closed-form solution for the expectation of a quadratic form exponent (Lemma G.2).

Lemma G.1 (Chernoff bound, [180])**.** *Let χ_1 and χ_2 be arbitrary random variables. Then, for any $u, v > 0$, the following bounds hold:*

$$\Pr\left[\chi_1 \geq 0\right] \leq \mathbb{E}_{\chi_1}\left[\exp\left(u\chi_1\right)\right]$$

and

$$\Pr\left[\chi_1 \geq 0, \chi_2 \geq 0\right] \leq \mathbb{E}_{\chi_1,\chi_2}\left[\exp\left(u\chi_1 + v\chi_2\right)\right].$$

Lemma G.2 (Theorem 3.2a.1, [181])**.** *Let $\boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_n)$, $\mathbf{b}$ be an arbitrary vector of length n, $\boldsymbol{A}$ be a symmetric matrix of size $n \times n$. If $\boldsymbol{I}_n - 2\boldsymbol{A}$ is positive definite, then*

$$\begin{aligned}&\mathbb{E}_{\boldsymbol{\eta}} \exp\left[\boldsymbol{\eta}^T \boldsymbol{A} \boldsymbol{\eta} + \mathbf{b}^T \boldsymbol{\eta}\right] = \\ &= \exp\left[-\frac{1}{2}\log\det\left(\boldsymbol{I}_n - 2\boldsymbol{A}\right) + \frac{1}{2}\mathbf{b}^T\left(\boldsymbol{I}_n - 2\boldsymbol{A}\right)^{-1}\mathbf{b}\right].\end{aligned}$$

Corollary G.1. *Let $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_n)$, and let $\boldsymbol{b}$ be a fixed vector of length n. Then, for any $a > 0$ and $\gamma > -\frac{1}{2a}$, we have*

$$\mathbb{E}_{\mathbf{z}} \exp\left[-\gamma \left\|\sqrt{a}\mathbf{z} + \mathbf{b}\right\|_2^2\right] = \frac{\exp\left[-\frac{\gamma\|\mathbf{b}\|^2}{1+2a\gamma}\right]}{\left(1+2a\gamma\right)^{\frac{n}{2}}}.$$

Corollary G.2. *Let $\mathbf{z} = [\mathsf{z}_1, \mathsf{z}_2] \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Gamma})$ with*

$$\boldsymbol{\Gamma} = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}.$$

Then, for any λ_1, λ_2 satisfying $\lambda_1\lambda_2\rho^2 > (\lambda_1 - 1)(1 + \lambda_2)$, we have

$$\mathbb{E}_{\mathsf{z}_1,\mathsf{z}_2} \exp\left[\frac{\lambda_1}{2}\mathsf{z}_1^2 - \frac{\lambda_2}{2}\mathsf{z}_2^2\right] = \frac{1}{\left(1 - \lambda_1 + \lambda_2 - \lambda_1\lambda_2\left(1-\rho^2\right)\right)^{\frac{1}{2}}}.$$

Lemma G.3 ([181])**.** *Let* $\mathbf{x} \sim \mathcal{CN}(0, \boldsymbol{\Sigma})$*, and let* $\boldsymbol{A}$ *be a Hermitian matrix. If* $\boldsymbol{I} - \boldsymbol{A}\boldsymbol{\Sigma}$ *is positive definite, then*

$$\mathbb{E}_{\mathbf{x}}\left[\exp\left(\mathbf{x}^H \boldsymbol{A}\mathbf{x}\right)\right] = |\boldsymbol{I} - \boldsymbol{A}\boldsymbol{\Sigma}|^{-1}.$$

Bibliography

[1] Ericsson. Ericsson mobility report. November 2023. Technical report, Ericsson, 2023.

[2] Dean Anthony Gratton. *The handbook of personal area networking technologies and protocols*. Cambridge University Press, USA, 2013.

[3] Alireza Maleki, Ha H. Nguyen, Ebrahim Bedeer, and Robert Barton. A tutorial on chirp spread spectrum modulation for LoRaWAN: basics and key advances. *IEEE Open Journal of the Communications Society*, 5:4578–4612, 2024.

[4] Patrick Agostini, Jean-Francois Chamberland, Federico Clazzer, Johannes Dommel, Gianluigi Liva, Andrea Munari, Krishna Narayanan, Yury Polyanskiy, Slawomir Stanczak, and Zoran Utkovski. Evolution of the 5G new radio two-step random access towards 6G unsourced MAC, 2024.

[5] Yury Polyanskiy. A perspective on massive random-access. In *Proc. IEEE Int. Symp. Inf. Theory*, pages 2523–2527, 2017.

[6] Claude E. Shannon. Two-way communication channels. in Proc. 4th Berkeley Symp. Math. Stat. Probab. 1, 611-644 (1961)., 1961.

[7] R. Gallager. A perspective on multiaccess channels. *IEEE Trans. Inf. Theory*, 31(2):124–142, 1985.

[8] Rudolf Ahlswede. Multi-way communication channels. In *Proc. IEEE Int. Symp. Inf. Theory*, Tsahkadsor, Armenia, USSR, 1973.

[9] H Liao. A coding theorem for multiple access communications. In *Proc. IEEE Int. Symp. Inf. Theory*, 1972.

[10] Hosein Nikopour and Hadi Baligh. Sparse code multiple access. In *Proc. IEEE 24th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications*, pages 332–336, 2013.

[11] Lawrence Roberts. ALOHA packet system with and without slots and capture. *ACM SIGCOMM Computer Communication Review*, 5:28–42, 04 1975.

[12] J. Capetanakis. Tree algorithms for packet broadcast channels. *IEEE Trans. Inf. Theory*, 25(5):505–515, 1979.

[13] B. S. Tsybakov and N. B. Likhanov. Some new random multiple-access algorithms. *Prob. Peredachi Inform.*, 21:134–153, 1985.

[14] Enrico Casini, Riccardo De Gaudenzi, and Oscar Del Rio Herrero. Contention resolution diversity slotted ALOHA (CRDSA): an enhanced random access scheme for satellite access packet networks. *IEEE Trans. Commun.*, 6(4):1408–1419, 2007.

[15] Gianluigi Liva. Graph-based analysis and optimization of contention resolution diversity slotted ALOHA. *IEEE Trans. Commun.*, 59(2):477–487, 2011.

[16] Enrico Paolini, Gianluigi Liva, and Marco Chiani. Coded slotted ALOHA: a graph-based method for uncoordinated multiple access. *IEEE Trans. Inf. Theory*, 61(12):6815–6832, 2015.

[17] P. Erdös and A. Rényi. On two problems of information theory. *Publications of Mathematical Institute of Hngarian Academy of Sciences*, 8:241–254, 1963.

[18] Staffan Söderberg and H. S. Shapiro. A combinatory detection problem. *The American Mathematical Monthly*, 70(10):1066–1070, 1963.

[19] B. Lindström. On a combinatory detection problem. *I. A Magyar Tudományos Akadémia Matematikai Kutató Intézetének Közleményei*, 9 (1-2):195–207, 1964.

[20] Shih-Chun Chang and E. Weldon. Coding for T-user multiple-access channels. *IEEE Trans. Inf. Theory*, 25(6):684–691, 1979.

[21] Gèrard Cohen, Simon Litsyn, and Gilles Zèmor. Binary B_2-sequences: a new upper bound. *Journal of Combinatorial Theory, Series A*, 94(1):152–155, 2001.

[22] P. Erdös and P. Turán. On a problem of Sidon in additive number theory, and on some related problems. *Journal of the London Mathematical Society*, s1-16(4):212–215, 1941.

[23] Bernt Lindström. Determination of two vectors from the sum. *Journal of Combinatorial Theory*, 6(4):402–407, 1969.

[24] Bernt Lindström. On B_2-sequences of vectors. *Journal of Number Theory*, 4(3):261–265, 1972.

[25] I. Bar-David, E. Plotnik, and R. Rom. Forward collision resolution – a technique for random multiple-access to the adder channel. *IEEE Trans. Inf. Theory*, 39(5):1671–1675, 1993.

[26] T. Ericson and L. Gyorfi. Superimposed codes in $\mathbb{R}^n$. *IEEE Trans. Inf. Theory*, 34(4):877–880, 1988.

[27] Z. Furedi and M. Ruszinko. Superimposed codes are almost big distance ones. In *Proc. IEEE Int. Symp. Inf. Theory*, pages 118–, 1997.

[28] A.. D'yachkov and V. Rykov. On a coding model for a multiple-access adder channel. *Prob. Peredachi Inform.*, 17(2):94–104, 1981.

[29] Or Ordentlich and Ofer Shayevitz. A VC-dimension-based outer bound on the zero-error capacity of the binary adder channel. In *Proc. IEEE Int. Symp. Inf. Theory*, pages 2366–2370, 2015.

[30] Yury Polyanskiy, H. Vincent Poor, and Sergio Verdú. Channel coding rate in the finite blocklength regime. *IEEE Trans. Inf. Theory*, 56(5):2307–2359, 2010.

[31] Yury Polyanskiy, H. Vincent Poor, and Sergio Verdú. Minimum energy to send k bits through the gaussian channel with and without feedback. *IEEE Trans. Inf. Theory*, 57(8):4880–4902, 2011.

[32] Ebrahim MolavianJazi and J Nicholas Laneman. On the second-order cost of TDMA for Gaussian multiple access. In *Proc. IEEE Int. Symp. Inf. Theory*, pages 266–270, 2014.

[33] Abbas El Gamal and Young-Han Kim. *Network information theory*. Cambridge University Press, 2011.

[34] David Tse and Pramod Viswanath. *Fundamentals of wireless communication*. Cambridge University Press, 2005.

[35] S. Verdú. Minimum probability of error for asynchronous Gaussian multiple-access channels. *IEEE Trans. Inf. Theory*, 32(1):85–96, 1986.

[36] Ruxandra Lupas and Sergio Verdú. Linear multiuser detectors for synchronous code-division multiple-access channels. *IEEE Trans. Inf. Theory*, 35(1):123–136, 1989.

[37] Michael Honig, Upamanyu Madhow, and Sergio Verdú. Blind adaptive multiuser detection. *IEEE Trans. Inf. Theory*, 41(4):944–960, 1995.

[38] M. Rupf and J.L. Massey. Optimum sequence multisets for synchronous code-division multiple-access channels. *IEEE Trans. Inf. Theory*, 40(4):1261–1266, 1994.

[39] P. Viswanath and V. Anantharam. Optimal sequences and sum capacity of synchronous CDMA systems. *IEEE Trans. Inf. Theory*, 45(6):1984–1991, 1999.

[40] Mahesh K Varanasi and Tommy Guess. Optimum decision feedback multiuser equalization with successive decoding achieves the total capacity of the Gaussian multiple-access channel. In *Conference Record of the Thirty-First Asilomar Conference on Signals, Systems & Computers*, volume 2, pages 1405–1409, 1997.

[41] D.N.C. Tse and S.V. Hanly. Linear multiuser receivers: effective interference, effective bandwidth and user capacity. *IEEE Trans. Inf. Theory*, 45(2):641–657, 1999.

[42] Sergio Verdú and S. Shamai. Spectral efficiency of CDMA with random spreading. *IEEE Trans. Inf. Theory*, 45(2):622–640, 1999.

[43] Xiaodong Wang and H.V. Poor. Iterative (turbo) soft interference cancellation and decoding for coded CDMA. *IEEE Trans. Commun.*, 47(7):1046–1061, 1999.

[44] A. Montanari and D. Tse. Analysis of belief propagation for non-linear problems: the example of CDMA (or: how to prove Tanaka's formula). In *Proc. IEEE Inf. Theory Workshop*, pages 160–164, 2006.

[45] Reza Hoshyar, Ferry P. Wathan, and Rahim Tafazolli. Novel low-density signature for synchronous CDMA systems over AWGN channel. *IEEE Trans. Signal Process.*, 56(4):1616–1626, 2008.

[46] Bixio Rimoldi and Rüdiger Urbanke. A rate-splitting approach to the Gaussian multiple-access channel. *IEEE Trans. Inf. Theory*, 42(2):364–375, 1996.

[47] E. Şaşoğlu, E. Telatar, and E. M. Yeh. Polar codes for the two-user multiple-access channel. *IEEE Trans. Inf. Theory*, 59(10):6583–6592, 2013.

[48] E. Arıkan. Polar coding for the Slepian-Wolf problem based on monotone chain rules. In *Proc. IEEE Int. Symp. Inf. Theory*, pages 566–570, 07 2012.

[49] S. Önay. Successive cancellation decoding of polar codes for the two-user binary-input MAC. In *Proc. IEEE Int. Symp. Inf. Theory*, pages 1122–1126, 07 2013.

[50] E. Abbe and E. Telatar. Polar codes for the m-user multiple access channel. *IEEE Trans. Inf. Theory*, 58(8):5437–5448, 2012.

[51] H. Mahdavifar, M. El-Khamy, J. Lee, and I. Kang. Achieving the uniform rate region of general multiple access channels by polar coding. *IEEE Trans. Commun.*, 64(2):467–478, 2016.

[52] A. Amraoui, S. Dusad, and R. Urbanke. Achieving general points in the 2-user Gaussian MAC without time-sharing or rate-splitting by means of iterative coding. In *Proc. IEEE Int. Symp. Inf. Theory*, pages 334–, 2002.

[53] Shrinivas Kudekar and Kenta Kasai. Spatially coupled codes over the multiple access channel. In *Proc. IEEE Int. Symp. Inf. Theory*, pages 2816–2820, 2011.

[54] Norman M. Abramson. The ALOHA system: another alternative for computer communications. In *Proc. AFIPS Fall Joint Computing Conference*, volume 37 of *AFIPS Conference Proceedings*, pages 281–285. AFIPS / ACM, 1970.

[55] Anders E. Kalor, Osama A. Hanna, and Petar Popovski. Random access schemes in wireless systems with correlated user activity. In *Proc. IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pages 1–5, 2018.

[56] Federico Clazzer, Enrico Paolini, Iacopo Mambelli, and Čedomir Stefanović. Irregular repetition slotted ALOHA over the Rayleigh block fading channel with capture. In *Proc. IEEE International Conference on Communications*, pages 1–6, 2017.

[57] M. Kaplan. A sufficient condition of nonergodicity of a Markov chain (corresp.). *IEEE Trans. Inf. Theory*, 25(4):470–471, 1979.

[58] B. Hajek and T. van Loon. Decentralized dynamic control of a multiaccess broadcast channel. *IEEE Trans. Autom. Control*, 27(3):559–569, 1982.

[59] J. Hayes. An adaptive technique for local distribution. *IEEE Trans. Commun.*, 26(8):1178–1186, 1978.

[60] B. S. Tsybakov and V. A. Mikhailov. Free synchronous packet access in a broadcast channel with feedback. *Prob. Peredachi Inform.*, 14:259–280, 1978.

[61] James L. Massey. *Collision-resolution algorithms and random-access communications*, pages 73–137. Springer Vienna, Vienna, 1981.

[62] B. S. Tsybakov and V. A. Mikhailov. Random multiple packet access: part-and-try algorithm. *Prob. Peredachi Inform.*, 16:305–317, 1980.

[63] B. S. Tsybakov and N. B. Likhanov. Upper bound on the capacity of a random multiple-access system. *Prob. Peredachi Inform.*, 23:224–236, 1987.

[64] Yingqun Yu and Georgios B. Giannakis. High-throughput random access using successive interference cancellation in a tree algorithm. *IEEE Trans. Inf. Theory*, 53(12):4628–4639, 2007.

[65] Tom Richardson and Ruediger Urbanke. *Modern coding theory*. Cambridge University Press, New York, NY, USA, 2008.

[66] L. Kleinrock and F. Tobagi. Packet switching in radio channels: part i - carrier sense multiple-access modes and their throughput-delay characteristics. *IEEE Trans. Commun.*, 23(12):1400–1416, 1975.

[67] Xu Chen and Dongning Guo. Gaussian many-access channels: definition and symmetric capacity. In *Proc. IEEE Inf. Theory Workshop*, pages 1–5, 2013.

[68] Xu Chen and Dongning Guo. Many-access channels: the Gaussian case with random user activities. In *Proc. IEEE Int. Symp. Inf. Theory*, pages 3127–3131, 2014.

[69] Xu Chen, Tsung-Yi Chen, and Dongning Guo. Capacity of Gaussian many-access channels. *IEEE Trans. Inf. Theory*, 63(6):3516–3539, 2017.

[70] Y Polyanskiy. Modern aspects of multiple access for IoT (4 lectures). `https://people.lids.mit.edu/yp/homepage/data/SkolTech18-MAC-lectures.pdf`, 2018. [Online].

[71] Khac-Hoang Ngo, Alejandro Lancho, Giuseppe Durisi, and Alexandre Graell i Amat. Massive uncoordinated access with random user activity. In *Proc. IEEE Int. Symp. Inf. Theory*, pages 3014–3019, 2021.

[72] Khac-Hoang Ngo, Alejandro Lancho, Giuseppe Durisi, and Alexandre Graell i Amat. Unsourced multiple access with random user activity. *IEEE Trans. Inf. Theory*, 69(7):4537–4558, 2023.

[73] V. K. Amalladinne, K. R. Narayanan, J. Chamberland, and D. Guo. Asynchronous neighbor discovery using coupled compressive sensing. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4569–4573, 2019.

[74] Alexander Fengler, Alejandro Lancho, Krishna Narayanan, and Yury Polyanskiy. On the advantages of asynchrony in the unsourced MAC, 2023.

[75] Xu Chen, Lina Liu, Dongning Guo, and Gregory W. Wornell. Asynchronous massive access and neighbor discovery using OFDMA. *IEEE Trans. Inf. Theory*, 69(4):2364–2384, 2023.

[76] Alexis Decurninge, Paul Ferrand, and Maxime Guillaud. Massive random access with tensor-based modulation in the presence of timing offsets. In *Proc. IEEE Global Telecommunications Conference*, pages 1061–1066, 2022.

[77] D.L. Donoho. Compressed sensing. *IEEE Trans. Inf. Theory*, 52(4):1289–1306, 2006.

[78] Emmanuel J. Candès, Justin K. Romberg, and Terence Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics*, 59(8):1207–1223, 2006.

[79] Galen Reeves and Michael C. Gastpar. Approximate sparsity pattern recovery: information-theoretic lower bounds. *IEEE Trans. Inf. Theory*, 59(6):3451–3465, 2013.

[80] J.-F. Chamberland, K. Narayanan, and Y Polyanskiy. Unsourced multiple access (UMAC): information theory and coding (tutorial). `https://people.lids.mit.edu/yp/homepage/data/isit2021_umac_tutorial.pdf`, 2021. [Online].

[81] Paul Erdős and Joel Spencer. *Probabilistic methods in combinatorics*. Academic Press, Inc., New York, 1974.

[82] C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, 1948.

[83] R. Gallager. A simple derivation of the coding theorem and some applications. *IEEE Trans. Inf. Theory*, 11(1):3–18, 1965.

[84] R.M. Fano. *Transmission of information: a statistical theory of communication*. MIT Press Classics. MIT Press, 1961.

[85] Anton Glebov, Pavel Rybin, Kirill Andreev, and Alexey Frolov. Energy efficiency of unsourced random access over the binary-input gaussian channel. *IEEE Commun. Lett.*, 27(9):2313–2317, 2023.

[86] Ranmji Venkataramanan, Sekhar Tatikonda, and Andrew Barron. Sparse regression codes. *Foundations and Trends in Communications and Information Theory*, 15(1-2):1–195, 2019.

[87] Ilias Zadik, Yury Polyanskiy, and Christos Thrampoulidis. Improved bounds on Gaussian MAC and sparse regression via Gaussian inequalities. In *Proc. IEEE Int. Symp. Inf. Theory*, 2019.

[88] Y. Gordon. On Milman's inequality and random subspaces which escape through a mesh in $\mathbb{R}^n$. In Joram Lindenstrauss and Vitali D. Milman, editors, *Geometric Aspects of Functional Analysis*, pages 84–106, Berlin, Heidelberg, 1988. Springer Berlin Heidelberg.

[89] Venkat Chandrasekaran, Benjamin Recht, Pablo A. Parrilo, and Alan S. Willsky. The convex geometry of linear inverse problems. *Foundations of Computational Mathematics*, 12(6):805–849, 2012.

[90] Or Ordentlich and Yury Polyanskiy. Low complexity schemes for the random access Gaussian channel. In *Proc. IEEE Int. Symp. Inf. Theory*, pages 2528–2532, 2017.

[91] Li Ping, Lihai Liu, Keying Wu, and W.K. Leung. Interleave division multiple-access. *IEEE Trans. Wireless Commun.*, 5(4):938–947, 2006.

[92] Asit Kumar Pradhan, Vamsi K. Amalladinne, Avinash Vem, Krishna R. Narayanan, and Jean-Francois Chamberland. Sparse IDMA: A joint graph-based coding scheme for unsourced random access. *IEEE Trans. Commun.*, 70(11):7124–7133, 2022.

[93] Vamsi K. Amalladinne, Jean-Francois Chamberland, and Krishna R. Narayanan. A coded compressed sensing scheme for unsourced multiple access. *IEEE Trans. Inf. Theory*, 66(10):6509–6533, 2020.

[94] Kirill Andreev, Pavel Rybin, and Alexey Frolov. Coded compressed sensing with list recoverable codes for the unsourced random access. *IEEE Trans. Commun.*, 70(12):7886–7898, 2022.

[95] Alexander Fengler, Peter Jung, and Giuseppe Caire. SPARCs for unsourced random access. *IEEE Trans. Inf. Theory*, 67(10):6894–6915, 2021.

[96] Vamsi K. Amalladinne, Asit Kumar Pradhan, Cynthia Rush, Jean-Francois Chamberland, and Krishna R. Narayanan. Unsourced random access with coded compressed sensing: integrating AMP and belief propagation. *IEEE Trans. Inf. Theory*, 68(4):2384–2409, 2022.

[97] Asit Kumar Pradhan, Vamsi K. Amalladinne, Krishna R. Narayanan, and Jean-Francois Chamberland. Polar coding and random spreading for unsourced multiple access. In *Proc. IEEE International Conference on Communications*, pages 1–6, 2020.

[98] Mohammad Javad Ahmadi and Tolga M. Duman. Random spreading for unsourced MAC with power diversity. *IEEE Commun. Lett.*, 25(12):3995–3999, 2021.

[99] Asit Kumar Pradhan, Vamsi K. Amalladinne, Krishna R. Narayanan, and Jean-Francois Chamberland. LDPC codes with soft interference cancellation for uncoordinated unsourced multiple access. In *Proc. IEEE International Conference on Communications*, pages 1–6, 2021.

[100] Avinash Vem, Krishna R. Narayanan, Jean-Francois Chamberland, and Jun Cheng. A user-independent successive interference cancellation based coding scheme for the unsourced random access Gaussian channel. *IEEE Trans. Commun.*, 67(12):8258–8272, 2019.

[101] E. Marshakov, G. Balitskiy, K. Andreev, and A. Frolov. A polar code based unsourced random access for the Gaussian MAC. In *Proc. IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*, pages 1–5, 2019.

[102] Bobak Nazer and Michael Gastpar. Compute-and-forward: harnessing interference through structured codes. *IEEE Trans. Inf. Theory*, 57(10):6463–6486, 2011.

[103] M. Ghanbarinejad and C. Schlegel. Irregular repetition slotted ALOHA with multiuser detection. In *Proc. 10th Annual Conference on Wireless On-demand Network Systems and Services*, pages 201–205, 03 2013.

[104] Č. Stefanović, E. Paolini, and G. Liva. Asymptotic performance of coded slotted ALOHA with multipacket reception. *IEEE Commun. Lett.*, 22(1):105–108, 2018.

[105] R Tanner. A recursive approach to low complexity codes. *IEEE Trans. Inf. Theory*, 27(5):533–547, 1981.

[106] A. Glebov, N. Matveev, K. Andreev, A. Frolov, and A. Turlikov. Achievability bounds for T-fold irregular repetition slotted ALOHA scheme in the Gaussian MAC. In *Proc. IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–6, 2019.

[107] K. R. Narayanan and H. D. Pfister. Iterative collision resolution for slotted ALOHA: An optimal uncoordinated transmission policy. In *Proc. 7th International Symposium on Turbo Codes and Iterative Information Processing*, pages 136–139, 4 2012.

[108] J.H. Conway, N.J.A. Sloane, E. Bannai, R.E. Borcherds, J. Leech, S.P. Norton, A.M. Odlyzko, R.A. Parker, L. Queen, and B.B. Venkov. *Sphere packings, lattices and groups*. Grundlehren der mathematischen Wissenschaften. Springer New York, 2013.

[109] J. Boutros and G. Caire. Iterative multiuser joint decoding: unified framework and asymptotic analysis. *IEEE Trans. Inf. Theory*, 48(7):1772–1793, 2002.

[110] F. R. Kschischang, B. J. Frey, and H. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Trans. Inf. Theory*, 47(2):498–519, 2001.

[111] Xiaojie Wang, Sebastian Cammerer, and Stephan Ten Brink. Near-capacity detection and decoding: code design for dynamic user loads in Gaussian multiple access channels. *IEEE Trans. Commun.*, 67(11):7417–7430, 2019.

[112] G. Liva and M. Chiani. Protograph LDPC codes design based on EXIT analysis. In *Proc. IEEE Global Telecommunications Conference*, pages 3250–3254, 11 2007.

[113] A. Bennatan and D. Burshtein. Design and analysis of nonbinary LDPC codes for arbitrary discrete-memoryless channels. *IEEE Trans. Inf. Theory*, 52(2):549–583, 2006.

[114] E. Arıkan. Channel polarization: a method for constructing capacity-achieving codes for symmetric binary-input memoryless channels. *IEEE Trans. Inf. Theory*, 55(7):3051–3073, 2009.

[115] Ido Tal and Alexander Vardy. List decoding of polar codes. *IEEE Trans. Inf. Theory*, 61(5):2213–2226, 2015.

[116] A. Goupil, M. Colas, G. Gelle, and D. Declercq. FFT-based BP decoding of general LDPC codes over abelian groups. *IEEE Trans. Commun.*, 55(4):644–649, 2007.

[117] Alexis Decurninge, Ingmar Land, and Maxime Guillaud. Tensor-based modulation for unsourced massive random access. *IEEE Wireless Commun. Lett.*, 10(3):552–556, 2021.

[118] V. K. Amalladinne, A. Vem, D. K. Soma, K. R. Narayanan, and J. Chamberland. A coupled compressive sensing scheme for unsourced multiple access. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6628–6632, 2018.

[119] G. Cormode and S. Muthukrishnan. Combinatorial algorithms for compressed sensing. In *Proc. 40th Annual Conference on Information Sciences and Systems*, pages 198–201, 2006.

[120] Hung Q. Ngo, Ely Porat, and Atri Rudra. Efficiently decodable compressed sensing by list-recoverable codes and recursion. In *Proc. 29th International Symposium on Theoretical Aspects of Computer Science*, volume 14, pages 230–241, 2012.

[121] Anna C. Gilbert, Martin J. Strauss, Joel A. Tropp, and Roman Vershynin. One sketch for all: fast algorithms for compressed sensing. In *Proc. of the 39th Annual ACM Symposium on Theory of Computing*, pages 237–246, 2007.

[122] P. Indyk and M. Ruzic. Near-optimal sparse recovery in the L_1 norm. In *Proc. 49th Annual IEEE Symposium on Foundations of Computer Science*, pages 199–207, 2008.

[123] K. Zigangirov, S. Popov, and V. Chepyzhov. Nonbinary convolutional coding in channels with jamming. *Prob. Peredachi Inform.*, 31(2):169–183, 1995.

[124] Venkatesan Guruswami. *List decoding of error-correcting codes*. Lecture Notes in Computer Science (LNCS, volume 3282), Springer Berlin, Heidelberg, 2002.

[125] Shih-Chun Chang and J. Wolf. On the T-user M-frequency noiseless multiple-access channel with and without intensity information. *IEEE Trans. Inf. Theory*, 27(1):41–48, 1981.

[126] L. A. Bassalygo and V. V. Rykov. Multiple-access hyperchannel. *Prob. Peredachi Inform.*, 49(4):299–307, 2013.

[127] L. A. Bassalygo and M. S. Pinsker. Evaluation of the asymptotics of the summarized capacity of an M-frequency T-user noiseless multiple-access channel. *Prob. Peredachi Inform.*, 36(2):91–97, 2000.

[128] Ronald L. Graham, Donald Ervin Knuth, and Oren Patashnik. *Concrete mathematics: a foundation for computer science*. Addison-Wesley, Reading, MA, second edition, 1994.

[129] V. Guruswami and M. Sudan. Improved decoding of Reed-Solomon and algebraic-geometry codes. *IEEE Trans. Inf. Theory*, 45(6):1757–1767, 1999.

[130] Vladimir Sidorenko and Robert Fischer. Low-complexity list decoding of Reed-Solomon coded pulse position modulation. In *Proc. 9th International ITG Conference on Systems, Communication and Coding*, pages 1–6, 2013.

[131] R. Koetter and A. Vardy. Algebraic soft-decision decoding of Reed-Solomon codes. *IEEE Trans. Inf. Theory*, 49(11):2809–2825, 2003.

[132] Jamison R. Ebert, Vamsi K. Amalladinne, Jean-Francois Chamberland, and Krishna R. Narayanan. A hybrid approach to coded compressed sensing where coupling takes place via the outer code. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4770–4774, 2021.

[133] Jamison R. Ebert, Vamsi K. Amalladinne, Stefano Rini, Jean-Francois Chamberland, and Krishna R. Narayanan. Coded demixing for unsourced random access. *IEEE Trans. Signal Process.*, 70:2972–2984, 2022.

[134] Alejandro Lancho, Alexander Fengler, and Yury Polyanskiy. Finite-blocklength results for the A-channel: applications to unsourced random access and group testing. In *Proc. 58th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 1–8, 2022.

[135] William W. Zheng, Jamison R. Ebert, Stefano Rini, and Jean-Francois Chamberland. Coding for the unsourced A-channel with erasures: the linked loop code. In *Proc. 31st European Signal Processing Conference (EUSIPCO)*, pages 1534–1538, 2023.

[136] Ehsan Nassaji and Dmitri Truhachev. Dynamic compressed sensing approach for unsourced random access. *IEEE Commun. Lett.*, 28(7):1644–1648, 2024.

[137] A. Lampe and J.B. Huber. On improved multiuser detection with iterated soft decision interference cancellation. In *Proc. IEEE Communications Theory Mini-Conference (Cat. No.99EX352)*, pages 172–176, 1999.

[138] H. El Gamal and E. Geraniotis. Iterative multiuser detection for coded CDMA signals in AWGN and fading channels. *IEEE J. Sel. Areas Commun.*, 18(1):30–41, 2000.

[139] G. Caire and R. Müller. The optimal received power distribution of IC-based iterative multiuser joint decoders. In *Proc. 39th Annu. Allerton Conf. on Communications, Control and Computing*, 2001.

[140] A. Korhan Tanc and Tolga M. Duman. Massive random access with trellis-based codes and random signatures. *IEEE Commun. Lett.*, 25(5):1496–1499, 2021.

[141] Gianluigi Liva and Yury Polyanskiy. On coding techniques for unsourced multiple-access. In *Proc. 55th Asilomar Conference on Signals, Systems, and Computers*, pages 1507–1514, 2021.

[142] Alexander Fengler, Gianluigi Liva, and Yury Polyanskiy. Sparse graph codes for the 2-user unsourced MAC. In *Proc. 56th Asilomar Conference on Signals, Systems, and Computers*, pages 682–686, 2022.

[143] Robert Calderbank and Andrew Thompson. CHIRRUP: a practical algorithm for unsourced multiple access. *Information and Inference: A Journal of the IMA*, 9(4):875–897, 2019.

[144] Mert Ozates, Mohammad Kazemi, and Tolga M. Duman. Unsourced random access using ODMA and polar codes. *IEEE Wireless Commun. Lett.*, 13(4):1044–1047, 2024.

[145] Ishay Haviv and Oded Regev. The restricted isometry property of subsampled Fourier matrices. In *Proc. of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '16, page 288–297, USA, 2016. Society for Industrial and Applied Mathematics.

[146] T. T. Cai and L. Wang. Orthogonal matching pursuit for sparse signal recovery with noise. *IEEE Trans. Inf. Theory*, 57(7):4680–4688, 2011.

[147] Suhas S. Kowshik and Yury Polyanskiy. Fundamental limits of many-user MAC with finite payloads and fading. *IEEE Trans. Inf. Theory*, 67(9):5853–5884, 2021.

[148] S. S. Kowshik, K. Andreev, A. Frolov, and Y. Polyanskiy. Energy efficient coded random access for the wireless uplink. *IEEE Trans. Commun.*, 68(8):4694–4708, 2020.

[149] Ehsan Nassaji, Murwan Bashir, and Dmitri Truhachev. Unsourced random access over fading channels via data repetition, permutation, and scrambling. *IEEE Trans. Commun.*, 70(2):1029–1042, 2022.

[150] Alexander Fengler, Saeid Haghighatshoar, Peter Jung, and Giuseppe Caire. Non-Bayesian activity detection, large-scale fading coefficient estimation, and unsourced random access with a massive MIMO receiver. *IEEE Trans. Inf. Theory*, 67(5):2925–2951, 2021.

[151] Junyuan Gao, Yongpeng Wu, Shuo Shao, Wei Yang, and H. Vincent Poor. Energy efficiency of massive random access in MIMO quasi-static Rayleigh fading channels with finite blocklength. *IEEE Trans. Inf. Theory*, 69(3):1618–1657, 2023.

[152] Junyuan Gao, Yongpeng Wu, Tianya Li, and Wenjun Zhang. Energy efficiency of MIMO massive unsourced random access with finite blocklength. *IEEE Wireless Commun. Lett.*, 12(4):743–747, 2023.

[153] Kirill Andreev, Daria Ustinova, and Alexey Frolov. Unsourced random access with the MIMO receiver: projection decoding analysis. *IEEE Wireless Commun. Lett.*, pages 1–1, 2023.

[154] Alexander Fengler, Osman Musa, Peter Jung, and Giuseppe Caire. Pilot-based unsourced random access with a massive MIMO receiver, interference cancellation, and power control. *IEEE J. Sel. Areas Commun.*, 40(5):1522–1534, 2022.

[155] Mohammad Javad Ahmadi and Tolga M. Duman. Unsourced random access with a massive MIMO receiver using multiple stages of orthogonal pilots. In *Proc. IEEE Int. Symp. Inf. Theory*, pages 2880–2885, 2022.

[156] Vamsi K. Amalladinne, Jean-Francois Chamberland, and Krishna R. Narayanan. Coded compressed sensing with successive cancellation list decoding for unsourced random access with massive MIMO, 2021.

[157] Volodymyr Shyianov, Faouzi Bellili, Amine Mezghani, and Ekram Hossain. Massive unsourced random access based on uncoupled compressive sensing: another blessing of massive MIMO. *IEEE J. Sel. Areas Commun.*, 39(3):820–834, 2021.

[158] Michail Gkagkos, Krishna R. Narayanan, Jean-Francois Chamberland, and Costas N. Georghiades. FASURA: a scheme for quasi-static fading unsourced random access channels. *IEEE Trans. Commun.*, 71(11):6391–6401, 2023.

[159] K. Andreev, S. S. Kowshik, A. Frolov, and Y. Polyanskiy. Low complexity energy efficient random access scheme for the asynchronous fading MAC. In *Proc. IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*, pages 1–5, 9 2019.

[160] Wei Yang, Giuseppe Durisi, Tobias Koch, and Yury Polyanskiy. Quasi-static multiple-antenna fading channels at finite blocklength. *IEEE Trans. Inf. Theory*, 60(7):4232–4265, 2014.

[161] Te Sun Han. An information-spectrum approach to capacity theorems for the general multiple-access channel. *IEEE Trans. Inf. Theory*, 44(7):2773–2795, 1998.

[162] I Bettesh and S Shamai. Outages, expected rates and delays in multiple-users fading channels. In *Proc. of the 2000 Conference on Information Science and Systems*, volume 1, 2000.

[163] Jørgen Nielsen. The distribution of volume reductions induced by isotropic random projections. *Advances in Applied Probability*, 31(4):985–994, 1999.

[164] Wei Yang, Giuseppe Durisi, Tobias Koch, and Yury Polyanskiy. Quasi-static SIMO fading channels at finite blocklength. In *Proc. IEEE Int. Symp. Inf. Theory*, pages 1531–1535, 2013.

[165] Yury Polyanskiy. *Channel coding: non-asymptotic fundamental limits*. Princeton University, 2010.

[166] Jie Chen and Xiaoming Huo. Theoretical results on sparse representations of multiple-measurement vectors. *IEEE Trans. Signal Process.*, 54(12):4634–4643, 2006.

[167] S. S. Kowshik, K. Andreev, A. Frolov, and Y. Polyanskiy. Energy efficient random access for the quasi-static fading MAC. In *Proc. IEEE Int. Symp. Inf. Theory*, pages 2768–2772, 2019.

[168] K. Andreev, E. Marshakov, and A. Frolov. A polar code based TIN-SIC scheme for the unsourced random access in the quasi-static fading MAC. In *Proc. IEEE Int. Symp. Inf. Theory*, pages 3019–3024, 2020.

[169] S. Sparrer and R. F. H. Fischer. MMSE-based version of OMP for recovery of discrete-valued sparse signals. *Electronics Letters*, 52(1):75–77, 2016.

[170] Alexander Fengler, Alejandro Lancho, Krishna Narayanan, and Yury Polyanskiy. On the advantages of asynchrony in the unsourced MAC. In *Proc. IEEE Int. Symp. Inf. Theory*, pages 2523–2528, 2023.

[171] J. Hou, J.E. Smee, H.D. Pfister, and S. Tomasin. Implementing interference cancellation to increase the EV-DO Rev A reverse link capacity. *IEEE Commun. Mag.*, 44(2):58–64, 2006.

[172] N. Abramson. VSAT data networks. *Proc. of the IEEE*, 78(7):1267–1274, 1990.

[173] Oscar del Rio Herrero and Riccardo De Gaudenzi. A high efficiency scheme for quasi-real-time satellite mobile messaging systems. In *Proc. 10th International Workshop on Signal Processing for Space Communications*, pages 1–9, 2008.

[174] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge University Press, 03 2004.

[175] Christopher M. Bishop. *Pattern recognition and machine learning (information science and statistics)*. Springer, 1 edition, 2007.

[176] David L. Donoho, Arian Maleki, and Andrea Montanari. Message passing algorithms for compressed sensing: I. motivation and construction. In *Proc. IEEE Inf. Theory Workshop*, pages 1–5, 2010.

[177] Sundeep Rangan. Generalized approximate message passing for estimation with random linear mixing. In *Proc. IEEE Int. Symp. Inf. Theory*, pages 2168–2172, 2011.

[178] Recep Can Yavas, Victoria Kostina, and Michelle Effros. Gaussian multiple and random access channels: finite-blocklength analysis. *IEEE Trans. Inf. Theory*, 67(11):6983–7009, 2021.

[179] Junyuan Gao, Yongpeng Wu, Giuseppe Caire, Wei Yang, and Wenjun Zhang. Unsourced random access in MIMO quasi-static Rayleigh fading channels with finite blocklength. In *Proc. IEEE Int. Symp. Inf. Theory*, pages 3213–3218, 2024.

[180] T.M. Cover and J.A. Thomas. *Elements of information theory*. Wiley-Interscience, New York, second edition, 2006.

[181] A. M. Mathai and S.B. Provost. *Quadratic forms in random variables: theory and applications*. Marcel Dekker, New York, 1992.