Politecnico di Torino

ScuDo
Scuola di Dottorato ~ Doctoral School
WHAT YOU ARE, TAKES YOU FAR

Doctoral Dissertation

Doctoral Program in Electrical, Electronics and Communications Engineering (37^{th} cycle)

Machine-Type Communications: Coding, Multiple Access and Synchronization

By

Riccardo Schiavone

Supervisors:
Prof. Monica Visintin, Supervisor, Politecnico di Torino
Prof. Roberto Garello, Co-Supervisor, Politecnico di Torino
Dr. Gianluigi Liva, Co-Supervisor, German Aerospace Center

Doctoral Examination Committee:
Prof. Valerio Bioglio, Referee, Università di Torino
Prof. Marco Dalai, Referee, Università di Brescia
Dr. Massimo Battaglioni, Universitá Politecnica delle Marche
Prof. Guido Montorsi, Politecnico di Torino
Prof. Giorgio Taricco, Politecnico di Torino

Politecnico di Torino
2025

Declaration

I hereby declare that the contents and organization of this dissertation constitute my own original work and does not compromise in any way the rights of third parties, including those relating to the security of personal data.

Riccardo Schiavone
2025

* This dissertation is presented in partial fulfillment of the requirements for **Ph.D. degree** in the Graduate School of Politecnico di Torino (ScuDo).

In loving memory of my grandmothers.

Acknowledgements

Mi sembra giusto prendermi un momento per dire grazie a tutte le persone che, in questi anni in cui il dottorato e la mia vita si sono intrecciati, mi sono state vicine, mi hanno aiutato a crescere, mi hanno ascoltato o semplicemente condiviso il loro tempo con me.

Parto dalla persona con cui ho trascorso più tempo: me stesso. Ringrazio quel me bambino, creativo, curioso, fragile e fiducioso nel mondo, e quel me giovane adulto, più accorto, organizzato e pragmatico. In questi anni si sono riavvicinati, affrontando insieme le diverse sfide — fisiche e mentali — che la vita e il dottorato hanno posto sul cammino. E anche per godersi, insieme, i piccoli e grandi traguardi raggiunti. (Un ringraziamento sentito va anche alla mia psicoterapista, che con professionalità e sensibilità mi ha aiutato a dare spazio a entrambi e ad attraversare questo percorso con degli strumenti e gentilezza verso me stesso.)

Il secondo grazie va senza dubbio ai miei genitori e a mio fratello, che mi sono stati accanto, soprattutto in alcune sfide che per me sono state enormi — e immagino non facili nemmeno per loro. Non lo sono state per me, perché ho dovuto imparare a chiedere aiuto e ad accettare i miei limiti. E forse non lo sono state per loro, che hanno dovuto riorganizzare la propria vita per riuscire ad esserci. Grazie per i piccoli e grandi sforzi, per le passeggiate attorno a casa che passo dopo passo mi hanno aiutato a rialzarmi. Le parole non bastano per esprimere ciò che provo, ma grazie davvero dal profondo del cuore.

Un grazie speciale ai miei supervisori: Monica, Roberto e Gianluigi. Questo manoscritto e questi ringraziamenti non esisterebbero senza di loro, che mi hanno spinto a intraprendere questo percorso. Al di là delle tante ore passate tra riunioni, progetti e altre attivitá, mi resteranno impressi i loro valori e la loro passione per gli ambiti sia scientifici che non, e i momenti più leggeri e umani. Come le chiacchiere davanti a un caffé nell'ufficio di Monica, il sederci con Gianluigi a un tavolo e pensare insieme a problemi e come affrontarli, o la volta in cui Roberto venne a trovarmi a casa con un fumetto :)

Ringrazio anche i revisori, che hanno dedicato tempo, gratuitamente, alla lettura dettagliata della tesi, lasciando spunti preziosi e suggerimenti importanti.

Un grazie sentito va anche ai tantissimi amici che ho incontrato o ritrovato lungo questi anni. Non li cito tutti per nome, ma sanno che gli voglio bene.

Dagli amici del Dipartimento, in particolare Greta, Barbara, Giuseppe, Franco, Antonio, Federico, Riccardo e Gabriel. Ai vecchi amici della triennale come Matteo, Gianluca, Stefano e Luana. Agli amici di Eurecom: Federico, Francesco, Piera, Viola. Un grazie anche alla famiglia della Croce Rossa di Giaveno — in particolare Marco, Michele, Melissa, Valentina, Nataly e tante altre bellissime persone — con cui in questi anni abbiamo condiviso giorni e notti a fare piccole e grandi cose, non per ridisegnare il futuro, ma per migliorare il presente.

Non posso non ringraziare poi gli amici di Monaco, che mi hanno supportato (e sopportato) anche nei momenti in cui scrivevo questa tesi. A partire da Jessica, che mi ha sempre spinto a provarci nelle cose nuove, con i miei limiti, aiutandomi a fare cose che magari avrei fatto lo stesso ma con tempi diversi, e che, invece, farle insieme ha dato un altro colore a quei momenti. A Stefano, per le pause caffè, le ultime sere d'estate con anche Violetta, e quei momenti di semplice relax. A Violetta, per i cibetti buonissimi e i tanti abbracci. E poi Ludovica, Pietro, Edoardo, Lorenzo e Gaetano, per tutte le esperienze insieme come le cene, le feste, le passeggiate in montagna, le giornate al sole lungo fiumi e laghi, i festival e non solo.

Un ringraziamento anche ai colleghi, più amici che colleghi, di DLR, per le pause caffè, le birre, gli spritz e tutto il tempo condiviso. In particolare grazie a Davide (e a Francesca), Estefania e Andrea, Alexander, Luca, Federico, Marcel e Manuel.

Infine, un grazie al mio amico Riccardo, che ogni volta che torno a Torino mi porta a sbranare 10 kg di cibi buonissimi e per i pomeriggi passati a ridere del più e del meno.

Grazie!

Abstract

Machine-type communications (MTC) encompass a broad range of applications—such as smart metering, industrial IoT, and autonomous vehicles—that require the reliable transmission of short to moderate-length packets under stringent energy, latency, and complexity constraints, often in networks with thousands to millions of intermittently active devices. In this regime, any protocol overhead—such as preamble or pilot insertion—impacts significantly the overall system performance, sharply reducing both spectral and energy efficiency. Although Shannon's capacity theorem, derived in the limit of large packets, establishes fundamental performance limits, it does not account for these finite-length effects. To address this gap, non-asymptotic finite-blocklength bounds have been developed and shown to be remarkably tight down to very short packets. Nevertheless, practical coding schemes—such as low-density parity check and turbo codes, which perform well at moderate to large blocklengths—suffer substantial performance degradation when blocklengths fall below a few hundred bits. The resulting losses are further emphasized in massive scenarios. Against this backdrop of massive connectivity and short-packet overhead, this dissertation tackles three important challenges in MTC system design: (1) channel coding optimized for short packets, (2) uncoordinated multiple-access protocols that scale to massive device populations, and (3) frame synchronization methods for direct-sequence spread-spectrum links.

Chapter 3 investigates the concatenation of convolutional codes with outer polynomial codes (poly+CC) of CCSDS telemetry recommendations and LTE control channels. By analyzing the trellis of the corresponding poly+CC and computing its distance spectrum, tight upper bounds on the block-error probability under maximum-likelihood decoding are derived. The analysis yields an approximate 3 dB coding gain for the maximum-likelihood decoding of poly+CCs over Viterbi decoding of the convolutional code alone. Numerical evaluations for both CCSDS telemetry and LTE systems demonstrate that list-Viterbi decoding of the poly+CC scheme can recover a large fraction of the theoretical 3 dB coding gain, with moderate list sizes and practical decoder complexity.

Chapter 4 adapts the enhanced spread-spectrum ALOHA (E-SSA) protocol—widely used in satellite MTC—for the unsourced multiple-access channel (UMAC) framework. A wrap-around framing model permits direct comparison with finite-blocklength UMAC achievability

bounds. Modifications in the E-SSA protocol include the integration of short low-rate polar codes with outer polynomial codes and the exploitation of the transmission timing as an auxiliary error-detection channel. Performance comparisons show that the optimized E-SSA approaches state-of-the-art UMAC schemes for moderate numbers of active users, while retaining a simple transmitter architecture and a receiver whose complexity scales linearly with the number of users.

Chapter 5 formulates sequential frame synchronization in direct-sequence spread-spectrum systems as a binary hypothesis test under coherent and non-coherent additive white Gaussian noise channel models. The optimal likelihood-ratio test is derived via the Neyman-Pearson criterion, and simplified metrics are proposed. Compared to traditional detectors based on preamble-only detection, the new metrics that incorporate spreading sequence information into the synchronization process significantly improve detection accuracy. Moreover, the simplified tests maintain robust performance while offering reduced implementation complexity, in low-SNR regimes and in the presence of phase uncertainty in non-coherent scenarios. When embedded into the E-SSA receiver, the new frame synchronizer is capable of working with a shorter preamble length, which results in improved energy and spectral efficiency, while achieving up to 20% fewer channel-decoder calls, and while reducing the necessary per-user signal-to-noise ratio by 0.4 dB to meet a target per-user error probability and a 33% increase in supported simultaneous users under moderate load.

Taken together, the coding, access-protocol, and synchronization methods developed in this work advance the design of energy- and spectrally-efficient MTC systems, helping to approach the finite-blocklength performance limits with feasible implementation complexity.

Contents

List of Figures

List of Tables

List of Acronyms

AOS advanced orbiting systems

ARQ automatic retransmission request

ASM attached synchronization marker

AWGN additive white Gaussian noise

bi-AWGN binary-input additive white Gaussian noise

BPSK binary phase-shift keying

CC convolutional code

CCSDS Consultative Committee for Space Data Systems

c.d.f. cumulative distribution function

CRC cyclic redundancy check

DSSS direct-sequence spread spectrum

E-SSA enhanced spread spectrum Aloha

FA false alarm

FBL finite blocklength

FER frame error rate

FSM finite state machine

GMAC Gaussian multiple access channel

i.i.d. independent and identically distributed

LDPC low-density parity-check

LLR log-likelihood ratio

LRT likelihood ratio test

LTE long term evolution

LVA list Viterbi algorithm

MAC medium access control

MD missed detection

ML maximum likelihood

mMTC massive machine type communication

MRA massive random access

MTC machine-type communication

NR new radio

OID only idle data

OSI open systems interconnections

p.d.f. probability density function

PLVA parallel-list Viterbi algorithm

PUPE per-user probability of error

QPSK quadrature phase-shift keying

RA random access

RCU random-coding union

ROC receiver operating characteristic

RS Reed-Solomon

r.v. random variable

SC successive cancellation

SCL successive cancellation list

SIC successive interference cancellation

SLVA serial-list Viterbi algorithm

SNR signal-to-noise ratio

TBCC tail-biting terminated convolutional code

TC telecommand

TF transfer frame

TM telemetry

TSB tangential sphere bound

TTLVA tree-trellis list Viterbi algorithm

UB union bound

UFER undetected frame error rate

UMAC unsourced multiple access channel

URLLC ultra-reliable and low-latency communication

VA Viterbi algorithm

WAVA wrap-around Viterbi algorithm

WE weight enumerator

WEF weight enumerating function

ZTCC zero-tail terminated convolutional code

Nomenclature

X, x random variable (uppercase) and its realization (lowercase)

$\mathbf{X}, \mathbf{x}$ random vector (bold uppercase) and its realization (bold lowercase)

$\mathsf{E}[X], \mathrm{Var}[X]$ expectation and variance of the random variable X

$\mathbb{C}\mathcal{N}, \mathcal{N}$ complex (circularly symmetric) and real normal distributions

$\mathcal{U}$ uniform distribution

$\langle \cdot, \cdot \rangle$ inner product between two vectors

$\Re, \Im$ real and imaginary part of a complex scalar or vector

E_b, E_s, E_c energy per information bit, per modulated symbol, per chip

N_0 single-sided noise power spectral density

n, k, R_0, ν convolutional encoder output size, input size, nominal rate and memory

N, K, R blocklength, information size and code rate

M spreading factor

K_a number of active users

$L_{\mathsf{p}}, L_{\mathsf{d}}$ preamble length (in symbols) and data part length (in chip)

Chapter 1

Introduction

1.1 Machine-Type Communications

In 1948, Claude Elwood Shannon, in his seminal paper "A mathematical theory of communication" [1], introduced a mathematical model of communication systems, describing the exchange of information between a *transmitter* and a *receiver* in a noisy environment. Shannon modeled the statistical influence of the noise on the transmitted data with the *channel* and he showed that reliable communication is possible for *communication rates R* smaller than the *channel capacity C*. Furthermore, Shannon proved that, in the limit of large blocks, it is optimal to separate the transmission into two parts: *source coding* and *channel coding*. However, Shannon did not provide an explicit method for achieving the capacity, neither doing it efficiently. Additionally, his theoretical results were inherently asymptotic, meaning they required large data packets to closely approach optimal performance.

The first large-scale digital mobile communication systems emerged during the '90s. They primarily served human interactions, involving the transmission of variable data volumes such as voice, video, and web browsing, with specific latency and reliability constraints necessary for real-time interactions, like voice or video calls. Typically, the number of users per cell was relatively limited, with predictable traffic peaks, leading to a strong focus on effective mobility management within cellular networks.

In contrast, modern communication systems increasingly target machine-to-machine communications, involving direct communication between devices with minimal or no human intervention. Human involvement, when present, is generally limited to configuration, monitoring, or management activities. Machine-type communications (MTCs) typically entail transmitting sporadic, relatively short data packets containing sensor measurements, telemetry data, or control commands. Additionally, the communication can become massive, involving numerous devices transmitting simultaneously, albeit individually with low data volumes.

Requirements for machine-type communications differ significantly depending on specific applications. Prominent applications of MTCs include: **smart metering**, the automatic collection of energy consumption data (electricity, gas, water) from smart meters; **smart cities**, which encompasses environmental monitoring (air quality, noise), traffic management, intelligent public lighting, smart parking, optimized waste collection, urban security; **industrial IoT** (or industry 4.0), which encloses monitoring and control of industrial processes, predictive maintenance, factory automation, supply chain management, collaborative robotics; **smart agriculture**, which is the monitoring of environmental conditions in fields (soil moisture, temperature, weather conditions), automated irrigation, livestock monitoring, precision agriculture (targeted application of fertilizers and pesticides); **logistics and asset tracking**, which comprises tracking the location and condition of goods in transit (containers, packages, vehicles), fleet management, cold chain monitoring; **healthcare**, which involves remote patient monitoring (vital signs, physical activity), connected medical devices, remote medicine, smart home care, medication management; **autonomous vehicles**, which includes vehicles that utilize sensors and onboard intelligence to drive themselves, demanding ultra-reliable, low-latency communication for safe navigation, real-time control, and interaction with their surroundings. Some of the mentioned applications, such as environmental monitoring, tolerate high latency and occasional packet loss, whereas others, like industrial control, demand extremely low latency and high reliability. Common priorities also include energy efficiency and low device costs. Machine-type communication scenarios often envision an enormous number of devices (from tens to billions) within a geographical area, necessitating efficient network management and scalability. Device mobility varies widely: many devices are static or minimally mobile (e.g., sensors in buildings or smart meters), while others exhibit mobility patterns distinctly different from those of humans (e.g., asset tracking or autonomous driving). Table 1.1 summarizes key requirements for these applications.

Table 1.1 Requirements of selected machine-type communications applications.

Application	Data Volume	Latency	Reliability	Energy Efficiency
Smart meters	Very small	Tolerant	Important	High
Smart cities	Variable	Application-specific	Important	Medium to high
Industrial IoT	Highly variable	Critical or tolerant	Extremely critical	Medium
Smart agriculture	Small	Tolerant	Medium	High
Logistics	Medium	Moderate	Important	High
Asset tracking	Small	Moderate	Important	Medium to high
Healthcare	Small to medium	Critical	Extremely critical	High
Autonomous vehicles	Highly variable	Extremely critical	Extremely critical	Medium to high

According to [2], expected application categories can be grouped into:

- Massive IoT or massive machine type communications (mMTCs): scenarios involving large numbers of intermittently transmitting devices, emphasizing energy efficiency, extended coverage, and cost-effectiveness. Typical IoT applications include smart metering, healthcare monitoring, and smart building management.

- Ultra-reliable and low-latency communications (URLLCs): applications requiring extremely low latency and high reliability, such as real-time industrial control, advanced collaborative robotics, autonomous vehicles, remote surgeries, and mission-critical tasks. The strict latency constraints prevent the use of long packets, yet high reliability remains crucial.

A distinct category of machine-type communications is classical space communication, including control or monitoring messages between devices in space and ground stations, with varying reliability and latency requirements depending on mission criticality and application type (e.g., deep-space versus near-Earth missions).

Several standards and communication systems have been developed or adapted specifically to support machine-type communications, including:

- LTE-mMTC (3GPP Releases 13 and 14): optimized to support massive numbers of IoT devices, offering efficient communications for applications like smart cities, asset tracking, and environmental monitoring.
- LTE NB-IoT (Narrowband IoT, 3GPP Releases 13 and 14): designed for narrowband communications with extremely low power consumption, exceptional coverage, and reduced costs, ideal for static sensor applications.
- LoRaWAN (Long Range Wide Area Network) [3, 4]: an open-source LPWAN technology known for long-range, low-power consumption, and cost-effectiveness, extensively used in smart cities and agriculture.
- BLE (Bluetooth Low Energy) [5]: highly energy-efficient, supporting applications such as indoor localization, medical devices, and personal area networks.
- WirelessHART, Zigbee, ISA100.11a, Miwi, 6LoWPAN, Thread, SNAP: based on IEEE 802.15.4 [6], these standards enable low-rate wireless personal area networks, primarily in industrial settings.
- Proprietary satellite-based systems (e.g., Solera Omnitracs, and others): supporting various MTC applications, especially in remote or satellite-dependent areas.

Machine-type communications applications inherently rely on short to moderate-length data packets —-ranging from very small to medium sizes-— with stringent requirements on energy efficiency and reliability. This stands in stark contrast to Shannon's framework in the limit of large packets to achieve capacity. The need for finite-length bounds thus arises from the practical limitations of real-world systems, where short packets are essential for low latency and energy-constrained operation. The shared challenge in machine-type communications is to

design efficient finite-length communication schemes that balance the energy required for reliable transmission and reception against the complexity of on-board processing. This has spurred significant advances in finite-length information theory. To precisely quantify the performance of practical codes under these constraints, researchers have developed *finite blocklength bounds* -—an area that has seen renewed interest in recent years. Building on the initial work by Fano (1961) and Gallager (1965) [7, 8], Polyanskiy, Poor and Verdú introduced tighter upper and lower bounds in 2010 [9] to capture the performance limits of the best finite-length codes. Approaching these bounds in a practical context remains an open challenge. In the single-user setting, efforts such as *code concatenation* (first studied by Forney in 1965 [10]) and *list decoding* (introduced by Elias in 1957 [11]) have emerged as promising strategies to approach these theoretical limits with manageable complexity. In parallel, the problem of coordinating a vast number of devices transmitting over a shared channel has led to the development of *unsourced multiple access* [12], which recasts multiuser communication in a "coding-like" formulation to derive finite-blocklength limits in a multiuser context. Despite these advances, designing practical schemes that closely approach finite-length bounds with low complexity remains an open research challenge in both single-user and multiuser settings.

1.2 Original Contributions

This doctoral thesis makes the following original contributions to the field, specifically focusing on concatenated convolutional codes with outer polynomial codes and their list decoding, unsourced multiple access protocols, and frame synchronization in direct-sequence spread spectrum systems. The primary original contributions of this dissertation are detailed below:

- **Analysis of concatenated convolutional codes with outer polynomial codes and their performance under list Viterbi decoding for some existing communication protocols.**

  In Chapter 3, this thesis presents an in-depth investigation into the analysis of concatenated inner convolutional codes with outer polynomial codes and the application of list Viterbi decoding to the concatenated coding scheme of two existing protocols adopting these concatenated codes: the satellite telemetry recommendation of the Consultative Committee for Space Data Systems and the 3GPP long term evolution (LTE) cellular mobile standard. The original contribution of this thesis lies in the analysis of the trellis of the concatenated scheme to derive properties of the distance spectrum of these codes. Furthermore, the use of analytical methods to compute the distance spectrum of the concatenation are explored, aiming at deriving tight upper bounds on the error probability under maximum-likelihood decoding of the concatenated schemes. Through this analysis, the thesis demonstrates and quantifies the significant coding gain for the two existing protocols w.r.t. the decoding of the inner convolutional code only. The gain is quantified in approximately

3 dB and it is achievable by employing list Viterbi decoding over the trellis of the inner convolutional code. Furthermore, the work highlights that this performance enhancement can be realized with moderate list sizes, thus maintaining a practical level of decoding complexity for certain received power values. This contribution underscores the potential for substantial power savings and performance improvements in both terrestrial and space communications systems adopting convolutional codes. These gains are achievable with a manageable increase in base station or ground segment complexity.

- **Design and performance analysis of enhanced spread spectrum Aloha protocol for the unsourced multiple access channel.**

  In Chapter 4, this thesis analyzes the performance of enhanced spread spectrum Aloha (E-SSA) in the framework of unsourced multiple access channel (UMAC). E-SSA is a prominent random access protocol in satellite-based MTC networks. Despite its popularity, little is know about the performance of E-SSA, when compared with recently-introduced UMAC bounds. The primary contribution of this thesis is the adaptation and analysis of E-SSA for the UMAC model, including modifications to its asynchronous transmission scheme to facilitate a direct comparison with framed UMAC protocols. Furthermore, the thesis proposes and evaluates specific enhancements to E-SSA to improve its energy efficiency, notably through the incorporation of a short polar code and a timing channel. The work provides a comprehensive analysis of the trade-offs between the design parameters of E-SSA, receiver complexity, and overall energy efficiency. The results demonstrate that a carefully optimized E-SSA protocol can achieve performance levels competitive with state-of-the-art UMAC schemes, while retaining a simple transmitter architecture and linear receiver complexity in the number of simultaneously transmitting users.

- **Likelihood-based frame synchronization for coherent and non-coherent direct-sequence spread spectrum systems.**

  In Chapter 5, this dissertation presents the analysis of the frame synchronization problem in direct-sequence spread spectrum (DSSS) communication systems, considering both coherent and non-coherent detection scenarios. The frame synchronization problem is cast to a binary-hypotheses testing formulation, and, by leveraging the knowledge of preambles and spreading sequences, the optimal likelihood ratio test is derived using the Neyman-Pearson framework. The performance of the test is then characterized analytically and via simulation for the different scenarios. The results demonstrated that incorporating spreading sequence information into the synchronization process significantly improves detection accuracy compared to traditional approaches that utilizes the preamble only. To address practical implementation constraints, in this thesis we propose and evaluate several approximations of the likelihood ratio tests as computationally efficient alternatives, demonstrating their robust performance and suitability for real-world DSSS communication

systems, even in low power conditions and under phase uncertainty. The proposed test metric is also incorporated into the modified E-SSA protocol for the UMAC setting, further improving its original performance.

1.3 Scientific Publications

The following section provides an overview of the published research contributions during the course of the doctoral study. The listed papers have been published in peer-reviewed journals and conferences. The dissertation is mainly based on the following articles:

[R1] R. Schiavone, R. Garello, and G. Liva, "Application of list Viterbi algorithms to improve the performance in space missions using convolutional codes", in *Proc. 9th International Workshop on Tracking, Telemetry and Command Systems for Space Applications (TT&C)*, Noordwijk, Netherlands, pp. 1–8, November 2022.

[R2] R. Schiavone, R. Garello, and G. Liva, "Performance improvement of space missions using convolutional codes by CRC-aided list Viterbi algorithms", *IEEE Access*, vol. 11, pp. 55925-55937, June 2023.

[R3] R. Schiavone, G. Liva, and R. Garello, "Design and performance of enhanced spread spectrum Aloha for unsourced multiple access".*IEEE Communications Letters*, vol. 28, no. 8, pp. 1790-1794, August 2024.

Other published articles include:

[R4] R. Schiavone, F. Galati, and M. A. Zuluaga, "Binary domain generalization for sparsifying binary neural networks", in *Proc. Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, Torino, Italy, pp. 123–140, September 2023.

[R5] M. Visintin, R. Schiavone, and R. Garello, "Performance analysis of uplink code division multiplexing for LEO satellite constellations under nonlinear power amplifiers", *Sensors*, vol. 24, no. 21-6879, October 2024.

[R6] A. Mirri, D. Forlivesi, R. Schiavone, R. Garello, M. Chiani, and E. Paolini, "A flexible scheme for critical mMTC", in *Proc. IEEE 8th Forum on Research and Technologies for Society and Industry Innovation*, Milano, Italy, pp. 426–431, November 2024.

[R7] R. Garello, M. Visintin, R. Schiavone, A. Compagnoni, and C. F. Chiasserini, "AES and mixed AES/Gold spreading sequences for satellite uplink code division multiplexing", *IEEE Transactions on Communications*, March 2025.

[R8] M. Ferro, R. Schiavone, G. Liva and M. Magarini, "A decoding algorithm for terminated convolutional codes over the blockwise noncoherent channel", in *Proc. 13th International Symposium on Topics in Coding*, Los Angeles, USA, August 2025.

Other articles, currently in preparation, include:

[R8] R. Schiavone, G. Liva, R. Garello, and M. Visintin, "On coherent and non-coherent frame synchronization in direct-sequence spread spectrum systems", submitted to *IEEE Transactions on Communications*.

1.4 Thesis Outline

The material covered in this thesis is organized as follows:

- Chapter 2 is devoted to provide preliminaries on communication models and binary linear codes.
- Basics of convolutional codes, and the analysis of concatenated convolutional codes with outer polynomial codes are presented in Chapter 3, together with the analysis and the application of list decoding of convolutional codes.
- Chapter 4 introduces the unsourced multiple access channel model and it addresses the design and performance of enhanced spread spectrum Aloha protocol in the unsourced multiple access channel framework.
- The study of frame synchronization for direct-sequence spread spectrum systems is carried out in Chapter 5, with applications of new metrics for the improvement of the enhanced spread spectrum Aloha protocol designed in Chapter 4.
- Conclusions follow in Chapter 6.

Chapter 2

Preliminaries

Preliminaries and definitions of the communication model, used as a reference throughout the thesis, are provided in Section 2.1. Basic definitions for binary linear block codes are summarized in Section 2.2.

2.0.1 Notation

Random variables (r.v.s) are denoted by uppercase letters, e.g. X. The corresponding realizations are denoted by lowercase letters, e.g., x. Random vectors are expressed in bold uppercase, e.g., $\mathbf{X} = (X_0, X_1, \ldots)$, whereas the corresponding realizations are in bold lowercase, e.g., $\mathbf{x} = (x_0, x_1, \ldots)$. The symbols $\mathsf{E}[X]$ and $\mathrm{Var}[X]$ denote the expectation and variance of the random variable X, respectively. $\mathcal{N}(\mu, \sigma^2)$ represents the normal distribution with mean μ and variance σ^2, while $\mathcal{U}([a,b])$ represents the uniform distribution over the range $[a,b]$. Given two vectors $\mathbf{a} = (a_0, \ldots, a_{N-1})$ and $\mathbf{b} = (b_0, \ldots, b_{N-1})$, their inner product is denoted as $\langle \mathbf{a}, \mathbf{b} \rangle$. $\Re$ and $\Im$ denote the real part and the imaginary part of a complex scalar or vector, respectively. E_b is the energy per information bit, E_s is the energy per modulated symbol, and (in spread spectrum signals), E_c is the energy per chip. Finally, N_0 is the single-sided noise power spectral density.

2.1 Communication Model

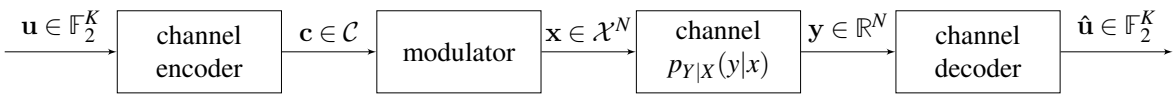


Fig. 2.1 Transmission chain example.

In this thesis, we primarily work with *bits* (binary digits), a term coined by John Wilder Tukey [1]. More generally, we operate within the binary field $\mathbb{F}_2$ (or Galois field of order 2),

which consists of the elements 0 and 1, and the addition operation is defined using the logical XOR operation, whereas the multiplication corresponds to the logical AND operation. The vector space composed of all K-bit sequences, with scalar multiplication and vector (element-wise) addition defined over $\mathbb{F}_2$ is denoted as $\mathbb{F}_2^K$. Sometimes, we will refer to $\mathbb{F}_2^K$ as *Hamming space*. To clarify the setup used, Figure 2.1 shows the transmission chain for the reference user. Let us consider a scenario with one transmitter and one receiver. At the transmitter, we start with a binary information *message* $\mathbf{u} = (u_0, u_1, \ldots, u_{K-1})$ of length K bits. This message is encoded using an (N,K) binary code $\mathcal{C} \subset \mathbb{F}_2^N$. The code $\mathcal{C}$ contains 2^K possible codewords. The encoder maps the message $\mathbf{u}$ to a binary *codeword* $\mathbf{c} = (c_0, c_1, \ldots, c_{N-1})$ of length N bits. The codeword $\mathbf{c}$ is then modulated using binary phase-shift keying (BPSK). In particular, denoting by $\mathcal{X} = \{-1, +1\}$ the channel input alphabet, the modulation maps each binary bit c_i to a symbol x_i as follows:

$$x_i = (-1)^{c_i} = \begin{cases} +1, & \text{if } c_i = 0, \\ -1, & \text{if } c_i = 1. \end{cases}$$

With a slight abuse of notation, we will denote by $\mathcal{C}$ both the set of codewords $\mathbf{c}$ with elements in $\mathbb{F}_2$, and the set of codewords after binary modulation, that is, of the vectors $\mathbf{x} \in \mathcal{X}^N$ that correspond to codewords with elements in $\mathbb{F}_2$. Thus, when writing $\mathbf{x} \in \mathcal{C}$, the set $\mathcal{C}$ has to be intended as the set of modulated codewords. The proper definition of $\mathcal{C}$ will be made clear by the context.

The signal $\mathbf{x}$ is then corrupted by the memoryless additive white Gaussian noise (AWGN) channel, which means that the n-long real-valued received message $\mathbf{y}$, after ideal demodulation, has the i-th component equal to

$$y_i = x_i + z_i,$$

with z_i noise terms modeled as samples of independent and identically distributed (i.i.d.) zero mean Gaussian random variables, $Z_i \sim \mathcal{N}\left(0, \sigma^2\right)$. The probability density function (p.d.f.) of the random variable Y given X is

$$p_{Y|X}(y_i|x_i) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y_i - x_i)^2}{2\sigma^2}}.$$

Then, the aim of the receiver is to output an estimate $\hat{\mathbf{u}} \in \mathbb{F}_2^K$ of the transmitted binary message.

The signal-to-noise ratio (SNR) is defined as the ratio between the signal energy E_s and the single-sided noise power spectral density N_0. We have

$$\text{SNR} = \frac{E_s}{N_0} = \frac{\mathsf{E}_X\left[X^2\right]}{2\sigma^2} = \frac{1}{2\sigma^2} = \frac{1}{N_0}.$$

Here, $\mathsf{E}[\cdot]$ refers to the expected value of a random variable. An alternative SNR definition is provided in terms of E_b/N_0, where E_b is the average energy per information bit, that is $E_s = E_b R$, where $R = K/N$ is the rate of the (N/K) code, hence

$$\frac{E_b}{N_0} = \frac{1}{2R\sigma^2}.$$

2.2 Binary Linear Codes

Definition 2.1: An (N,K) *binary linear code* $\mathcal{C}$ is a *linear* $K-$dimensional subspace of the $N-$dimensional vector space $\mathbb{F}_2^N$.

The *linearity* property implies that the sum of any two elements in $\mathcal{C}$ is also an element of $\mathcal{C}$; as a consequence, the all zero elements vector $\mathbf{0}$ is always an element of a binary linear code $\mathcal{C}$. The elements of $\mathcal{C}$ are called *codewords*.

We can define an (N,K) binary linear block code $\mathcal{C}$ using the *generator matrix* $\mathbf{G}$, a $K \times N$ binary matrix that contains in each row one of the basis vectors of $\mathcal{C}$. Equivalently, the rows of $\mathbf{G}$ span $\mathcal{C}$, which means that

$$\forall \mathbf{c} \in \mathcal{C}, \exists \mathbf{u} \in \mathbb{F}_2^K : \mathbf{c} = \mathbf{u}\mathbf{G}.$$

Alternatively, a binary linear block code can be defined by its $(N-K) \times N$ *parity check matrix* $\mathbf{H}$, a matrix whose null space is $\mathcal{C}$, i.e., $\mathbf{G}\mathbf{H}^T = \mathbf{0}$, where $\mathbf{0}$ denotes the all zero elements matrix or vector, which means that

$$\mathbf{c}\mathbf{H}^T = \mathbf{0} \;\; \forall \mathbf{c} \in \mathcal{C}.$$

The rows of $\mathbf{H}$ span the subspace orthogonal to $\mathcal{C}$, denoted as $\mathcal{C}^\perp$. The vectors in $\mathcal{C}^\perp$ form a $(N, N-K)$ binary linear code called the *dual code*.

Definition 2.2: Given a vector $\mathbf{v} \in \mathbb{F}_2^N$, its *syndrome* w.r.t. $\mathbf{H}$ is given by $\mathbf{s} = \mathbf{v}\mathbf{H}^T$, and by definition $\mathbf{s}$ is $\mathbf{0}$ if and only if $\mathbf{v} \in \mathcal{C}$.

Definition 2.3: Given a binary vector $\mathbf{v} \in \mathbb{F}_2^N$, its *Hamming weight* $w_H(\mathbf{v})$ corresponds to the number of non-zero entries in the vector

$$w_H(\mathbf{v}) = |\{i | v_i \neq 0\}|.$$

Definition 2.4: The *Hamming distance* between two binary vectors $d_H(\mathbf{v_1}, \mathbf{v_2})$ is defined to be the Hamming weight of the sum over the binary field of the two vectors,

$$d_H(\mathbf{v_1}, \mathbf{v_2}) = w_H(\mathbf{v_1} + \mathbf{v_2}).$$

An important characterization of a code is its minimum distance $d_{\min}$, since it provides information about the error detection and error correction capabilities of the code.

Definition 2.5: The *minimum distance* $d_{\min}$ of a binary linear code $\mathcal{C}$ is

$$d_{\min}(\mathcal{C}) = \min_{\substack{\mathbf{c} \neq \mathbf{c}' \\ \mathbf{c}, \mathbf{c}' \in \mathcal{C}}} d_H(\mathbf{c}, \mathbf{c}') = \min_{\substack{\mathbf{c} \neq \mathbf{0} \\ \mathbf{c} \in \mathcal{C}}} w_H(\mathbf{c}). \tag{2.1}$$

A more complete characterization of the distance properties of a linear block code is given by its weight enumerator (WE) or *distance spectrum*.

Definition 2.6: The WE of an (N, K) linear block code $A_d(\mathcal{C})$ corresponds to the multiplicity of all codewords in $\mathcal{C}$ with Hamming weight equal to d,

$$A_d(\mathcal{C}) = |\{\mathbf{c} \in \mathcal{C}, w_H(\mathbf{c}) = d\}|. \tag{2.2}$$

Hereafter, we will refer to $A_d(\mathcal{C})$ as A_d. The WE can also be expressed via the weight enumerating function (WEF) $A(\mathtt{X})$, a polynomial whose coefficient of power d is A_d, i.e.,

$$A(\mathtt{X}) = \sum_{d=0}^{N} A_d \mathtt{X}^d \implies \text{coeff}\left(A(\mathtt{X}), \mathtt{X}^d\right) = A_d. \tag{2.3}$$

Different criteria can be applied at the channel decoder to decide which codeword was transmitted. In this work, we do assume that the transmitted codewords have all the same probability, and our target is to minimize the block error probability P_B,

$$P_B = P(\hat{\mathbf{X}} \neq \mathbf{X}),$$

where $\mathbf{X}$ is the r.v. associated with the transmitted modulated codeword $\mathbf{x}$, whereas $\hat{\mathbf{X}}$ is the r.v. associated with the decoder decision $\hat{\mathbf{x}}$. The rule that minimizes P_B under the assumption that all codewords are transmitted with uniform probability is called the block-wise maximum likelihood (ML) decoding rule.

Definition 2.7: For the binary-input additive white Gaussian noise (bi-AWGN) channel model, the ML decision $\hat{\mathbf{x}}_{ML}$ is given by

$$\begin{aligned}\hat{\mathbf{x}}_{ML} &= \arg\max_{\mathbf{x}\in\mathcal{C}} p_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}) \\ &= \arg\max_{\mathbf{x}\in\mathcal{C}} \prod_{i=0}^{N-1} p_{Y|X}(y_i|x_i) \\ &= \arg\min_{\mathbf{x}\in\mathcal{C}} d_E^2(\mathbf{y},\mathbf{x}) \\ &= \arg\max_{\mathbf{x}\in\mathcal{C}} \langle \mathbf{y},\mathbf{x}\rangle .\end{aligned}$$

Here, $d_E^2(\cdot,\cdot)$ and $\langle\cdot,\cdot\rangle$ represent the squared Euclidean distance and the inner product between their two arguments, respectively. A consequence of this decision rule is that all the received outputs $\mathbf{y}$ will be assigned to the $\mathbf{x}$ vector that maximizes the likelihood $p_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x})$. Thus, the space $\mathbb{R}^N$ is partitioned in so-called decision regions or Voronoi regions [13]. In particular, the decision region of a codeword $\mathbf{x}$ is defined as

$$\mathcal{D}(\mathbf{x}) = \left\{ \mathbf{y} \,\middle|\, p_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}) > p_{\mathbf{Y}|\mathbf{X}}\left(\mathbf{y}\middle|\mathbf{x}'\right), \forall \mathbf{x}' \in \mathcal{C}\backslash\mathbf{x} \right\}.$$

For binary linear block codes, over the bi-AWGN channel, these regions have the same geometry [14], which implies that the block error probability is independent of the transmitted codeword. Hence, by fixing $\mathbf{x}$ as the transmitted codeword, over the bi-AWGN channel, we can compute an upper bound to the block error probability under ML decoding of a linear code with WE A_d by means of the union bound (UB)

$$\begin{aligned}P_B = P(\hat{\mathbf{X}} \neq \mathbf{x}|\mathbf{x}\text{ transmitted}) &= P\left(\bigcup_{\mathbf{x}'\in\mathcal{C}\backslash\mathbf{x}} \left\{\mathbf{x}\to\mathbf{x}'\right\}\right) \\ \leq P_{UB} = \sum_{\mathbf{x}'\in\mathcal{C}\backslash\mathbf{x}} P\left(\left\{\mathbf{x}\to\mathbf{x}'\right\}\right) &= \frac{1}{2}\sum_{d=d_{\min}}^{N} A_d \operatorname{erfc}\left(\sqrt{d\,R\,\frac{E_b}{N_0}}\right)\end{aligned} \tag{2.4}$$

where we introduced the shorthand

$$\left\{\mathbf{x}\to\mathbf{x}'\right\} \triangleq \left\{ p_{\mathbf{Y}|\mathbf{X}}\left(\mathbf{Y}|\mathbf{x}'\right) \geq p_{\mathbf{Y}|\mathbf{X}}(\mathbf{Y}|\mathbf{x}) \right\}.$$

The term $P\left(\left\{\mathbf{x}\to\mathbf{x}'\right\}\right)$ is known as *pairwise error probability*. Moreover, in (2.4) $\operatorname{erfc}(\cdot)$ denotes the complementary error function defined as

$$\operatorname{erfc}(a) = 1 - \frac{2}{\sqrt{\pi}}\int_0^a e^{-b^2}\mathrm{d}b. \tag{2.5}$$

When the SNR is large, by using only the $A_{d_{\min}}$ term in the UB, we can obtain a tight approximation to the performance of the linear code as

$$P_B \approx \frac{1}{2} A_{d_{\min}} \operatorname{erfc}\left(\sqrt{d_{\min} R \frac{E_b}{N_0}}\right).$$

Chapter 3

Analysis and Performance of Convolutional Code-Based Protocols

This Chapter addresses the performance analysis of inner convolutional codes concatenated with outer polynomial codes. The problem of designing good performing codes for short blocklength is introduced in Section 3.1. Section 3.2 reviews the preliminaries of convolutional codes, followed by Section 3.3 which is devoted to the concatenation of inner convolutional codes with outer polynomial codes, and the decoding via list Viterbi algorithms. Then, the analysis and performance improvement of existing protocols based on convolutional codes is carried out in Section 3.4 for satellite systems, and in Section 3.5 for LTE systems. The improvements are based on incorporating the outer polynomial codes, already included by the protocols for error detection, into the error correction mechanism using list Viterbi decoding.

3.1 Coding for Short Block Codes

As machine-type communications (MTCs) entail the transmission of short packets, it is necessary to consider performance limits that account for finite blocklength effects when benchmarking different channel codes. When finite blocklength is considered, performance benchmarks can be obtained via lower and upper bounds on the block error probability achievable by the best codes. Notable examples include Shannon's 1959 sphere packing bound [15] – a lower bound on the block error probability – and Gallager's random coding bound [8], an upper bound that becomes tight when the blocklength is large (e.g. $N > 10^5$ bits). More recently, a new class of bounds was derived in [9], which has been shown to be tight down to short blocklengths. In what follows, we review the random-coding union (RCU) [9], an upper bound on the block error probability achievable by the best (N, K) block code.

Theorem 3.1 (RCU bound [9, Theorem 16]): For an arbitrary $p_{\mathbf{X}}(\mathbf{x})$, there exists an (N,K) code $\mathcal{C}$ such that, under ML decoding,

$$P_B(\mathcal{C}) \leq \mathsf{E}\left[\min\left\{1, \left(2^K-1\right) P\left(i\left(\bar{\mathbf{X}};\mathbf{Y}\right) \geq i\left(\mathbf{X};\mathbf{Y}\right) |\mathbf{X},\mathbf{Y}\right)\right\}\right],$$

with $p_{\mathbf{X},\mathbf{Y},\bar{\mathbf{X}}}(\mathbf{x},\mathbf{y},\bar{\mathbf{x}}) = p_{\mathbf{X}}(\mathbf{x})p_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x})p_{\bar{\mathbf{X}}}(\bar{\mathbf{x}})$, where $i(\mathbf{x};\mathbf{y})$ denotes the *information density*,

$$i(\mathbf{x};\mathbf{y}) = \log_2 \frac{p_{\mathbf{X},\mathbf{Y}}(\mathbf{x},\mathbf{y})}{p_{\mathbf{X}}(\mathbf{x})p_{\mathbf{Y}}(\mathbf{y})}.$$

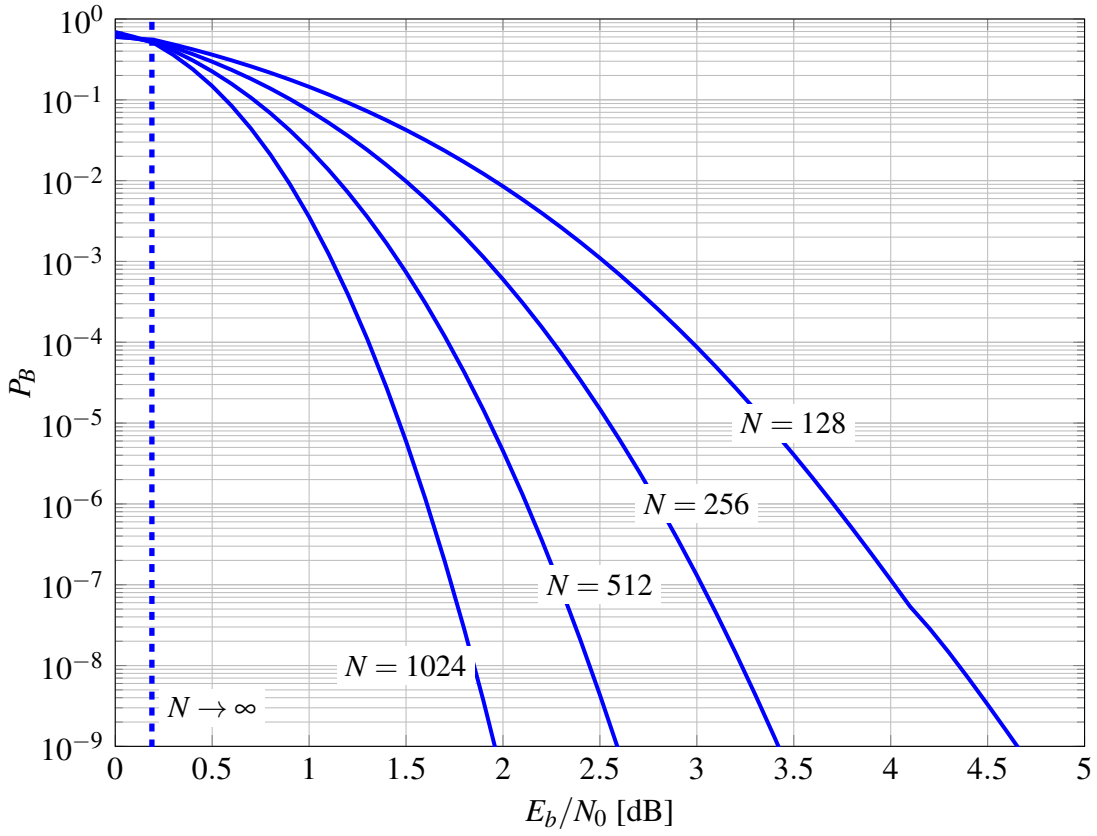


Fig. 3.1 RCU bound on the block error probabilities for rate $R = 1/2$ binary codes over the bi-AWGN channel for different blocklengths versus Shannon's channel capacity.

Figure 3.1 shows the RCU bound on the block error probabilities for rate $R = 1/2$ binary codes over the bi-AWGN channel for different blocklengths (128, 256, 512, 1024) and the channel capacity predicted by Shannon at $E_b/N_0 = 0.18$ dB. As can be seen, the required E_b/N_0 to achieve a given block error probability increases as the blocklength decreases. Moreover, the performance of short blocklength codes deviates significantly from Shannon's capacity prediction. In very short blocklength scenarios (e.g., 128 bits), well-established iterative coding schemes such as low-density parity-check (LDPC) codes [16] and turbo codes [17] – which typically perform very well at moderate and long blocklengths (e.g., above 1000 bits) – exhibit

a notable performance degradation of more than 1 dB relative to the RCU bound [18]. In contrast, convolutional codes (CCs) [19] have demonstrated remarkable performance for very short blocklengths, approaching the RCU bound, though at the expense of increased decoder complexity [20]. Given the rising interest in short blocks for MTC, considerable effort was placed, during the past few years, in approaching the bounds at short blocks with low complexity.

A prominent technique to address this regime is to adopt a concatenation scheme consisting of an inner linear block code (e.g., polar [21] or convolutional code) with an outer binary linear code, with decoding performed via list decoding techniques [22, 23], as depicted in Figure 3.2.

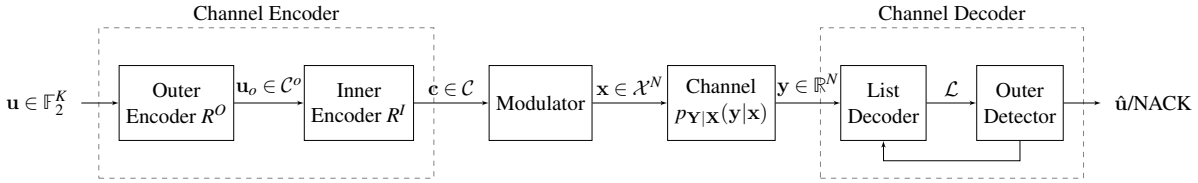


Fig. 3.2 List decoding example.

3.2 Basic of Convolutional Codes

3.2.1 Convolutional Encoders

We will focus only on binary convolutional codes (CCs). Let $\mathbf{u}_t$ and $\mathbf{c}_t$ be the encoder input and the encoder output at time t, respectively. Then

$$\mathbf{c}_t = \sum_{i=0}^{\nu} \mathbf{u}_{t-i}\mathbf{G}_i, \quad \mathbf{G} = \begin{pmatrix} \mathbf{G}_0 & \mathbf{G}_1 & \dots & \mathbf{G}_\nu & & \\ & \mathbf{G}_0 & \mathbf{G}_1 & \dots & \mathbf{G}_\nu & \\ & & \ddots & \ddots & & \ddots \end{pmatrix},$$

where $\mathbf{G}_i$, $i = 0, \dots, \nu$ are $k \times n$ binary matrices and ν is the encoder memory; alternatively, we say that the code has a constraint length $(\nu + 1)$. The encoder can be implemented using ν memory cells; an example is given in Figure 3.3.

Definition 3.1: The *free distance* of an (n, k, ν) binary CC $\mathcal{C}$ is the minimum Hamming distance between two distinct sequences produced by its encoder,

$$d_{\text{free}}(\mathcal{C}) = \min_{\substack{\mathbf{c} \neq \mathbf{c}' \\ \mathbf{c}, \mathbf{c}' \in \mathcal{C}}} d_H(\mathbf{c}, \mathbf{c}'). \tag{3.1}$$

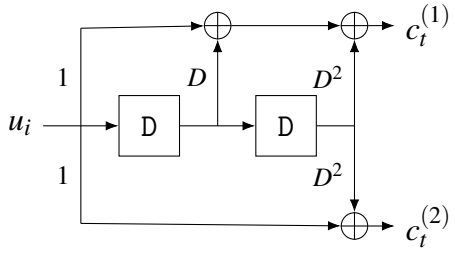


Fig. 3.3 Convolutional encoder for an $(n=2,k=1,\nu=2)$ code with $\mathbf{G}(D)=[1+D+D^2,1+D^2]$. In the block diagram, `D` represents a single delay block.

The largest possible *free distance* of a CC depends on its memory ν, as shown by well-known upper bounds to the largest possible free distance for a given memory ν CC, such as the Heller's upper bound [24] and the tighter Griesmer's upper bound [25].

Theorem 3.2 (Heller's bound [26, Corollary 3.18]): The free distance d_{free} for any (n,k,ν) binary CC satisfies

$$d_{\text{free}} \leq \min_{i\geq 1} \left\lfloor \frac{(\nu+i)n}{2\left(1-2^{-ki}\right)} \right\rfloor . \tag{3.2}$$

The nominal rate of a CC encoder is $R_0 = k/n$. Next, we will focus only on rate-$1/2$ convolutional encoders with $k=1$, i.e., $(2,1,\nu)$ CCs.

We can view the CC encoder as a finite state machine (FSM) with 2^ν states (all possible binary values contained in the ν memory cells) and 2 possible transitions from each state. We can represent such FSM using a *state-transition diagram* as in Figure 3.4b, or its equivalent *state-transition table* as in Figure 3.4a. We denote the states with σ_i, where i is the decimal representation of the content of the memory cells of the encoder, i.e. if $\nu=2$, then $\sigma_0=(0,0)$, $\sigma_1=(0,1)$, $\sigma_2=(1,0)$, and $\sigma_3=(1,1)$. The input-output relationships of a convolutional encoder can be represented also using the so-called D-transform [26]. The sequence $\mathbf{x} = (\ldots,\mathbf{x}_{-1},\mathbf{x}_0,\mathbf{x}_1,\mathbf{x}_2,\ldots)$ has D-transform

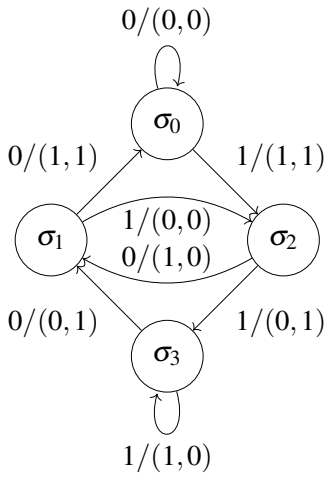
$$\mathbf{x}(D) = \sum_t \mathbf{x}_t D^t = \ldots + \mathbf{x}_{-1}D^{-1} + \mathbf{x}_0 + \mathbf{x}_1 D + \mathbf{x}_2 D^2 + \ldots$$

with $\mathbf{x}_t = \left(x_t^{(0)}, x_t^{(1)}, \ldots, x_t^{(n-1)}\right)$. With reference to the example of Figure 3.3, the encoder equations

$$\begin{aligned}\mathbf{c}_t &= \left[c_t^{(0)}, c_t^{(1)}\right] \\ &= [u_t + u_{t-1} + u_{t-2}, u_t + u_{t-2}]\end{aligned}$$

Current State (u_{t-1}, u_{t-2})	**Input** u_t	**Next State** (u_t, u_{t-1})	**Output** $\mathbf{c}_t = \left(c_t^{(0)}, c_t^{(1)}\right)$
$\sigma_0 = (0,0)$	0	$\sigma_0 = (0,0)$	$(0,0)$
	1	$\sigma_2 = (1,0)$	$(1,1)$
$\sigma_1 = (0,1)$	0	$\sigma_0 = (0,0)$	$(1,1)$
	1	$\sigma_2 = (1,0)$	$(0,0)$
$\sigma_2 = (1,0)$	0	$\sigma_1 = (0,1)$	$(1,0)$
	1	$\sigma_3 = (1,1)$	$(0,1)$
$\sigma_3 = (1,1)$	0	$\sigma_1 = (0,1)$	$(0,1)$
	1	$\sigma_3 = (1,1)$	$(1,0)$

(a) State transition table



(b) State transition diagram

Fig. 3.4 State transition table (a) and state transition diagram (b) of the $(n = 2, k = 1, \nu = 2)$ convolutional code with $\mathbf{G}(D)$=$[1 + D + D^2, 1 + D^2]$.

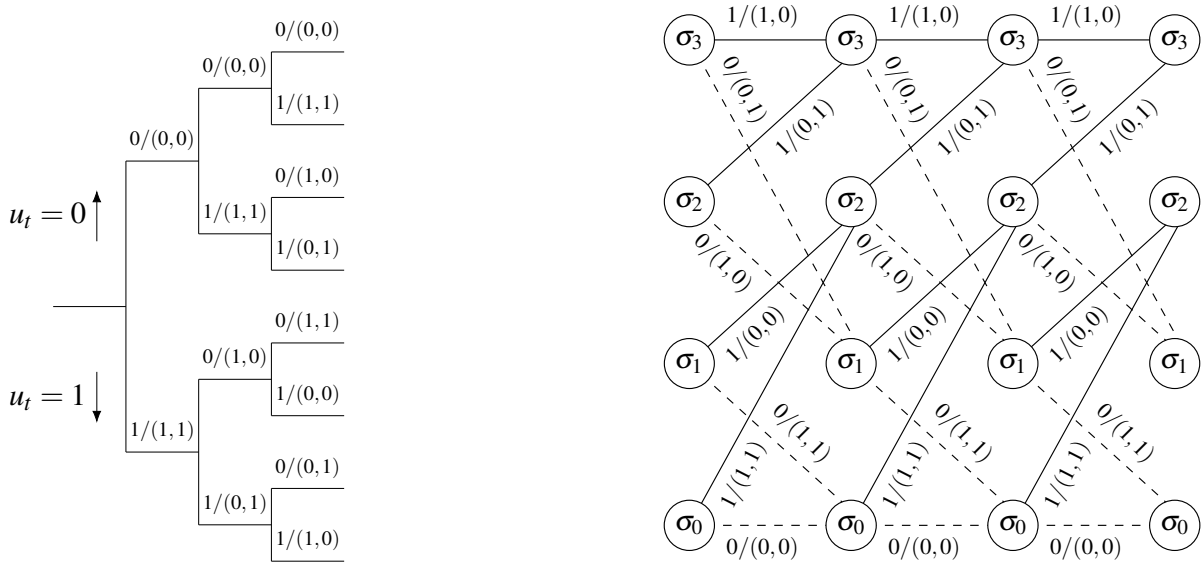
can be expressed as

$$\begin{aligned} c(D) &= \left[c^{(0)}(D), c^{(1)}(D)\right] \\ &= u(D)\left[(1 + D + D^2), (1 + D^2)\right] \\ &= u(D)\,\mathbf{G}(D), \end{aligned}$$

where $\mathbf{G}(D) = [(1 + D + D^2), (1 + D^2)]$ is called the *transfer function*. In the general case, where the convolutional encoder can be recursive, each entry of the transfer function can be written as a rational polynomial function in the form $f(D)/q(D)$. When we encode an information sequence $\mathbf{u}$ of length K by a CC encoder, we can graphically represent the space of possible transmitted sequences $\mathbf{c}_t$ through a *tree* structure (Figure 3.5a), or in a more compact way with a *trellis* representation (Figure 3.5b) [27]. Each *tree* node at depth t is split into 2 different branches, according to all possible information sequences u_t. The *trellis* diagram has K sections, each having 2^ν nodes, and each node at time t is connected to the nodes in the following section according to the rules specified by the state-transition diagram.

The encoding of a CC can be terminated to obtain an (N, K) linear block code, where N is the number of encoded bits. Among the possible termination strategies, we consider the following ones.

Zero-Tail Terminated Convolutional Codes An (N, K) zero-tail terminated convolutional code (ZTCC) has $N = (K + \nu)/R_0$, and its codebook is built from all the paths that at $t = 0$ start from σ_0, the all-zero state, and terminate at time $t = K + \nu$ in the same state σ_0. In practice, the encoder state is forced into the all-zero state by injecting ν consecutive zeros into the encoder,



(a) Tree diagram

(b) Trellis diagram

Fig. 3.5 Tree representation (a) and trellis representation (b) for $K = 3$ of the $(2,1,2)$ convolutional code with $\mathbf{G}(D)$=$[1+D+D^2, 1+D^2]$. In the trellis diagram, dotted edges correspond to transitions caused by an information bit 0, whereas solid edges are associated with transitions caused by an information bit 1.

after encoding the K information bits (see Figure 3.6a). This means that the code rate of the resulting block code is

$$R = \frac{K}{N} = R_0 \frac{K}{K+\nu}$$

i.e., it is smaller than the nominal rate of the CC encoder, R_0. Moreover, for a sufficiently long information sequence K, $R \to R_0$.

The generator matrix of the (N,K) block code is

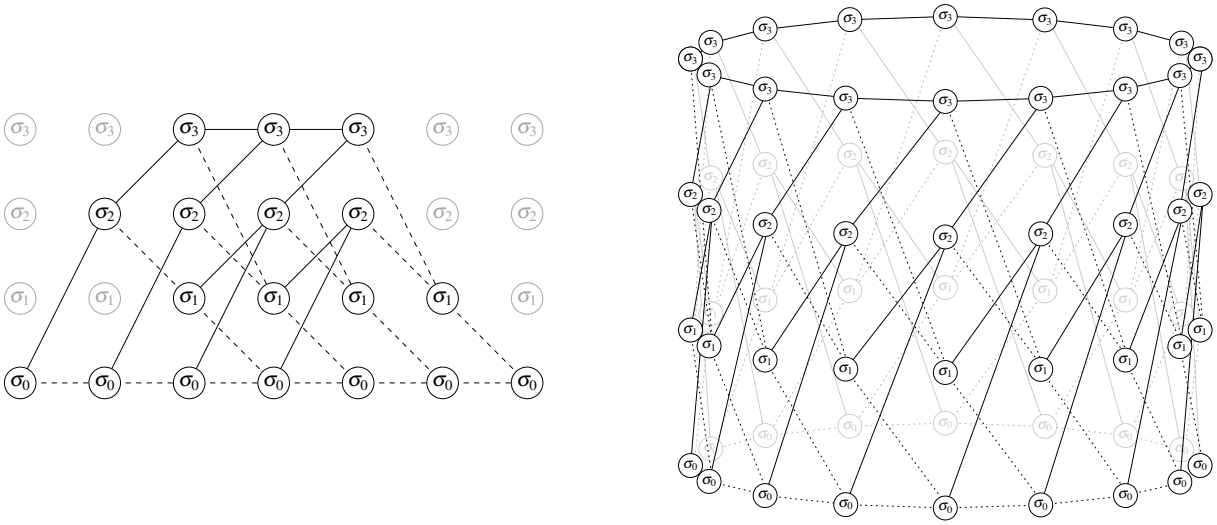
$$\mathbf{G}_{ZT} = \begin{pmatrix} \mathbf{G}_0 & \mathbf{G}_1 & \dots & \mathbf{G}_\nu & & \\ & \mathbf{G}_0 & \mathbf{G}_1 & \dots & \mathbf{G}_\nu & \\ & & \ddots & \ddots & & \ddots \\ & & & \mathbf{G}_0 & \mathbf{G}_1 & \dots & \mathbf{G}_\nu \end{pmatrix}.$$

Tail-Biting Terminated Convolutional Codes An (N,K) tail-biting terminated convolutional code (TBCC) with $N = K/R_0$ is defined as the set of all paths over the trellis whose starting and ending states are the same. Often, the trellis is drawn in a circular way. An example of a circular trellis is shown in Figure 3.6b. In this case, the coding rate is equal to the CC nominal rate, i.e.,

$$R = K/N = R_0.$$

The generator matrix of the (N,K) block code is

$$\mathbf{G}_{TB} = \begin{pmatrix} \mathbf{G}_0 & \mathbf{G}_1 & \dots & \mathbf{G}_\nu & & & \\ & \mathbf{G}_0 & \mathbf{G}_1 & \dots & \mathbf{G}_\nu & & \\ & & \ddots & \ddots & & \ddots & \\ & & & \mathbf{G}_0 & \mathbf{G}_1 & \dots & \mathbf{G}_\nu \\ \mathbf{G}_\nu & & & & \mathbf{G}_0 & \dots & \mathbf{G}_{\nu-1} \\ \vdots & \ddots & & & & \ddots & \vdots \\ \mathbf{G}_1 & \dots & \mathbf{G}_\nu & & & & \mathbf{G}_0 \end{pmatrix}.$$



(a) ZTCC (b) TBCC

Fig. 3.6 Terminated trellises of a memory $\nu = 2$ convolutional code.

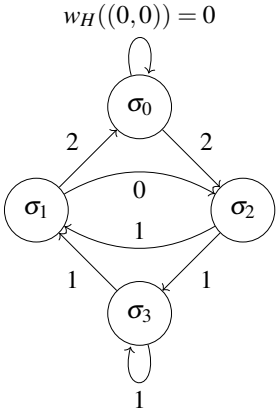
3.2.2 Weight Enumerator of Terminated Convolutional Codes

To find the weight enumerator (WE) of a terminated CC we can use the *state-transition matrix* $\mathbf{T}$ of its convolutional encoder. The (i,j) element of $\mathbf{T}$ is $T_{i,j} = \mathtt{X}^d$ where d is the Hamming weight of the encoder output for the transition $\sigma_i \to \sigma_j$. An example of such a matrix is shown in Figure 3.7.

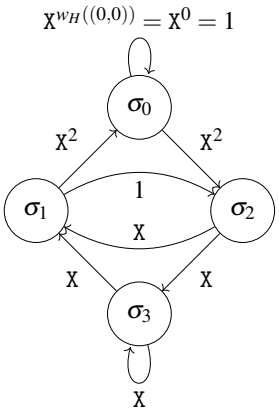
The weight enumerating function (WEF) of a ZTCC can be obtained as

$$A(\mathtt{X}) = \sum_{d=0}^{N} A_d \cdot \mathtt{X}^d = \left(\mathbf{T}^{K+\nu}\right)_{0,0} \tag{3.3}$$

where $\left(\mathbf{T}^{K+\nu}\right)_{0,0}$ is the entry in the first row and column of power $K+\nu$ of the matrix $\mathbf{T}$. The WEF of a TBCC is instead



(a) State transition diagram with weights



(b) State transition diagram with polynomials

$$\mathbf{T} = \begin{pmatrix} 1 & 0 & \mathtt{X}^2 & 0 \\ 0 & \mathtt{X} & 0 & \mathtt{X} \\ \mathtt{X}^2 & 0 & 1 & 0 \\ 0 & \mathtt{X} & 0 & \mathtt{X} \end{pmatrix}$$

(c) State transition matrix

Fig. 3.7 State transition diagram with the Hamming weight of the corresponding output vector (a), with corresponding output monomial (b) and the state transition matrix (c) of the CC with $\mathbf{G}(D)$=$[1+D+D^2, 1+D^2]$.

$$A(\mathtt{X}) = \sum_{d=0}^{N} A_d \, \mathtt{X}^d \tag{3.4}$$

$$= \mathrm{tr}\left(\mathbf{T}^K\right). \tag{3.5}$$

The trellis diagram can be exploited to reduce the computation of the WEF through simple sums of polynomials and multiplications of polynomials and monomials [28]. An example of the algorithm over the trellis can be found in Appendix A.

It is important to note that, due to the exponential number of states w.r.t. the encoding memory ν, it is impossible to compute the full weight enumerator using this approach, even for moderate memory sizes. One way to mitigate this problem is to compute only the first terms of the WEF, by exploiting properties of the state diagram [28–31].

3.2.3 Decoding of Terminated Convolutional Codes

The Viterbi Algorithm

The Viterbi algorithm (VA) is an efficient ML decoding algorithm that can be conveniently described over the code trellis [32]. Its optimality [33] holds for general memory-less channels

where block-wise likelihood $p_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x})$ can be factored as

$$p_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}) = \prod_{i=0}^{N-1} p_{Y|X}(y_i|x_i). \tag{3.6}$$

For ZTCCs, the algorithm stores at each trellis section t and each state σ_i a *cumulative path metric* $\Lambda_t^{\sigma_i}$, and at every edge from every state σ_i at section t to every connected state σ_j at section $(t+1)$ a weight $\lambda_t^{\sigma_j \to \sigma_j}$ called *branch metric*, where

$$\lambda_t^{\sigma_j \to \sigma_i} = \log p_{\mathbf{Y}|\mathbf{X}} \left(\mathbf{y}_t | \mathbf{x}_t^{\sigma_i \to \sigma_j} \right)$$

with $\mathbf{x}_t^{\sigma_i \to \sigma_j}$ being the output symbols when the encoder transitions from state σ_i to state σ_j. The cumulative metric $\Lambda_t^{\sigma_i}$ is computed recursively, starting from state σ_0 at section $t = 0$ with $\Lambda_0^{\sigma_0} = 0$, and following the rule

$$\Lambda_t^{\sigma_j} = \max_{\sigma_i \in \mathcal{I}_{\sigma_j}} \left(\Lambda_{t-1}^{\sigma_i} + \lambda_t^{\sigma_i \to \sigma_j} \right), \tag{3.7}$$

where $\mathcal{I}_{\sigma_j}$ is the set of states σ_i that lead to state σ_j. The information about the state σ_i at time $t-1$ that maximizes the metric (3.7) is also stored at σ_j. At $t = K + \nu$, the ML path and its corresponding information message estimate $\hat{\mathbf{u}}$ are reconstructed by traversing backward the trellis along the stored state sequence, starting from state σ_0, because we know that the final state of the encoder corresponds to σ_0. For ZTCCs, the complexity of the VA is proportional to the number of states, 2^ν, and linear in the information size K.

In principle, to implement ML decoding on TBCCs, one should

1. Run the VA for each *subtrellis* with the same initial and final state σ_i, which means to run Viterbi 2^ν times.
2. Select the most likely path among all the 2^ν found most likely paths of the different subtrellises.

Such algorithm becomes impractical for TBCCs with already small memory ν. In practice, near-ML, but suboptimal algorithms are used, like the wrap-around Viterbi algorithm (WAVA) [34].

Wrap-Around Viterbi Algorithm

The WAVA consists of a round of the VA over the trellis, starting at time $t = 0$ from all possible 2^ν states σ_i with $\Lambda_0^{\sigma_i} = 0$. When reaching the last section, a check is performed to verify if the ML path is also tail-biting (i.e., if the initial and final states coincide). If the ML path is

tail-biting, we output the corresponding input message $\hat{\mathbf{u}}$. Otherwise, we re-initialize the metrics $\Lambda_{t=0}^{\sigma_i} = \Lambda_K^{\sigma_i}$ and re-run the VA. This process is repeated until the ML tail-biting path is found or a maximum number of iterations is reached. If the algorithm does not converge to a valid decision, an erasure (NACK) is declared.

3.3 Concatenation of Convolutional Codes with Outer Polynomial Codes

3.3.1 Polynomial Codes

Definition 3.2 ([35]): A binary linear *polynomial code* $\mathcal{C}$ is a $(K+m, K)$ code for which the D-transform polynomials of its codewords are all divisible without remainder by the same degree-m generator polynomial $g(D)$

$$g(D) = g_0 + g_1 D + g_2 D^2 + \ldots + g_m D^m = \sum_{i=0}^{m} g_i D^i. \tag{3.8}$$

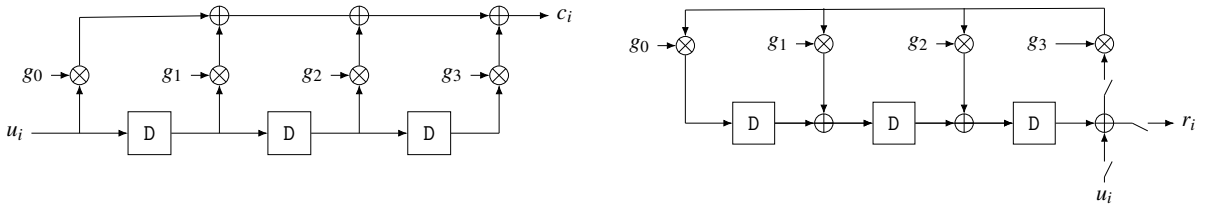
Polynomial codes are often used for *error detection*. From the generator polynomial $g(D)$, the same codebook can be constructed in two different ways, that is

$$c(D) = u(D)g(D), \tag{3.9}$$

which yields a non systematic encoder, and

$$\begin{aligned} c(D) &= u(D)D^m + r(D) = q(D)g(D), \\ r(D) &= u(D)D^m \bmod g(D), \end{aligned} \tag{3.10}$$

which gives a systematic encoding rule. In (3.10), $r(D)$ and $q(D)$ are referred to as the *remainder* and the *quotient* of the division between $u(D)D^m$ and $g(D)$. Note that $\deg(r(D)) < m$, $\deg(q(D)) \leq K$, $\deg(u(D)) \leq K$, and $\deg(u(D)D^m) \leq K+m$. The equations correspond to the two binary linear circuits [36] shown in Figure 3.8. Polynomial codes are often erroneously referred to as cyclic redundancy check (CRC) codes [37], however, as Peterson pointed out in [35, 37], CRC codes are *cyclic* polynomial codes or *shortened* cyclic polynomial codes [38], which is only a subset of the more general polynomial code class.



(a) Non-systematic circuit

(b) Systematic circuit, to compute the remainder $r(D)$.

Fig. 3.8 Example of polynomial encoder with degree-3 generator polynomial $g(D)$.

Definition 3.3: A *cyclic code* is a binary linear code with the added property that the circular shift of any codeword is also a codeword. An (N, K) polynomial code is cyclic if $g(D)$ divides without remainder $(D^N - 1)$.

3.3.2 Convolutional Codes with Outer Polynomial Codes

The method of combining an inner terminated CC with an outer polynomial code has gained widespread acceptance in the context of protocols with packet retransmission policies, i.e., automatic retransmission request (ARQ) protocols [39]. Typically, the CC acts as an inner error-correcting code, while the polynomial code functions as an outer error-detecting code that validates the correctness of the decoded codeword. Alternatively, the outer polynomial code can be used to improve the error correction capability. This can be achieved by (1) treating the concatenation of the polynomial code with the terminated CC as a binary linear block code, and (2) by leveraging list decoding of the inner CC. This approach was introduced in [40, 23] and later optimized in [41–53] where efficient decoding through the utilization of list Viterbi algorithms (LVAs) [54] was demonstrated. Hereafter, we refer to the code obtained by concatenating an inner terminated CC with an outer polynomial code as "poly+CC" code.

3.3.3 List Viterbi Decoding

We are now going to review some decoders which do not output only the path over the trellis that maximizes the likelihood, but a list $\mathcal{L}$ containing the L most likely paths. They are based on the VA, for this reason they are called LVAs. They were first introduced in [54]. We may distinguish two types of LVAs: the parallel-list Viterbi algorithm (PLVA), which outputs at once the full list, and the serial-list Viterbi algorithm (SLVA), which works sequentially, and outputs at iteration i the $i-$th most likely path.

Parallel-List Viterbi Algorithm

PLVA requires a minor modification to the VA, where at every state σ_i and at every trellis section t, we do not store only the cumulative metric $\Lambda_t^{\sigma_i}$ and information about the ML path reaching that state, but we also store the metrics and the relevant information of the L most likely paths reaching that state [54]. The sorted list of the L-th most likely paths at section t for state σ_i is denoted by $\mathcal{L}_t^{\sigma_i}$. The cumulative metric of its ℓ-th element, corresponding to the path with the ℓ-th largest likelihood, is denoted by $\Lambda_t^{\sigma_i,(\ell)}$ and it is

$$\Lambda_t^{\sigma_j,(\ell)} = \max_{\substack{\sigma_i \in \mathcal{I}_{\sigma_j} \\ z=1,\dots,L}}^{(\ell)} \left(\Lambda_{t-1}^{\sigma_i,(z)} + \lambda_{t-1}^{\sigma_j \to \sigma_i} \right) \tag{3.11}$$

where $\max^{(\ell)}$ is an operator that returns the ℓ-th largest argument. Figure 3.9 shows an example of the decoding step of the algorithm. In practice, the $2L$ metrics $\mathcal{L}_{t-1}^{\sigma_a} + \lambda_t^{\sigma_a \to \sigma_c}$ and $\mathcal{L}_{t-1}^{\sigma_b} + \lambda_t^{\sigma_b \to \sigma_c}$ in the figure are sorted from largest to smallest, and the first L (largest) values are sorted in $\mathcal{L}_t^{\sigma_c}$. For ZTCCs, the final list $\mathcal{L}$ containing the L most likely paths over the trellis corresponds to $\mathcal{L}_{K+\nu}^{\sigma_i}$, whereas for TBCCs $\mathcal{L}$ contains the L most likely paths among the $2^\nu L$ paths in $\left\{ \mathcal{L}_K^{\sigma_0}, \mathcal{L}_K^{\sigma_1}, \dots, \mathcal{L}_K^{\sigma_{2^\nu-1}} \right\}$. When PLVA is used to decode poly+CCs, the decoder returns the information message $\hat{\mathbf{u}}$ associated with the most likely path in $\mathcal{L}$ that satisfies the parity-check constraints of the polynomial code, if any. For TBCCs, the decoder will return the most likely tail-biting path in $\mathcal{L}$ that satisfies the parity-check constraints of the polynomial code, if any.

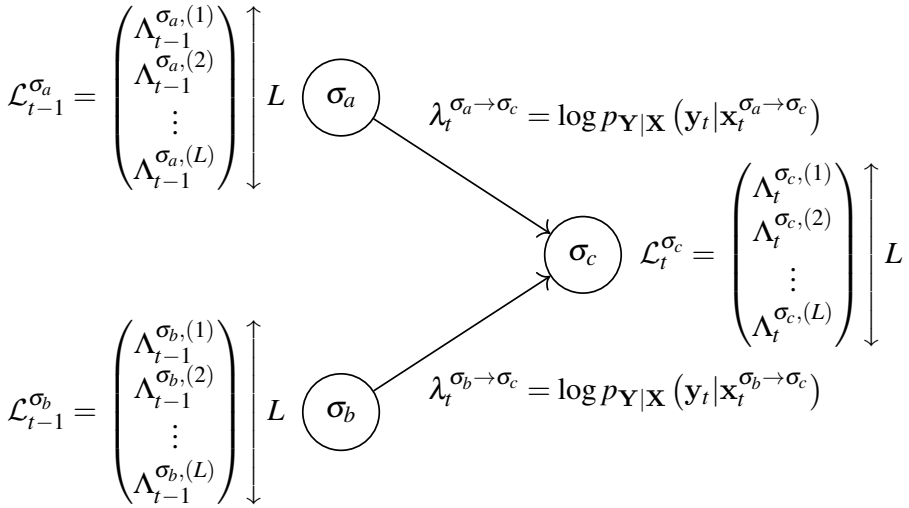


Fig. 3.9 Example of the update rule of the PLVA. Each entry of each list contains in position ℓ the parameters of the ℓ-th most likely path.

Serial-List Viterbi Algorithm

The SLVA is particularly useful when additional constraints—such as a tail-biting condition or an outer code (e.g., polynomial code) check—must be satisfied. The algorithm operates iteratively, generating candidate paths in order of likelihood while checking each one against the imposed constraints. The algorithm works as follows:

1. **Initial step:** the algorithm begins by performing a standard VA search to find the most likely path. This path is then tested against the required constraints (e.g., verifying a tail-biting condition, checking the parity-check constraints of the polynomial code). If the path meets the constraints, decoding stops immediately.

2. **Iterative candidate generation:** if the most likely path does not pass the constraints, the algorithm proceeds iteratively. At the ℓ-th iteration, the algorithm:

   (a) scans the trellis to identify possible new paths not yet been examined and inserts them, sorted by their likelihood, in a list $\mathcal{L}$ of possible candidates. The best candidate in $\mathcal{L}$, the one with largest likelihood, is then chosen. The list of possible candidates is constructed with paths having the smallest likelihood difference from the previously identified $(\ell-1)$ best paths (*local best paths*).

   (b) checks whether the chosen best candidate from the list satisfies the required constraints.

3. **Termination:** the process continues until a candidate path is found, that meets all the constraints, or until a maximum of L iterations is reached. By stopping early, when a valid candidate is found, the algorithm saves computational resources.

The local best path at trellis section τ of the ℓ-th most likely path corresponds to the path discarded by the VA during the initial step, which merges the ℓ-th most likely path at section τ [55], and that does not leave the ℓ-th most likely path for $t \geq \tau$. The algorithm's efficiency is further enhanced by using advanced data structures, such as heaps [56] or red-black trees [57], which manage the list insertions and deletions of new possible candidate paths with logarithmic time complexity. This efficient implementation of SLVA is known as tree-trellis list Viterbi algorithm (TTLVA) [55]. A visual example of the SLVA procedure is shown in Figure 3.10, where each dot corresponds to a trellis section t. In Figure 3.10.a, the blue path in the figure corresponds to the most likely path selected by the VA. Its local best paths, one for each trellis section, merge the best path at instant times τ from the discarded edge by VA which is depicted with a dashed green line. The likelihoods of the local best paths are computed, and the paths are sorted based on their likelihoods in the list $\mathcal{L}$. The local best path with the largest likelihood corresponds to the second global best path (solid green curve) in the picture. Then, as shown in

Figure 3.10.b the local best paths of the second global best path are extracted, and their merging time from the discarded edge by VA is depicted with a dashed red line. Similarly to what has been done for the best path, the corresponding likelihood of these local best paths is computed. The paths are then inserted, sorted by their likelihood, in the previous list $\mathcal{L}$. In the example, the most likely path in $\mathcal{L}$ corresponds to the third global best path (solid red curve) in the figure.

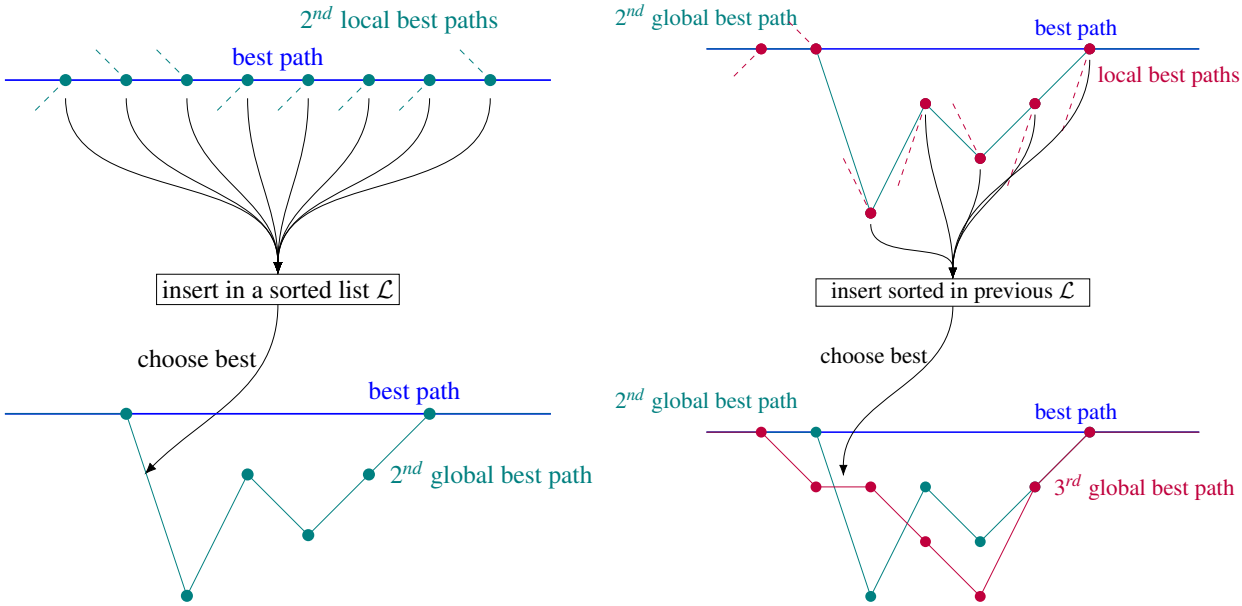


Fig. 3.10 Visual example of the SLVA.

3.4 Improving the Performance of CCSDS Telemetry Links Based on Convolutional Codes

The Consultative Committee for Space Data Systems (CCSDS) is an international organization whose task is to define the protocols used in satellite communications, to allow for interoperability among the world space agencies. CCSDS protocols do not match exactly the open systems interconnections (OSI) protocol stack, the main reason being that the assumption in the OSI protocols is that "the system elements are fixed in place and that they are continuous communication over what are nominally error-free communication channels that suffer from occasional disruption", which is in contrast with several space communication scenarios. In the early 1980s, telemetry (TM) and telecommand (TC) packets were defined by CCSDS, which only considered transmission of data. In particular, TC refers to the process of sending commands from Earth to space missions, enabling operators to control spacecraft operations remotely. On the other hand, TM involves transmitting data from spacecraft back to Earth, providing critical information about the spacecraft's health, status, environment, and on-board instruments (i.e. images). Both TM and TC rely on channel coding for robust and efficient communications, which are detailed in the synchronization and channel coding sublayers. Despite the availability in CCSDS telemetry synchronization and channel coding recommendation [58] of more performant error-correcting codes such as turbo codes [17] and low-density parity-check (LDPC) codes [16], CCs continue to be used in various space missions due to their minimal encoder complexity, especially for on-board applications in small satellites and cubesats. It is worth noting that when CCs are employed in a TM link, the CCSDS TM and the advanced orbiting systems (AOS) space data link protocol [59, 60] mandate the use of an outer polynomial code. The inclusion of a polynomial code is crucial for error detection following Viterbi decoding of the inner CC. However, by adopting the list decoding technique proposed in [55], the outer polynomial code becomes an integral part of the error correction system and is no longer solely used for error detection.

In this section, we compute analytic upper bounds on the block error probability P_B under ML decoding of the poly+CCs of CCSDS. The bounds show that a 3 dB gain is potentially available moving from convolutional code decoding to concatenated code decoding. Then, by means of Monte Carlo simulations, we evaluate the frame error rate (FER) of the poly+CC scheme when decoded by means of list Viterbi algorithms, comparing the results for various list sizes. We analyze the performance of the codes also in terms of undetected error probability P_U, that is, the probability that the decoder outputs an erroneous message, without signalling the decoding failure. Is important to observe that, by using the outer code in a poly+CC scheme to enhance the error correction performance of the inner code, the capability of detecting decoding failures is, in general, reduced with respect to the case where the outer polynomial code is used to simply validate the output of the inner Viterbi decoder. It is hence essential to study the trade-off between P_B and P_U, for different decoder architectures. Note that P_U is always upper bounded by

P_B, with $P_U = P_B$ holding for a *complete* decoder of the poly+CC code, whereas $P_U < P_B$ for any *incomplete* decoders [61].

Simulation results demonstrate that the aforementioned 3 dB gain can be achieved with a reasonable list size, while incurring a small penalty in terms of undetected frame error rates compared with the VA decoding of the inner CC only. We analyze the decoding complexity and show that it is manageable at medium and high SNR, with a slight increase with respect to the plain VA complexity. Then, we compare the poly+CC scheme with modern coding options of the CCSDS TM recommendation. Results show that the gap in performance between poly+CC with LVA and LDPC/turbo codes can be surprisingly limited, especially at high rates. Since the encoder complexity of convolutional codes is extremely low the results suggest that the CCSDS poly+CCs represent an extremely appealing solution for small satellites with limited on-board computational complexity.

3.4.1 CCSDS Telemetry Recommendation

Transfer Frames

The TM and the AOS recommendations from the CCSDS [59, 60] provide essential functionalities for data transfer utilizing a protocol data unit known as the transfer frame (TF), which is generated by upper layers and contains information bits. Within the synchronization and channel coding sublayer, auxiliary functions are offered to facilitate TF transmission across the space link. These functions encompass error-control coding and decoding, TF delimiting and synchronization, as well as bit transition generation and removal.

Various families of channel codes can be chosen as coding options, including convolutional codes, parallel/serial turbo codes [17], Reed-Solomon codes [62], concatenated Reed-Solomon and convolutional codes, and low-density parity-check codes [16]. It is worth mentioning that the decoders for the latter three code families possess a native transfer frame validation property, enabling error detection. In contrast, this capability is not inherent in CC and turbo codes. Consequently, when utilizing the first two code families, the polynomial code (denoted as Transfer Frame Error Control Field in the protocol) defined in [59, 60] is compulsory; it is, instead, optional for the last three code families.

With the exception of CCs, the TF is encoded into a codeword, and then an attached synchronization marker (ASM) is appended as prefix. The ASM is a fixed binary sequence that is used for frame synchronization. For CCs, the procedure differs, and the TF is first preceded by a 32-bit long ASM pattern, equal to $(1ACFFC1D)_{\mathrm{HEX}}$, and the ASM is also encoded. It is valuable to highlight that when there are no TFs available, a frame containing dummy data, and called only idle data (OID) frame, is encoded to carry on the link availability. Subsequently,

before being convolutionally coded, the information sequences are arranged as continuous sequences comprising data/OID TFs separated by ASM patterns. A high-level view of the CCSDS convolutionally coded systems is shown in Figure 3.11, where the synchronization and channel coding operations of a TM CCSDS compatible transmitter are depicted.

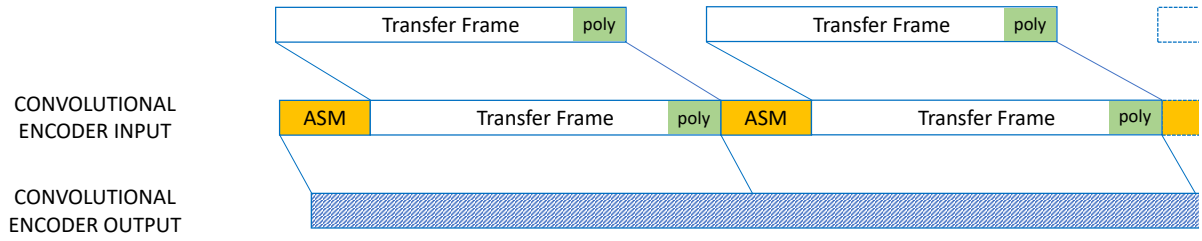


Fig. 3.11 High-level view of the CCSDS TM synchronization and channel coding option of a convolutional coded transmission. "poly" indicates the parity bits of the Transfer Frame Error Control Field

Convolutional and Polynomial Codes

The recursive and systematic outer polynomial encoder with generator polynomial

$$g(D) = 1 + D^5 + D^{12} + D^{16}$$

is used in [59, 60] to generate the 16−bit vector denoted as *Transfer Frame Error Control Field* and whose encoder circuit is depicted in Figure 3.12. The circuit is loaded with all ones at time $t = 0$, i.e., at the beginning of each TF. The switch remains in position s_{in} for the entire frame duration and is then switched to position s_{out} (and s_m is open) to generate the 16 parity bits.

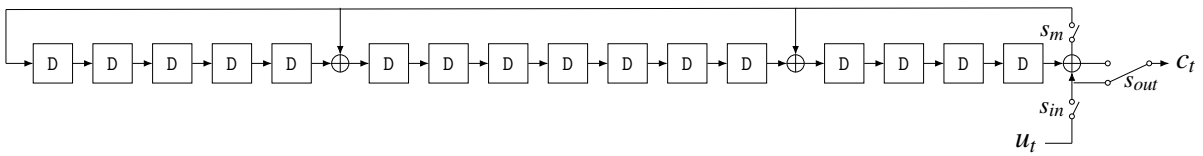


Fig. 3.12 Encoder circuit of the Transfer Frame Error Control Field of a CCSDS TM compatible transmitter.

The memory $\nu = 6$ (64 states) non-recursive inner CC encoder with polynomial generator matrix

$$\mathbf{G}(D) = \left[1 + D + D^2 + D^3 + D^6, 1 + D^2 + D^3 + D^5 + D^6\right]$$

is adopted in [58]. Its encoding circuit is depicted in Figure 3.13.

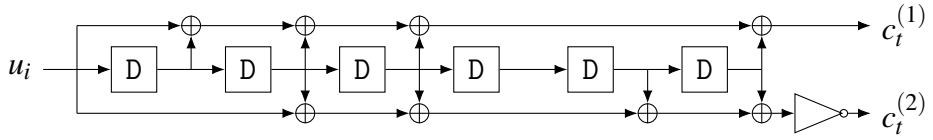


Fig. 3.13 Encoder circuit of the convolutional encoder of a CCSDS TM compatible transmitter.

Note that the CC encoder is not terminated. However, in practice, the encoding of the ASM forces the start/end states to have fixed values. Let us denote by σ_i the i-th state of the trellis. Since the CC encoder is feedforward, the CC encoder is forced to the binary state $\sigma_{\text{start}} = (101110)$ at the beginning of a frame and to $\sigma_{\text{end}} = (110101)$ at the end of it. Those bits are the last 6 bits of ASM preceding the TF and the first 6 bits of the ASM following it, respectively. In the remainder of the section, we denote by K the length of the TF in bits, prior to polynomial encoding. Denoting by $m = 16$ and $h = 32$ the number of parity bits of the polynomial encoder and the length of the ASM, respectively, we get that the blocklength N of the corresponding block code is $(K+m+h)/R_0$ and its code rate is given by

$$R = \frac{K}{N} = \frac{K}{K+m+h} R_0$$

where R_0 is the nominal code rate of the convolutional encoder (equal to 1/2 for the non-punctured case).

The convolutional encoder of [58] has a rate of $R_0 = 1/2$, but it is possible to increase the encoder rate by using a puncturer in cascade with the encoder itself. The periodic puncturing patterns with the corresponding rates are shown in Table 3.1. In the periodic pattern, a "1" indicates a bit which is transmitted, while a "0" indicates a punctured bit which is not transmitted. The bits of the puncturing patterns at even positions, beginning with the position 0, refer to the output bits of the first polynomial of the generator matrix, while the bits at odd positions refer to the output bits of the second polynomial of the generator matrix. For instance, given the CC encoder output

$$\left[c_t^{(1)}, c_t^{(2)}, c_{t+1}^{(1)}, c_{t+1}^{(2)}, c_{t+2}^{(1)}, c_{t+2}^{(2)}, c_{t+3}^{(1)}, c_{t+3}^{(2)}, c_{t+4}^{(1)}, c_{t+4}^{(2)}, c_{t+5}^{(1)}, \ldots\right]$$

by setting the encoder rate to $R_0 = 2/3$ through the application of the periodic puncturing pattern $[1,1,0,1]$ we obtain the transmitted sequence

$$[c_t^{(1)}, c_t^{(2)}, c_{t+1}^{(2)}, c_{t+2}^{(1)}, c_{t+2}^{(2)}, c_{t+3}^{(2)}, c_{t+4}^{(1)}, c_{t+4}^{(2)}, \ldots].$$

This means that the mother rate-1/2 encoder is the same for all convolutional codes of the CCSDS standard and only the correct puncturing pattern is needed to adjust the rate of the encoder.

Table 3.1 CCSDS convolutional encoder puncturing patterns with corresponding convolutional encoder rates. 1 indicates that the bit is transmitted, 0 that the bit is punctured.

Rate R_0	**Puncturing Pattern**
2/3	[1,1,0,1]
3/4	[1,1,0,1,1,0]
5/6	[1,1,0,1,1,0,0,1,1,0]
7/8	[1,1,0,1,0,1,0,1,1,0,0,1,1,0]

Remark 3.1: The poly+CC, having the termination conditions imposed by the ASM and the polynomial encoder initialized to the all-one state, is not strictly a binary linear code. However, the codewords generated by poly+CC form a coset of the poly+CC binary linear code, i.e., the code is affine. In this study, we examine the performance of the code over the bi-AWGN channel. Due to the channel's symmetry, analyzing the affine poly+CC concatenation can be reduced to analyzing the linear poly+CC concatenation. This reduction is achieved by (i) initializing the polynomial circuit by loading only zeros, (ii) setting the ASM to the all-zero sequence, which implies enforcing the CC starting and ending states to be the all-zero state. For the remainder of this manuscript, we will leverage this observation and focus our analysis on the linear poly+CC concatenation.

3.4.2 Iterative Parallel-List Viterbi Algorithm

In this section, we present an implementation approach based on the PLVA with a decoder that exhibits decreasing algorithmic complexity as the SNR increases. The iterative PLVA is employed by running multiple instances, starting with a small list size L (e.g., $L = 1$), and incrementing L whenever the polynomial code constraints are not met for all paths in the list or when L exceeds the maximum list size $L_{\max}$, similar to the adaptive successive cancellation list (SCL) decoder of [63]. This approach emulates the behavior of the SLVA and results in reduced average algorithmic complexity with increasing SNR.

The iterative PLVA incorporates a **scheduler** that determines how L is increased whenever the parity-check equations of the polynomial code are not satisfied. The scheduler, denoted as $sched(\cdot)$, takes the current value of L at the i-th iteration ($L^{(i)}$) and outputs the value of L at the $(i+1)$-th iteration ($L^{(i+1)}$), where $L^{(i+1)} > L^{(i)}$. For instance, a straightforward constant increase in the list size can be achieved with $L^{(i+1)} = sched(L^{(i)}) = L^{(i)} + 1$, or a doubling of the list size at each iteration with $L^{(i+1)} = sched(L^{(i)}) = 2L^{(i)}$. It is important to note that the chosen scheduler impacts both the algorithm's delay and complexity in the worst-case scenario, which occurs when L reaches $L_{\max}$.

The procedure of the iterative PLVA is described in Algorithm 1[1]. Given a received sequence $\mathbf{y}$, a maximum list size $L_{\max}$, and the scheduler $sched(\cdot)$, the algorithm produces a list $\mathcal{L}$ that contains the messages associated with the L most likely paths over the trellis. The list is sorted in decreasing order of likelihood. Additionally, a flag named NACK is used to indicate whether none of the found messages satisfies the parity-check equations of the polynomial code (NACK = 1) or if at least one does (NACK = 0). Lastly, ℓ represents the position of the most likely message that satisfies the polynomial code constraints, if any, within the list.

Algorithm 1 Step-by-step mechanism of the iterative parallel-list Viterbi algorithm applied to the received sequence $\mathbf{y}$ under the constraint of a maximum list size $L_{\max}$ and utilizing the list increment function $sched(\cdot)$.

procedure ITERATIVE_PLVA($\mathbf{y}, L_{\max}, sched(\cdot)$)
    $L \leftarrow 1$
    $\mathcal{L} \leftarrow \emptyset$
    NACK$\leftarrow 1$
    while $(L \leq L_{\max}$ and NACK $= 1)$ **do**
        $(\mathcal{L}, \text{NACK}, \ell) = \text{PLVA}(\mathbf{y}, L)$
        $L \leftarrow sched(L)$
    end while
    return $(\mathcal{L}, \text{NACK}, \ell)$
end procedure

Remark 3.2: The combination of the polynomial code and CCs was initially introduced in the CCSDS telemetry synchronization and channel coding recommendation [58] to serve as an error detection mechanism for the receiver. However, when using the decoders described in this section, the role of the outer polynomial code has evolved beyond pure error detection and has become integrated into the error correction process [40]. Despite this transformation, the approach still maintains a degree of error detection capability. Specifically, a decoding error can be identified when none of the paths included in the final list satisfies the parity-check equations of the polynomial code. It is important to note that the size of the list, denoted as L, significantly impacts the undetected frame error rate. As L increases, the likelihood of undetected errors rises. It is worth mentioning that regardless of the list size, the undetected error probability of the list decoder remains upper bounded by the block error probability under ML decoding of the poly+CC. In Section 3.4.4, we present simulation results to complement these discussions.

[1] In Algorithm 1, at each iteration, the PLVA is run from scratch without reusing computations from previous iterations. However, it is worth noting that more efficient implementations are possible, which can take advantage of intermediate results from earlier iterations.

3.4.3 Distance Spectrum of poly+CCs

Distance Spectrum Calculation

As observed in [40], when both the CC and the polynomial code encoders are non-systematic, the poly+CC can be described by the encoder of a CC with larger memory. More specifically, the memory of the resulting CC is $(\nu + m)$, where m is the degree of the generator polynomial of the polynomial code and ν is the memory of the inner convolutional code. The generator matrix transfer function of the poly+CC is

$$\begin{aligned} \mathbf{G}'(D) &= g(D)\,\mathbf{G}(D) \\ &= [g(D)G^{(0)}(D), \ldots, g(D)G^{(n-1)}(D)], \end{aligned} \tag{3.12}$$

where $\mathbf{G}(D) = [G^{(0)}(D), \ldots, G^{(n-1)}(D)]$ is the generator matrix transfer function of the inner CC, while $\mathbf{g}(D)$ is the generator polynomial of the polynomial code. Hence, we compute the first terms of the distance spectrum of the poly+CC using the trellis representation of $\mathbf{G}'(D)$ by employing the algorithm detailed in Appendix A.

Despite the fact that the polynomial code of the CCSDS poly+CC is systematic, this is not a problem for the computation of the distance spectrum and the above-mentioned analysis remains valid. This is due to the fact that both a systematic and a non-systematic polynomial code, with equal generator polynomial, generate the same codebook, but with different input-output relationships. This means that the set of possible input messages entering the convolutional code is the same for both the systematic and the non-systematic polynomial codes; thus, their weight enumerators are also unmodified. For the CCSDS poly+CC, the memory amounts to $(\nu + m) = 22$, rendering the straightforward application of the above-mentioned technique challenging. The approach can be simplified by performing a search limited to the first terms of the WEF over the code trellis [28].

Asymptotic Coding Gain Analysis

In this section, we delve into the analysis of the achievable asymptotic coding gain provided by the ML decoding of CCSDS poly+CC. The analysis revolves around establishing a union upper bound on the block error probability of both the CC alone and the poly+CC. We first derive the inner CC distance spectrum. The WEF of the code (limited to the lower tail of the distance spectrum) for various values of K is presented in Table 3.2. A notable observation is the doubling of the minimum distance of the inner CC achieved through concatenation with the outer code. Furthermore, it is worth noting that the multiplicity terms exhibit linear growth in relation to the frame length K solely for the CC, as noted in [64], whereas they display quadratic growth for the poly+CC scheme. By leveraging the doubled minimum distance and the controlled rate loss

introduced by the outer polynomial code, asymptotically the poly+CC concatenation achieves a coding gain of approximately 3 dB over the inner CC under ML decoding. This highlights the effectiveness of the poly+CC concatenation in improving the overall performance of the system.

Table 3.2 Distance spectrum of the CC and of the poly+CC of the CCSDS standard for different TF length K.

code	K	$d_{\min}$	$A_{d_{\min}}, A_{d_{\min}+1}, A_{d_{\min}+2}, \ldots$
CC	1768	10	19580, 0, 67477, 0, 342205, …
poly+CC	1768	20	7431, 0, 28005, 0, 175576, …
CC	3552	10	39204, 0, 135269, 0, 686517, …
poly+CC	3552	20	16351, 0, 91945, 0, 610136, …
CC	8904	10	98076, 0, 338645, 0, 1719453, …
poly+CC	8904	20	59091, 0, 557162, 0, 3581187, …
CC	16368	10	180180, 0, 622277, 0, 3160005, …
poly+CC	16368	20	197358, 0, 1800329, 0, 11847522, …

For a specific distance spectrum, we can establish an upper bound on the error probability of ML decoding by using the union bound in (2.4). In Figure 3.14, we present truncated versions of the upper bounds on P_B for both the CC and poly+CC, corresponding to different code lengths K. The curves in Figure 3.14 are obtained by truncating the summation in (2.4) up to $d = 120$. By examining frame error rates below 10^{-6}, where the bound is expected to be highly accurate, we can already observe a significant coding gain of approximately 3 dB that can be achieved through the poly+CC concatenation.

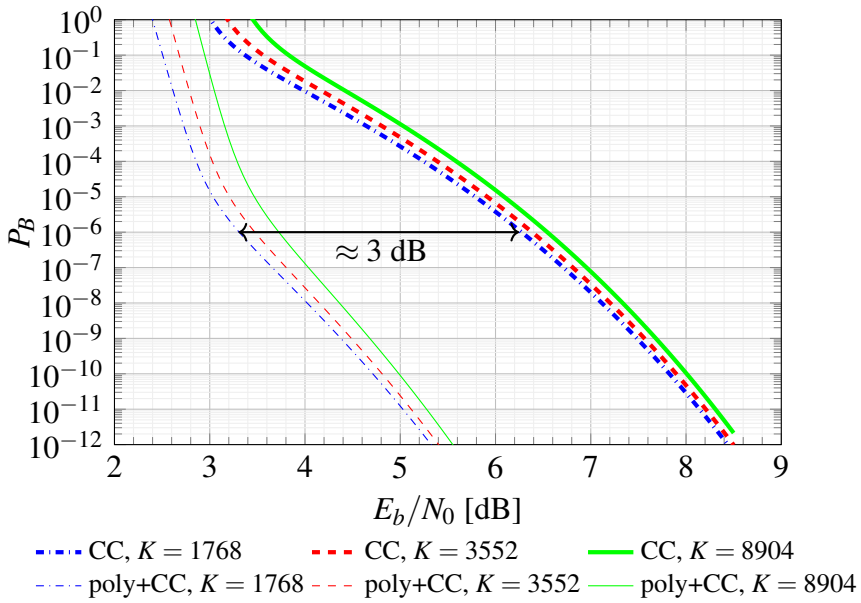


Fig. 3.14 Comparison of truncated union bounds on P_B under ML decoding between the CCSDS TM recommended CCs and poly+CCs under BPSK modulation and AWGN channel.

Punctured Codes

In [58] the output of the CC encoder described in Section 3.4.1 may be punctured to achieve higher code rates. Also, for these higher-rate codes we have computed the lower tail of the distance spectra, which is reported in Table 3.3, and the truncated union bounds on P_B under ML decoding which are depicted in Figure 3.15.

Table 3.3 Distance spectra of the punctured CCs and poly+CCs of the CCSDS standard for various rates of the encoder. The TF length is fixed to $K = 1768$.

code	R_0	$d_{\min}$	$A_{d_{\min}}, A_{d_{\min}+1}, A_{d_{\min}+2}, \ldots$
CC	2/3	6	891, 14229, 42607, 139960, ...
poly+CC	2/3	14	1756, 21066, 76351, 341467, ...
CC	3/4	5	4738, 18328, 94331, 524544, ...
poly+CC	3/4	10	808, 2646, 15199, 80484, ...
CC	5/6	4	4971, 24449, 230378, 1754473, ...
poly+CC	5/6	8	787, 4618, 36036, 317668, ...

Also in these cases, the minimum distance is doubled (and even more than doubled for the $R_0 = 2/3$ case) when compared with the distance of the corresponding inner CCs of the same rates, resulting also in these cases in up to approximately 3 dB coding gain for the poly+CC.

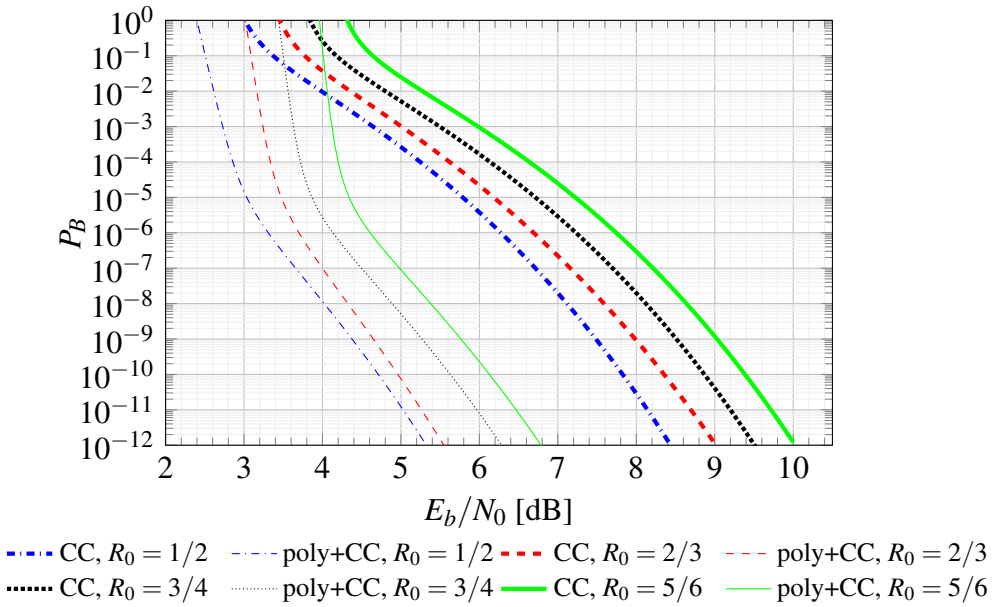


Fig. 3.15 Comparison of truncated union bounds on P_B under ML decoding between the CCSDS TM recommended punctured CCs and punctured poly+CCs under BPSK modulation and AWGN channel conditions. The TFs have length $K = 1768$.

3.4.4 Numerical Results

This section presents the results of Monte Carlo simulations for the poly+CC concatenation of the CCSDS standard. The simulations are performed considering different lengths of the uncoded TF, specifically $K \in \{1768, 3552, 8904\}$ bits, which correspond to commonly encountered CCSDS input lengths of 1784, 3568, and 8920 bits when the additional 16 parity bits of the polynomial encoder are included. The simulations assume BPSK modulation over the AWGN channel.

Figure 3.16 reports the simulation outcomes for $K = 1768$ (additional results for $K = 3552$ and $K = 8904$ are provided in Appendix B, see Figure B.1 and Figure B.2). The results are extended to various list sizes of the LVAs. Note that for the same list size, the performance of SLVA, PLVA and the iterative PLVA is identical. Thus, we do not specify the type of list decoder in the figures. The Viterbi decoding of the inner CC corresponds to $L = 1$ in the figures. On each chart, the (truncated) union bounds on P_B under ML decoding are illustrated for both the CC and the poly+CC concatenation. In the figures, we also depict the RCU bound [9] for the given R and K. Examining Figure 3.16, with $K = 1768$, it becomes evident that higher SNR values allow for a reduction in the list size L while still approaching the ML decoding bound. For example, at $E_b/N_0 = 4$ dB, a maximum list size of $L = 64$ is sufficient to limit the loss to only 0.5 dB from the (truncated) union bound for the poly+CC. Similarly, at $E_b/N_0 = 4.5$ dB, the same level of performance is achieved with a reduced list size of $L = 32$, yielding a coding gain of approximately 2.5 dB compared to the FER of the plain CC under Viterbi decoding. Comparable trends can be observed for other transfer frame lengths.

Punctured Codes Results on the FER of several punctured codes are provided in Figure 3.17 for $R_0 = 2/3$. In Appendix B, specifically in Figure B.3 and Figure B.4, we report the results for $R_0 = 3/4$ and $R_0 = 5/6$, respectively. Here, $K = 1768$ is chosen as TF length. On the same chart, union bounds on the FER under ML decoding are depicted as reference. The results are similar to the non-punctured case, and they show coding gains w.r.t. the use of the VA on the inner CC that are larger than 2.5 dB at a FER of 10^{-7}, when the maximum list size is 32. It is worth mentioning that the coding loss of the poly+CC performance with respect to the RCU diminishes for increasing code rate. For instance, for a target FER of 10^{-5}, for the $R_0 = 1/2$ code, the coding loss is about 1.75 dB, while it decreases to 1.4 dB when $R_0 = 2/3$.

Undetected Frame Error Rates

While it is true that poly+CCs under LVA decoding yield better error rates than those achieved by the inner CC under VA decoding, the gains come at the price of reducing the error detection capabilities of the system, thus increasing the undetected error probability P_U. Nevertheless, the list Viterbi decoding of the poly+CC always produces either the decision yielded by the ML

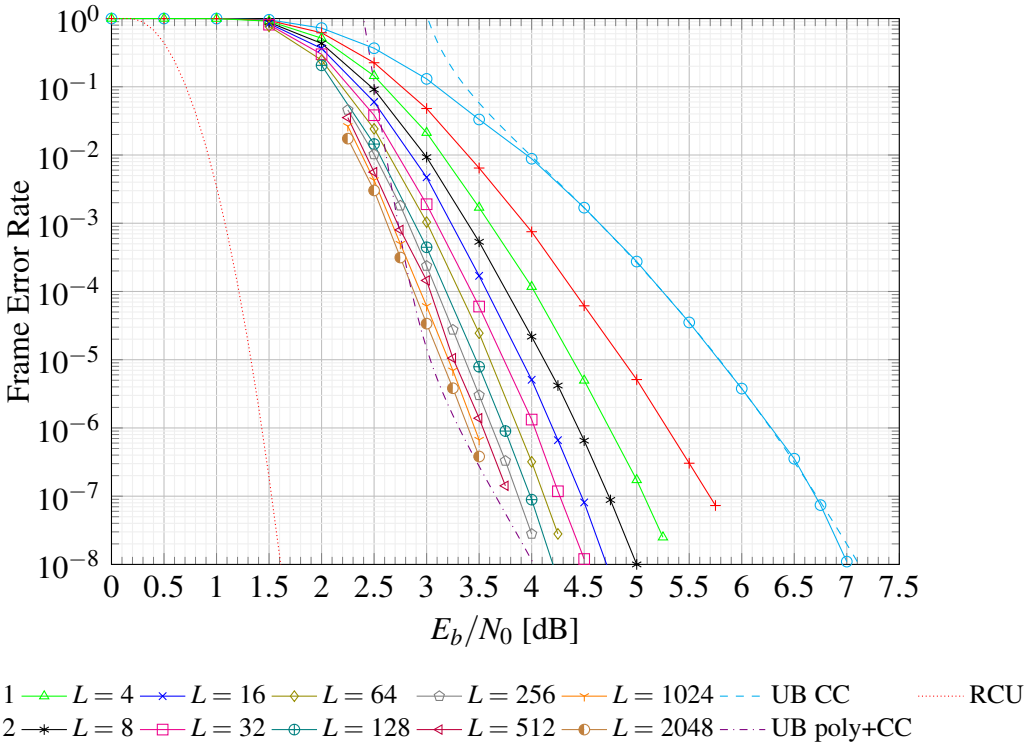


Fig. 3.16 Performance comparison of the CCSDS poly+CC using BPSK modulation over an AWGN channel. The chart depicts the FER as a function of the SNR. The evaluation focuses on a TF of length $K = 1768$ bits, employing a CC encoder with a code rate of $R_0 = 1/2$.

decoder or an erasure when the list is too small (known as rank ordered property of the LVAs in [54]). Consequently, the P_U of the LVAs is guaranteed to be no greater than the P_B under ML decoding of the poly+CC.

We have investigated the UFER of the CC encoder of rate $R_0 = 1/2$ for a TF of length $K = 1768$ for various list sizes of polynomial code-aided LVAs via Monte-Carlo simulations. We have simulated up to counting 100 undetected errors for different values of E_b/N_0. Figure 3.18 shows the obtained results which confirm that higher list sizes penalize the UFER, but the penalty decreases when the SNR increases. For instance, when the VA (which is equivalent to a LVA with $L = 1$) is employed at the decoder side, the penalty at $E_b/N_0 = 3.5$ dB is less than 0.5 dB from the truncated union bound on P_B under ML decoding of the poly+CC, while at $E_b/N_0 = 4$ dB the margin from the bound is reduced to approximately 0.1 dB. This result suggests that the union bound on the P_B of the poly+CC scheme can be regarded as a relatively tight bound for low values of the UFER. Being the P_B union bound an upper bound on the undetected error probability P_U of a LVA for all list sizes, this also means that at medium and high SNR there is little penalty in the UFER when using a LVA instead of the VA. This is true even for very large values of L. In certain scenarios, the required UFER may be very low. Although the constraints are much more pronounced for the uplink than for the downlink, let us suppose very extreme requirements like FER=10^{-9} and UFER=10^{-12}. Looking at Figure 3.14 for a TF of

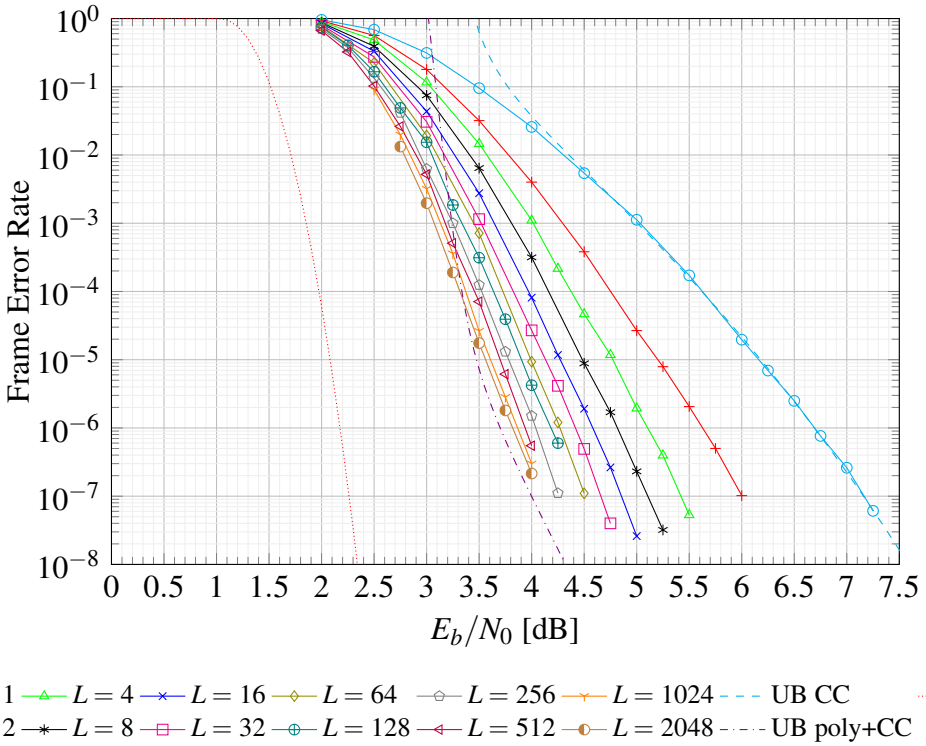


Fig. 3.17 Performance comparison of the CCSDS poly+CC using BPSK modulation over an AWGN channel. The chart depicts the FER as a function of the SNR. The evaluation focuses on a TF of length $K = 1768$ bits, employing a CC encoder with a code rate of $R_0 = 2/3$.

length $K = 1768$, we have a FER of 10^{-9} at around $E_b/N_0 = 7.5$ dB with the VA, but the UFER of 10^{-12} can already be reached at $E_b/N_0 = 5.2$ dB, meaning that the use of a LVA decoder can still provide more than 2 dB coding gain with respect to the Viterbi decoding without exceeding the target UFER.

Complexity of the Iterative PLVA

In Figure 3.19 we report the outcomes of the complexity of the application of the iterative PLVA. We use $\overline{C}_{\text{iPLVA}}$ as a complexity metric, which represents the average complexity normalized to that of the VA. We indicate with $L_{\max}$ the maximum list size and we use a scheduler that doubles the list size each time none of the codewords in the list satisfy the parity-check equations of the polynomial code.

To compute $\overline{C}_{\text{iPLVA}}$, we introduce the variable I that defines the number of iterations of the iterative PLVA. Then, due to the fact that the complexity of the $j-$th iteration is $L^{(j)}$ times larger than the VA, we compute the normalized complexity as

$$\overline{C}_{\text{iPLVA}} = \mathbb{E}_I \left[\sum_{j=1}^{I} L^{(j)} \right],$$

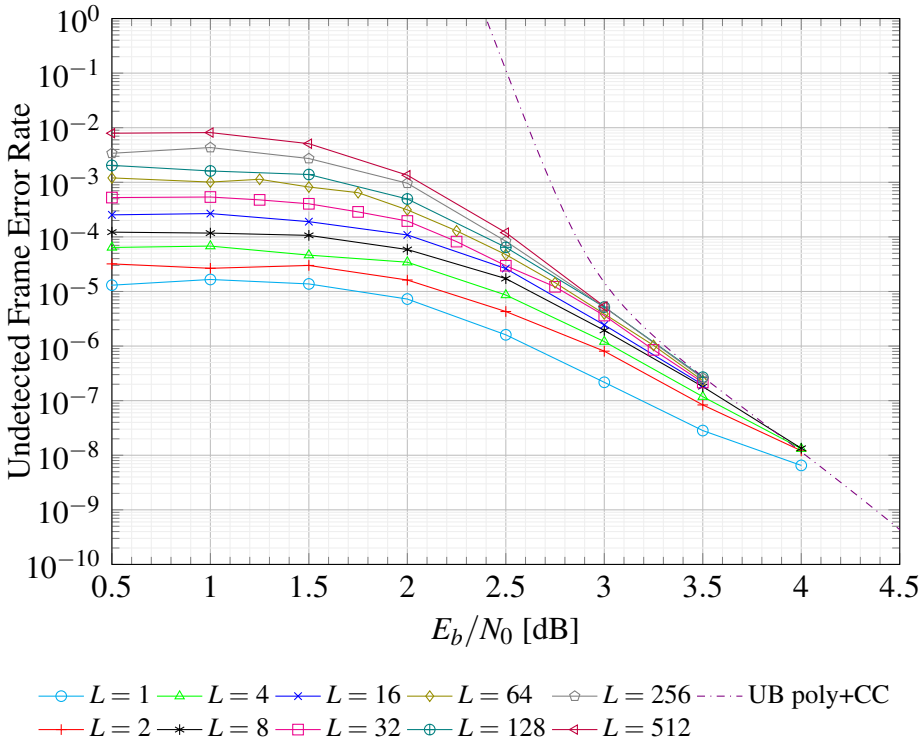


Fig. 3.18 Performance comparison of the CCSDS poly+CC using BPSK modulation over an AWGN channel. The chart depicts the UFER as a function of the SNR. The evaluation focuses on a TF of length $K = 1768$ bits, employing a CC encoder with a code rate of $R_0 = 1/2$.

where the expectation is over I. We estimate the expression via Monte Carlo simulations.

When examining the computed values of $\overline{C}_{\text{iPLVA}}$ for a TF of length $K = 1768$, as illustrated in Figure 3.19, and for various maximum list sizes $L_{\max}$, it is apparent that at low E_b/N_0 values, $\overline{C}_{\text{iPLVA}}$ corresponds to the summation of all powers of two up to the employed maximum list size. This implies that, regardless of the list size, very few messages can be accurately decoded during the initial stages. However, for a target FER of 10^{-7}, where $E_b/N_0 > 4$ dB for ML decoding of the poly+CC scheme, the average complexity of the iterative PLVA decoder approaches a value of 1 for all maximum list sizes $\overline{C}_{\text{iPLVA}}$, indicating that nearly all codewords in the concatenated scheme are correctly decoded during the first iteration, with only a small fraction requiring additional iterations. In particular, when focusing on Figure 3.16 ($K = 1768$), by looking at the FER results at $E_b/N_0 = 4.5$ dB, the Viterbi decoder achieves a FER of $2 \cdot 10^{-3}$. This suggests that with the use of the iterative PLVA decoder, only approximately 0.2% of messages necessitate the execution of PLVA decoders with a list $L \geq 2$.

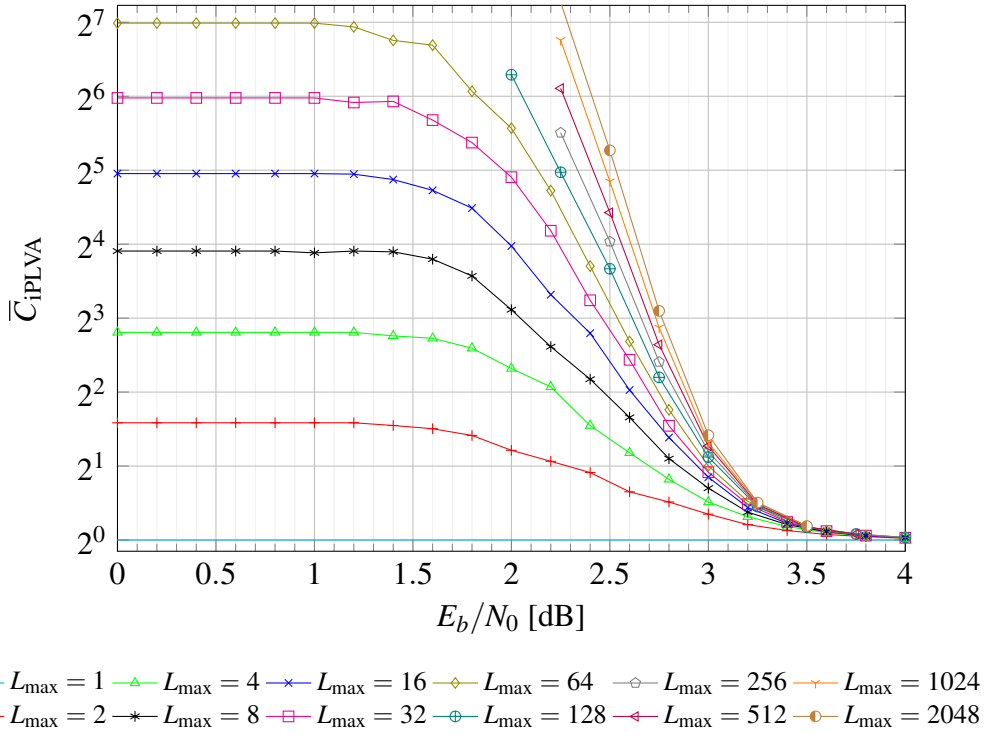


Fig. 3.19 Complexity of the CCSDS poly+CC using BPSK modulation over an AWGN channel. The chart depicts the $\overline{C}_{\text{iPLVA}}$ as a function of the SNR. The evaluation focuses on a TF of length $K = 1768$ bits, employing a CC encoder with a code rate of $R_0 = 1/2$ which is decoded using the iterative PLVA with $L_{\max}$ in the legend.

3.4.5 Performance Comparison for the Codes of the CCSDS Telemetry Reccomendation

In this section we compare the performance of CCSDS poly+CCs under LVA decoding with those of other channel coding schemes for the CCSDS TM recommendation with similar lengths and rates. The comparison is performed on the AWGN channel with BPSK modulated signals. We consider the CCSDS poly+CCs decoded via LVAs with $L_{\max} = 2048$ and compare them with the performance results reported in the CCSDS TM Green Book [65]. Table 3.4 reports the rates of the various analyzed codes. The simulation results are depicted in Figure 3.20, Figure 3.21 and Figure 3.22 for a frame length of $K = 1768$ bits. We use the notation RS+CC-R_0 to identify the $(255, 223)$ Reed-Solomon (RS) codes with error correction capability $E = 16$ concatenated with CC having nominal rate R_0. In all cases, the interleaver has a length of $I = 5$ blocks. Note that this code is more complex and longer than the poly+CC with a frame size of 8920.
The LDPC codes used in the comparison have a frame size of 1024 bits and they are decoded via the belief propagation algorithm with 200 iterations, while the turbo code has a frame size of 1784 bits and it is decoded with 10 iterations of the BCJR algorithm.

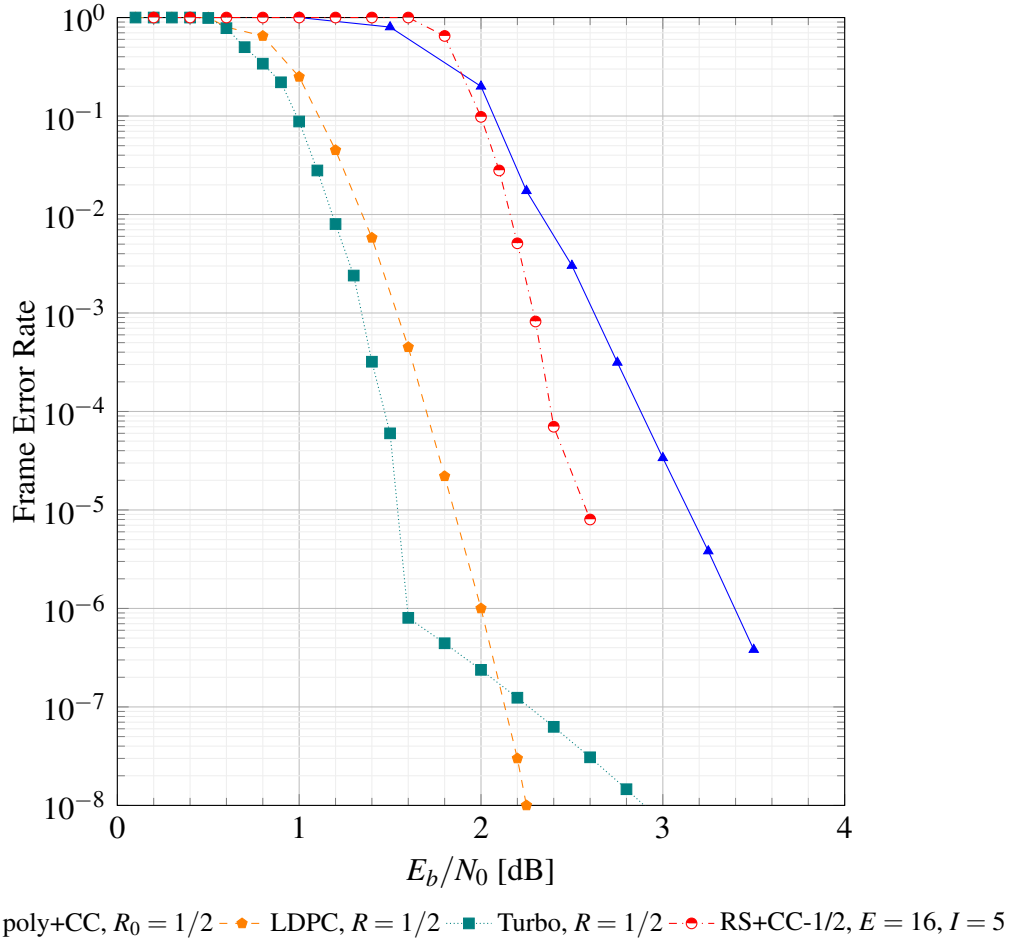


Fig. 3.20 Comparison of the Transfer Frame Error Rate for some channel coding options of the CCSDS versus the poly+CC concatenations of the CCSDS TM recommendation decoded via LVAs with $L_{\max} = 2048$ over the AWGN channel and BPSK modulation. The channel codes have rate $R \leq 1/2$, and the poly+CC TFs have a length $K = 1768$.

Looking at Figure 3.20, we can see that the coding gain achieved by LDPC codes over the poly+CC decreases as the code rate increases. For example, at FER 10^{-5}, and with a code rate of $\approx 1/2$, the coding gain is approximately 1.2 dB. The gain reduces, in Figure 3.21, to 0.7 dB at a rate of $\approx 2/3$, and in Figure 3.22 it is only 0.4 dB when the code rate is $4/5$ for the LDPC code (and slightly higher for the poly+CC).

When compared with the RS+CCs, with similar rates, the poly+CCs perform slightly better while possessing shorter frames. The rate $1/2$ turbo code with $K = 1768$ exhibits reduced gains as the SNR increases, due to a high error floor [66] caused by its low minimum distance ($d_{\min} = 17$), which is smaller than that of the poly+CC. At FER 10^{-5}, the coding gain is approximately 1.5 dB and it decreases down to 0.7 dB at FER of 10^{-10}. To summarize the results, we have compared the achievable rate-SNR pairs for the various code families with the capacity of the binary input AWGN channel. For the poly+CCs, we have taken into account the performance of a LVA with $L_{\max} = 2048$. The results are shown in Figure 3.23 for the target FER of 10^{-5}.

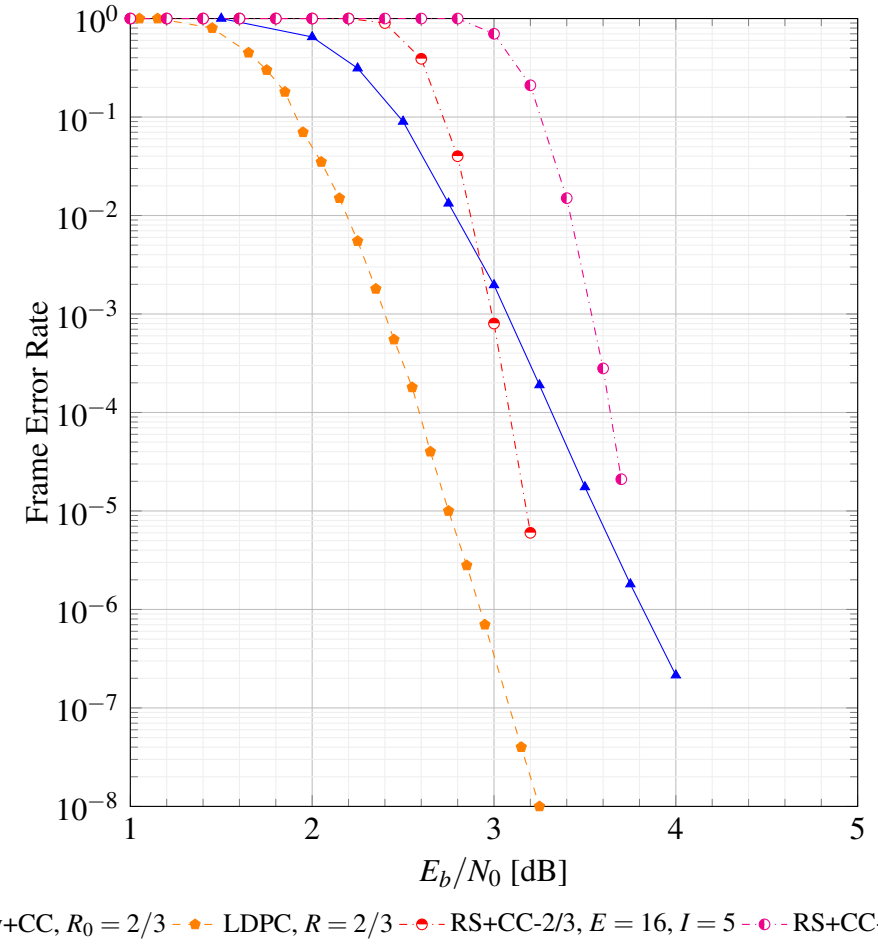


Fig. 3.21 Comparison of the Transfer Frame Error Rate for some channel coding options of the CCSDS versus the poly+CC concatenations of the CCSDS TM recommendation decoded via LVAs with $L_{\max} = 2048$ over the AWGN channel and BPSK modulation. The channel codes have rate $1/2 < R \leq 2/3$, and the poly+CC TFs have a length $K = 1768$.

3.5 Improving the Performance of LTE Control Channels

The 3GPP LTE [67] standard (often referred to as 4G, i.e., 4th generation) has been designed to provide higher data rates, lower latency, and improved spectral efficiency w.r.t. the previous mobile standard generations. To ensure reliability, TBCCs are used as channel codes in the control channels, which are communication channels used to transmit control information, i.e., system information, connection setup, handover commands, etc. Similar to CCSDS, LTE frames are protected by polynomial codes that are intended for error detection. In particular, the standard describes four polynomials with degrees $8, 16$ and 24. Typically, only the degree 16 polynomial is used for the control channel. Its recursive and systematic polynomial encoder has generator polynomial

$$g(D) = 1 + D^5 + D^{12} + D^{16}$$

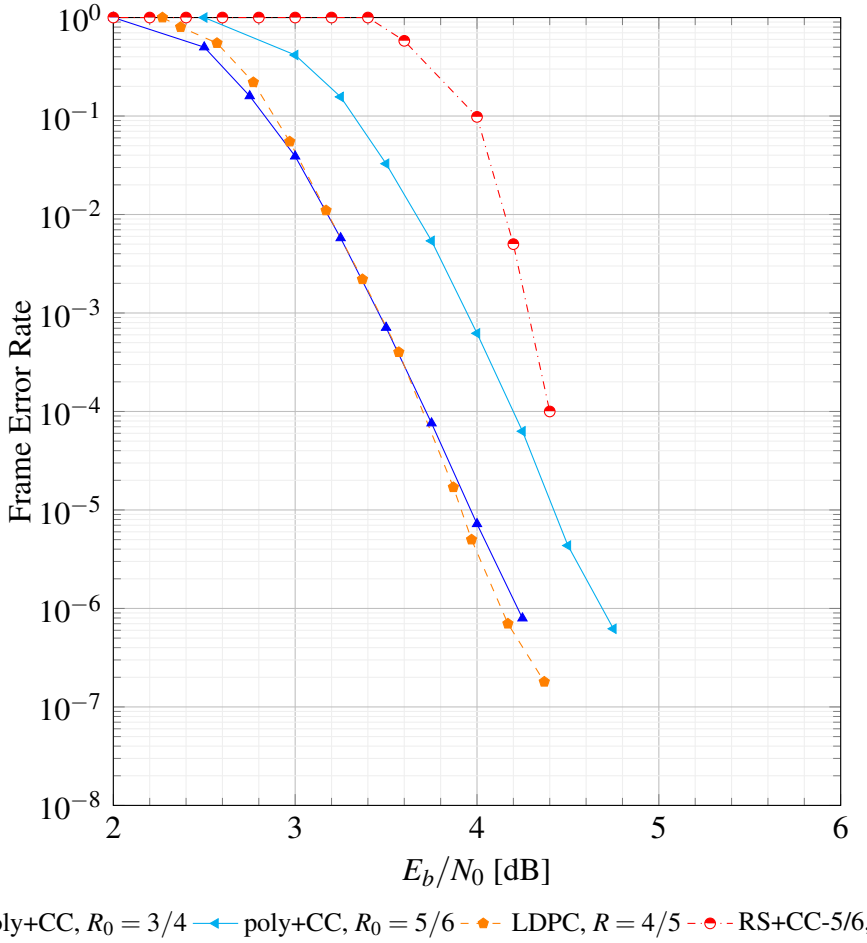


Fig. 3.22 Comparison of the Transfer Frame Error Rate for some channel coding options of the CCSDS versus the poly+CC concatenations of the CCSDS TM recommendation decoded via LVAs with $L_{\max} = 2048$ over the AWGN channel and BPSK modulation. The channel codes have rate $R > 2/3$, poly+CC TFs have a length $K = 1768$.

that is the same as the CCSDS standard. The inner TBCC has memory $\nu = 6$ (64 states) non-recursive encoder with polynomial generator matrix

$$\mathbf{G}(D) = \left[1 + D + D^2 + D^3 + D^6, 1 + D^2 + D^3 + D^5 + D^6, 1 + D + D^2 + D^4 + D^6\right].$$

3.5.1 Distance Spectrum

Similarly to the CCSDS case, we compute the lower tail of the spectrum of the poly+CC, and we report the result in Table 3.5.

In the following, the performance achievable by the LTE poly+CCs under ML decoding is analyzed by means of Poltyrev's tangential sphere bound (TSB) [68]. Note that, while the UB of (2.4) is typically tight at high SNRs, it yields inaccurate estimates of the block error probability

Table 3.4 Rates of the CCSDS telemetry channel coding options compared in Figure 3.20, Figure 3.21 and Figure 3.22.

code	rate	input frame size
RS+CC-1/2	0.437	8920
Turbo code, $R = 1/2$	0.495	1768
poly+CC, $R_0 = 1/2$	0.495	1768
LDPC code, $R = 1/2$	0.500	1024
RS+CC-2/3	0.583	8920
RS+CC-3/4	0.656	8920
poly+CC, $R_0 = 2/3$	0.661	1768
LDPC code, $R = 2/3$	0.667	1024
RS+CC-5/6	0.729	8920
poly+CC, $R_0 = 3/4$	0.743	1768
LDPC code, $R = 4/5$	0.800	1024
poly+CC, $R_0 = 5/6$	0.826	1768

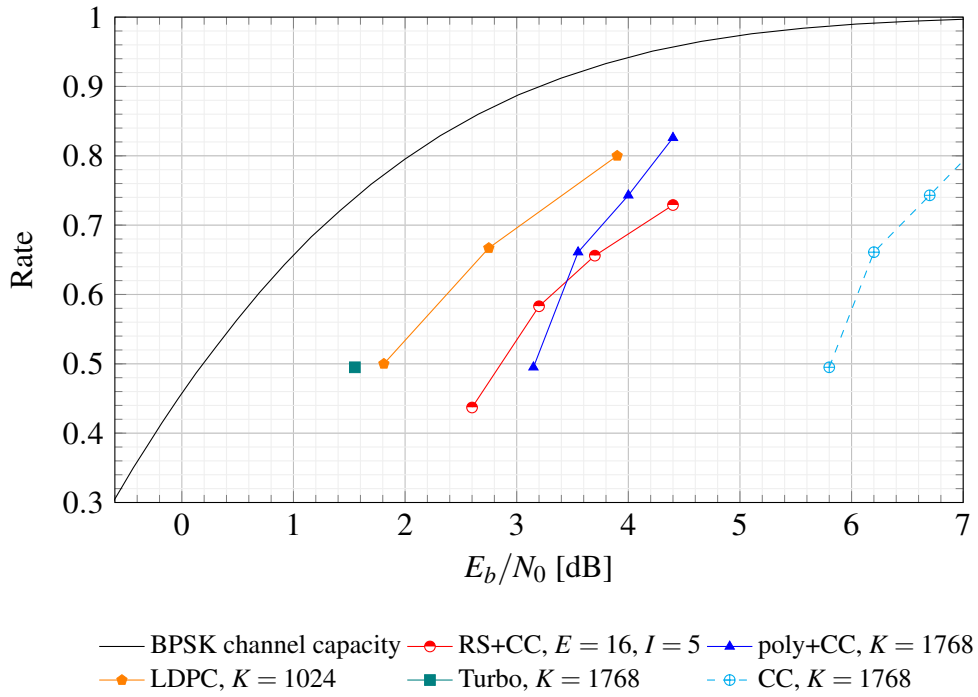


Fig. 3.23 Comparison of the achievable SNR-rate pairs for a target Transfer Frame Error Rate of 10^{-5} for some coding options of the CCSDS TM recommendation versus the poly+CC concatenations of the CCSDS TM recommendation decoded via LVAs with $L_{\max} = 2048$ over the AWGN channel with BPSK modulation.

at low SNRs. In contrast, the TSB is tight at both low and high SNRs and provides a tight upper bound on the block error probability under ML decoding. Figure 3.24 depicts the result of the TSB for different frame lengths, and even at low to moderate SNRs, the bound predicts important coding gains for the ML decoding of the LTE poly+CCs w.r.t. the Viterbi decoding of the CCs alone.

Table 3.5 Distance spectrum of the CCs and of the poly+CCs of the LTE standard for different frame lengths.

LTE protocol

coding scheme	K	$d_{\min}$	$A_{d_{\min}}, A_{d_{\min}+1}, A_{d_{\min}+2}, \ldots$
CC	64	15	192, 192, 384, 576, 256, ...
poly+CC	64	28	5, 0, 1, 0, 72, ...
CC	128	15	384, 384, 768, 1152, 512, ...
poly+CC	128	28	3, 0, 1, 0, 129,...
CC	256	15	768, 768, 1536, 2304, 1024, ...
poly+CC	256	30	2, 0, 377, 0, 377, ...
CC	512	15	1536, 1536, 3072, 4608, 2048, ...
poly+CC	512	30	7, 0, 1034, 0, 1227, ...
CC	1024	15	3072, 3072, 6144, 9216, 4096,...
poly+CC	1024	32	2706, 0, 4134, 0, 10032, ...

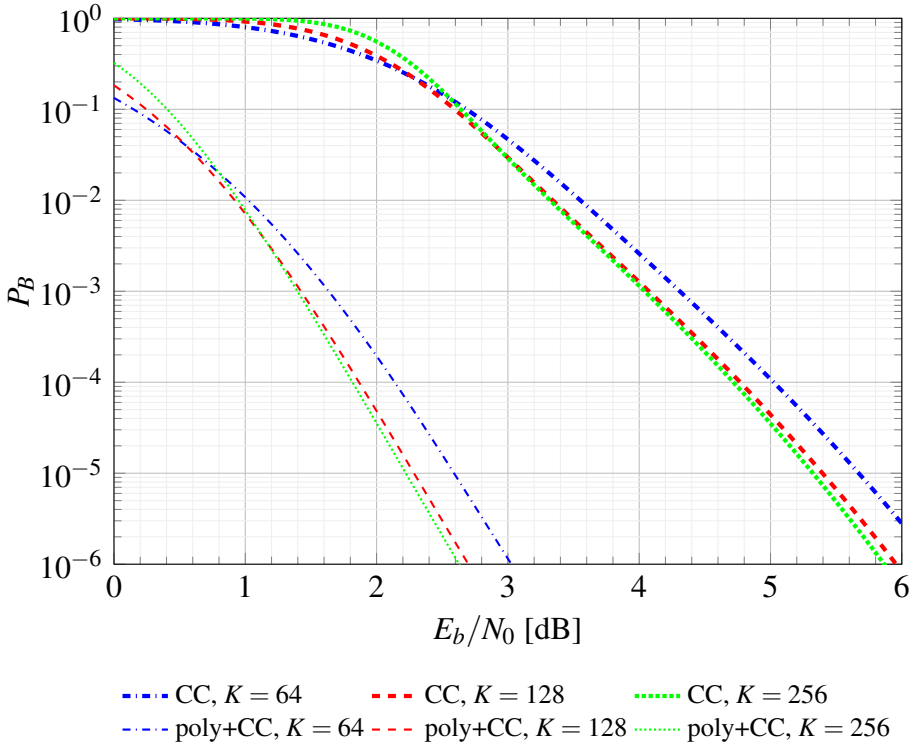


Fig. 3.24 Comparison of TSB on the block error probability under ML decoding between the LTE CCs and the poly+CCs under BPSK modulation and AWGN channel conditions.

3.5.2 Numerical Results

In Section 3.2.3 has been mentioned that a near-ML decoder named wrap-around Viterbi algorithm (WAVA) can be used to decode TBCCs. Another approach can rely on list decoding. Interestingly, for decoding poly+CC convolutional codes, the combination of WAVA and LVAs is also possible. This topic has been investigated, for instance, in [45, 69–71] where both PLVA and SLVA have been tested. In the following simulations, for the decoding of LTE codes, the SLVA has been used. In particular, the TTLVA [55] has been adapted to work over the trellis used by the WAVA. Figures 3.25 and 3.26 present simulation results of the FER versus E_b/N_0 for various list sizes. The WAVA decoder is used with a maximum number of iterations $I_{\max} = 1$ in Figure 3.25, and with $I_{\max} = 2$ in Figure 3.26. The RCU bound [9] for the specified rate R and blocklength N is provided as a reference. Notably, unlike the CCSDS example of Figure 3.16, the

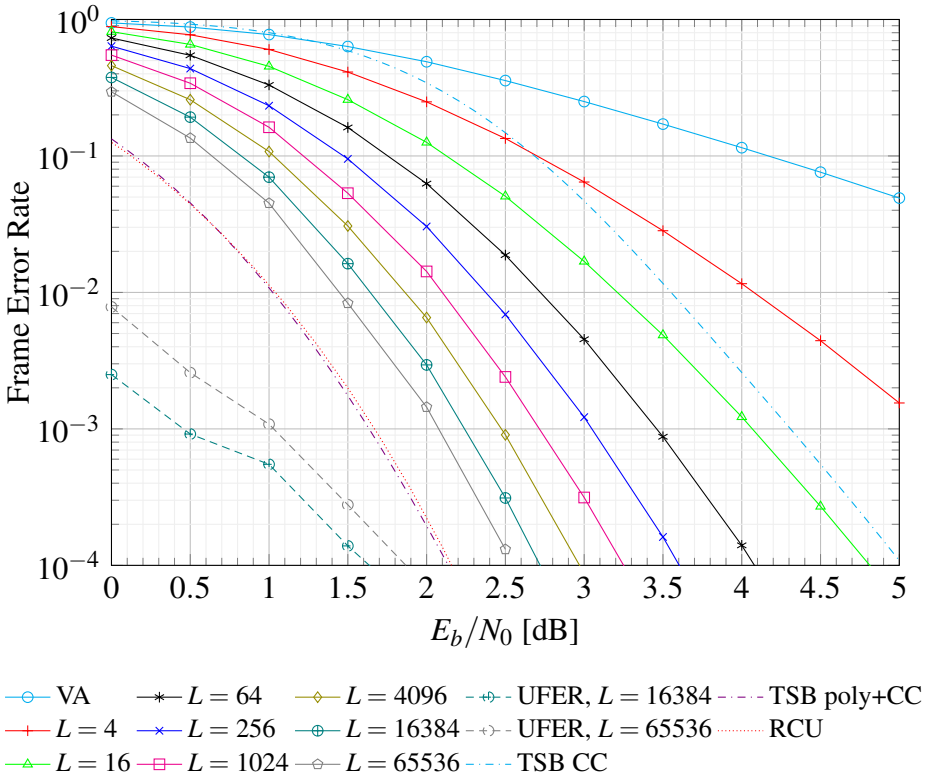


Fig. 3.25 Performance comparison of the LTE poly+CC using BPSK modulation over an AWGN channel. The chart depicts the FER as a function of the SNR. We have adopted $K = 64$ and $I_{\max} = 1$.

LTE codes require larger list sizes to achieve large coding gains with respect to the performance of ML decoding of the inner code. This is a well-known result in the literature [45, 43, 70] and is mainly due to the suboptimal decoding performance of the WAVA. Another interesting finding is that, at least for $K = 64$, LTE codes under ML decoding can closely approach the RCU bound. However, achieving this result with LVA is impractical since it requires working with a very large maximum list size. It is also worth noting that the performance achieved by

WAVA with $I_{\max} = 2$ without list decoding can be matched by a LVA with a list size of roughly 16. Furthermore, for large list sizes (e.g., $L \geq 64$), the performance of WAVA with $I_{\max} = 2$ and with $I_{\max} = 1$ becomes very similar. Finally, it is important to emphasize that when $I_{\max} \geq 2$, the

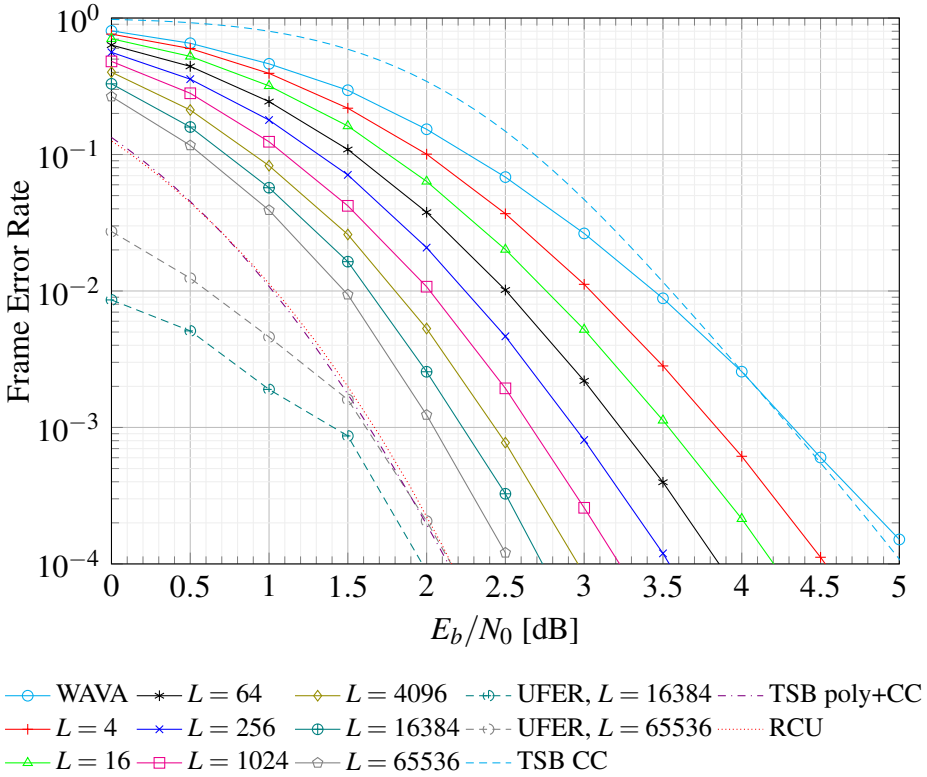


Fig. 3.26 Performance comparison of the LTE poly+CC using BPSK modulation over an AWGN channel. The chart depicts the FER as a function of the SNR. We have adopted $K = 64$ and $I_{\max} = 2$.

undetected error probability of the WAVA decoder with list decoding is not guaranteed to be bounded by the block error rate under ML decoding of the poly+CC[2]. For instance, in Figure 3.26, even with $L = 65536$, the UFER (represented by the dashed gray line) reaches the TSB on $P_B(\mathcal{C})$ of the poly+CC. In contrast, with $I_{\max} = 1$ (i.e., using the classical LVA), the upper bound holds, as seen in Figure 3.25, where the UFER for $L = 65536$ is lower than that observed for $I_{\max} = 2$. This distinction is particularly significant because convolutional codes are used in the LTE control channel, which has stricter requirements for undetected error probabilities.

Using the TTLVA, the average list size can be much lower than $L_{\max}$. Note that differently from PLVA, the relationship between the complexity of TTLVA w.r.t. VA is not linear in the list size L. Different works have proposed methods to better characterize it [23, 70], showing a non-linear behavior, due to the different complexities between the forward and the backward steps of the algorithm. Nevertheless, we can compute the expected list size $\overline{L}$ when the last

[2]The list decoders over the trellis used by the WAVA, when $I_{\max} \geq 2$, do not rank the paths according to their likelihood but instead use a suboptimal metric. Consequently, the output of these list decoders is not guaranteed to be either the decision of the ML decoder of the poly+CC or an erasure when the list is insufficiently large.

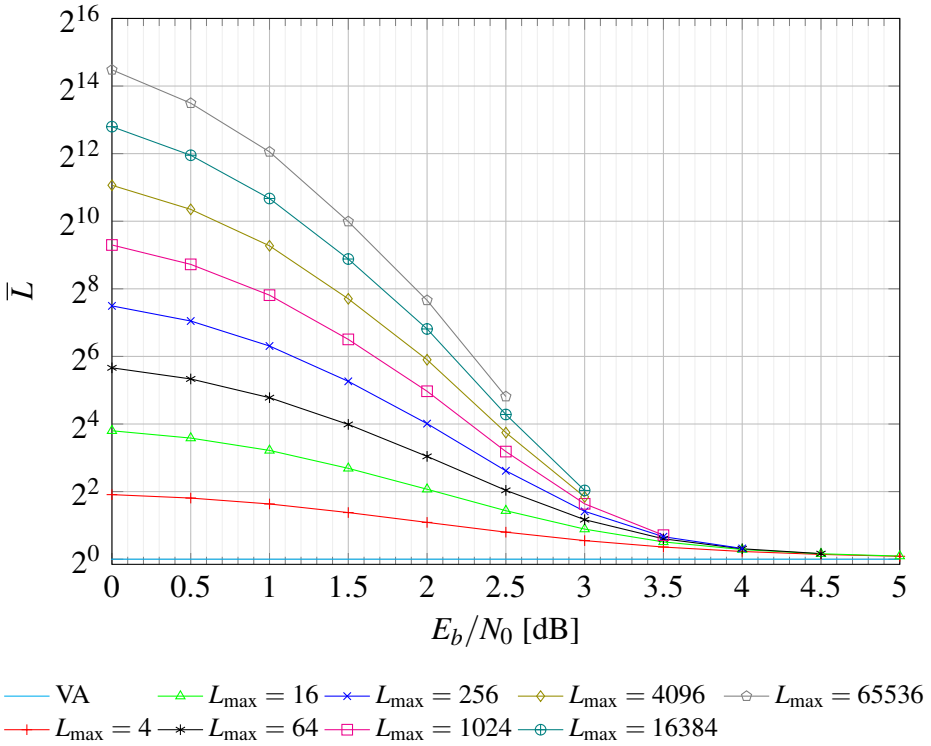


Fig. 3.27 Complexity of the LTE poly+CC using BPSK modulation over an AWGN channel. The chart depicts $\overline{L}$ as a function of the SNR. We have adopted $K = 64$ and $I_{\max} = 1$.

WAVA iteration occurs. We estimate $\overline{L}$ via Montecarlo simulations, according to

$$\overline{L} = \mathbb{E}_I \left[\sum_{j=1}^{I} L^{(j)} \right],$$

where j denotes the $j-$th iteration of the TTLVA. The resulting $\overline{L}$ for $K = 64$ is reported in Figure 3.27 for $I_{\max} = 1$ and in Figure 3.28 for $I_{\max} = 2$, where we can observe that as the SNR increases, $\overline{L}$ decreases. The decrease in the average list size is similar for the two figures.

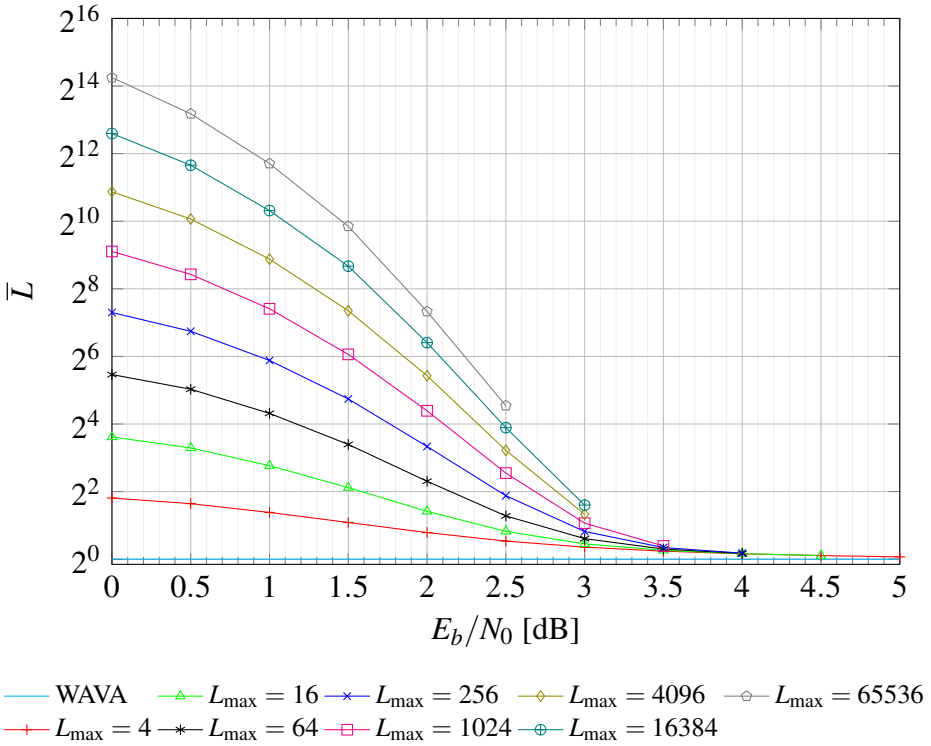


Fig. 3.28 Complexity of the LTE poly+CC using BPSK modulation over an AWGN channel. The chart depicts $\overline{L}$ as a function of the SNR. We have adopted $K = 64$ and $I_{\max} = 2$.

3.6 Final Remarks

In this chapter, an in-depth analysis of the concatenated coding scheme formed by inner convolutional codes and outer polynomial codes has been provided. The investigation involved examining the trellis structure of the concatenated scheme and its effect on the distance spectrum, which ultimately governs the error probability performance. In the analysis, both the CCSDS telemetry recommendation and the LTE standard were considered. In each case, it has been demonstrated that decoding poly+CC with LVA can yield approximately 3 dB coding gain over Viterbi decoding the inner convolutional code alone. For telemetry applications, the results show that employing a list Viterbi decoder allows the system to achieve this coding gain with a manageable increase in complexity. In contrast, for LTE applications, the potential gains are even more pronounced. However, the result comes at the expense of larger list sizes. The difference in complexity is mainly attributed to the termination applied to the inner convolutional encoder. These observations raise open questions regarding how to optimize list decoding to reconcile performance improvements with practical limitations in complexity. Overall, the findings of this chapter underscore the promise of concatenated convolutional and polynomial coding techniques for short-packet scenarios typical of machine-type communications.

.

Chapter 4

Enhanced Spread Spectrum Aloha over the Unsourced Multiple Access Channel

In this chapter, the design and performance of enhanced spread spectrum Aloha (E-SSA) [72] in the unsourced multiple access channel (UMAC) framework are explored. E-SSA is based on spread Aloha [73], and it is a purely asynchronous protocol that combines Aloha [74] with direct-sequence spread spectrum to suppress multiuser interference.

The objective of this chapter is twofold. On one hand, the performance of E-SSA in the UMAC setting has been characterized, showing how a well-established random access protocol that is already deployed in real systems can yield performance that competes with state-of-the-art UMAC solutions. On the other hand, specific E-SSA design choices that stem from the adaptation to the UMAC setting are addressed, with emphasis on the short packet transmission regime.

To the best of the author's knowledge, no effort has been made to analyze the performance of E-SSA in the UMAC setting. An obstacle to this task is represented by the asynchronous/unframed nature of E-SSA, which collides with the fixed frame size analysis of [12]. This problem is circumvented by casting a wrap-around version of E-SSA. By doing so, a fair basis to compare E-SSA with finite block length limits [12] and with advanced UMAC schemes can be provided. The focus is on the Gaussian multiple access channel (GMAC) channel, which serves as a realistic model for satellite communications. It is shown that a judicious design of E-SSA allows approaching the bounds up to moderate-size user populations, competing with some of the best known schemes [75]. Considering the transmission of small data units (in the order of a few tens of bits), the result is achieved by modifying the original E-SSA design according to the following observations. First, the adoption of codes that perform close to finite-length over single-user channels [9] allows keeping the gap from the UMAC bound [12] small at low channel loads. Second, the use of protocol information [76] to carry part of the message content improves

energy efficiency.[1] To address the first point, polar codes [21] concatenated with an outer CRC are employed, together with successive cancellation list (SCL) decoding [80, 63] at the receiver side. For the second point, the *timing channel* associated with the packet transmission time is utilized. The use of timing channels to convey information was pioneered in [76, 81] (see also [82] for an insightful review of the topic). In the context of random access protocols, timing channels were used in [83] to improve the efficiency of slotted Aloha. In this construction, the timing channel is used to improve the error detection capability of the channel decoder, thus reducing the false alarm rate (which can have a detrimental effect on the successive interference cancellation (SIC) algorithm performance). The resulting scheme retains the simple transmitter structure of the original E-SSA, with a decoding complexity that scales linearly in the number of active users.

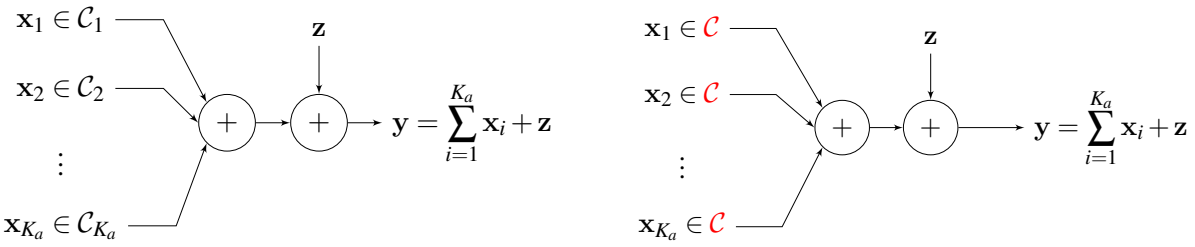
4.1 Massive Random Access

Addressing the connectivity of a massive number of intermittently transmitting devices—a setting often referred to as massive random access (MRA)—poses a significant challenge for the upcoming generations of wireless networks. This challenge has garnered increased attention in recent years due to emerging IoT applications. The interest in random access (RA) protocols is shared both in terrestrial [84] and non-terrestrial networks [85]. While the design of efficient (coordinated) medium access control (MAC) protocols is a well-studied problem in communication systems, the MRA setting reveals several crucial aspects that require new solutions [12]. These include finite blocklength effects [9], the fact that only a small fraction of users is active at a given moment, and the absence of pre-agreed resource allocation with the base station. From an information-theoretic viewpoint, the latter point can be captured by constraining users to adopt the same codebook, leading to the definition of UMAC protocols. The development of an UMAC achievability bound by Polyanskiy [12] provided a fundamental benchmark, triggering the design of new schemes capable of approaching the bound [75, 77, 86, 78, 87, 88].

4.1.1 Unsourced Multiple Access Channel

A perspective on the topic of unsourced multiple access channel (UMAC) is presented in [89], where first a historical survey details the different periods regarding MAC techniques, followed by a classification of various UMAC schemes.

[1] This observation was exploited, for example, in [77, 78] by mapping a portion of the message onto the preambles / signature sequences envisaged by the scheme, in contrast with the random preamble choice adopted in the 5GNR random access protocol [79].



(a) MAC (b) Unsourced MAC

Fig. 4.1 Classical coordinated MAC model (left) and UMAC model (right).

Following the model of [12], the main differences between coordinated MAC and the UMAC model are as follows:

- Users employ a common codebook $\mathcal{C}$ to encode their message. This is particularly relevant for massive random access scenarios in which the number of active users is much smaller than the total number of devices in the whole network, and the use of the same codebook $\mathcal{C}$ avoids the need for coordination between the base station and each device to allocate individual codebooks (see Figure 4.1) and resources (*grant-free* access).
- The receiver is only interested in recovering the set of transmitted messages up to a permutation, without resolving user identities and, for this reason, the framework is named *unsourced*.
- Unlike classical MAC, where the derivation of the rate region requires to decode all transmissions with a (vanishing) small error probability, in the UMAC setting, the error event is defined on a per-user basis.

Indeed, in many network theoretic studies, the MAC layer is seen primarily as a mechanism for packet delivery rather than sender identification, since payloads typically include header information that serves to identify the source. When small payloads are considered, the details of how the identification is performed can significantly impact performance, thereby complicating fair evaluations and comparisons of different random access schemes.

In the following, the transmission of K_a active users over the Gaussian multiple access channel (GMAC) is considered. The $i-$th active user transmits its message in the form of a codeword $\mathbf{X}_i \in \mathcal{C}$ of length n real channel uses within a given frame. The channel is affected by additive white Gaussian noise (AWGN), so that the received vector $\mathbf{Y}$ is given by

$$\mathbf{Y} = \mathbf{X}_1 + \mathbf{X}_2 + \cdots + \mathbf{X}_{K_a} + \mathbf{Z}, \quad \mathbf{Z} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_n),$$

where $\mathbf{I}_\mathrm{n}$ denotes the $\mathrm{n} \times \mathrm{n}$ identity matrix and σ^2 is the noise variance. Each transmitted codeword $\mathbf{X}_i$ must satisfy the power constraint $\|\mathbf{X}_i\|_2^2 \leq \mathrm{n}P$. The code $\mathcal{C}$ contains $M = 2^K$

codewords. Hence, each codeword carries K bits of information. The per-user signal-to-noise ratio (SNR) is denoted by E_b/N_0 and is given by

$$\frac{E_b}{N_0} = \frac{\mathrm{n}P}{2K\sigma^2}.$$

The decoder, upon observing $\mathbf{y}$, outputs a list $\mathtt{D}(\mathbf{y})$ of K_a distinct messages from $\mathcal{C}$ and the per-user probability of error (PUPE) is defined as

$$\varepsilon := \frac{1}{K_a}\sum_{i=1}^{K_a} P\big(\mathbf{X}_i \notin \mathtt{D}(\mathbf{Y})\big).$$

4.2 Low-Rate Small-Blocklength Codes

Short, low-rate channel codes play an important role in the UMAC framework since they allow reliable decoding in presence of strong multiuser interference identification of active users from a large, unknown population using very short messages. In [12, Theorem 1], an achievability bound utilizing Gaussian codebooks at low coding rates is presented. The use of low-rate codes allows for a more favorable balance between short message length and decoding reliability, thereby efficiently managing interference arising from simultaneous transmissions.

4.2.1 Convolutional Codes Limitations at Low Rate

In Chapter 3, the excellent performance of convolutional codes (CCs) in the finite blocklength regime has been analyzed, especially when concatenated with outer polynomial codes and decoded via list Viterbi algorithms. However, it is shown in the following section that, despite terminated CCs with code rates greater than $1/2$ perform incredibly well at short blocklengths, the design of good lower rate CCs (i.e., $R_0 = 1/n$, with $n > 2$) can be challenging and not effective. In particular, one can rely on Heller's upper bound on the free distance (Theorem 3.2). Following the bound, it is possible to verify that when $\nu \to \infty$, then

$$\lim_{\nu\to\infty} \frac{d_{\text{free}}}{n\nu} \leq \frac{1}{2}. \tag{4.1}$$

as shown in [26, Corollary 3.19].

Observing that, for a terminated CC with a sufficiently long encoding sequence, $d_{\text{free}} = d_{\min}$, by (4.1) one can notice that $R_0\, d_{\text{free}}$ scales at most as $\nu/2$, i.e., the product $R_0\, d_{\text{free}}$ is independent of n. Consider now the union bound truncated to the minimum distance term,

$$P_B \approx \frac{1}{2} A_{\min}\, \mathrm{erfc}\left(\sqrt{d_{\min} R \frac{E_b}{N_0}}\right)$$

which yields an accurate estimate of the block error probability at large SNR[2]. The coding gain at high SNR w.r.t. uncoded transmission is $10\log_{10}(d_{\min}R)$ dB. Noting that for rate-$1/n$ CCs with memory ν, the product $d_{\min}R$ is insensitive to n, it can be concluded that the coding gain achieved by a rate-$1/n$ ($n > 2$) CC over a rate-$1/2$ CC vanishes at high SNR. This result implies that limited or no performance gains can be expected by lowering the code rate below $1/2$.

To confirm this result, the Heller's bound has been computed for several memory values and different code rates $1/n$, and in Figure 4.2 the ratio d_{free}/n versus the memory ν is depicted. In the figure, all the curves representing different code rates have approximately the same d_{free}/n for the same memory value, except for some rate-$1/2$ codes that are slightly lower. Interestingly, the authors in [90] have computed upper bounds on the bit error probability of some low-rate convolutional codes attaining the Heller's bound for different memories and code rates, and despite some coding gain is visible at moderate E_b/N_0 values when decreasing the rate from $1/2$ to $1/4$, the coding gain tends to disappear when the rate is further reduced w.r.t. the rate-$1/4$ case.

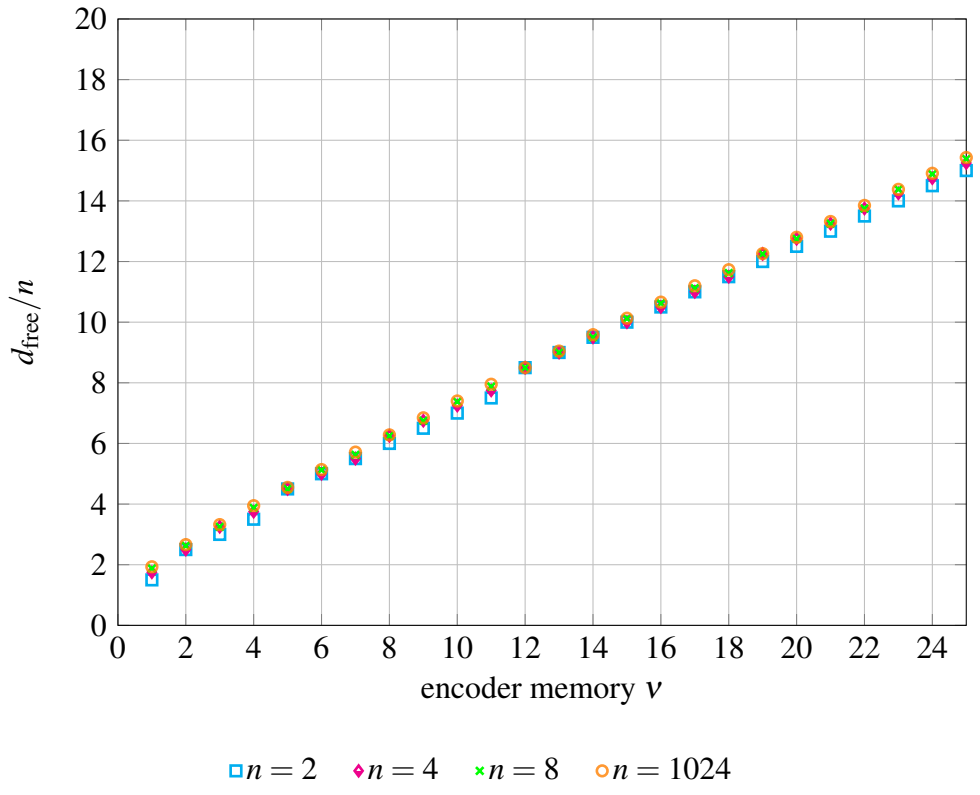


Fig. 4.2 Heller's bound, normalized to n, as a function of the memory ν for some values of n.

Owing to the limitations of CCs at low rates, a different class of codes is next considered, that is polar codes concatenated with outer polynomial codes. It will be shown that, while the

[2]Recall that R is the code rate of the terminated CC (i.e., $R = R_0$ for a TBCC, $R = R_0 K/(K+\nu)$ for a ZTCC).

performance of CCs and polar codes is comparable at rates $\geq 1/2$, at low rates polar codes exhibit visible gains.[3]

4.2.2 Polar Codes with Successive Cancellation List Decoding

Polar codes [91, 21] have been proven to be a class of capacity-achieving error-correcting codes over binary-input symmetric memoryless channels. They are based on the recursive Plotkin construction [92] and their name *polar* stems from the phenomenon of *channel polarization*, where synthetic subchannels evolve in the recursive construction to either completely noiseless or completely noisy channels as the code length increases, enabling efficient capacity-achieving transmission. While polar codes exhibit optimal performance in the limit of large blocklengths, their finite-length performance is far from optimal. However, the introduction of successive cancellation list (SCL) decoding by Tal and Vardy in [80], and the concatenation with outer linear codes, significantly improves the short blocklength performance, making polar codes competitive at short blocks. This result led to their adoption in the 5G new radio (NR) standard [79, 93] for the control channel. Their encoding and decoding structure also allows efficient hardware implementations.

Polar Codes

An (N,K) polar code [91, 21] is specified by the set of indices of the information bits $\mathcal{A} \subseteq [N] = \{0,1,...,N-1\}$. The set of *frozen* bits is denoted by $\mathcal{F} = [N] \backslash \mathcal{A}$. From an information vector $\mathbf{u} \in \mathbb{F}_2^K$, the corresponding codeword $\mathbf{c}$ is obtained by mapping the K bits in $\mathbf{u}$ to the $\mathcal{A}$ positions of a binary vector $\mathbf{v}$ and freezing to zero the bits of $\mathbf{v}$ with indices in $\mathcal{F}$. The codeword is generated as $\mathbf{c} = \mathbf{v}\mathbf{F}^{\otimes m}$ where $\mathbf{F}^{\otimes m}$ denotes the m-fold Kronecker power of Arikan's 2×2 polarization kernel [21]

$$\mathbf{F} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}.$$

The blocklength N is equal to 2^m. Efficient encoding follows by means of a fast Fourier transform-like architecture, with m layers. At each layer, $N/2$ operations are performed. An example of the polar encoder of a $(8,4)$ polar code with $m = 3$, $\mathcal{F} = \{0,1,2,4\}$ and $\mathcal{A} = \{3,5,6,7\}$ is shown in Figure 4.3

The performance of polar codes highly depends on the position of the frozen bits, and different methods have been proposed to optimize these positions [21, 94, 95].

[3] One could consider letting the memory ν of a CC grow proportionally with n (i.e., as the rate $R_0 = 1/n$ decreases), but this would lead to a significant increase in decoding complexity. Since one typically aims to maintain bounded complexity, this approach is generally not desirable.

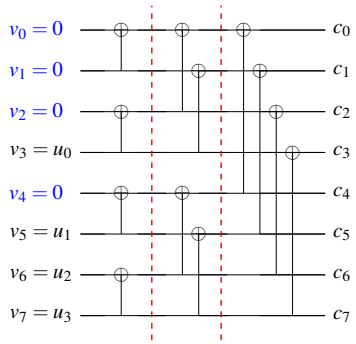


Fig. 4.3 Example of the $(8,4)$ polar encoder with $\mathcal{F} = \{0,1,2,4\}$ and $\mathcal{A} = \{3,5,6,7\}$.

Successive Cancellation Decoding

An efficient algorithm for decoding polar codes is successive cancellation (SC). It has the advantage of having a $O(N\log_2 N)$ complexity. Let $\mathbf{y}$ denote the received channel output, then SC decoder sequentially estimates each bit in $\mathbf{v}$, the encoder input vector, using log-likelihood ratios (LLRs) that incorporate both the channel observations and the decisions made on previously estimated bits. For the i-th bit v_i, the LLR is defined as

$$L_i = \log \frac{P_{\mathbf{Y}\mathbf{V}_1^{i-1}|V_i}\left(\mathbf{y}, \hat{\mathbf{v}}_1^{i-1} \mid v_i = 0\right)}{P_{\mathbf{Y}\mathbf{V}_1^{i-1}|V_i}\left(\mathbf{y}, \hat{\mathbf{v}}_1^{i-1} \mid v_i = 1\right)}$$

and a decision is taken as

$$\hat{v}_i = \begin{cases} 0, & \text{if } L_i \geq 0 \\ 1, & \text{if } L_i < 0 \end{cases}$$

if $i \in \mathcal{A}$, whereas $\hat{v}_i = 0$ if $i \in \mathcal{F}$. Here $\hat{\mathbf{v}}_1^{i-1} = (\hat{v}_1, \ldots, \hat{v}_{i-1})$. Due to the structure of polar codes, the computation of L_i is carried out recursively.

Successive Cancellation List Decoding

Building upon the SC decoder, successive cancellation list (SCL) decoding [80] maintains a list $\mathcal{L}$ of L candidate paths over a decoding tree, each corresponding to a different sequence of decisions $\hat{\mathbf{v}}_1^i$ up to the current bit index i. For each candidate path $\ell \in \mathcal{L}$, a path metric Λ_ℓ is defined to quantify its reliability.

The performance of polar codes can be improved by concatenating an inner polar code with an outer high-rate code under SCL decoding [80]. In this work, an outer $(K+m, K)$ polynomial code with generator polynomial of degree m is employed, resulting in a concatenated poly+polar code with blocklength N and dimension K. In the rest of the work, the polar code design of the 5GNR standard [79, 93] is adopted. This code construction is referred to as the 5GNR polar code.

The complexity of SCL w.r.t. the basic SC decoding is proportional to the list size L adopted. In order to decrease the average complexity, it is possible to adopt an adaptive SCL [63] procedure, where decoding $\mathbf{y}$ is first attempted using SC decoding, but if the decoded message $\hat{\mathbf{u}}$ does not satisfy the outer code constraint, the receiver tries SCL decoding with $L = 2$ and if none of the decoded messages in the list meets the outer code constraints, the list size is doubled and so on, until a solution that satisfies the outer parity-check equations is found, or a maximum list size $L_{\max}$ is reached.

4.2.3 Comparison Between Convolutional and Polar Codes

Figure 4.4 presents a performance comparison in terms of frame error rate (FER) between convolutional and polar codes in the low-rate regime. In the experimental setup, the information length has been fixed to $K = 100$. The poly+CCs are composed of a zero-tail terminated convolutional encoder with memory $\nu = 6$ and an outer polynomial code defined by a generator polynomial of degree 11 optimized as in [40]. These codes are decoded using the tree-trellis list Viterbi algorithm [55] with a maximum list size $L_{\max} = 2048$. The nominal rate is $R_0 = 1/n$, with $n = 2, 5$ and 8 (due to termination, the actual code rate is slightly lower than the nominal one). The blocklengths of the corresponding block codes are $234, 585$, and 936. The binary generator polynomials are taken from [96], and their generator polynomials expressed in octal notation[4] correspond to $[133, 171]$ for the rate$-1/2$ code, $[117, 127, 133, 153, 171]$ for the rate$-1/5$ code, and $[117, 127, 133, 133, 137, 153, 171, 171]$ for the rate$-1/8$ code. Note that each inner convolutional code achieves a free distance equal to the corresponding Heller's bound.

The 5GNR polar codes in this comparison are constructed with the same overall code rate as their convolutional counterparts. They are composed of the outer polynomial code with a polynomial generator of degree 11 from the 5G standard specifications. The polar codes are decoded with an adaptive SCL decoder, which uses a maximum list size of $L = 256$.

The performance curves in Figure 4.4 clearly indicate that while both families of codes exhibit comparable performance for the code rate $\approx 1/2$, polar codes outperform the CCs as the rate decreases. Moreover, no coding gain is visible when the convolutional code rate decreases from $R_0 = 1/5$ to $R_0 = 1/8$, as predicted by the analysis provided in Section 4.2.1. On the contrary, polar codes demonstrate coding gains when the code rate decreases, in line with the RCU bound predictions for the same rates (solid curves).

[4]The following example is made to clarify what is the bit ordering for the octal notation considered in this thesis. I.e., the value $[13]$ in octal notation denotes the binary vector $[1, 0, 1, 1]$ which corresponds to the binary polynomial $p(D) = 1 + D^2 + D^3$.

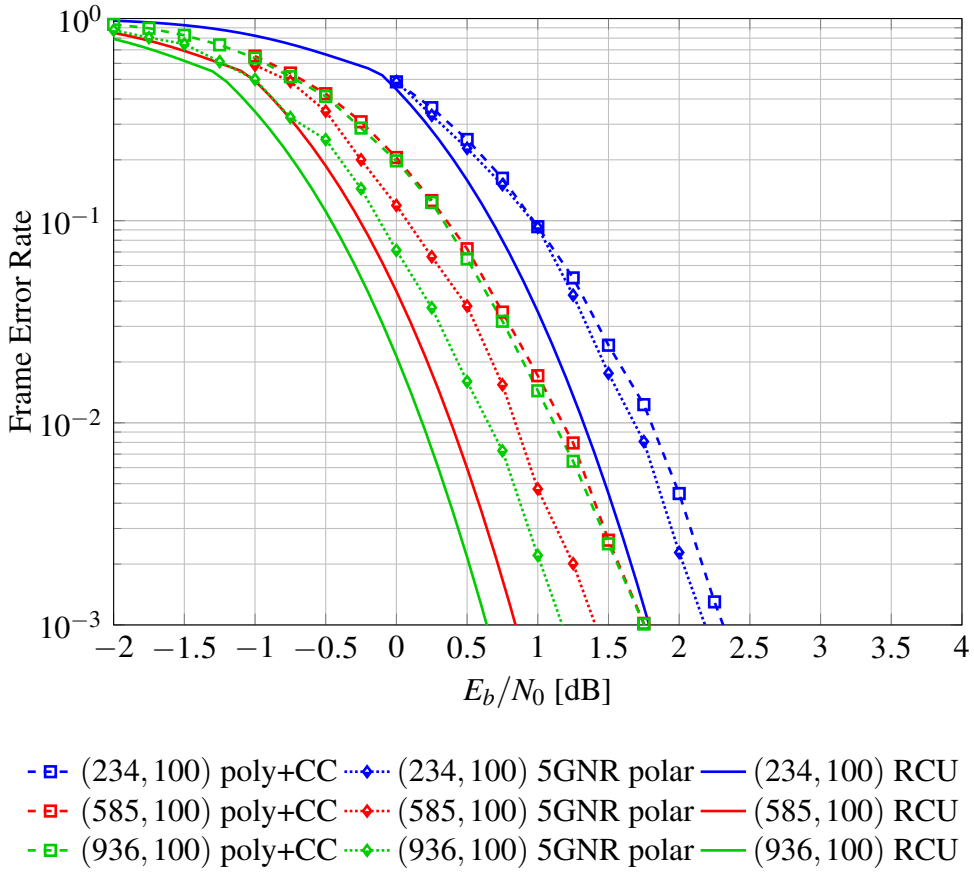


Fig. 4.4 Performance comparison in terms of frame error rate versus E_b/N_0 between convolutional and polar codes in the low-rate regime.

4.3 Enhanced Spread Spectrum ALOHA

Enhanced spread spectrum Aloha (E-SSA) [72] is a random access scheme designed to efficiently accommodate asynchronous transmissions from a large number of uncoordinated users. It represents an evolution of traditional spread spectrum Aloha [73] techniques, incorporating advanced coding and improving the multipacket reception capability of spread spectrum Aloha by canceling the interference contribution of decoded packets. The design of E-SSA is based on the 3GPP W-CDMA air interface and is optimized to harvest the gains that successive interference cancellation (SIC) can provide. Due to its outstanding performance and lean transmitter/receiver design, E-SSA has emerged as a high-efficiency random access solution, currently adopted in satellite IoT networks [97].

4.3.1 Transmitter Architecture

The transmitter in an E-SSA system primarily consists of four main functional blocks: encoding, spreading, modulation, and insertion of preamble and pilot symbols. Initially, user messages

are encoded using channel codes to facilitate robust detection and decoding in the presence of interference. Subsequently, the encoded symbols are spread using a pseudo-random spreading sequence with a period of at least the duration of the entire codeword, which provides interference mitigation capabilities in case of user collisions. Then, the spread symbols are modulated onto the physical channel, employing modulation schemes like binary phase-shift keying or quadrature phase-shift keying. Finally, known symbol sequences of preamble and pilot symbols are inserted into the modulated block to enable synchronization and detection procedures.

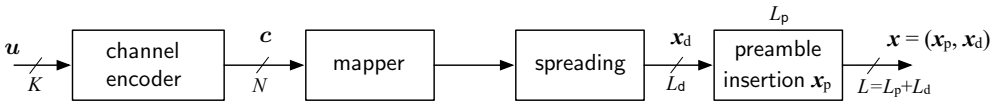


Fig. 4.5 Structure of E-SSA transmission chain.

4.3.2 Protocol Operation

E-SSA is an asynchronous and unframed protocol where users, independently, transmit data packets without prior coordination or synchronization. Each user randomly selects the transmission time according to their local clock, causing transmissions from different users to partially overlap in time. Due to the asynchronous and unframed nature, on one hand, the receiver faces significant challenges in distinguishing user transmissions from each other and from background noise; on the other hand, the asynchronous nature allows the use of a single spreading sequence and a single preamble sequence shared among all users to mitigate interference. In fact, users who transmit at different times experience mutual interference that closely resembles that of a synchronous spread spectrum system with random spreading sequences assigned to the different users [74].

4.3.3 Receiver Design

The receiver architecture for E-SSA involves advanced multi-user detection algorithms designed to resolve asynchronous transmissions effectively. Initially, the receiver attempts to locate within a time window the start of transmitted packets by performing a threshold test on the preamble correlation. Then, the received signals pass through a matched filtering stage matched to the spreading sequence. Following matched filtering, channel decoding, error detection, and SIC are applied to decode user packets.

4.4 Protocol Design for the Unsourced Multiple Access Channel

In this section, a specific E-SSA design tailored to the UMAC framework of [12] is introduced. First, the transmission protocol (Section 4.4.1), including the modifications applied to E-SSA to comply with the fixed frame size setting of [12] is defined. Then, the algorithms performed at the receiver (Section 4.4.2) are outlined. Finally, the role played by the timing channel in Section 4.4.3 is discussed.

4.4.1 Message Transmission

An example of E-SSA packet before zero-padding and delay shift is depicted in Figure 4.7, and the structure of the modified E-SSA transmission chain for UMAC is presented in Figure 4.6. A representation of K_a active users transmitting during the same frame over the GMAC is shown in Figure 4.8.

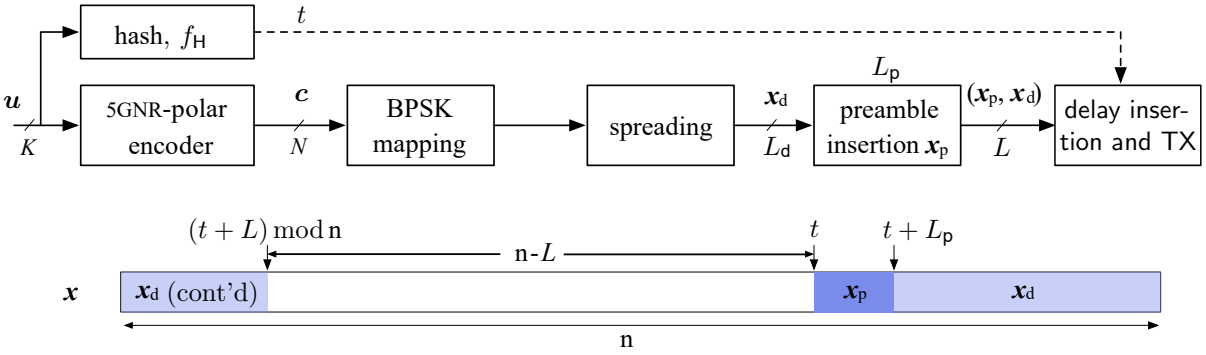


Fig. 4.6 Structure of the E-SSA UMAC transmission chain.

The transmission of a message resembles closely the one employed originally by E-SSA [72], with a few important differences. First, in order to retain the key advantage of asynchronous E-SSA systems–namely, the ability to use a single spreading and preamble sequence shared among users–a wrap-around approach is adopted in order to turn the continuous, unframed transmission behavior of E-SSA into a framed one. This means that the message part of an E-SSA packet that should be transmitted after the end of the frame length n is appended instead at the beginning of the frame. This choice is dictated by the interest in comparing the protocol performance with available performance bounds, which rely on a fixed frame size n [12]. Although framing introduces synchronization among users and aligns their transmissions within a common structure, a preamble is still required. Similarly to asynchronous E-SSA systems, in the proposed framed setting the receiver does not know a priori the starting positions of transmitted packets within the frame. The preamble is thus needed for packet detection.

However, the introduction of a common frame also enables a new opportunity: the definition of a timing channel, where the start time of a packet within the frame is determined as a function of the message. This additional degree of freedom is later exploited, as shown in Section 4.4.3, to improve the reliability of error detection and mitigate the false alarm problem. Second, owing to their excellent performance in the short blocklength regime, 5GNR polar codes are used instead of the turbo codes adopted in [72].

Transmission takes place by encoding the information message with the (N,K) 5GNR polar code, by spreading the codeword $\mathbf{c}$ through a spreading sequence

$$\mathbf{s} = (s_0, s_1, ..., s_{L_\mathsf{d}-1})$$

with a period L_d and spreading factor M, and by appending to the spread packet, denoted by $\mathbf{x}_\mathsf{d}$, a preamble $\mathbf{x}_\mathsf{p}$. The transmission proceeds with a start time obtained by hashing the information message. As for the original E-SSA design, both the preamble and the spreading sequence are unique, i.e., all users employ the same preamble and the same spreading sequence. Denote by

$$\mathbf{x}_\mathsf{p} = (x_{\mathsf{p},0}, x_{\mathsf{p},1}, \ldots, x_{\mathsf{p},L_\mathsf{p}-1})$$

the length-L_p preamble, with symbols $x_{\mathsf{p},i} \in \{-1,+1\}$ for $i = 0,\ldots,L_\mathsf{p}-1$. Furthermore, denote by

$$\mathbf{s} = (s_0, s_1, ..., s_{L_\mathsf{d}-1})$$

the spreading sequence with spreading factor $M = L_\mathsf{d}/N$ (the length of the spreading sequence is an integer multiple of the 5GNR polar code blocklength N). As for the preamble, the spreading sequence is binary ($s_i \in \{-1,+1\}$ for $i = 0,\ldots,L_\mathsf{d}-1$). The spreading sequence is partitioned into N disjoint blocks, one per coded bit, of M chips each, i.e.,

$$\mathbf{s} = (\mathbf{s}_0, \mathbf{s}_1, ..., \mathbf{s}_{N-1}),$$

where

$$\mathbf{s}_i = (s_{0+iM}, ..., s_{M-1+iM})$$

is the portion of the spreading sequence associated to the $i-$th modulated bit. Given the spreading sequence $\mathbf{s}$, for $\mathbf{c} \in \mathbb{F}_2^N$ the spread symbols are generated as

$$f_\mathsf{s}(\mathbf{c};\mathbf{b}) := ((-1)^{c_0}\mathbf{s}_0, (-1)^{c_1}\mathbf{s}_1, \ldots, (-1)^{c_{N-1}}\mathbf{s}_{N-1}),$$

where $\mathbf{x}_{\mathsf{d},i} = (-1)^{c_i}\mathbf{s}_i$ corresponds to the spread and modulated $i-$th bit.

Lastly, $f_\mathsf{H} : \mathbb{F}_2^K \mapsto [\mathrm{n}]$ denotes a uniform hash function. The transmission of a message proceeds as follows.

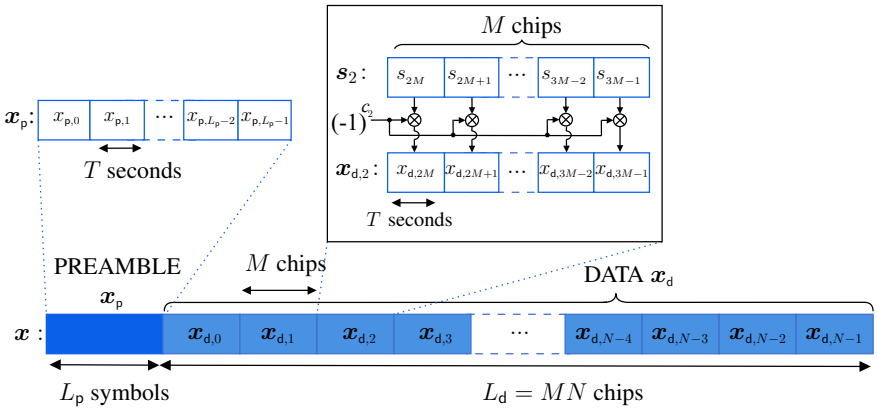


Fig. 4.7 Example of preamble insertion and spreading of a codeword $\mathbf{c}$.

1. *Polar Encoding:* The K-bits information vector $\mathbf{u}$ is encoded via the (N,K) 5GNR polar encoder, resulting in the N-bits codeword $\mathbf{c}$.

2. *Spreading:* The N-bits codeword is BPSK modulated and spread via the sequence $\mathbf{s}$, resulting in the length-L_{d} vector $\mathbf{x}_{\mathsf{d}} = f_{\mathsf{s}}(\mathbf{c};\mathbf{b})$.

3. *Preamble Insertion and Zero-Padding:* The preamble $\mathbf{x}_{\mathsf{p}}$ is appended to the vector $\mathbf{x}_{\mathsf{d}}$. The resulting the length-L vector $(\mathbf{x}_{\mathsf{p}},\mathbf{x}_{\mathsf{d}})$ is padded with $(\mathrm{n}-L)$ zeros, resulting into length-n vector $\mathbf{x}' = (\mathbf{x}_{\mathsf{p}},\mathbf{x}_{\mathsf{d}},0,0,\ldots,0)$.

4. *Hashing and Time Selection:* The starting time of the packet is obtained by hashing the information message as $t = f_{\mathsf{H}}(\mathbf{u})$.

5. *Delay Shift and Transmission.* The transmitted vector $\mathbf{x}$ is given by the circular-right shift of $\mathbf{x}'$ by t positions.

Due to the choice of the binary alphabet, the average power of an E-SSA packet is $P = L/\mathrm{n}$. Note that the system employs a pre-defined preamble and a single, pre-defined spreading sequence, i.e., each user employs the same preamble and the same spreading sequence, which are known at the receiver.

4.4.2 Detection and Decoding

For each received frame, the receiver runs iteratively. In each iteration, a list of candidate start-of-packet positions is produced through a preamble search. For each candidate position, a decoding attempt is performed via adaptive SCL decoding [63]. If the decoder list contains a codeword that satisfies the CRC code constraints, the associated information message is hashed with the function f_{H}, and the output is checked with the start time of the detected preamble. If the output of the hash function matches the start time, the decision is considered correct and the

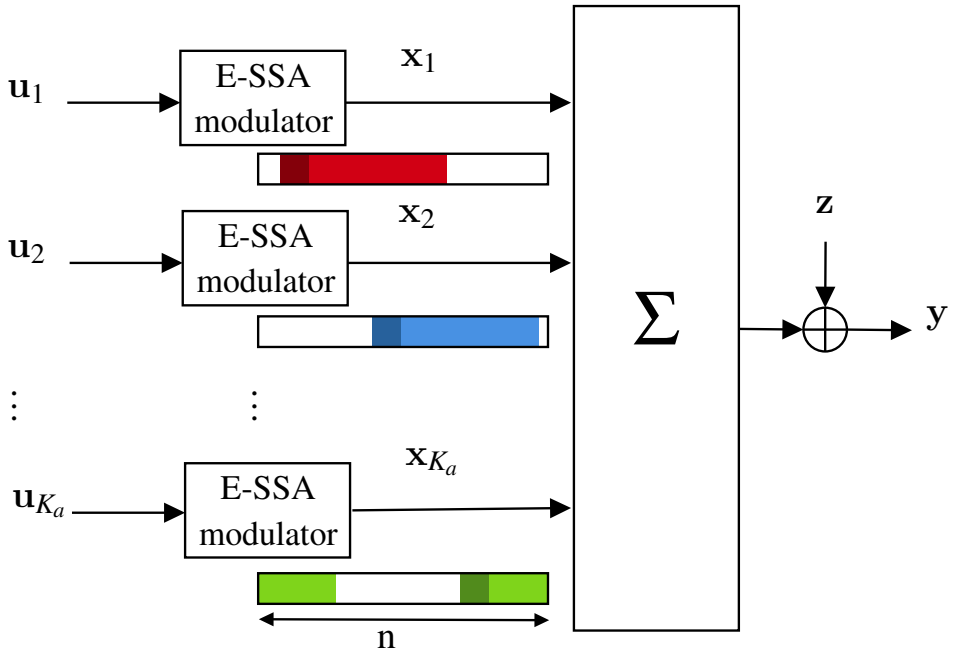


Fig. 4.8 Representation of the transmission over the GMAC of K_a simultaneously active users who are employing the modified E-SSA transmission scheme.

decoded packet interference contribution is removed from the received signal. The behavior of the receiver is detailed next.

Initialization: The iteration count is set to $I = 0$, and $\widetilde{\mathbf{y}} = \mathbf{y}$.

Iterative Decoding and SIC: At the ℓ-th iteration, the following steps are performed:

1. *Preamble Detection:* The correlation between the preamble and the vector $\widetilde{\mathbf{y}}$ is computed at each $t \in [\mathrm{n}]$ as
$$\Lambda_t = \sum_{j=0}^{L_\mathrm{p}-1} x_{\mathrm{p},j} \widetilde{y}_{(j+t) \bmod \mathrm{n}}.$$
The times associated with the W largest values of Λ_t are stored in the list $\mathcal{T}$.

2. *Despreading, Decoding, and SIC:* For each $t \in \mathcal{T}$

   a. The observation vector $\mathbf{r}$ is extracted as $\mathbf{r} = (\widetilde{y}_{t'}, \ldots, \widetilde{y}_{t''})$ where $t' = (t + L_\mathrm{p}) \bmod \mathrm{n}$ and $t'' = (t + L - 1) \bmod \mathrm{n}$.

   b. A soft-estimate of the jth codeword bit is obtained by matched filtering (despreading) as
$$\tilde{r}_j = \frac{1}{M} \sum_{i=0}^{M-1} r_{jM+i} s_{jM+i}.$$

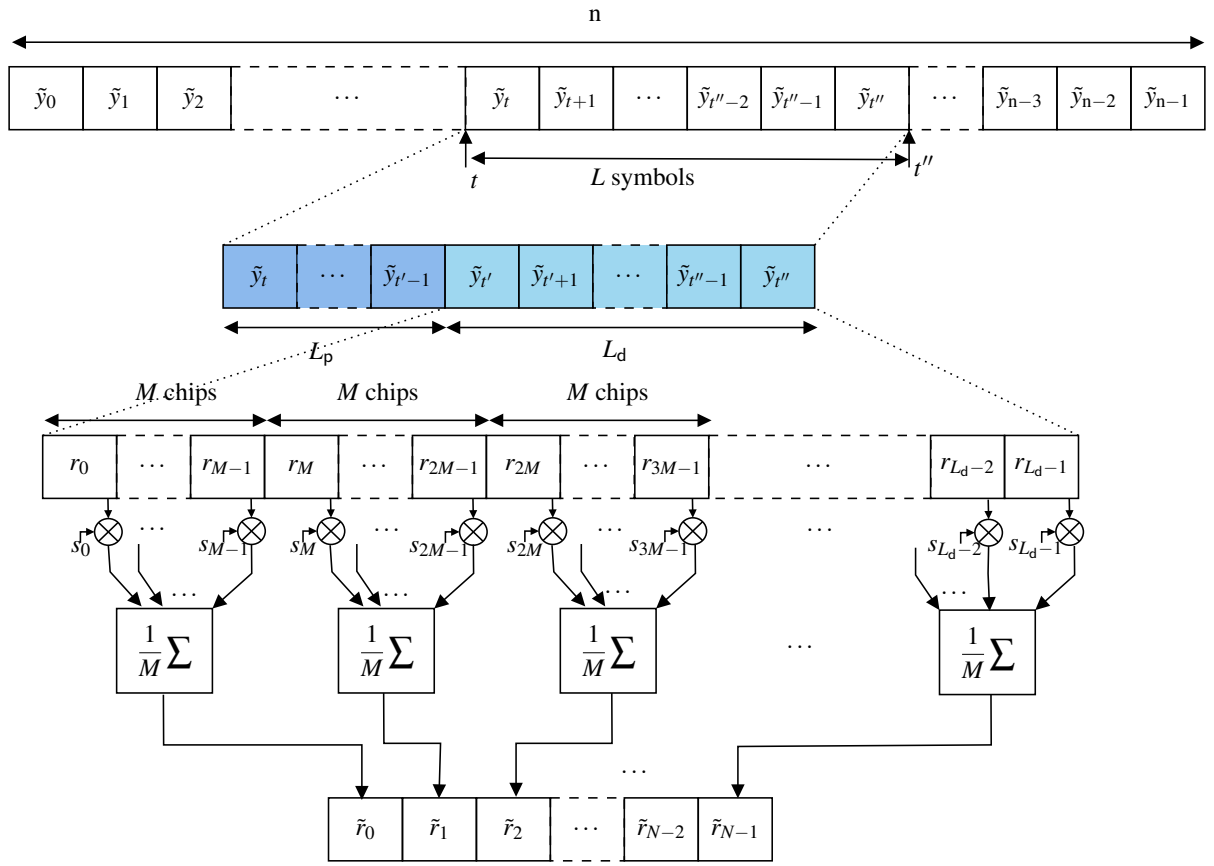


Fig. 4.9 Example of extraction and despreading of the content of a detected possible transmitted packet which starts at time t.

c. The soft-estimate vector $\tilde{\mathbf{r}} = (\tilde{r}_0, \tilde{r}_1, \ldots, \tilde{r}_{N-1})$ is input to the polar code adaptive SCL decoder [63]. The decoder returns a valid codeword $\hat{\mathbf{c}}$ or an erasure flag (if no polar code word in the SCL decoder list satisfies the CRC code constraints).

d. If the decoder outputs a decision $\hat{\mathbf{c}}$, the corresponding information message $\hat{\mathbf{u}}$ is hashed generating the time index $\hat{t} = f_{\mathsf{H}}(\hat{\mathbf{u}})$. If $\hat{t} = t$, then the message $\hat{\mathbf{u}}$ is deemed as correct and it is included in the output list $\mathsf{D}(\mathbf{y})$.

e. If the check at point 2.d succeeds, the message is re-encoded according to the transmission procedure described in Section 4.4.1, resulting in the vector $\mathbf{x}(\hat{\mathbf{u}})$, which is subtracted from the vector $\mathbf{y}$

$$\widetilde{\mathbf{y}} = \widetilde{\mathbf{y}} - a\mathbf{x}(\hat{\mathbf{u}})$$

where a is the estimate of the channel amplitude[5] obtained by normalizing the soft-correlation between $\mathbf{y}$ and $\mathbf{x}(\hat{\mathbf{u}})$ by the vectors' length L.

[5]Note that the detection/decoding algorithm described in this section does not make use of any prior information on the unitary channel amplitude.

Steps 1 and 2 are iterated for a maximum number of iterations $I_{\max}$. As an early stopping criterion, the process ends if—within an iteration—no decoding attempts succeed according to the test described in step 2.d. It is important to note that, if the maximum number of iterations $I_{\max}$ is fixed and the size W, of the time instants with the largest correlation, is scaled with the number of active users K_a, the detection/decoding algorithm outlined above entails a complexity that is linear in K_a.

It is important to remark that when two users transmit with the same starting time t, the combined preambles help identify the starting time, whereas the use of a low-rate channel code can reduce the mutual interference. This enables, under certain conditions, the correct recovery of one of the messages of the two users, and its interference is then canceled using the SIC mechanism, allowing the message of the non-decoded user to be recovered in subsequent iterations.

4.4.3 Error Detection via Timing Channel

The use of the timing channel for error detection, as described in the algorithm presented in Section 4.4.2 (step 2.d) allows for a drastic reduction of the *false alarm rate*, i.e., the rate at which the decoder outputs packets that were not transmitted. False alarms have a detrimental effect on the efficiency of the system since they introduce artificial interference through the SIC process. It is hence of paramount importance to keep the false alarm rate some orders of magnitude lower than the target PUPE. In simulations, for instance, with 125 active users, exactly three false alarms were counted in 800 simulated frames, whereas with $K_a = 100$ and $K_a = 75$ no false alarms were detected. As an alternative to the use of the timing channel, one may improve the error detection capability of the SCL decoder. The result can be achieved by limiting the SCL list size or by introducing a stronger polynomial code. In either case, the price to be paid is a deterioration of the error correction capability of the decoder, resulting in a loss of coding gain, thus in a loss of energy efficiency. A detailed quantification of the trade-off between error detection and error correction capability of the system in the unsourced setting is nontrivial and highly dependent on code parameters and system configuration. Related analyses on how different error detection strategies affect both the undetected errors and the total error rate in the single-user scenario with polar codes is presented in [98, 99] where some of these effects are studied in depth. Finally, An obvious question relates to the practicality of using the timing information in a real system. Answering this question thoroughly goes beyond the scope of this analysis. However, a possible direction to explore is the use of a beacon signal (transmitted periodically by the base station) to announce the start of the frame, which is used by the terminals as a reference to compute the transmission time.

4.5 Numerical Results

In this section, numerical results obtained via Monte Carlo simulations are provided. The results are obtained for a frame size n = 30000. A $(1000, 100)$ 5GNR polar code is used, and the number of iterations is set to $I_{\max} = 50$. The target PUPE is $\varepsilon^\star = 5 \times 10^{-2}$. For the simulations, a single randomly generated spreading sequence was used, with symbols that are independent and uniformly distributed in $\{\pm 1\}$.

4.5.1 Parameter Optimizazion

A first set of results deals with the choice of the spreading factor. In Figure 4.10, the SNR required to achieve the target PUPE ε^* is reported as a function of the spreading factor M. The results are provided for various numbers of active users K_a and are obtained under genie-aided preamble detection. That is, the preamble is omitted and the receiver has perfect knowledge of the starting time of each user. The chart shows how small spreading factors tend to penalize system performance. For larger spreading factors, and at low channel loads (small K_a), the required SNR closely approaches that of the single-user case. The result holds up to $K_a = 75$. Above this value, a rapid deterioration in performance is observed.

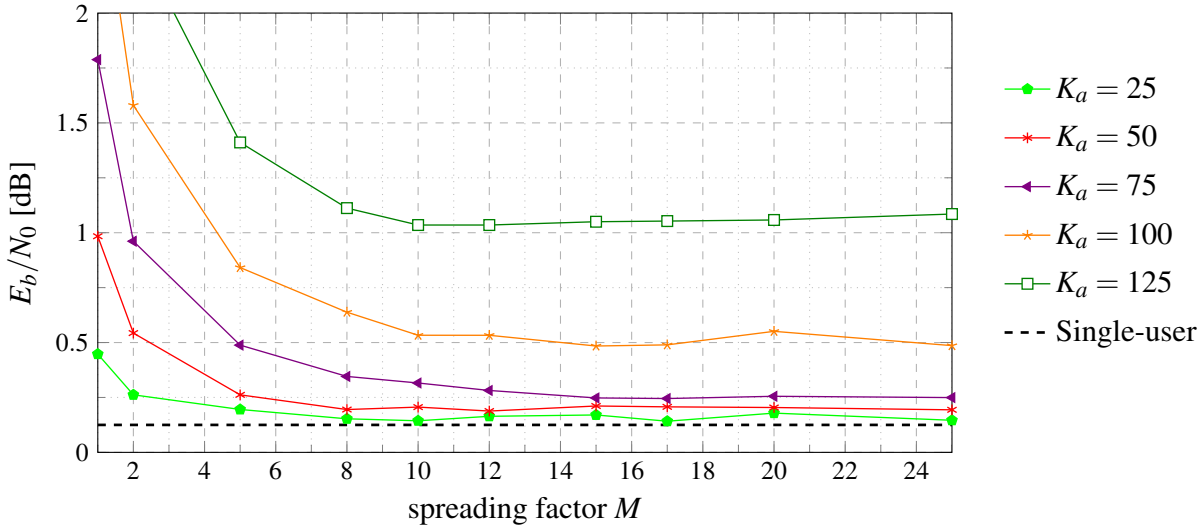


Fig. 4.10 Effect of the spreading factor M on the minimum E_b/N_0 required for achieving the target PUPE $\varepsilon^\star = 5 \times 10^{-2}$ for some values of active users K_a. Genie-aided preamble detection. $(1000, 100)$ 5GNR polar code.

This behavior is analyzed in Figure 4.11. Here, the PUPE is depicted as a function of the SNR, for various numbers of active users. The spreading factor is set to 25. On the same chart, the RCU [9] for $(1000, 100)$ codes is provided, as well as the single-user $(1000, 100)$ 5GNR polar code performance. The PUPE approaches the single-user performance at low enough error probabilities; the SNR at which this happens varies with the number of active users. When the

number of active users is sufficiently low, the PUPE tightly matches the single-user probability of error already at error probabilities larger than the target $\varepsilon^\star$. As the number of users grows, the convergence happens at PUPE values that are below $\varepsilon^\star$, giving rise to a visible increase of the required SNR. An analysis of this phenomenon, which is common in multiuser systems, was provided in [100].

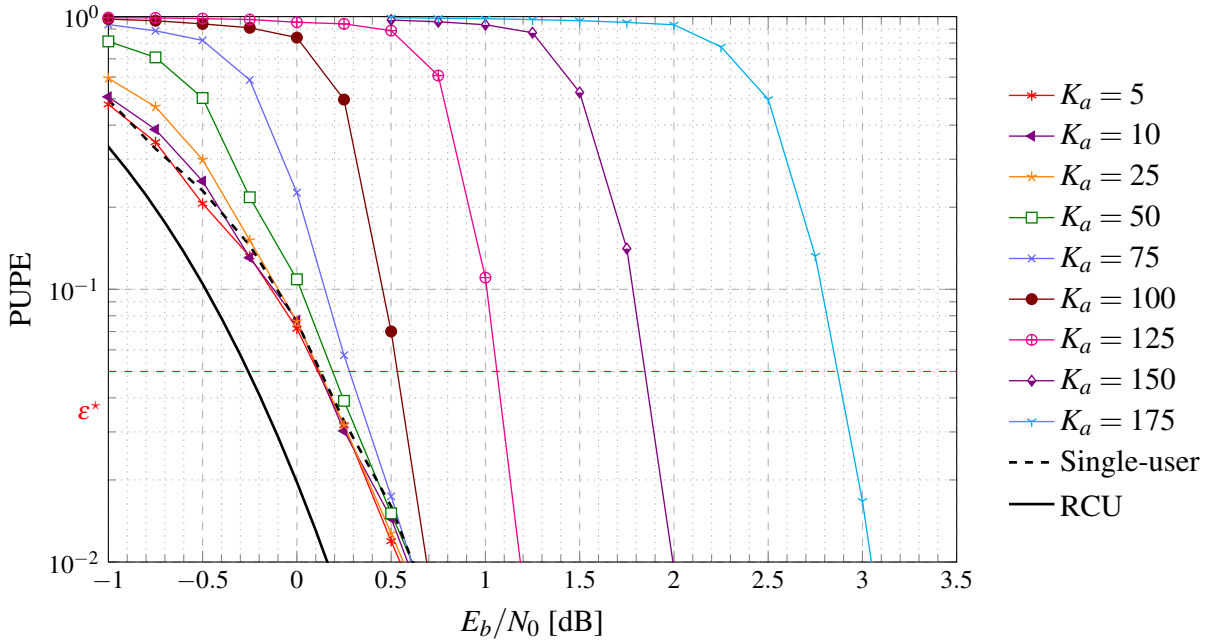


Fig. 4.11 Effect of the channel load K_a on the PUPE in a system with no preamble. Spreading factor $M = 25$. Genie-aided preamble detection. $(1000, 100)$ 5GNR polar code.

A second set of results analyzes the impact on the system performance of the number W of start-of-packet candidates used at step 1 of the algorithm described in Section 4.4.2. The results, obtained for spreading factor $M = 15$ and for various preamble lengths L_p, are depicted in Figure 4.12 in terms of SNR required to achieve the target PUPE as a function of the energy overhead introduced by the preamble, given by $\Delta E := 1 + L_\mathsf{p}/L$. As expected, a larger preamble overhead (larger L_p) enables accurate preamble detection already with small values of W, resulting in a smaller average number of decoding attempts. However, the result comes at the expense of an increase in the energy cost entailed by the preamble. Operating the system with lower preamble lengths allows limiting the energy loss. Nevertheless, the preamble miss-detection probability can be kept low only by increasing W, with an obvious implication in complexity due to the larger number of decoding attempts. The analysis highlights the trade-off that exists at fixed SNR/PUPE between a reduction of the preamble overhead and the corresponding increase in decoding complexity, suggesting that the preamble length should be selected by finding an acceptable compromise between energy overhead and the size of the list of start-of-packet candidates W.

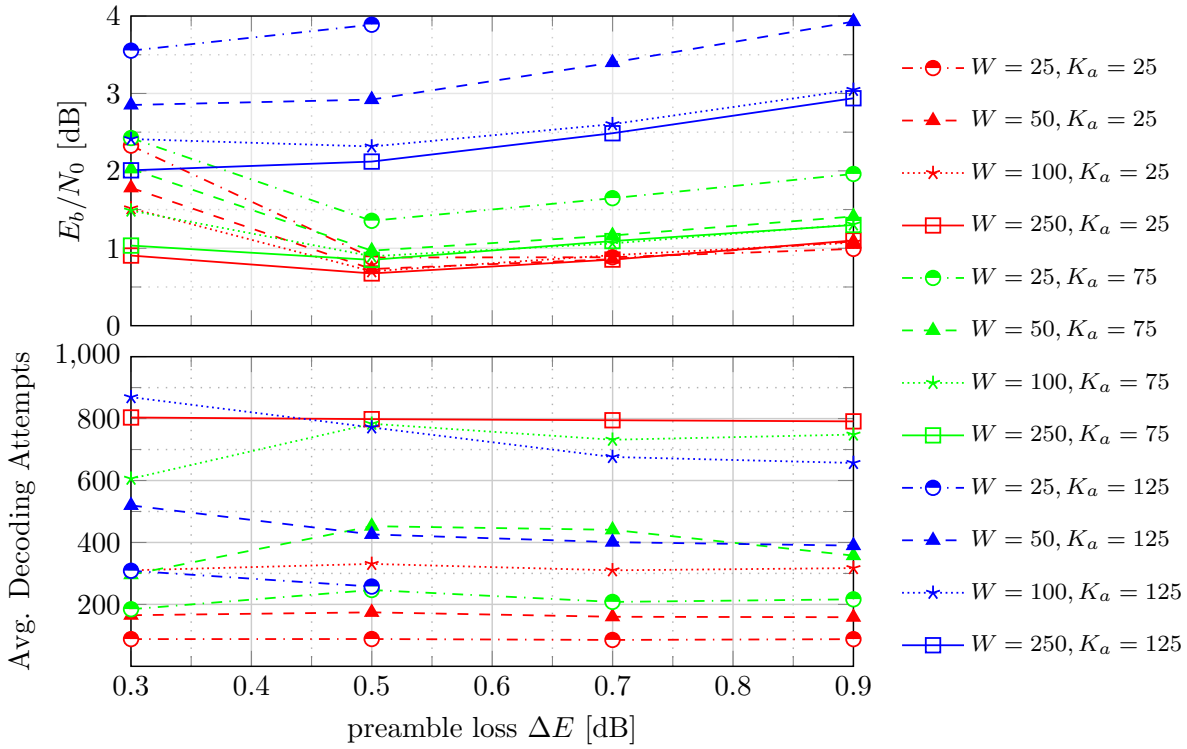


Fig. 4.12 Minimum E_b/N_0 (above) and average number of decoding attempts (below) for $\varepsilon^\star = 5 \times 10^{-2}$ as a function of the preamble loss, for spreading factor $M = 15$.

4.5.2 Performance Comparison with Polar and Convolutional Codes

Figure 4.13 compares the performance of two E-SSA schemes against the UMAC achievability bound [12]. A first scheme employs a 5G NR polar code (E-SSA, 5GNR polar), and a second scheme relies on a convolutional code (E-SSA, poly+CC). Both schemes use a E-SSA preamble length that translates to an energy overhead of approximately $\Delta E \approx 0.5\,\text{dB}$. The system is configured with a spreading factor $M = 25$ and $W = 250$. For the polar-coded E-SSA, a $(1000, 100)$ 5GNR polar code is used. Decoding is performed by an adaptive SCL decoder [80, 63] with a maximum list size of $L = 256$. In contrast, the convolutionally coded E-SSA employs a $(936, 100)$ poly+CC design, using an optimized [40] outer polynomial code of degree 11 and an inner zero-tail terminated convolutional code whose generator polynomials in octal notation are $[117, 127, 133, 133, 137, 153, 171, 171]$ [96]. This convolutional code has a nominal encoder rate $R_0 = 1/8$. At the receiver, a tree-trellis list Viterbi algorithm [55] with a maximum list size of $L = 2048$ is used for decoding. Figure 4.13 highlights the critical role played by the selected code. The scheme based on polar codes provides a superior single-user coding gain relative to the convolutionally coded version, which brings two main advantages:

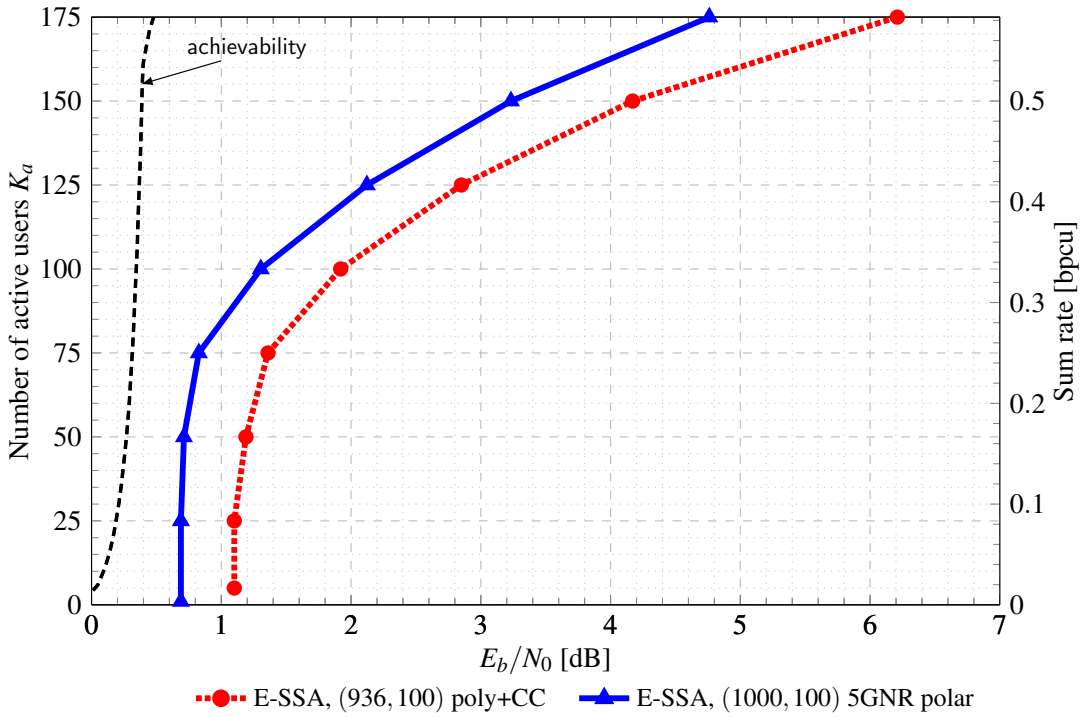


Fig. 4.13 E_b/N_0 required to achieve the target PUPE $\varepsilon^\star = 5 \times 10^{-2}$ for polar vs convolutional coded E-SSA for different number of active users K_a.

1. Lower energy requirements at small to moderate channel loads: for up to about 50–75 active users, the polar-coded solution operates at roughly 0.4 dB less E_b/N_0 than the convolutionally coded scheme.

2. Higher user capacity: at a fixed E_b/N_0, polar code supports more users. For instance, at $E_b/N_0 = 1.4$ dB, the poly+CC scheme supports 75 active users, while the 5GNR polar version accommodates 100, an increase of approximately 33%.

4.5.3 State-Of-The-Art Comparison

Figure 4.14 provides a comparison of E-SSA scheme based on $(1000, 100)$ 5GNR polar codes with the UMAC achievability bound and several UMAC schemes. The results are obtained by setting the E-SSA preamble length to $L_\mathsf{p} = 3050$ and a spreading factor $M = 25$, yielding an energy overhead $\Delta E \approx 0.5$ dB. The E-SSA performance is provided for $W = 100$ and for $W = 250$. The performance with genie-aided preamble detection is given as a reference, together with the UMAC achievability bound from [12]. The comparison includes the 5GNR two-step random access procedure [79] under SIC with parameters that have been chosen to match the simulation setup [89], the sparse Kronecker-product coding of [101], the synchronous spread spectrum scheme of [75], the enhanced irregular repetition slotted Aloha scheme of [86], sparse interleave-division multiple access [77] (which exploits joint multiuser decoding), and sparse

regression codes / coded compressive sensing [87]. The performance of E-SSA shows to be competitive, especially in the moderate load regime (up to 100 active users), requiring only single-user detection and decoding blocks at the receiver side. Among the competitors, only the schemes [75, 101] outperform E-SSA over the entire channel load range, with a gain that is nevertheless limited to 0.2 dB for [75] and 0.5 dB for [101] up to 75 users.

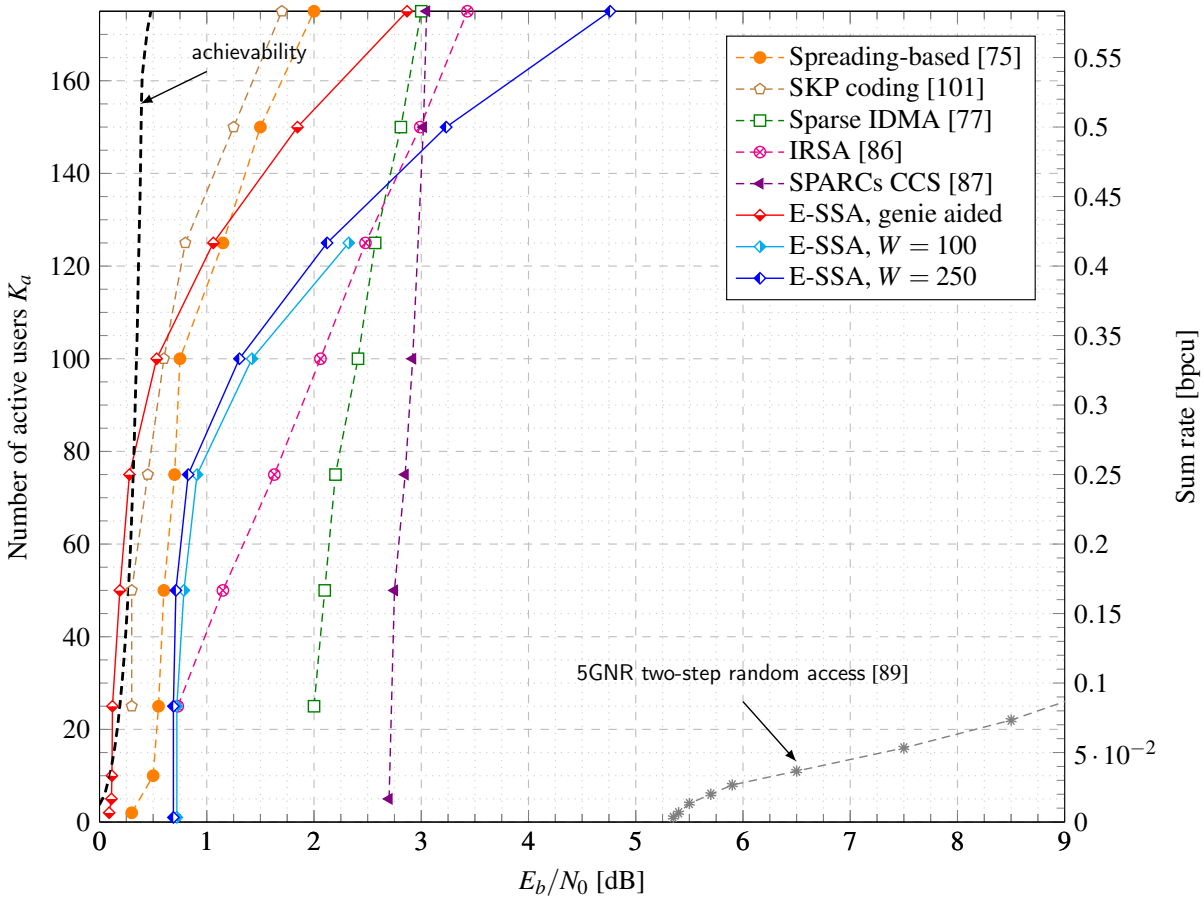


Fig. 4.14 E_b/N_0 required to achieve the target PUPE $\varepsilon^\star = 5 \times 10^{-2}$.

4.6 Final Remarks

We have investigated the limitations of short low-rate convolutional codes and compared these codes with short low-rate polar codes for unsourced multiple access channel (UMAC). We analyzed the performance of the enhanced spread spectrum Aloha (E-SSA) protocol in the framework of UMAC. The asynchronous, unframed transmission of E-SSA has been modified to enable a comparison with framed UMAC schemes. We have improved the energy efficiency of the scheme by introducing a short polar code and a timing channel. We have shown how the design of the different components of E-SSA affects the receiver complexity and the energy efficiency of the system. Results show that a careful design of E-SSA yields a performance that

is competitive with state-of-the-art UMAC schemes, with a simple transmitter and linear receiver complexity in the number of active users.

Chapter 5

Frame Synchronization of Direct-Sequence Spread Spectrum Systems

Recognizing the energy overhead introduced by the preamble in enhanced spread spectrum Aloha (E-SSA), this chapter addresses a frame synchronization strategy that may substantially mitigate such cost. In particular, this chapter addresses the sequential frame synchronization problem of direct-sequence spread spectrum (DSSS). The frame synchronization problem is introduced in Section 5.1. Section 5.2 formalizes the sequential frame synchronization problem for a coherent and a non-coherent channel model. Section 5.3 introduces for the coherent channel model the associated optimum and simplified suboptimal tests. Section 5.4 provides analytical performance results of the test proposed in Section 5.3 and validates them numerically. Section 5.5 provides a derivation of the optimum and simpler suboptimal tests for the non-coherent channel model setup, verifying their effectiveness through numerical simulations. Then, Section 5.6 applies the new tests to improve the enhanced spread spectrum Aloha protocol over the unsourced multiple access channel studied in Chapter 4. Final remarks follow in Section 5.7.

5.1 Frame Synchronization

Frame synchronization plays an important role in communication systems, ensuring that received signals are properly segmented into frames before further processing and decoding. The accuracy of this step directly influences system performance: a missed detected transmission results in lost data, while a falsely detected one activates subsequent receiver stages unnecessarily, increasing energy consumption and potentially leading to the decoding of non-existent packets. Therefore, establishing a reliable synchronization strategy is essential. This holds true also in DSSS systems where the choice of the spreading factor, the design of the preamble sequence, and the detection strategy must be jointly optimized to withstand adverse channel conditions.

Foundational works have demonstrated that the widely adopted correlation-based approach in communication systems, while simple, can be suboptimal even in standard formulations of the problem. This is the case, for example, of sequential frame synchronization on the additive white Gaussian noise (AWGN) channel [102]. Massey's seminal study [103] laid the groundwork for optimal frame synchronization in AWGN channels, showing that metrics derived from likelihood ratio tests yield the best performance. Subsequent research by Robertson [104] and by Chiani and Martini [102, 105, 106] introduced optimal and computationally efficient suboptimal methods for different communication scenarios, offering performance close to the theoretical optima at reduced complexity. Building on these insights, this paper develops and compares optimal and suboptimal metrics for frame synchronization in DSSS systems with BPSK modulation over the AWGN channel, including the non-coherent channel setup [107]. The focus is on the sequential frame synchronization setting [102]. Results, validated through analytical performance evaluations and numerical simulations, show that the proposed metrics surpass optimum frame synchronization algorithms that make use of the preamble only, ignoring the structure of the DSSS signal.

5.2 Problem Statement

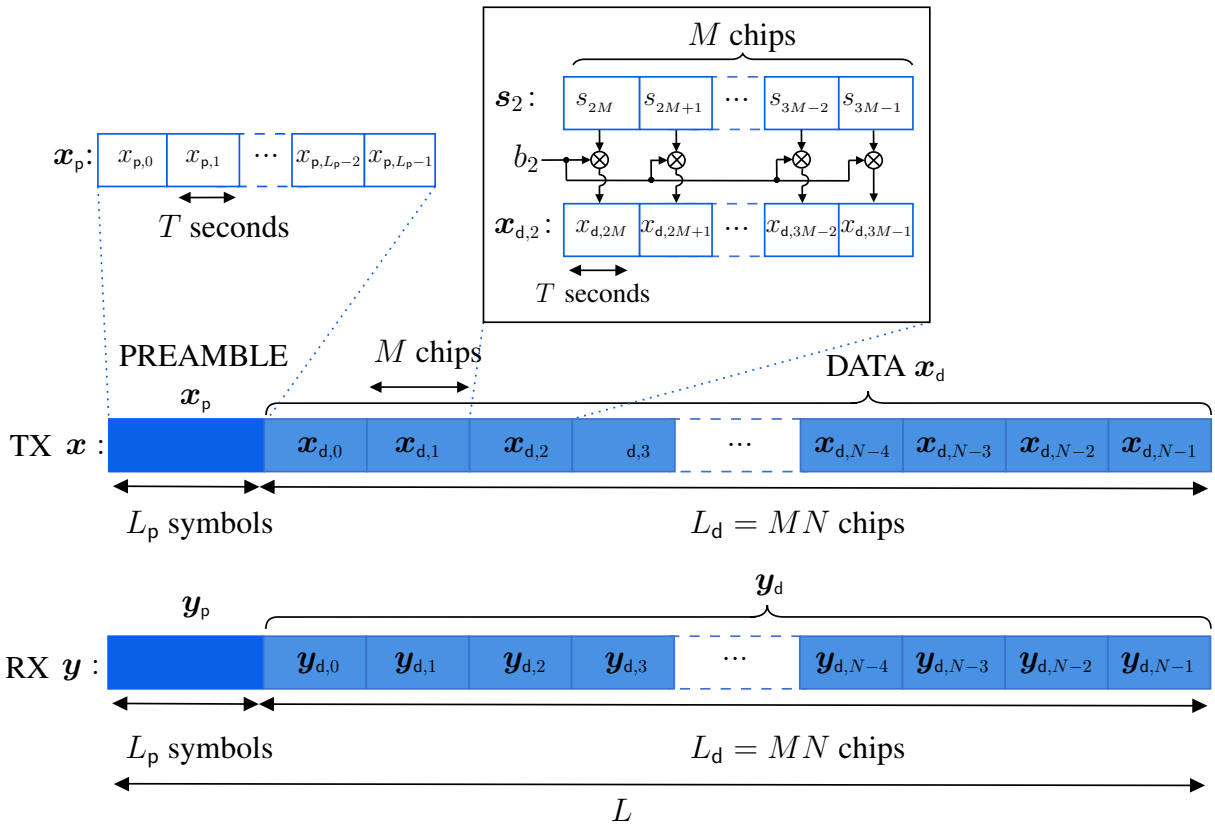


Fig. 5.1 Frame structure.

In this section, the system model and the sequential frame synchronization problems are introduced, considering two primary use cases. The first scenario involves detecting whether a packet is present or not (useful in bursty transmissions). The second scenario considers identifying the start of a frame within a continuous stream of data (useful when the transmission is continuous and frame boundaries need to be determined).

5.2.1 System Model

Figure 5.1 shows an example of the transmitted and received frames. The transmission of a BPSK modulated packet $\mathbf{x} = (x_0, x_1, \ldots, x_{L-1})$ composed of a preamble sequence $\mathbf{x}_\mathsf{p}$ and a data part $\mathbf{x}_\mathsf{d}$ is considered, hence $\mathbf{x} = (\mathbf{x}_\mathsf{p}, \mathbf{x}_\mathsf{d})$. In particular, it is

$$x_i = \begin{cases} x_{\mathsf{p},i} & \text{if } 0 \leq i < L_\mathsf{p} \\ x_{\mathsf{d},i-L_\mathsf{p}} & \text{if } L_\mathsf{p} \leq i < L. \end{cases}$$

The sequence $\mathbf{x}_\mathsf{p} = (x_{\mathsf{p},0}, x_{\mathsf{p},1}, \ldots, x_{\mathsf{p},L_\mathsf{p}-1})$, $x_{\mathsf{p},i} \in \{\pm 1\}$, represents the preamble of L_p symbols, whereas the data vector $\mathbf{x}_\mathsf{d} = (x_{\mathsf{d},0}, x_{\mathsf{d},1}, \ldots, x_{\mathsf{d},L_\mathsf{d}-1})$, $x_{\mathsf{d},i} \in \{\pm 1\}$, is a vector of $L_\mathsf{d} = MN$ chips, obtained by spreading N data bits $\mathbf{b} = (b_0, \ldots, b_{N-1})$ with a spreading sequence $\mathbf{s}$ of length L_d and spreading factor M. The total packet length is $L = L_\mathsf{p} + L_\mathsf{d}$. Throughout the text, to simplify the notation, we are going to slightly abuse the notation and refer to b_i as the associated BPSK symbol for the $i-$th bit, thus $b_i \in \{\pm 1\}$. It follows that

$$x_{\mathsf{d},j} = b_i s_j, \quad i = \lfloor j/M \rfloor.$$

Moreover, the $L_\mathsf{d}-$ chips spreading sequence $\mathbf{s}$ is partitioned into N subsequences of M chips each, i.e.,

$$\mathbf{s} = (s_0, s_1, \ldots, s_{L_\mathsf{d}-1}) = (\mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_{N-1}),$$

where

$$\mathbf{s}_k = (s_{0+kM}, \ldots, s_{M-1+kM})$$

is the portion of the spreading sequence associated with the $k-$th bit. Similarly,

$$\mathbf{x}_\mathsf{d} = \left(\mathbf{x}_{\mathsf{d},0}, \mathbf{x}_{\mathsf{d},1}, \ldots, \mathbf{x}_{\mathsf{d},N-1}\right)$$

where $\mathbf{x}_{\mathsf{d},k}$ denotes the portion of the spread data associated with the k-th bit, that is

$$\mathbf{x}_{\mathsf{d},k} = (x_{\mathsf{d},0+kM}, \ldots, x_{\mathsf{d},M-1+kM}) = b_k \mathbf{s}_k.$$

The structure of the transmitted packet is shown in Figure 5.1.

The signal is transmitted through an AWGN channel. The *sequential* frame synchronization setting of [102] is investigated, which consists of a receiver that continuously monitors the incoming signal using a sliding observation window. Each received window contains L samples, corresponding to the length of the transmitted DSSS signal $\mathbf{x}$. At each step, a synchronization metric—such as the correlation with the known preamble—is computed over the relevant portion of the window. If the computed metric exceeds a predefined threshold, a frame start is declared at the current position; otherwise, the observation window is shifted forward by one sample, and the process repeats. Under this framework, the problem has been reformulated as a binary hypothesis testing problem, where upon observing L samples of the channel output, one has to decide whether a packet transmission occurred. In particular, $\mathcal{H}_0$ denotes the hypothesis that the packet is transmitted. This manuscript, focuses on two channel setups, the first one, typical of a coherent receiver, works under the assumption of a known carrier frequency and perfect carrier frequency and phase synchronization (i.e., any carrier frequency offset is negligible or corrected), and after matched filtering and ideal carrier frequency and timing recovery, and assuming no inter-symbol interference, the discrete-time model of the received signal under $\mathcal{H}_0$ can be expressed as

$$\mathbf{Y} = \mathbf{X} + \mathbf{Z}, \tag{5.1}$$

with $Z_i \sim \mathcal{N}\left(0, \sigma^2\right)$ i.i.d. Gaussian. Whereas, in the second channel setup, a not-known uniform random phase rotation that is constant over the L observed channel outputs affects the transmitted signal. In this second model, the received signal under $\mathcal{H}_0$ can be expressed as

$$\mathbf{Y} = \mathbf{X}e^{j\theta} + \mathbf{Z}, \tag{5.2}$$

with $Z_i \sim \mathbb{C}\mathcal{N}\left(0, 2\sigma^2\right)$ i.i.d. circularly symmetric complex Gaussian and $\theta \sim \mathcal{U}\left([-\pi, \pi]\right)$ the uniform random phase rotation.

Moreover, X_i is the $i-$th element of the random vector $\mathbf{X}$, which results from modeling the data bits as N i.i.d. Bernoulli r.v.s $B_0, B_1, \ldots, B_{N-1}$ with

$$P(B_i = +1) = P(B_i = -1) = \frac{1}{2}.$$

An example of the received frame structure under hypothesis $\mathcal{H}_0$ is depicted in Figure 5.1.
For the hypothesis $\mathcal{H}_1$, two cases are considered:

Bursty Transmission (Scenario 1)

In a first scenario, representative of bursty packet transmission, under $\mathcal{H}_1$ the channel output is a sequence of L Gaussian noise samples. In the **coherent** channel setup, the hypothesis testing

problem is therefore defined by

$$\begin{aligned} \mathcal{H}_0 &: \mathbf{Y} = \mathbf{X} + \mathbf{Z} \\ \mathcal{H}_1 &: \mathbf{Y} = \mathbf{Z} \end{aligned} \tag{5.3}$$

where $Z_i \sim \mathcal{N}(0, \sigma^2)$ are i.i.d. r.v.s which model real-valued Gaussian noise. Whereas, in the **non-coherent** system model, the two hypothesis are

$$\begin{aligned} \mathcal{H}_0 &: \mathbf{Y} = \mathbf{X}e^{j\theta} + \mathbf{Z} \\ \mathcal{H}_1 &: \mathbf{Y} = \mathbf{Z}. \end{aligned} \tag{5.4}$$

with $Z_i \sim \mathbb{C}\mathcal{N}\left(0, 2\sigma^2\right)$ i.i.d. circularly symmetric complex Gaussian and $\theta \sim \mathcal{U}([-\pi, \pi])$ a uniform random phase rotation.

Continuous Transmission (Scenario 2)

In the second scenario, representative of continuous data transmission, under $\mathcal{H}_1$, the channel output is the superposition of Gaussian noise and a sequence of L chips, $\mathbf{c} = (c_0, c_1, \ldots, c_{L-1})$ where the chips are modeled as realizations of i.i.d. Bernoulli r.v.s, with

$$P(C_i = +1) = P(C_i = -1) = \frac{1}{2}.$$

Hence, in the **coherent** channel setup, the two hypotheses are

$$\begin{aligned} \mathcal{H}_0 &: \mathbf{Y} = \mathbf{X} + \mathbf{Z} \\ \mathcal{H}_1 &: \mathbf{Y} = \mathbf{C} + \mathbf{Z} \end{aligned} \tag{5.5}$$

where $Z_i \sim \mathcal{N}(0, \sigma^2)$ are i.i.d. r.v.s which model real-valued Gaussian noise. On the contrary, for the **non-coherent** system model, the hypothesis testing problem is defined by

$$\begin{aligned} \mathcal{H}_0 &: \mathbf{Y} = \mathbf{X}e^{j\theta} + \mathbf{Z} \\ \mathcal{H}_1 &: \mathbf{Y} = \mathbf{C}e^{j\theta} + \mathbf{Z}, \end{aligned} \tag{5.6}$$

with $Z_i \sim \mathbb{C}\mathcal{N}\left(0, 2\sigma^2\right)$ i.i.d. circularly symmetric complex Gaussian (with $N_0 = 2\sigma^2$) and $\theta \sim \mathcal{U}([-\pi, \pi])$ a uniform random phase rotation.

Given a channel observation realization $\mathbf{y} = (y_0, y_1, \ldots, y_{L-1})$ and a noise realization $\mathbf{z} = (z_0, z_1, \ldots, z_{L-1})$, the first L_p elements of $\mathbf{y}$, corresponding to the corrupted pilot sequence, and the first L_p elements of $\mathbf{z}$ are denoted by $\mathbf{y}_\mathsf{p}$ and $\mathbf{z}_\mathsf{p}$, respectively. The remaining L_d elements, corresponding to the data part, are denoted by $\mathbf{y}_\mathsf{d}$ and $\mathbf{z}_\mathsf{d}$, respectively. Similarly to $\mathbf{d}_k$ and $\mathbf{s}_k$, $\mathbf{y}_{\mathsf{d},k}$ is defined as

$$\mathbf{y}_{\mathsf{d},k} = (y_{\mathsf{d},0+kM}, \ldots, y_{\mathsf{d},M-1+kM})$$

and $\mathbf{z}_{\mathsf{d},k}$ is defined as

$$\mathbf{z}_{\mathsf{d},k} = (z_{\mathsf{d},0+kM}, \ldots, z_{\mathsf{d},M-1+kM}).$$

5.2.2 Optimum Sequential Frame Synchronization

The sequential frame synchronization problem can be cast as a binary hypothesis testing problem, which can be optimally solved by resorting to the Neymann-Pearson lemma [108], where the optimal test for the two hypotheses $\mathcal{H}_0$ and $\mathcal{H}_1$ is the likelihood ratio test (LRT) and it is given by

$$\Lambda(\mathbf{y}) = \frac{f_{\mathbf{Y}|\mathcal{H}_0}(\mathbf{y}|\mathcal{H}_0)}{f_{\mathbf{Y}|\mathcal{H}_1}(\mathbf{y}|\mathcal{H}_1)} \underset{\mathcal{D}_1}{\overset{\mathcal{D}_0}{\gtrless}} \lambda \tag{5.7}$$

where λ is a threshold, and $\mathcal{D}_0$ and $\mathcal{D}_1$ are the decisions for $\mathcal{H}_0$ and $\mathcal{H}_1$, respectively, and where $f_{\mathbf{Y}|\mathcal{H}_i}(\mathbf{y}|\mathcal{H}_i)$ is the probability density function of $\mathbf{Y}$ given the hypotheses $\mathcal{H}_i$.

In this context, missed detection (also known as a Type II error) occurs when, under hypotheses $\mathcal{H}_0$, the test in (5.7) decides for $\mathcal{H}_1$, which means that the decision rule fails to detect the presence of a frame. The missed detection probability corresponds to

$$P_{\mathrm{MD}} = P\left(\Lambda(\mathbf{Y}) < \lambda | \mathcal{H}_0\right). \tag{5.8}$$

Conversely, a false alarm (or Type I error) happens when, under the hypotheses $\mathcal{H}_1$, the test in (5.7) decides for $\mathcal{H}_0$, which means that the decision rule incorrectly indicates the presence of a frame. The false alarm rate equals

$$P_{\mathrm{FA}} = P\left(\Lambda(\mathbf{Y}) \geq \lambda | \mathcal{H}_1\right). \tag{5.9}$$

5.2.3 Optimum Frame Synchronization Only Preamble

If the case in which only the first L_{p} symbols of the $\mathbf{y}$ sequence are used for the sequential frame synchronization test is considered, the two previously described scenarios can be further simplified. In this section, a coherent channel model is analyzed as an example, and the reader can refer to [106] for the non-coherent case.

Bursty Transmission (Scenario 1)

The two hypotheses reduce to

$$\begin{aligned} \mathcal{H}_0 &: \mathbf{Y}_{\mathsf{p}} = \mathbf{x}_{\mathsf{p}} + \mathbf{Z}_{\mathsf{p}} \\ \mathcal{H}_1 &: \mathbf{Y}_{\mathsf{p}} = \mathbf{Z}_{\mathsf{p}} \end{aligned} \tag{5.10}$$

where $Z_{\mathsf{p},i} \sim \mathcal{N}(0,\sigma^2)$ are i.i.d. r.v.s which model real-valued Gaussian noise. The LRT is known [103] and the metric of the test corresponds to

$$\Lambda_{\mathsf{p}}^{(1)}(\mathbf{y}_{\mathsf{p}}) = \exp\left\{-\frac{L_{\mathsf{p}} - 2\langle \mathbf{y}_{\mathsf{p}}, \mathbf{x}_{\mathsf{p}}\rangle}{2\sigma^2}\right\}. \tag{5.11}$$

By applying the logarithm and including additive or multiplicative constants into the threshold test, an alternative metric can be written according to

$$\widetilde{\Lambda}_{\mathsf{p}}^{(1)}(\mathbf{y}) = \langle \mathbf{y}_{\mathsf{p}}, \mathbf{x}_{\mathsf{p}}\rangle. \tag{5.12}$$

Continuous Transmission (Scenario 2)

The two hypotheses are

$$\begin{aligned} \mathcal{H}_0 &: \mathbf{Y}_{\mathsf{p}} = \mathbf{x}_{\mathsf{p}} + \mathbf{Z}_{\mathsf{p}} \\ \mathcal{H}_1 &: \mathbf{Y}_{\mathsf{p}} = \mathbf{C}_{\mathsf{p}} + \mathbf{Z}_{\mathsf{p}} \end{aligned} \tag{5.13}$$

where again $Z_{\mathsf{p},i} \sim \mathcal{N}(0,\sigma^2)$ is the real-valued Gaussian noise. Also for this scenario, the LRT is known [103] and its metric is given by

$$\Lambda_{\mathsf{p}}^{(2)}(\mathbf{y}_{\mathsf{p}}) = \frac{2^{L_{\mathsf{p}}}}{\prod_{i=0}^{L_{\mathsf{p}}-1}\left(1 + e^{-2y_{\mathsf{p},i}x_{\mathsf{p},i}/\sigma^2}\right)}. \tag{5.14}$$

Again, by applying the logarithm and including additive or multiplicative constants into the threshold test, an alternative formulation can be based on the metric

$$\widetilde{\Lambda}_{\mathsf{p}}^{(2)}(\mathbf{y}_{\mathsf{p}}) = -\sum_{i=0}^{L_{\mathsf{p}}-1} \log\left(1 + e^{-2y_{\mathsf{p},i}x_{\mathsf{p},i}/\sigma^2}\right). \tag{5.15}$$

5.3 Coherent Frame Synchronization Tests

In this section, the LRTs and some simplified suboptimal tests for the coherent channel model are derived, using not only the preamble part of the frame, but also the data, under the two scenarios of interest.

5.3.1 Likelihood Ratio Test - Bursty Transmission (Scenario 1)

In the problem, under the hypothesis $\mathcal{H}_0$, the received vector $\mathbf{y}$ depends on $\mathbf{b}$, $\mathbf{x}_{\mathsf{p}}$ and $\mathbf{s}$. However, the exact value of $\mathbf{b}$ is not known at the decoder, but it is known statistically, since the transmitted

bits can be 0 or 1 equiprobably. This means that

$$f_{\mathbf{Y}|\mathcal{H}_0}(\mathbf{y}|\mathcal{H}_0) = f_{\mathbf{Y}_\mathsf{p}|\mathcal{H}_0}(\mathbf{y}_\mathsf{p}|\mathcal{H}_0) f_{\mathbf{Y}_\mathsf{d}|\mathcal{H}_0}(\mathbf{y}_\mathsf{d}|\mathcal{H}_0),$$

where the first term is equal to

$$\begin{aligned} f_{\mathbf{Y}_\mathsf{p}|\mathcal{H}_0}(\mathbf{y}_\mathsf{p}|\mathcal{H}_0) &= \prod_{i=0}^{L_\mathsf{p}-1} f_{Y_{\mathsf{p},i}|\mathcal{H}_0}(y_{\mathsf{p},i}|\mathcal{H}_0) \\ &= \prod_{i=0}^{L_\mathsf{p}-1} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{(y_{\mathsf{p},i}-x_{\mathsf{p},i})^2}{2\sigma^2}\right\} \\ &= K_\mathsf{p}(\mathbf{y}_\mathsf{p}) \exp\left\{\frac{\langle \mathbf{y}_\mathsf{p}, \mathbf{x}_\mathsf{p}\rangle}{\sigma^2}\right\} \end{aligned}$$

where it is used the fact that $x_{\mathsf{p},i}^2 = 1$, and where

$$K_\mathsf{p}(\mathbf{y}_\mathsf{p}) = \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^{L_\mathsf{p}} \exp\left\{-\sum_{i=0}^{L_\mathsf{p}-1} \frac{y_{\mathsf{p},i}^2+1}{2\sigma^2}\right\}. \tag{5.16}$$

The second term, $f_{\mathbf{Y}_\mathsf{d}|\mathcal{H}_0}(\mathbf{y}_\mathsf{d}|\mathcal{H}_0)$, is

$$\begin{aligned} f_{\mathbf{Y}_\mathsf{d}|\mathcal{H}_0}(\mathbf{y}_\mathsf{d}|\mathcal{H}_0) &= \prod_{k=0}^{N-1} \frac{1}{2} \sum_{b\in\{\pm 1\}} f_{\mathbf{Y}_{\mathsf{d},k}|\mathcal{H}_0,B_k}(\mathbf{y}_{\mathsf{d},k}|\mathcal{H}_0, B_k = b) \\ &= K_\mathsf{d}(\mathbf{y}_\mathsf{d}) \prod_{k=0}^{N-1} \cosh\left\{\frac{\langle \mathbf{y}_{\mathsf{d},k}, \mathbf{s}_k\rangle}{\sigma^2}\right\} \end{aligned}$$

where

$$K_\mathsf{d}(\mathbf{y}_\mathsf{d}) = \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^{L_\mathsf{d}} \exp\left\{-\sum_{i=0}^{L_\mathsf{d}-1} \frac{y_{\mathsf{d},i}^2+1}{2\sigma^2}\right\}. \tag{5.17}$$

Note that, in the data part of the frame, the value of b_k is not known, but it is known that the transmitted sequence is either $\mathbf{s}_k$ or $-\mathbf{s}_k$. This knowledge is exploited to achieve better performance with respect to the case of observing the preamble alone.

In the first scenario, under the hypothesis $\mathcal{H}_1$, where there is only noise, the likelihood function is given instead by

$$\begin{aligned} f_{\mathbf{Y}|\mathcal{H}_1}(\mathbf{y}|\mathcal{H}_1) &= \prod_{i=0}^{L-1} f_{Y_i|\mathcal{H}_1}(y_i|\mathcal{H}_1) \\ &= \prod_{i=0}^{L-1} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{y_i^2}{2\sigma^2}} = \frac{K_\mathsf{p}(\mathbf{y}_\mathsf{p}) K_\mathsf{d}(\mathbf{y}_\mathsf{d})}{K_0} \end{aligned}$$

with $K_0 = \exp\left\{-\frac{L}{2\sigma^2}\right\}$. Thus, the LRT requires evaluating the metric

$$\begin{aligned}\Lambda^{(1)}(\mathbf{y}) &= \frac{f_{\mathbf{Y}|\mathcal{H}_0}(\mathbf{y}|\mathcal{H}_0)}{f_{\mathbf{Y}|\mathcal{H}_1}(\mathbf{y}|\mathcal{H}_1)} \\ &= K_0 e^{\frac{\langle \mathbf{y}_\mathsf{p}, \mathbf{x}_\mathsf{p}\rangle}{\sigma^2}} \prod_{k=0}^{N-1} \cosh\left\{\frac{\langle \mathbf{y}_{\mathsf{d},k}, \mathbf{s}_k\rangle}{\sigma^2}\right\}. \end{aligned} \tag{5.18}$$

Constant K_0 can be included in the threshold λ of the test, and the logarithm of the remaining metric can be used in (5.18), obtaining the equivalent metric

$$\widetilde{\Lambda}^{(1)}(\mathbf{y}) = \frac{\langle \mathbf{y}_\mathsf{p}, \mathbf{x}_\mathsf{p}\rangle}{\sigma^2} + \sum_{k=0}^{N-1} \log\cosh\left\{\frac{\langle \mathbf{y}_{\mathsf{d},k}, \mathbf{s}_k\rangle}{\sigma^2}\right\} \tag{5.19}$$

where $\langle \mathbf{y}_\mathsf{p}, \mathbf{x}_\mathsf{p}\rangle$ is the correlation term of the preamble. In Figure 5.2, a possible implementation of the coherent receiver architecture used to compute the simplified metric $\widetilde{\Lambda}^{(1)}(\mathbf{y})$ is illustrated.

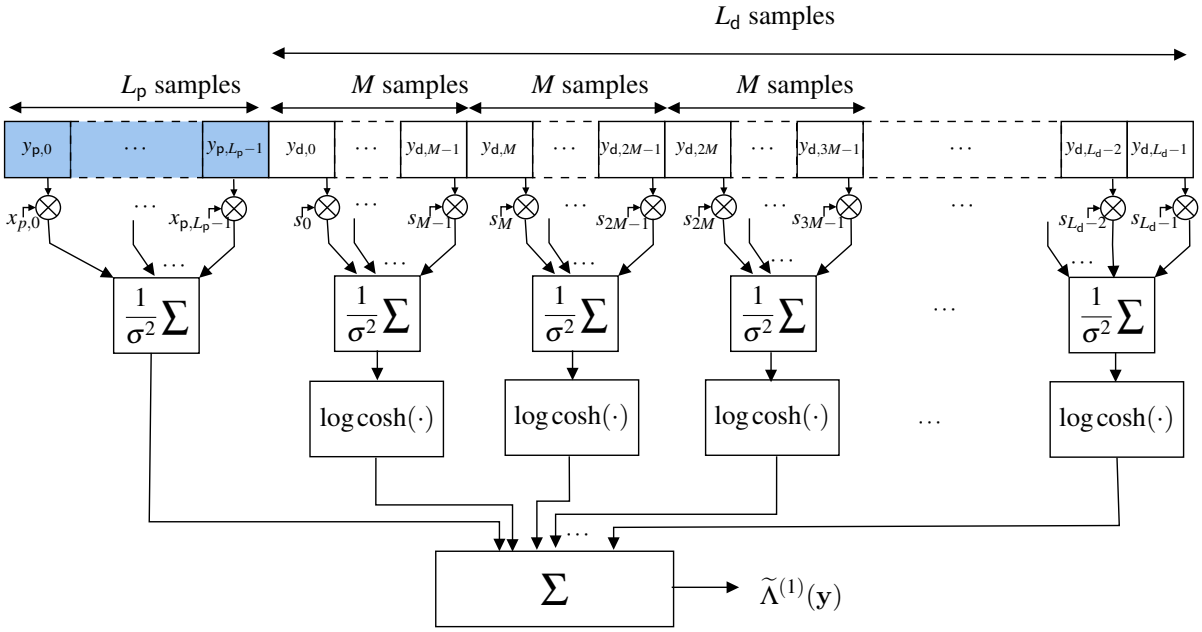


Fig. 5.2 Receiver architecture for the coherent channel model, under bursty transmissions (Scenario 1). Block diagram of the coherent receiver implementing the simplified metric $\widetilde{\Lambda}^{(1)}(\mathbf{y})$.

5.3.2 Likelihood Ratio Test - Continuous Transmission (Scenario 2)

In the second scenario, the likelihood of the hypothesis $\mathcal{H}_1$ can be expressed as

$$\begin{aligned} f_{\mathbf{Y}|\mathcal{H}_1}(\mathbf{y}|\mathcal{H}_1) &= \prod_{i=0}^{L-1} \frac{1}{2} \sum_{c\in\{\pm 1\}} f_{Y_i|\mathcal{H}_1,C_i}(y_i|\mathcal{H}_1, C_i = c) \\ &= K_{\mathsf{p}}(\mathbf{y}_{\mathsf{p}}) K_{\mathsf{d}}(\mathbf{y}_{\mathsf{d}}) \prod_{i=0}^{L-1} \cosh\left\{\frac{y_i}{\sigma^2}\right\} \end{aligned}$$

where $K_{\mathsf{p}}(\mathbf{y}_{\mathsf{p}})$ and $K_{\mathsf{d}}(\mathbf{y}_{\mathsf{d}})$ have been defined in 5.16 and (5.17).

If $\Lambda^{(2)}(\mathbf{y})$ denotes the likelihood ratio between the hypothesis $\mathcal{H}_0$ and $\mathcal{H}_1$, in the second scenario, $\Lambda^{(2)}(\mathbf{y})$ is

$$\Lambda^{(2)}(\mathbf{y}) = \frac{K_0^{-1}\Lambda^{(1)}(\mathbf{y})}{\prod_{i=0}^{L-1} \cosh\left\{\frac{y_i}{\sigma^2}\right\}} \tag{5.20}$$

where $K_0 = \exp\left\{-\frac{L}{2\sigma^2}\right\}$, and the corresponding simplified log-domain version is

$$\widetilde{\Lambda}^{(2)}(\mathbf{y}) = \widetilde{\Lambda}^{(1)}(\mathbf{y}) - \sum_{i=0}^{L-1} \log\cosh\left\{\frac{y_i}{\sigma^2}\right\}. \tag{5.21}$$

5.3.3 High and Low SNR Approximations

In the high and low SNR regimes, the $\log\cosh(a)$ function admits two simple approximations. In the first case, it can be approximated using $\log\cosh(a) \approx |a| - \log 2$ [106, Equation (19)] which allows rewriting (5.19) and (5.21) as

$$\widetilde{\Lambda}_{\mathsf{H}}^{(1)}(\mathbf{y}) = \langle \mathbf{y}_{\mathsf{p}}, \mathbf{x}_{\mathsf{p}} \rangle + \sum_{k=0}^{N-1} \left| \langle \mathbf{y}_{\mathsf{d},k}, \mathbf{s}_k \rangle \right| \tag{5.22}$$

$$\widetilde{\Lambda}_{\mathsf{H}}^{(2)}(\mathbf{y}) = \widetilde{\Lambda}_{\mathsf{H}}^{(1)}(\mathbf{y}) - \sum_{i=0}^{L-1} |y_i| \tag{5.23}$$

respectively. The advantage of $\widetilde{\Lambda}_{\mathsf{H}}^{(1)}(\mathbf{y})$ and $\widetilde{\Lambda}_{\mathsf{H}}^{(2)}(\mathbf{y})$ metrics is that there is no need to estimate the noise variance term. They correspond to the preamble correlation plus the non-coherent accumulations of the N spread bits. Note, however, that spread spectrum systems are typically operated at low SNR regimes. Thus, the use of $\widetilde{\Lambda}_{\mathsf{H}}^{(1)}(\mathbf{y})$ and $\widetilde{\Lambda}_{\mathsf{H}}^{(2)}(\mathbf{y})$ may be questionable in typical applications.

In the low SNR case instead, using the Taylor expansion of $\log\cosh(a)$, at $a=0$, $\log\cosh(a) = a^2/2 + o(a^2)$ is obtained, which gives

$$\widetilde{\Lambda}_{\mathsf{L}}^{(1)}(\mathbf{y}) = \frac{\langle \mathbf{y}_{\mathsf{p}}, \mathbf{x}_{\mathsf{p}} \rangle}{\sigma^2} + \frac{1}{2}\sum_{k=0}^{N-1}\left(\frac{\langle \mathbf{y}_{\mathsf{d},k}, \mathbf{s}_k \rangle}{\sigma^2}\right)^2 \tag{5.24}$$

$$\widetilde{\Lambda}_{\mathsf{L}}^{(2)}(\mathbf{y}) = \widetilde{\Lambda}_{\mathsf{L}}^{(1)}(\mathbf{y}) - \frac{1}{2}\sum_{i=0}^{L-1}\left(\frac{y_i}{\sigma^2}\right)^2. \tag{5.25}$$

5.4 Performance Analysis for Coherent Channel Model

To evaluate the performance of the various tests described in Section 5.3, it is necessary to examine the distributions of the corresponding random variables under the hypotheses $\mathcal{H}_0$ and $\mathcal{H}_1$. The following Lemma is utilized.

Lemma 5.1: Consider a random variable $\Xi = \log\cosh(\Psi)$ where $\Psi \sim \mathcal{N}\left(\mu, \sigma^2\right)$. The p.d.f. of Ξ is

$$f_{\Xi}(\xi) = \begin{cases} \frac{e^{\xi}}{\sqrt{e^{2\xi}-1}} \frac{\exp\left\{-g_1^2(\xi)\right\} + \exp\left\{-g_2^2(\xi)\right\}}{\sqrt{2\pi\sigma^2}} & \text{if } \xi \geq 0 \\ 0 & \text{if } \xi < 0 \end{cases} \tag{5.26}$$

and the c.d.f. of Ξ is

$$F_{\Xi}(\xi) = \begin{cases} \frac{1}{2}\operatorname{erfc}\left(-g_2(\xi)\right) - \frac{1}{2}\operatorname{erfc}\left(g_1(\xi)\right) & \text{if } \xi \geq 0 \\ 0 & \text{if } \xi < 0 \end{cases} \tag{5.27}$$

where

$$g_1(\xi) = \frac{\cosh^{-1}(e^{\xi}) - \mu}{\sqrt{2}\sigma},\ g_2(\xi) = \frac{\cosh^{-1}(e^{\xi}) + \mu}{\sqrt{2}\sigma}.$$

Proof. The proof is provided in Appendix D. □

An example of the distribution is depicted in Figure 5.3.

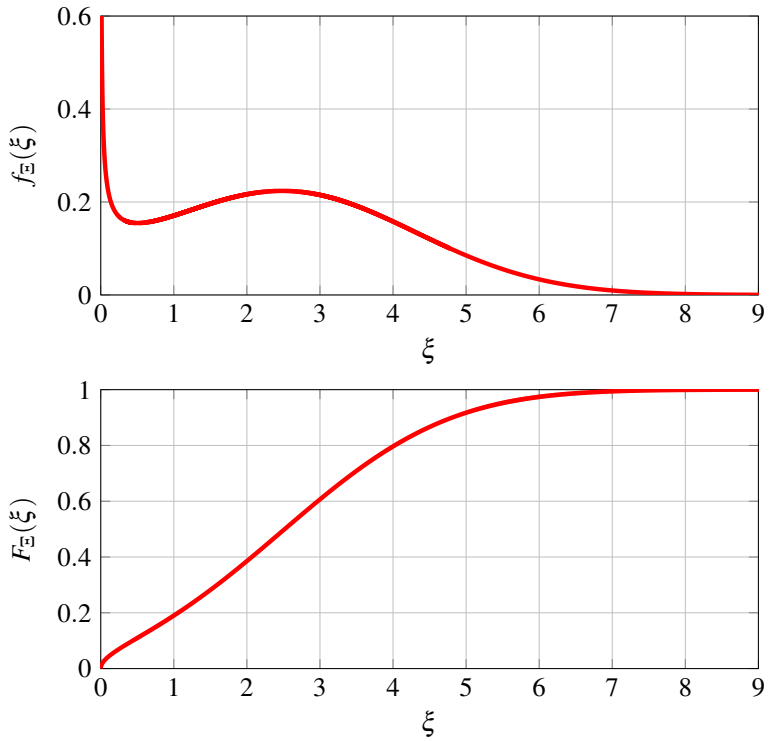


Fig. 5.3 Example of the p.d.f. and c.d.f. of Ξ, when $\mu = \sigma^2 = 3.2076$.

5.4.1 Analysis of Bursty Transmissions (Scenario 1)

Analysis of the Likelihood Ratio Test

The distribution of the log-domain metric (5.19) associated with the LRT has been analyzed first. Without loss of generality, it is assumed that each preamble $x_{\mathsf{p},i} = +1$ and each chip of the spreading sequence $s_i = +1$. It is defined

$$\Psi_{\mathsf{p}} = \frac{1}{\sigma^2} \langle \mathbf{Y}_{\mathsf{p}}, \mathbf{x}_{\mathsf{p}} \rangle, \quad \Psi_{\mathsf{d},k} = \frac{1}{\sigma^2} \langle \mathbf{Y}_{\mathsf{d},k}, \mathbf{s}_k \rangle,$$

and

$$\Xi_{\mathsf{d},k} = \log\cosh \Psi_{\mathsf{d},k}, \quad \Xi_{\mathsf{d}} = \sum_{k=0}^{N-1} \Xi_{\mathsf{d},k},$$

which means that

$$\begin{aligned}\widetilde{\Lambda}^{(1)}(\mathbf{Y}) &= \frac{1}{\sigma^2}\langle \mathbf{Y}_{\mathsf{p}}, \mathbf{x}_{\mathsf{p}}\rangle + \sum_{k=0}^{N-1} \log\cosh\left(\frac{1}{\sigma^2}\langle \mathbf{Y}_{\mathsf{d},k}, \mathbf{s}_k\rangle\right) \\ &= \Psi_{\mathsf{p}} + \sum_{k=0}^{N-1} \log\cosh \Psi_{\mathsf{d},k} \\ &= \Psi_{\mathsf{p}} + \sum_{k=0}^{N-1} \Xi_{\mathsf{d},k} \\ &= \Psi_{\mathsf{p}} + \Xi_{\mathsf{d}}.\end{aligned} \tag{5.28}$$

Under the hypothesis $\mathcal{H}_0$,

$$\Psi_{\mathsf{p}} \sim \mathcal{N}\left(\frac{L_{\mathsf{p}}}{\sigma^2}, \frac{L_{\mathsf{p}}}{\sigma^2}\right), \quad \Psi_{\mathsf{d},k} \sim \mathcal{N}\left(\frac{M}{\sigma^2}, \frac{M}{\sigma^2}\right),$$

whereas, under the hypothesis $\mathcal{H}_1$,

$$\Psi_{\mathsf{p}} \sim \mathcal{N}\left(0, \frac{L_{\mathsf{p}}}{\sigma^2}\right), \quad \Psi_{\mathsf{d},k} \sim \mathcal{N}\left(0, \frac{M}{\sigma^2}\right).$$

When the number of bits N is sufficiently large, by invoking the central limit theorem, the distribution of $\widetilde{\Lambda}^{(1)}(\mathbf{Y})$ can be approximated by $\mathcal{N}\left(\mu_{\Psi_{\mathsf{p}}} + N\mu_{\Xi_{\mathsf{d},k}}, \sigma^2_{\Psi_{\mathsf{p}}} + N\sigma^2_{\Xi_{\mathsf{d},k}}\right)$ where

$$\mu_{\Psi_{\mathsf{p}}} = \mathsf{E}\left[\Psi_{\mathsf{p}}\right], \ \sigma^2_{\Psi_{\mathsf{p}}} = \mathrm{Var}\left[\Psi_{\mathsf{p}}\right],$$

$$\mu_{\Xi_{\mathsf{d},k}} = \mathsf{E}\left[\Xi_{\mathsf{d},k}\right], \ \sigma^2_{\Xi_{\mathsf{d},k}} = \mathrm{Var}\left[\Xi_{\mathsf{d},k}\right]$$

and $\mu_{\Xi_{\mathsf{d},k}}$, $\sigma^2_{\Xi_{\mathsf{d},k}}$ can be computed numerically using the p.d.f. in Lemma 1. Under the normal approximation, the missed detection (MD) probability can be computed according to

$$\begin{aligned}P_{\mathrm{MD}} &= P(\widetilde{\Lambda}^{(1)}(\mathbf{Y}) \le \lambda | \mathcal{H}_0) \\ &\approx \frac{1}{2}\,\mathrm{erfc}\left(-\frac{\lambda - L_{\mathsf{p}}/\sigma^2 - N\mu_{\Xi_{\mathsf{d},k}}}{\sqrt{2\left(L_{\mathsf{p}}/\sigma^2 + N\sigma^2_{\Xi_{\mathsf{d},k}}\right)}}\right).\end{aligned} \tag{5.29}$$

whereas the false alarm (FA) probability is

$$P_{\mathrm{FA}} = P(\widetilde{\Lambda}^{(1)}(\mathbf{Y}) \geq \lambda | \mathcal{H}_1) \approx \frac{1}{2}\operatorname{erfc}\left(\frac{\lambda - N\mu_{\Xi_{\mathsf{d},k}}}{\sqrt{2\left(L_{\mathsf{p}}/\sigma^2 + N\sigma^2_{\Xi_{\mathsf{d},k}}\right)}}\right). \tag{5.30}$$

In the top part of Figure 5.4, the p.d.f. of $\widetilde{\Lambda}^{(1)}(\mathbf{Y})$ under the two hypotheses $\mathcal{H}_0$ and $\mathcal{H}_1$ is depicted and compared with respect to the p.d.f. of the corresponding normal approximation. In this case, $N = 32$ bits, the spreading factor is set to $M = 15$, thus $L_{\mathsf{d}} = 480$ chips, the preamble has $L_{\mathsf{p}} = 32$ symbols, and $E_c/N_0 = -15.5$ dB. Already with a small number of bits N, the normal approximation approaches the exact distribution well. The middle plot in Figure 5.4 shows the ROC curve in terms of detection probability $P_{\mathrm{D}} = 1 - P_{\mathrm{MD}}$ versus P_{FA} of (5.19) w.r.t. the use of the preamble correlation of (5.12). Monte Carlo simulation results for the LRT (represented by the red dashed curve) are compared with its corresponding normal approximation (dotted cyan line) and saddlepoint approximation (green circles). The plot illustrates the exceptional accuracy of the saddlepoint approximation in relation to the LRT performance. The normal approximation yields a small prediction error, that is more visible at small probability of detection. To better clarify the gains provided by the LRT based on the metric (5.18) w.r.t. the use of the preamble correlation (5.12), the P_{MD} versus E_c/N_0 for different N bits values is depicted at the bottom of Figure 5.4, while fixing $P_{\mathrm{FA}} = 10^{-2}$. Already for $N = 10$ bits, gains around 5 dB can be observed for $P_{\mathrm{FA}} = 10^{-2}$. In the same chart, the performance of both the normal and saddlepoint approximations of the LRT is shown, with both approximations tightly matching their respective LRT performances.

Analysis of the High SNR Approximation

The approximations carried out in this section are based on the assumption that the number of bits N is large, thus resorting to the central limit theorem. Defining

$$\Psi_{\mathsf{p}} = \langle \mathbf{Y}_{\mathsf{p}}, \mathbf{x}_{\mathsf{p}} \rangle, \quad \Psi_{\mathsf{d},k} = \left\langle \mathbf{Y}_{\mathsf{d},k}, \mathbf{s}_k \right\rangle,$$

and

$$\mathcal{M}_{\mathsf{d},k} = \left|\Psi_{\mathsf{d},k}\right|, \quad \mathcal{M}_{\mathsf{d}} = \sum_{k=0}^{N-1} \mathcal{M}_{\mathsf{d},k},$$

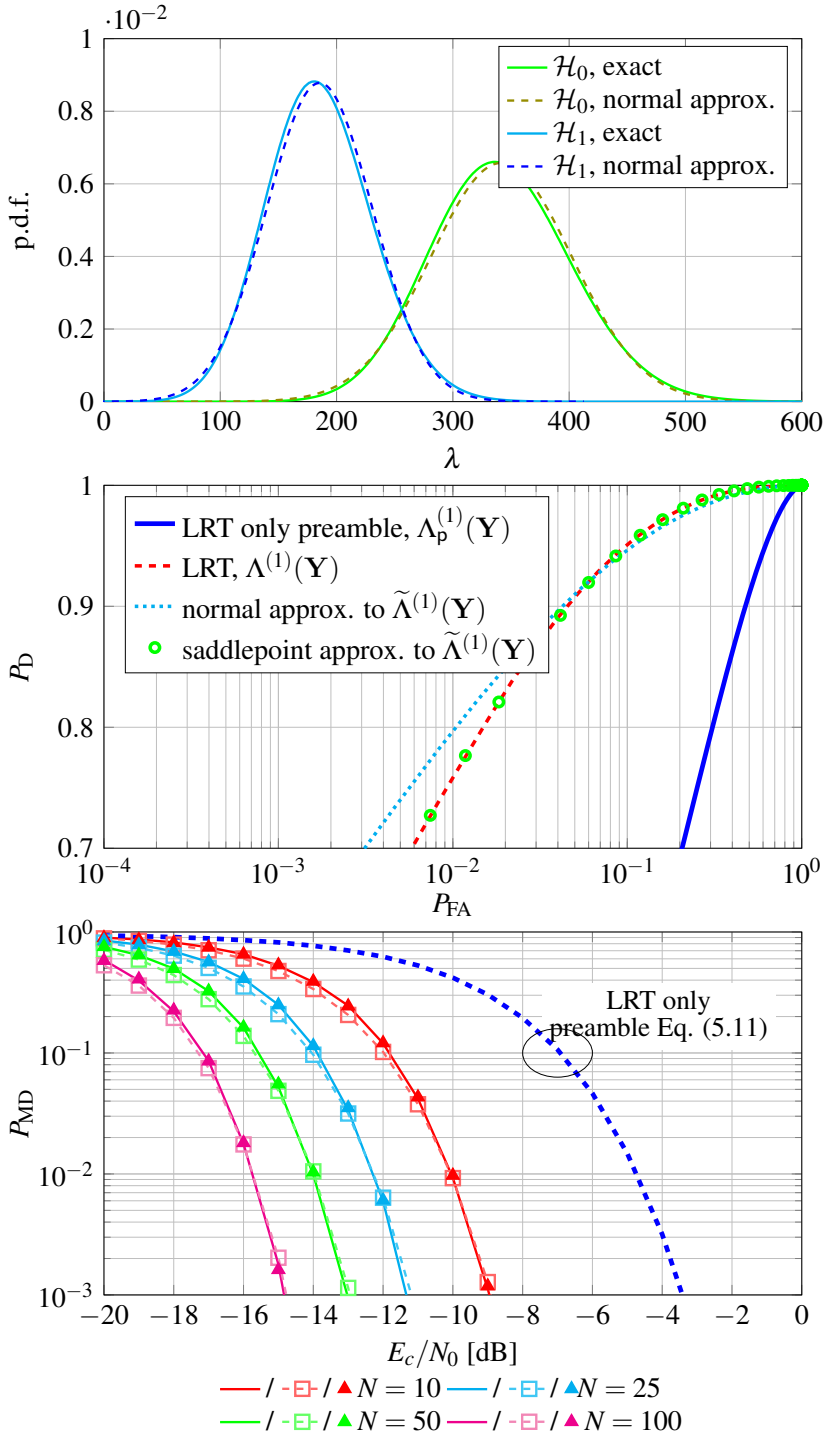


Fig. 5.4 Receiver for the coherent channel model that uses the log cosh function, under bursty transmissions scenario (Scenario 1). **[top]** Distributions of $\widetilde{\Lambda}^{(1)}(\mathbf{Y})$ under hypothesis $\mathcal{H}_0$ and $\mathcal{H}_1$, and their respective normal approximations. **[middle]** ROC curve for the threshold test applied to $\Lambda^{(1)}(\mathbf{Y})$ vs. the one relying on correlation with the preamble, $\Lambda_{\mathrm{p}}^{(1)}(\mathbf{Y})$. **[bottom]** MD probability versus E_c/N_0 for the LRT $\Lambda^{(1)}(\mathbf{y})$ (solid curves) for different values of bits N, with fixed $P_{\mathrm{FA}} = 10^{-2}$, vs. the respective normal (dashed lines with square markers) and saddlepoint approximations (filled triangles) of the threshold test based on the metric $\widetilde{\Lambda}^{(1)}(\mathbf{Y})$.

where $\mathcal{M}_{\mathsf{d},k}$ are i.i.d. folded normal random variables [109]. It follows that, under the bursty transmission scenario, the suboptimal metric $\widetilde{\Lambda}_{\mathsf{H}}^{(1)}(\mathbf{Y})$ in (5.22) can be written according to

$$\begin{aligned}\widetilde{\Lambda}_{\mathsf{H}}^{(1)}(\mathbf{Y}) &= \langle \mathbf{Y}_{\mathsf{p}}, \mathbf{x}_{\mathsf{p}} \rangle + \sum_{k=0}^{N-1} \left| \langle \mathbf{Y}_{\mathsf{d},k}, \mathbf{s}_k \rangle \right| \\ &= \Psi_{\mathsf{p}} + \sum_{k=0}^{N-1} \left| \Psi_{\mathsf{d},k} \right| \\ &= \Psi_{\mathsf{p}} + \sum_{k=0}^{N-1} \mathcal{M}_{\mathsf{d},k} \\ &= \Psi_{\mathsf{p}} + \mathcal{M}_{\mathsf{d}}.\end{aligned} \tag{5.31}$$

Under the hypothesis $\mathcal{H}_0$,

$$\Psi_{\mathsf{p}} \sim \mathcal{N}\left(L_{\mathsf{p}}, L_{\mathsf{p}}\sigma^2\right), \quad \Psi_{\mathsf{d},k} \sim \mathcal{N}\left(M, M\sigma^2\right),$$

whereas, under the hypothesis $\mathcal{H}_1$,

$$\Psi_{\mathsf{p}} \sim \mathcal{N}\left(0, L_{\mathsf{p}}\sigma^2\right), \quad \Psi_{\mathsf{d},k} \sim \mathcal{N}\left(0, M\sigma^2\right).$$

The normal approximation of $\widetilde{\Lambda}_{\mathsf{H}}^{(1)}(\mathbf{Y})$ corresponds to $\mathcal{N}\left(\mu_{\Psi_{\mathsf{p}}} + N\mu_{\mathcal{M}_{\mathsf{d},k}}, \sigma^2_{\Psi_{\mathsf{p}}} + N\sigma^2_{\mathcal{M}_{\mathsf{d},k}}\right)$, where

$$\mu_{\Psi_{\mathsf{p}}} = \mathsf{E}\left[\Psi_{\mathsf{p}}\right], \ \sigma^2_{\Psi_{\mathsf{p}}} = \mathrm{Var}\left[\Psi_{\mathsf{p}}\right],$$

$$\mu_{\mathcal{M}_{\mathsf{d},k}} = \mathsf{E}\left[\mathcal{M}_{\mathsf{d},k}\right], \quad \sigma^2_{\mathcal{M}_{\mathsf{d},k}} = \mathrm{Var}\left[\mathcal{M}_{\mathsf{d},k}\right],$$

and $\mu_{\mathcal{M}_{\mathsf{d},k}}$, $\sigma^2_{\mathcal{M}_{\mathsf{d},k}}$ can be computed according to the expressions provided in [109]. It follows that the MD probability is

$$\begin{aligned}P_{\mathrm{MD}} &= P(\widetilde{\Lambda}_{\mathsf{H}}^{(1)}(\mathbf{Y}) \leq \lambda \,|\, \mathcal{H}_0) \\ &\approx \frac{1}{2}\,\mathrm{erfc}\left(-\frac{\lambda - L_{\mathsf{p}} - N\mu_{\mathcal{M}_{\mathsf{d},k}}}{\sqrt{2\left(L_{\mathsf{p}}\sigma^2 + N\sigma^2_{\mathcal{M}_{\mathsf{d},k}}\right)}}\right),\end{aligned} \tag{5.32}$$

whereas the FA can be computed as

$$\begin{aligned}P_{\mathrm{FA}} &= P(\widetilde{\Lambda}_{\mathsf{H}}^{(1)}(\mathbf{Y}) \geq \lambda \,|\, \mathcal{H}_1) \\ &\approx \frac{1}{2}\,\mathrm{erfc}\left(\frac{\lambda - N\mu_{\mathcal{M}_{\mathsf{d},k}}}{\sqrt{2\left(L_{\mathsf{p}}\sigma^2 + N\sigma^2_{\mathcal{M}_{\mathsf{d},k}}\right)}}\right).\end{aligned} \tag{5.33}$$

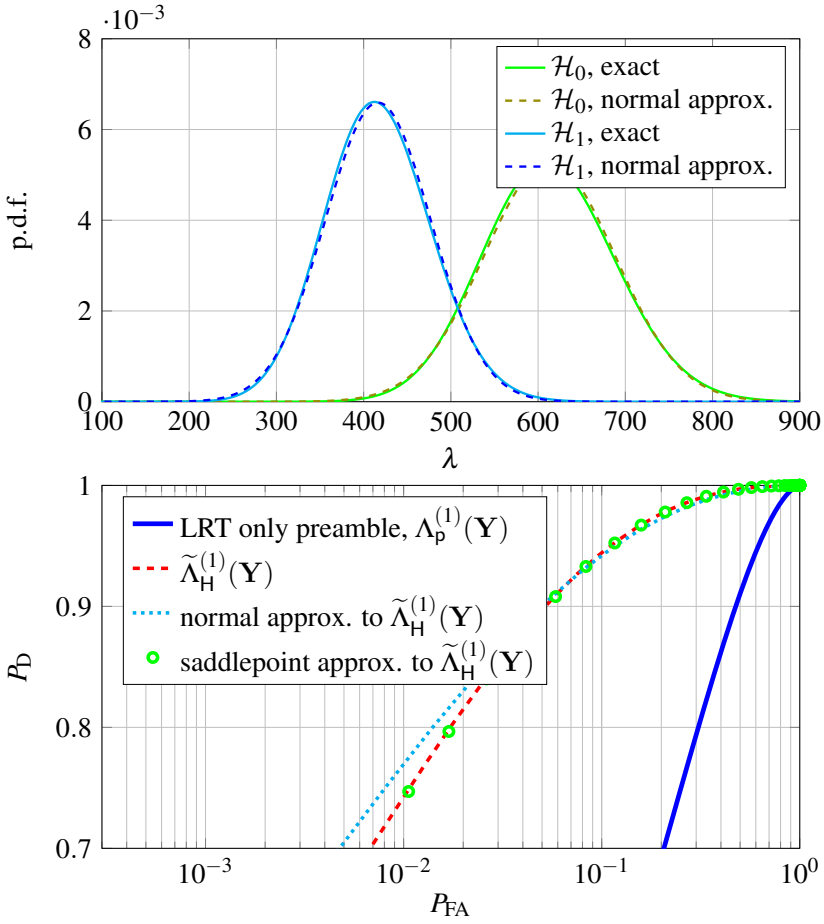


Fig. 5.5 Receiver for the coherent channel model that uses the high SNR approximation of the log cosh function, under bursty transmissions scenario (Scenario 1). **[top]** Distributions of $\widetilde{\Lambda}_{\mathsf{H}}^{(1)}(\mathbf{Y})$ under hypothesis $\mathcal{H}_0$ and $\mathcal{H}_1$, and their respective normal approximations. **[bottom]** ROC curve for the threshold test applied to $\widetilde{\Lambda}_{\mathsf{H}}^{(1)}(\mathbf{Y})$ vs. the one relying on correlation with the preamble, $\Lambda_{\mathsf{p}}^{(1)}(\mathbf{Y})$.

In the upper part of Figure 5.5, the p.d.f. of $\widetilde{\Lambda}_{\mathsf{H}}^{(1)}(\mathbf{Y})$ is shown under both $\mathcal{H}_0$ and $\mathcal{H}_1$, along with a comparison to the corresponding normal approximation. For this analysis $N = 32$ bits, a spreading factor $M = 15$, a preamble of $L_{\mathsf{p}} = 32$ symbols, and $E_c/N_0 = -15.5$ dB were used. Similarly to Figure 5.4, notably, even with a modest number of bits N, the normal approximation aligns closely with the true distribution. The lower part of Figure 5.5 presents the ROC curve, plotting the detection probability against the false alarm rate for the test described in (5.22). In this plot, the red dashed line represents Monte Carlo simulation results for $\widetilde{\Lambda}_{\mathsf{H}}^{(1)}(\mathbf{Y})$, which are compared with both the normal approximation (dotted cyan line) and the saddlepoint approximation (green circles). Likewise, in Figure 5.4, also in Figure 5.5 the remarkable precision of the saddlepoint approximation in capturing the test performance is illustrated, while the normal approximation, though slightly off—especially at lower detection probabilities—still provides a reasonably accurate prediction.

Analysis of the Low SNR Approximation

Let us define

$$\Psi_{\mathsf{p}} = \frac{1}{\sigma^2}\left\langle \mathbf{Y}_{\mathsf{p}}, \mathbf{x}_{\mathsf{p}} \right\rangle, \quad \Psi_{\mathsf{d},k} = \frac{1}{\sigma^2}\left\langle \mathbf{Y}_{\mathsf{d},k}, \mathbf{s}_k \right\rangle,$$

and

$$\Omega_{\mathsf{d}} = \frac{1}{2}\sum_{k=0}^{N-1} (\Psi_{\mathsf{d},k})^2,$$

which means that

$$\begin{aligned} \widetilde{\Lambda}^{(1)}(\mathbf{Y}) &= \frac{1}{\sigma^2}\left\langle \mathbf{Y}_{\mathsf{p}}, \mathbf{x}_{\mathsf{p}} \right\rangle + \frac{1}{2}\sum_{k=0}^{N-1}\left(\frac{1}{\sigma^2}\left\langle \mathbf{Y}_{\mathsf{d},k}, \mathbf{s}_k \right\rangle\right)^2 \\ &= \Psi_{\mathsf{p}} + \frac{1}{2}\sum_{k=0}^{N-1}\left(\Psi_{\mathsf{d},k}\right)^2 = \Psi_{\mathsf{p}} + \Omega_{\mathsf{d}}. \end{aligned} \tag{5.34}$$

Under the hypothesis $\mathcal{H}_0$, Ψ_{p} and Ψ_{p} correspond to

$$\Psi_{\mathsf{p}} \sim \mathcal{N}\left(\frac{L_{\mathsf{p}}}{\sigma^2}, \frac{L_{\mathsf{p}}}{\sigma^2}\right), \quad \Psi_{\mathsf{d},k} \sim \mathcal{N}\left(\frac{M}{\sigma^2}, \frac{M}{\sigma^2}\right).$$

Using the property

$$A = cB \Rightarrow f_A(a) = \frac{1}{c} f_B\left(\frac{b}{c}\right),$$

the p.d.f. of Ω_{d} can be expressed according to

$$f_{\Omega_{\mathsf{d}}}(\omega) = \frac{2\sigma^2}{M} f_{\chi^2}\left(\frac{2\sigma^2}{M}\omega\right)$$

where f_{χ^2} is the p.d.f. of a non-central χ^2-distribution with N degrees of freedom and non-centrality parameter $\gamma = NM/\sigma^2$ [110]. This means that the c.d.f. of the metric $\widetilde{\Lambda}^{(1)}_{\mathsf{L}}(\mathbf{Y})$ based on the low SNR approximation corresponds to

$$\begin{aligned} F_{\widetilde{\Lambda}^{(1)}_{\mathsf{L}}(\mathbf{Y})}(\lambda) &= \int_0^{\infty} f_{\Omega_{\mathsf{d}}}(\omega) \int_{-\infty}^{\lambda-\omega} f_{\Psi_{\mathsf{p}}}(\psi)\mathrm{d}\psi\mathrm{d}\omega \\ &= \frac{1}{2}\int_0^{\infty} f_{\Omega_{\mathsf{d}}}(\omega)\,\mathrm{erfc}\left(-\frac{\lambda-\omega-L_{\mathsf{p}}/\sigma^2}{\sqrt{2L_{\mathsf{p}}/\sigma^2}}\right)\mathrm{d}\omega. \end{aligned} \tag{5.35}$$

Under the hypothesis $\mathcal{H}_1$,

$$\Psi_{\mathsf{p}} \sim \mathcal{N}\left(0, \frac{L_{\mathsf{p}}}{\sigma^2}\right), \quad \Psi_{\mathsf{d},k} \sim \mathcal{N}\left(0, \frac{M}{\sigma^2}\right)$$

which implies that

$$f_{\Omega_\mathsf{d}}(\omega) = \frac{2\sigma^2}{M} f_{\chi^2}\left(\frac{2\sigma^2}{M}\omega\right)$$

where f_{χ^2} is the p.d.f. of a central χ^2-distribution with N degrees of freedom [111]. This allows for writing the c.d.f. of $\widetilde{\Lambda}_\mathsf{L}^{(1)}(\mathbf{Y})$ as

$$\begin{aligned} F_{\widetilde{\Lambda}_\mathsf{L}^{(1)}(\mathbf{Y})}(\lambda) &= \int_0^\infty f_{\Omega_\mathsf{d}}(\omega) \int_{-\infty}^{\lambda-\omega} f_{\Psi_\mathsf{p}}(\psi)\mathrm{d}\psi\mathrm{d}\omega \\ &= \frac{1}{2}\int_0^\infty f_{\Omega_\mathsf{d}}(\omega)\,\mathrm{erfc}\left(-\frac{\lambda-\omega}{\sqrt{2L_\mathsf{p}/\sigma^2}}\right)\mathrm{d}\omega. \end{aligned} \tag{5.36}$$

Using (5.35), the MD probability can be expressed as

$$P_\mathrm{MD} = P\left(\widetilde{\Lambda}_\mathsf{L}^{(1)}(\mathbf{Y}) \le \lambda | \mathcal{H}_0\right) = F_{\widetilde{\Lambda}_\mathsf{L}^{(1)}(\mathbf{Y})}(\lambda) \tag{5.37}$$

whereas the FA definition follows from (5.36) and corresponds to

$$P_\mathrm{FA} = P(\widetilde{\Lambda}_\mathsf{L}^{(1)}(\mathbf{Y}) \ge \lambda | \mathcal{H}_1) = 1 - F_{\widetilde{\Lambda}_\mathsf{L}^{(1)}(\mathbf{Y})}(\lambda). \tag{5.38}$$

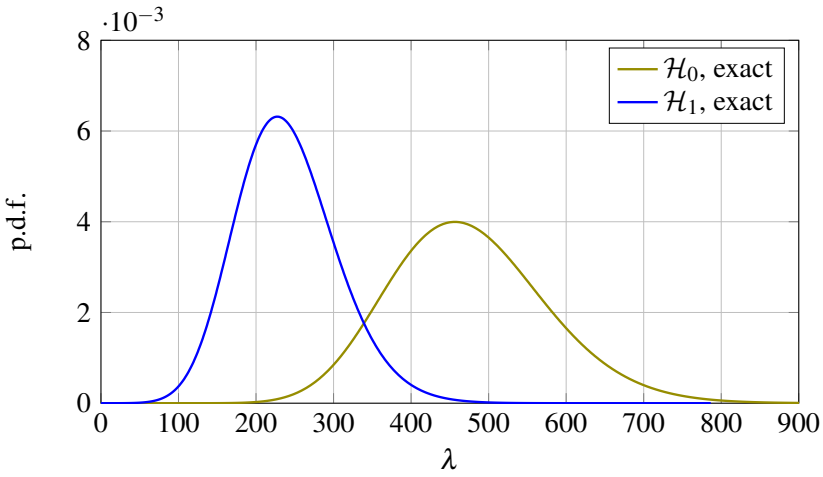


Fig. 5.6 Pdf of $\Lambda_\mathsf{L}^{(1)}(\mathbf{Y})$ under hypothesis $\mathcal{H}_0$ and $\mathcal{H}_1$, for a receiver for the coherent channel model that uses the low SNR approximation of the log cosh function, under bursty transmissions scenario (Scenario 1).

In order to compare the performance of the different tests, a FA rate 10^{-2} has been fixed and the MD performance versus E_c/N_0 has been reported in Figure 5.7 when the spreading factor $M = 15$, the preamble has a length of $L_\mathsf{p} = 32$ symbols and $N = 10, 25, 50, 100, 250, 500, 1000$ bits. The solid curves with the circle marker show the performance of $\Lambda^{(1)}(\mathbf{Y})$, the dashed lines with the star represent the performance of the high SNR approximation, $\widetilde{\Lambda}_\mathsf{H}^{(1)}(\mathbf{Y})$, whereas the

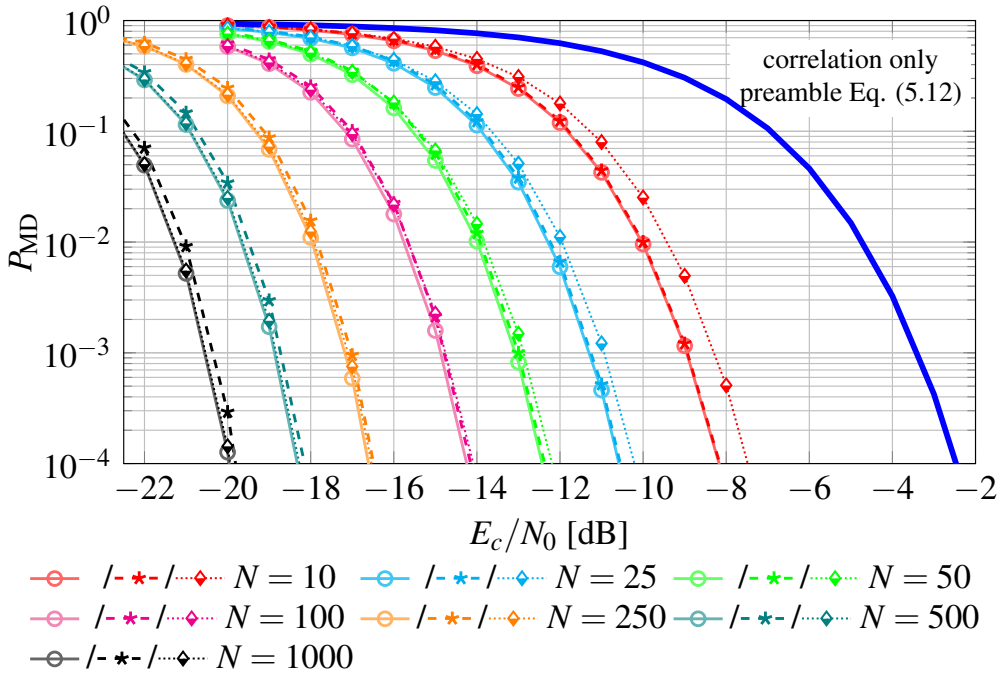


Fig. 5.7 Receiver for the coherent channel model under bursty transmissions (Scenario 1). MD probability versus E_c/N_0 for the LRT $\Lambda^{(1)}(\mathbf{y})$ (solid curves) for different lengths of transmitted bits N, with fixed $P_{\text{FA}} = 10^{-2}$, vs. the respective high SNR (dashed lines with star markers) and low SNR approximations (dotted lines with half filled diamonds) of the threshold test based on the metric $\widetilde{\Lambda}^{(1)}(\mathbf{Y})$.

dotted lines with the half filled diamond represent the performance of the low SNR approximation, $\widetilde{\Lambda}_{\text{L}}^{(1)}(\mathbf{Y})$. From the results, it can be seen that the high SNR approximation is tight both at high and low SNR values and does not require the estimation of the noise variance. The low SNR curve does not approximate well the $\log\cosh()$ function at high SNR, incurring a loss which is clearly visible above $E_c/N_0 = -12$ dB. Nevertheless, the exact and efficient computation of the MD probability and FA rate of $\widetilde{\Lambda}_{\text{L}}^{(1)}(\mathbf{Y})$ can be used for a preliminary design of the system parameters for a coherent spread spectrum communication system, being any suboptimal test an upper bound to the LRT performance.

5.4.2 Analysis of Continuous Transmissions (Scenario 2)

Closed formulas or approximations for the Scenario 2 result difficult to be computed analitycally. Nevertheless, one can measure via Monte Carlo simulations the p.d.f. and c.d.f. of independent r.v.s involved and resort to Fourier transforms or saddlepoint approximation to estimate the p.d.f. and c.d.f. of the test metrics, e.g. considering for the two hypotheses the random variables

$$\Psi_{\text{p},i} = Y_{\text{p},i} x_{\text{p},i} - |Y_{\text{p},i}|\,, \text{ and } \Psi_{\text{d},k} = \left|\langle \mathbf{y}_{\text{d},k}, \mathbf{s}_k \rangle\right| - \left|\mathbf{y}_{\text{d},k}\right|_1$$

where $|\cdot|_1$ denotes the L_1-norm.

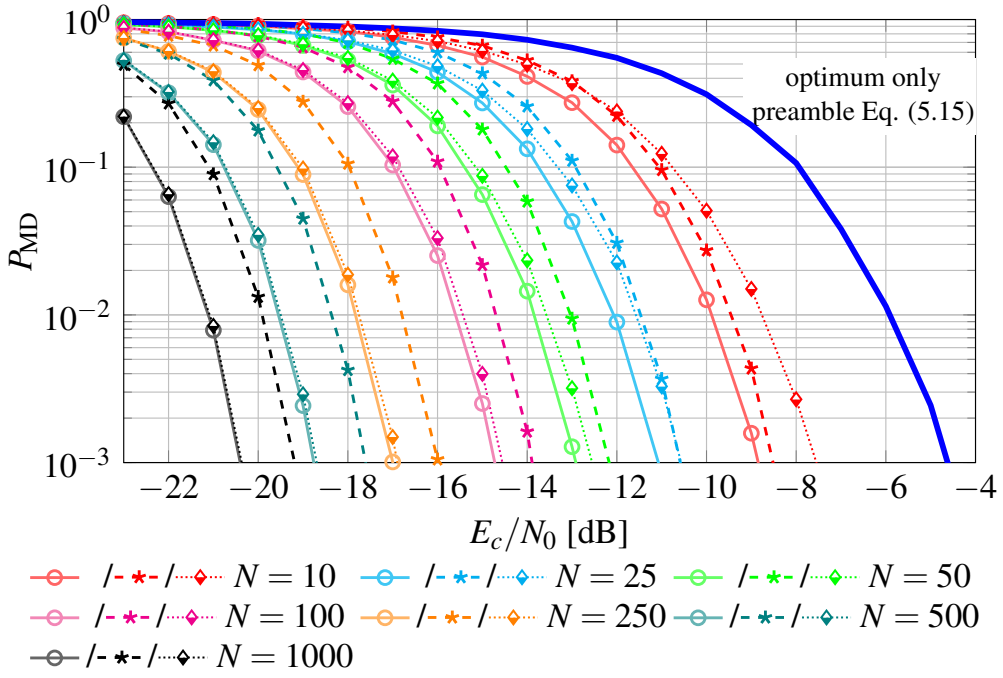


Fig. 5.8 Receiver for the coherent channel model under continuous transmissions (Scenario 2).MD probability versus E_c/N_0 for the LRT $\Lambda^{(2)}(\mathbf{y})$ (solid curves) for different values of N, with fixed $P_{\mathrm{FA}} = 10^{-2}$, vs. the respective high SNR (dashed lines with star markers) and low SNR approximations (dotted lines with half filled diamonds) of the threshold test based on the metric $\widetilde{\Lambda}^{(2)}(\mathbf{Y})$.

Using the same parameters of Figure 5.7, the probability in Scenario 2 of the different tests is compared in Figure 5.8. Differently from the previous case, the use of the high SNR approximation leads to losses at low SNR values. For instance, for values of E_c/N_0 smaller than -16 dB, the measured loss is approximately 1 dB and increases as SNR decreases.

5.5 Non-Coherent Frame Synchronization Tests

Similarly to what has been done in Section 5.3, in this section the LRTs and some simplified tests for the non-coherent channel model are derived. To simplify notation, symbols defined earlier are again used—e.g., Λ for the LRT and $\widetilde{\Lambda}$ for its simplified versions—but it is important to note that this notation does not apply to the equations in Section 5.3.

5.5.1 Likelihood Ratio Test - Bursty Transmission (Scenario 1)

In this case, under the hypothesis $\mathcal{H}_0$, for the non-coherent channel model, neither the exact data bits $\mathbf{b}$ nor the correct phase rotation are known; however, information on their statistical

distributions is available. This means that

$$f_{\mathbf{Y}|\mathcal{H}_0}(\mathbf{y}|\mathcal{H}_0) = \int_{-\pi}^{\pi} p_{\Phi}(\phi) f_{\mathbf{Y}|\mathcal{H}_0,\Phi}(\mathbf{y}|\mathcal{H}_0,\phi)\mathrm{d}\phi \tag{5.39}$$

where $\Phi \sim \mathcal{U}([-\pi,\pi])$ represents the phase offset and

$$f_{\mathbf{Y}|\mathcal{H}_0,\Phi}(\mathbf{y}|\mathcal{H}_0,\phi) = f_{\mathbf{Y}_\mathsf{p}|\mathcal{H}_0,\Phi}(\mathbf{y}_\mathsf{p}|\mathcal{H}_0,\phi) f_{\mathbf{Y}_\mathsf{d}|\mathcal{H}_0,\Phi}(\mathbf{y}_\mathsf{d}|\mathcal{H}_0,\phi)$$

The first term corresponds to

$$\begin{aligned} f_{\mathbf{Y}_\mathsf{p}|\mathcal{H}_0,\phi}(\mathbf{y}_\mathsf{p}|\mathcal{H}_0,\phi) &= \prod_{i=0}^{L_\mathsf{p}-1} f_{Y_{\mathsf{p},i}|\mathcal{H}_0,\phi}(y_{\mathsf{p},i}|\mathcal{H}_0,\phi) \\ &= \prod_{i=0}^{L_\mathsf{p}-1} \frac{1}{2\pi\sigma^2} \exp\left\{-\frac{\left|y_{\mathsf{p},i} - x_{\mathsf{p},i}e^{j\phi}\right|^2}{2\sigma^2}\right\} \\ &= K_\mathsf{p}^{'}(\mathbf{y}_\mathsf{p}) \exp\left\{\frac{\left\langle \mathbf{x}_\mathsf{p}, \Re\left\{\mathbf{y}_{\mathsf{p},i}e^{-j\phi}\right\}\right\rangle}{\sigma^2}\right\} \end{aligned} \tag{5.40}$$

where $\Re\{\cdot\}$ denotes the real part of a complex number (or vector) and

$$K_\mathsf{p}^{'}(\mathbf{y}_\mathsf{p}) = \left(\frac{1}{2\pi\sigma^2}\right)^{L_\mathsf{p}} \exp\left\{-\sum_{i=0}^{L_\mathsf{p}-1} \frac{|y_{\mathsf{p},i}|^2 + 1}{2\sigma^2}\right\}.$$

The term affected by the data is

$$\begin{aligned} f_{\mathbf{Y}_\mathsf{d}|\mathcal{H}_0,\phi}(\mathbf{y}_\mathsf{d}|\mathcal{H}_0,\phi) &= \prod_{k=0}^{N-1} \frac{1}{2} \sum_{b\in\{\pm 1\}} f_{\mathbf{Y}_{\mathsf{d},k}|\mathcal{H}_0,\phi,B_k}(\mathbf{y}_{\mathsf{d},k}|\mathcal{H}_0,\phi,B_k = b) \\ &= K_\mathsf{d}^{'}(\mathbf{y}_\mathsf{d}) \prod_{k=0}^{N-1} \cosh \frac{\left\langle \mathbf{s}_k, \Re\left\{\mathbf{y}_{\mathsf{d},k}e^{-j\phi}\right\}\right\rangle}{\sigma^2} \end{aligned} \tag{5.41}$$

$$K_\mathsf{d}^{'}(\mathbf{y}_\mathsf{d}) = \left(\frac{1}{2\pi\sigma^2}\right)^{L_\mathsf{d}} \exp\left\{-\sum_{i=0}^{L_\mathsf{d}-1} \frac{|y_{\mathsf{d},i}|^2 + 1}{2\sigma^2}\right\},$$

and $K^{'}(\mathbf{y}) = K_\mathsf{p}^{'}(\mathbf{y}_\mathsf{p}) K_\mathsf{d}^{'}(\mathbf{y}_\mathsf{d})$. Plugging (5.41) and (5.40) into (5.39),

$$f_{\mathbf{Y}|\mathcal{H}_0}(\mathbf{y}|\mathcal{H}_0) = \frac{K^{'}(\mathbf{y})}{\pi} \int_0^{\pi} \cosh \frac{\left\langle \mathbf{x}_\mathsf{p}, \mathbf{y}_{\mathsf{p},\phi}^R \right\rangle}{\sigma^2} \prod_{k=0}^{N-1} \cosh \frac{\left\langle \mathbf{s}_k, \mathbf{y}_{\mathsf{d},k,\phi}^R \right\rangle}{\sigma^2} \mathrm{d}\phi$$

is obtained, where

$$\mathbf{y}_{\mathsf{p},\phi}^R = \Re\left\{\mathbf{y}_{\mathsf{p},i}e^{-j\phi}\right\}, \mathbf{y}_{\mathsf{d},k,\phi}^R = \Re\left\{\mathbf{y}_{\mathsf{d},k}e^{-j\phi}\right\}.$$

Looking at hypothesis $\mathcal{H}_1$, where only noise is transmitted, it can be noted that ϕ does not affect $f_{Y|\mathcal{H}_1}(\mathbf{y}|\mathcal{H}_1)$, because of the assumption that the noise is a circularly symmetric Gaussian variable. This means that

$$f_{\mathbf{Y}|\mathcal{H}_1}(\mathbf{y}|\mathcal{H}_1) = \frac{K^{'}(\mathbf{y})}{K_0^{'}}$$

with $K_0^{'} = e^{-L/2\sigma^2}$, which gives the LRT of (5.42).

$$\Lambda^{(1)}(\mathbf{y}) = \frac{K_0^{'}}{\pi}\int_0^{\pi} \cosh\frac{\left\langle \mathbf{x}_{\mathsf{p}}, \mathbf{y}_{\mathsf{p},\phi}^{R} \right\rangle}{\sigma^2} \prod_{k=0}^{N-1} \cosh\frac{\left\langle \mathbf{s}_k, \mathbf{y}_{\mathsf{d},k,\phi}^{R} \right\rangle}{\sigma^2} \mathrm{d}\phi \tag{5.42}$$

5.5.2 Likelihood Ratio Test - Continuous Transmission (Scenario 2)

Under the continuous transmission scenario (Scenario 2), the likelihood of the hypothesis $\mathcal{H}_1$ corresponds to

$$f_{\mathbf{Y}|\mathcal{H}_1}(\mathbf{y}|\mathcal{H}_1) = \frac{K^{'}(\mathbf{y})}{\pi}\int_0^{\pi} \prod_{i=0}^{L-1} \cosh\left\{\frac{\Re\left\{y_i e^{-j\phi}\right\}}{\sigma^2}\right\} \mathrm{d}\phi$$

which gives the LRT in (5.43).

$$\Lambda^{(2)}(\mathbf{y}) = \frac{\pi\Lambda^{(1)}(\mathbf{y})}{K_0^{'}\int_0^{\pi} \prod_{i=0}^{L-1} \cosh\left\{\frac{\Re\left\{y_i e^{-j\phi}\right\}}{\sigma^2}\right\} \mathrm{d}\phi} \tag{5.43}$$

5.5.3 Simplified Tests - (Bursty Transmissions) Scenario 1

The LRT in (5.42) does not admit a closed form solution. For this reason, similarly to [106], this section resorts to approximations, which generate suboptimal tests whose complexity is greatly simplified.

Summation Rule

The integration is approximated by the sum of N_q rectangles with the same width $w_\ell = \pi/N_q$, resulting in

$$\int_0^{\pi} f(\phi)d\phi \approx \sum_{\ell=0}^{N_q-1} w_\ell f(\phi_\ell) = \frac{\pi}{N_q}\sum_{\ell=0}^{N_q-1} f(\phi_\ell)$$

The approximation of (5.42) with N_q rectangles is denoted by $\breve{\breve{\Lambda}}^{(1)}_{N_q}(\mathbf{y})$. In the case $N_q = 2$, only the real and imaginary components of the received vector are required. The approximation of $\breve{\breve{\Lambda}}^{(1)}(\mathbf{y})$ is

$$\begin{aligned}\breve{\breve{\Lambda}}^{(1)}_{N_q=2}(\mathbf{y}) = &\cosh\frac{\langle \mathbf{x}_\mathsf{p}, \Re\{\mathbf{y}_\mathsf{p}\}\rangle}{\sigma^2}\prod_{k=0}^{N-1}\cosh\frac{\langle \mathbf{s}_k, \Re\{\mathbf{y}_{\mathsf{d},k}\}\rangle}{\sigma^2}\\ &+\cosh\frac{\langle \mathbf{x}_\mathsf{p}, \Im\{\mathbf{y}_\mathsf{p}\}\rangle}{\sigma^2}\prod_{k=0}^{N-1}\cosh\frac{\langle \mathbf{s}_k, \Im\{\mathbf{y}_{\mathsf{d},k}\}\rangle}{\sigma^2}\end{aligned} \tag{5.44}$$

When using instead $N_q = 4$, it follows

$$\begin{aligned}\breve{\breve{\Lambda}}^{(1)}_{N_q=4}(\mathbf{y}) = &\breve{\breve{\Lambda}}^{(1)}_{N_q=2}(\mathbf{y}) + \cosh\frac{\langle \mathbf{x}_\mathsf{p}, \mathcal{S}\{\mathbf{y}_\mathsf{p}\}\rangle}{\sigma^2}\prod_{k=0}^{N-1}\cosh\frac{\langle \mathbf{s}_k, \mathcal{S}\{\mathbf{y}_{\mathsf{d},k}\}\rangle}{\sigma^2}\\ &+\cosh\frac{\langle \mathbf{x}_\mathsf{p}, \mathcal{D}\{\mathbf{y}_\mathsf{p}\}\rangle}{\sigma^2}\prod_{k=0}^{N-1}\cosh\frac{\langle \mathbf{s}_k, \mathcal{D}\{\mathbf{y}_{\mathsf{d},k}\}\rangle}{\sigma^2}\end{aligned} \tag{5.45}$$

where $\Im\{\cdot\}$ denotes the imaginary part of a complex value (or vector), and $\mathcal{S}$ and $\mathcal{D}$ denote the sum and difference of the real and imaginary parts, respectively,

$$\mathcal{S}\{a\} = \Re\{a\} + \Im\{a\}, \text{ and } \mathcal{D}\{a\} = \Re\{a\} - \Im\{a\}.$$

Equations (5.44) and (5.45) both involve sums of exponential-type terms—in this case hyperbolic cosine functions, $\cosh(\cdot)$. Such sums are dominated by their largest term. If the entire sum is approximated by its maximum component and then the natural logarithm is taken, the simplified metric $\widetilde{\Lambda}^{(1)}_{N_q=2}(\mathbf{y})$ and $\widetilde{\Lambda}^{(1)}_{N_q=4}(\mathbf{y})$ of (5.46) and (5.47) can be obtained, respectively.

$$\widetilde{\Lambda}^{(1)}_{N_q=2}(\mathbf{y}) = \max\{\ell_R(\mathbf{y}), \ell_I(\mathbf{y})\} \tag{5.46}$$

$$\ell_R(\mathbf{y}) = \log\cosh\frac{\langle \mathbf{x}_\mathsf{p}, \Re\{\mathbf{y}_\mathsf{p}\}\rangle}{\sigma^2} + \sum_{k=0}^{N-1}\log\cosh\frac{\langle \mathbf{s}_k, \Re\{\mathbf{y}_{\mathsf{d},k}\}\rangle}{\sigma^2}$$

$$\ell_I(\mathbf{y}) = \log\cosh\frac{\langle \mathbf{x}_\mathsf{p}, \Im\{\mathbf{y}_\mathsf{p}\}\rangle}{\sigma^2} + \sum_{k=0}^{N-1}\log\cosh\frac{\langle \mathbf{s}_k, \Im\{\mathbf{y}_{\mathsf{d},k}\}\rangle}{\sigma^2}$$

$$\widetilde{\Lambda}^{(1)}_{N_q=4}(\mathbf{y}) = \max\left\{\widetilde{\Lambda}^{(1)}_{N_q=2}(\mathbf{y}), \ell_S(\mathbf{y}), \ell_D(\mathbf{y})\right\} \tag{5.47}$$

$$\ell_S(\mathbf{y}) = \log\cosh\frac{\langle \mathbf{x}_\mathsf{p}, \mathcal{S}\{\mathbf{y}_\mathsf{p}\}\rangle}{\sqrt{2}\sigma^2} + \sum_{k=0}^{N-1}\log\cosh\frac{\langle \mathbf{s}_k, \mathcal{S}\{\mathbf{y}_{\mathsf{d},k}\}\rangle}{\sqrt{2}\sigma^2}$$

$$\ell_D(\mathbf{y}) = \log\cosh\frac{\langle \mathbf{x}_\mathsf{p}, \mathcal{D}\{\mathbf{y}_\mathsf{p}\}\rangle}{\sqrt{2}\sigma^2} + \sum_{k=0}^{N-1}\log\cosh\frac{\langle \mathbf{s}_k, \mathcal{D}\{\mathbf{y}_{\mathsf{d},k}\}\rangle}{\sqrt{2}\sigma^2}$$

Figure 5.9 illustrates the receiver architecture for the non-coherent channel model used to compute the simplified metrics $\widetilde{\Lambda}^{(1)}_{N_q=2}(\mathbf{y})$ and $\widetilde{\Lambda}^{(1)}_{N_q=4}(\mathbf{y})$. The components of the received sample vec-

tor $\mathbf{y}$ are processed into 4 parallel $\ell(\cdot)$ blocks, each producing one branch metric—$\ell_R(\mathbf{y}), \ell_I(\mathbf{y}), \ell_S(\mathbf{y})$ or $\ell_D(\mathbf{y})$—by taking inner products followed by a $\log\cosh()$ sum over the symbols in $\mathbf{y}$ associated with the preamble and the spread bits.

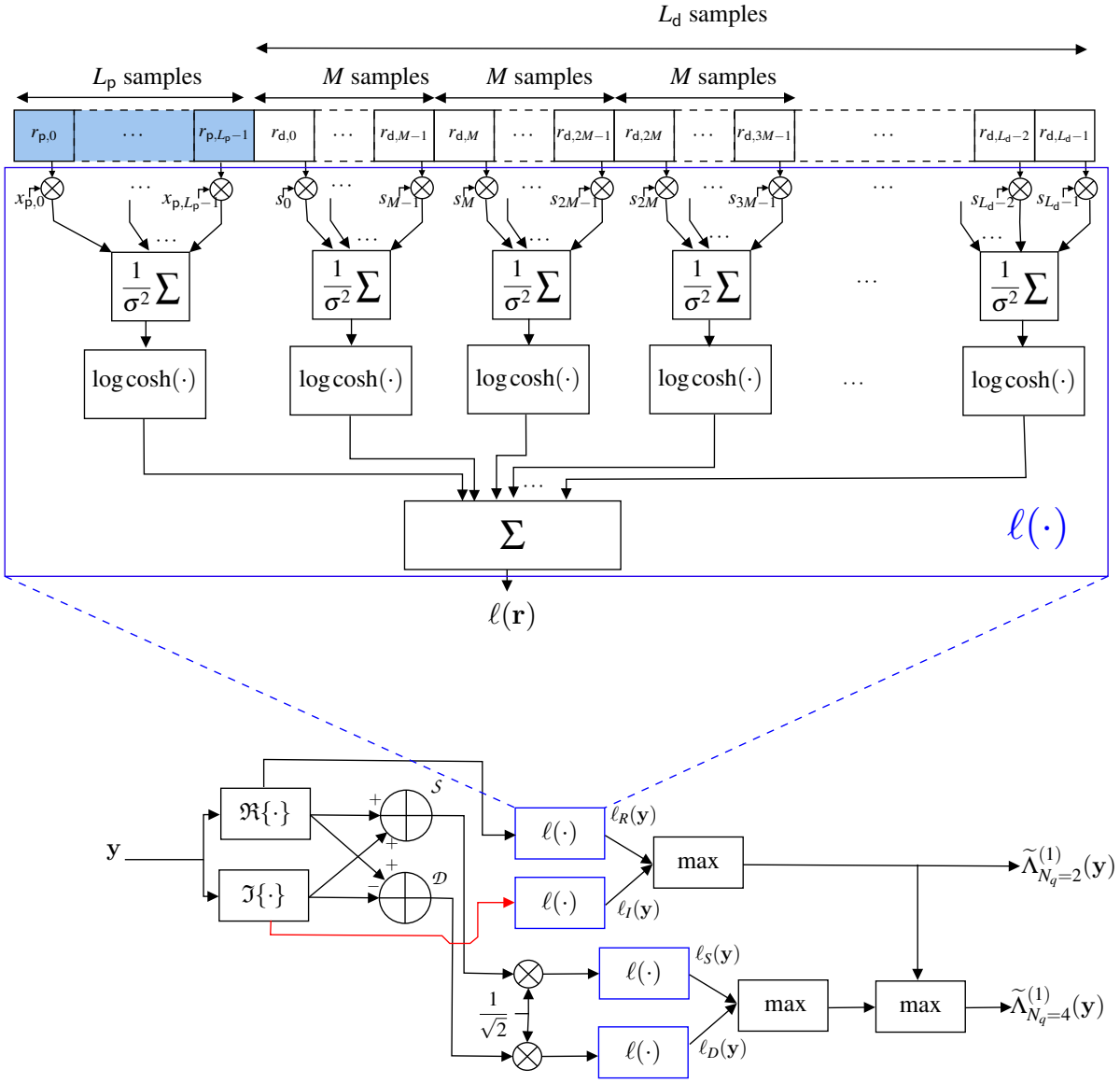


Fig. 5.9 Receiver architecture for the non-coherent channel model, under bursty transmissions (Scenario 1). Block diagram of the receiver implementing the simplified metrics $\widetilde{\Lambda}^{(1)}_{N_q=2}(\mathbf{y})$ and $\widetilde{\Lambda}^{(1)}_{N_q=4}(\mathbf{y})$.

Similarly to Section 5.3.3, the $\log\cosh()$ term can be approximated with its high and low SNR approximations, leading in both cases to tests that do not require the estimation of the noise variance. For the sake of brevity, the equations of those tests are not presented.

Non-coherent Accumulations

To avoid the integral computation in (5.42), a widespread technique resorts to non-coherent accumulations of the spread bits b_k and the preamble [112, 113]. The simplified metric avoids the need to estimate the variance term σ^2 and is expressed as

$$\widetilde{\Lambda}^{(1)}_{\mathrm{ACC}}(\mathbf{y}) = \left|\left\langle \mathbf{x}_{\mathsf{p}}, \mathbf{y}^{*}_{\mathsf{p}} \right\rangle\right| + \sum_{k=0}^{N-1} \left|\left\langle \mathbf{s}_k, \left(\mathbf{y}_{\mathsf{d},k}\right)^{*} \right\rangle\right|. \tag{5.48}$$

The receiver architecture for the non-coherent channel setup used to compute $\widetilde{\Lambda}^{(1)}_{\mathrm{ACC}}(\mathbf{y})$ is depicted in Figure 5.10. Unlike the non-coherent receiver in Figure 5.9 which processes only real-valued elements, the elements in Figure 5.10 carry out every operation on complex-valued signals.

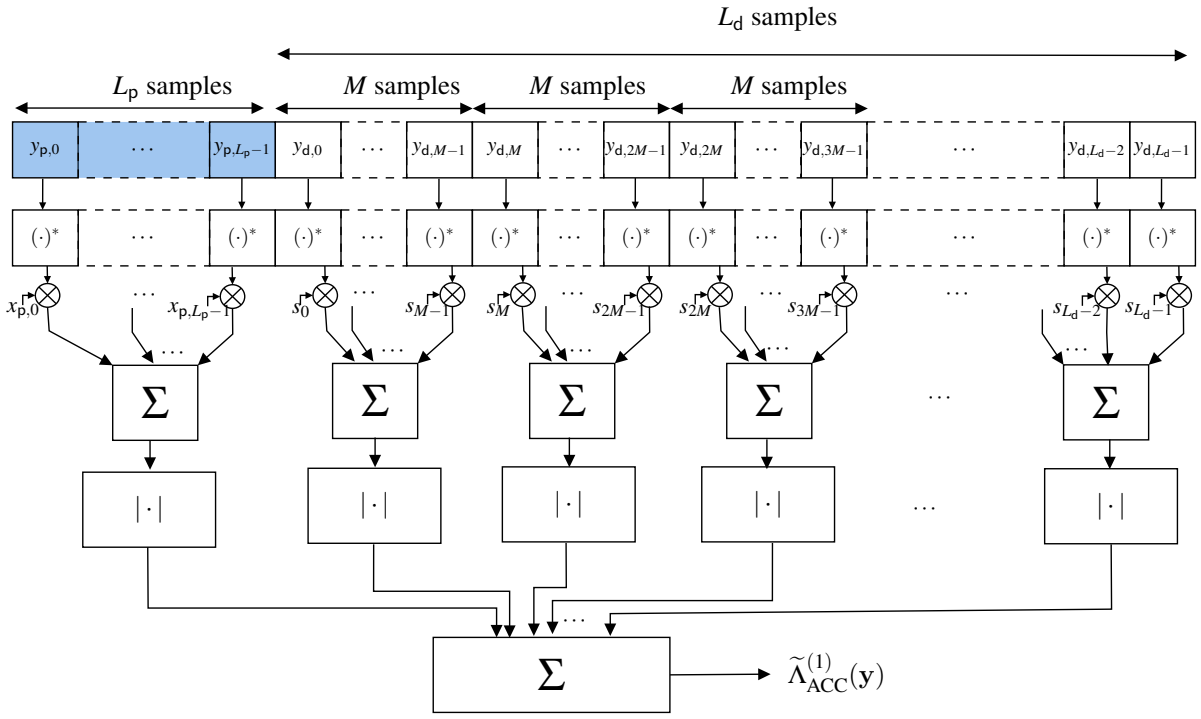


Fig. 5.10 Receiver architecture for the non-coherent channel model, under bursty transmissions (Scenario 1). Block diagram of the non-coherent receiver implementing the simplified non-coherent accumulator $\widetilde{\Lambda}^{(1)}_{\mathrm{ACC}}(\mathbf{y})$.

Performance Comparison

Figure 5.11 reports simulation results obtained by applying the proposed simplified metrics for receiver frame synchronization under the non-coherent channel model. In the upper panel, the performance is depicted in terms of the missed detection probability as a function of E_c/N_0,

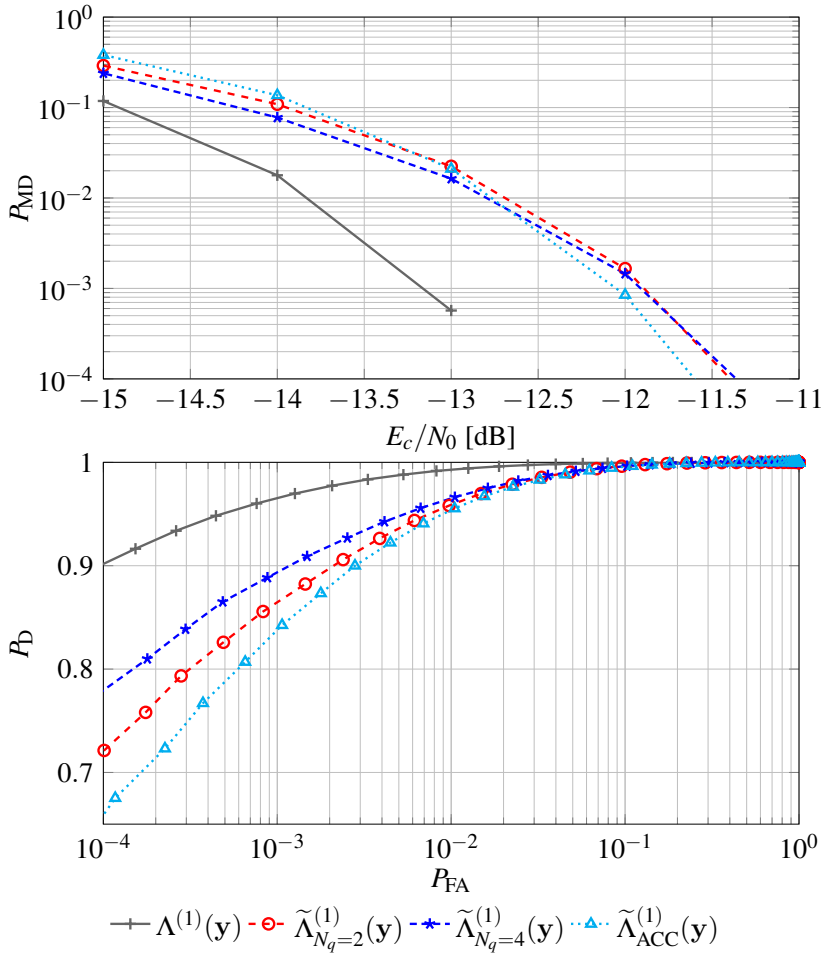


Fig. 5.11 Non-coherent channel model under bursty transmissions (Scenario 1). [top] MD probability versus E_c/N_0 for the LRT $\Lambda^{(1)}(\mathbf{y})$ (solid gray curve) for $N = 250$ bits, with fixed $P_{\mathrm{FA}} = 10^{-3}$, vs. the respective simplified tests. [bottom] ROC curve for the threshold test applied to $\Lambda^{(1)}(\mathbf{y})$ (solid gray curve) for $N = 100$ bits, with fixed $E_c/N_0 = -12$ dB, vs. the one relying on the respective simplified metrics.

considering the case with $N = 250$ bits and a false alarm rate fixed to 10^{-3}. In particular, the solid gray curve represents the performance of the optimal LRT $\Lambda^{(1)}(y)$, while the curves associated with the simplified metrics — obtained by approximating the integral part of the function (with $N_q = 2$ and $N_q = 4$), and the one based on non-coherent accumulations $\widetilde{\Lambda}^{(1)}_{\text{ACC}}(y)$ — show robust performance, although they exhibit a certain loss compared to the optimal test.

The lower panel in Figure 5.11 presents the ROC curves for $N = 100$ bits at $E_c/N_0 = -12$ dB, illustrating the trade-off between detection and false alarm probabilities for each method. Overall, the results indicate that, despite the reduced computational complexity, the simplified tests approach the performance of the LRT, making them appealing for practical applications where a compromise between accuracy and implementation simplicity is desired.

5.6 Improved Frame Synchronization for E-SSA over the UMAC

The frame synchronization metrics introduced in this chapter can be applied directly to the enhanced spread spectrum Aloha (E-SSA) frame detector in the unsourced multiple access channel (UMAC) scenario of Chapter 4, without any changes to the existing system parameters. By using these new metrics, more reliable synchronization can be achieved—reducing both missed detections and false alarms—and thereby avoiding unnecessary channel decoder activations. In addition, because these metrics allow shortening the preamble (which carries no payload and thus results in an energy loss), they improve overall energy efficiency and lower the interference each user generates. Concretely, within the Chapter 4 framework—a coherent receiver facing many independent interferers whose aggregate effect is treated as Gaussian—the original preamble-only based correlator of the E-SSA receiver is replaced by the metric introduced in (5.19):

$$\widetilde{\Lambda}^{(1)}(\mathbf{y}) = \frac{\langle \mathbf{y}_{\mathsf{p}}, \mathbf{x}_{\mathsf{p}} \rangle}{\sigma^2} + \sum_{k=0}^{N-1} \log\cosh\left\{ \frac{\langle \mathbf{y}_{\mathsf{d},k}, \mathbf{s}_k \rangle}{\sigma^2} \right\}.$$

The metric is restated here purely for the reader's convenience.

With the same system parameters used in Section 4.5.3—namely a frame length of $\mathrm{n} = 30000$, spreading factor $M = 25$, $K = 100$ bits of information, and the $(1000, 100)$ 5GNR polar code, the system design is modified utilizing a shorter preamble length of $L_{\mathsf{p}} = 582$ symbols instead of $L_{\mathsf{p}} = 3050$. This modification reduces the energy loss introduced by the preamble from $\Delta E = 0.5$ dB to $\Delta E = 0.1$ dB.

The receiver incorporating the new metric selects the $W = 200$ time instants $t \in [\mathrm{n}]$ with the largest value of $\widetilde{\Lambda}^{(1)}(\mathbf{y})$ metric, as opposed to $W = 250$ time instants $t \in [\mathrm{n}]$ with the largest correlation value of their preamble, as done previously in Chapter 4. Figure 5.12 shows the

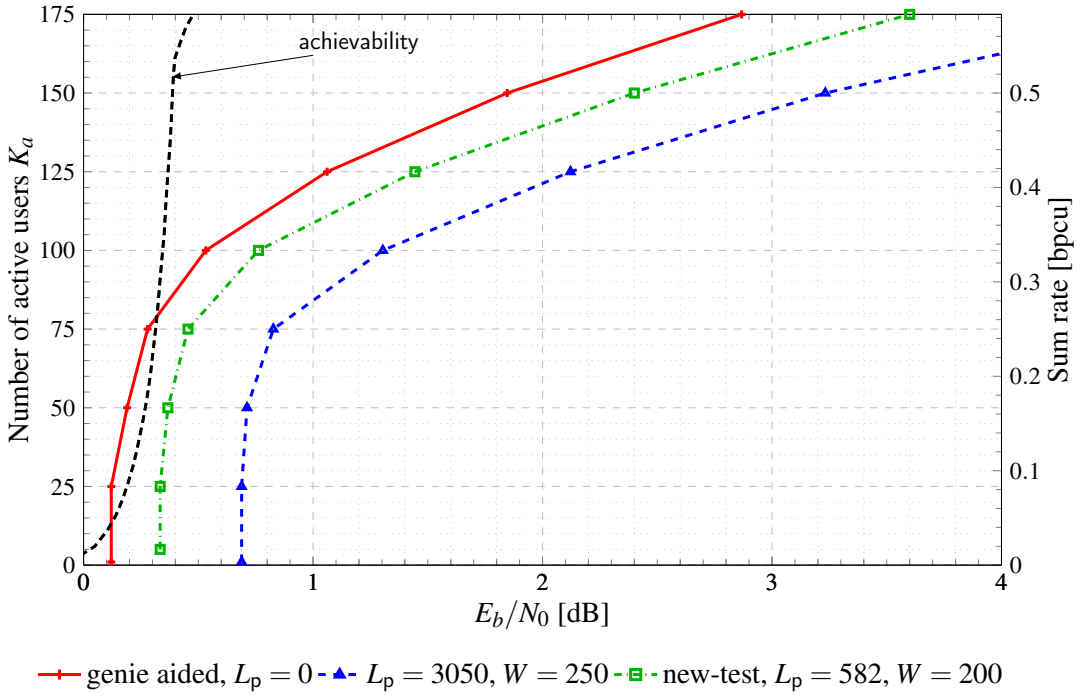


Fig. 5.12 Minimum required E_b/N_0 versus the number of active users K_a for achieving a PUPE of 5×10^{-2}. The plot compares the performance of the system analyzed in Chapter 4 based on the correlation of the preamble alone (blue curve), the improved system employing the new frame synchronization metric for the coherent receiver (green curve), and a genie-aided benchmark (red curve).

resulting system performance. Note that $W = 200$ results in approximately 20% fewer channel decoder calls. The figure reports the minimum E_b/N_0 required to maintain a PUPE of 5×10^{-2} for varying numbers of active users, K_a. At low to moderate user loads ($K_a \leq 75$), the old system (blue curve) suffers an approximately 0.4 dB penalty compared to the new metric-based system (green curve), which is consistent with the energy loss attributed to a longer preamble. As the number of users increases, the performance penalty of the conventional system becomes more pronounced due to increased preamble-induced interference.

It is also noteworthy that the new system, with $E_b/N_0 = 0.8$ dB, can support $K_a = 100$ users, whereas the old system could only support $K_a = 75$ users, resulting in a 33.3% improvement. Furthermore, the performance with the new metric is very similar to that of a genie-aided system (red curve), which knows the exact positions of the packets of various users and where no preamble is used, and up to $K_a = 75$, it almost matches the achievability bound in [12].

5.7 Final Remarks

This chapter presented an in-depth analysis of frame synchronization in direct-sequence spread spectrum systems, considering both coherent and non-coherent channel models. By leveraging

the knowledge of preambles and spreading sequences, the optimal likelihood ratio test has been derived using the Neyman-Pearson framework, and its performance has been characterized analytically. The results demonstrate that incorporating spreading sequence information into the synchronization process significantly improves detection accuracy compared to algorithms that make use of the preamble only. To address computational complexity, various simplified tests have been proposed and evaluated as practical alternatives. These simplified tests maintain robust performance while offering reduced implementation complexity, making them suitable for real-world communication systems. Moreover, the new metrics have been incorporated in the random access scheme analyzed in Chapter 4, showing that the new frame synchronization metrics allow the design of shorter preamble sequences, resulting in an improvement of the energy efficiency of the overall system. Furthermore, the new tests allow to reduce the channel decoding calls, to increase the number of users sustained by the system, tightly approaching the theoretical bounds for the considered system for moderate channel loads.

Chapter 6

Conclusions

This thesis has addressed some key challenges in machine-type communications (MTCs) by focusing on three main areas: the analysis of concatenated convolutional codes with outer polynomial codes, the design and performance evaluation of enhanced spread spectrum Aloha for unsourced multiple access channels, and likelihood-based sequential frame synchronization for direct-sequence spread spectrum systems.

In Chapter 3, a comprehensive analysis of convolutional codes concatenated with outer polynomial codes, poly+CC, was conducted. By investigating the trellis structure and the associated distance spectrum, it was shown that significant coding gains—on the order of 3 dB—can be achieved under maximum likelihood decoding of the poly+CC compared to Viterbi decoding of the convolutional code only. The analytical and numerical results obtained in both CCSDS telemetry and LTE scenarios confirm that the poly+CC approach yields a robust solution for short-packet communications, effectively reducing error probabilities while keeping the decoder complexity practical. Poly+CC with list Viterbi decoding is therefore a strong candidate for channel coding in MTCs, especially concerning point-to-point links.

Chapter 4 addressed the medium access control (MAC) layer of massive MTCs (mMTCs), where large terminal populations are required to transmit short information messages in a sporadic, unpredictable manner. In this context, the enhanced spread spectrum Aloha (E-SSA) protocol has been modified to comply with the unsourced multiple access channel (UMAC) framework. By incorporating low-rate short polar codes and leveraging a timing channel for error detection, the optimized protocol achieves performance levels that are competitive with state-of-the-art UMAC schemes. Moreover, its linear receiver complexity and the inherent simplicity of the transmitter highlight the practical benefits of E-SSA in both terrestrial and satellite-based mMTC systems.

In Chapter 5, recognizing the essential role played by the preamble length in the energy efficiency of E-SSA, the sequential frame synchronization problem in direct-sequence spread

spectrum systems was addressed. By formulating the synchronization task as a binary-hypothesis testing problem and exploiting the information present in the preamble and spreading sequences, optimal likelihood ratio tests were derived for both coherent and non-coherent channel models. The subsequent development of efficient approximations shows that incorporating spreading sequence information substantially enhances synchronization accuracy, even under low-power conditions and phase uncertainty. Furthermore, these new test metrics have been successfully integrated into the E-SSA protocol of Chapter 4, further boosting the overall performance to tightly approach the theoretical limits of the Gaussian UMAC channel.

The techniques developed in the context of this thesis can be applied to a wide class of MTC problems, addressing key components that directly define the energy-and-spectral efficiency of a system: channel coding, the MAC protocol, and frame synchronization. Several research directions can be considered for further investigation.

- **Concatenation of Convolutional and Polynomial Codes:** Although the structures and properties of convolutional codes and polynomial codes are well understood individually, a deeper theoretical investigation into their concatenation could reveal opportunities for novel design and decoding techniques. New insights in this area could lead to improved joint decoding strategies, and potentially incorporates poly+CC codes in more advanced serial or parallel turbo coding schemes.

- **Upper Bounds for List Decoding:** While tight bounds exist for the block error probability of channel codes under maximum likelihood decoding, the literature lacks tight upper bounds for both the total and undetected error probabilities of list decoders. Developing such bounds would provide a deeper understanding of the performance limits of practical list decoding algorithms.

- **Advanced List Decoding Algorithms for Convolutional Codes:** Although this thesis showed that moderate list sizes can yield significant coding gains for list Viterbi algorithms, further research in lower-complexity list decoders seems still necessary. New decoding techniques that could bridge the gap to theoretical limits with reduced complexity would be particularly desirable, especially for joint decoding of convolutional codes concatenated with outer polynomial codes.

- **Low-Rate Short Channel Codes:** Despite the fact that polar codes exhibit increasing coding gains when their rate decreases, even for moderate and big list sizes, their single-user performance cannot attain the theoretical performance bounds at finite blocklength, keeping open the question on how to design good codes in this regime.

- **Frame Synchronization for Non-coherent Channel:** Even though the simplified metrics for the sequential frame synchronization of direct-sequence spread spectrum systems,

proposed in Chapter 5 for the non-coherent channel model, can provide significant performance improvement in packet detection, a non-negligible performance gap arises w.r.t. the optimum test. New insights in this area could lead to improved frame synchronization strategies, which could also be robust to other channel impairments such as frequency offsets and Doppler frequency shifts.

References

[1] Claude Elwood Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, July 1948.

[2] 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Service requirements for the 5G system; Stage 1 (Release 20). TS 22.261, V. 20.1.0, 3GPP, December 2024.

[3] LoRaWAN L2 1.0.4 Specification. TS 001, V. 1.0.4, LoRa Alliance Technical Committee, October 2020.

[4] Low power protocol for wide area wireless networks. ITU-T Y.4480, International Telecommunication Union, January 2021.

[5] Martin Woolley and Ifti Anees. The Bluetooth low energy primer. Technical Report Version 1.2.0, Bluetooth, March 2024.

[6] IEEE standard for low-rate wireless networks. IEEE Std 802.15.4-2024, IEEE, December 2024.

[7] Robert Mario Fano. *Transmission of information: a statistical thoery of communications*. Cambridge, Mass.: M.I.T. Press, and New York: Wiley, 1961.

[8] Robert Gray Gallager. A simple derivation of the coding theorem and some applications. *IEEE Transactions on Information Theory*, 11(1):3–18, January 1965.

[9] Yury Polyanskiy, H Vincent Poor, and Sergio Verdú. Channel coding rate in the finite blocklength regime. *IEEE Transactions on Information Theory*, 56(5):2307–2359, May 2010.

[10] George David Forney Jr. *Concatenated codes*. PhD thesis, Massachusetts Institute of Technology, December 1965.

[11] Peter Elias. List decoding for noisy channels. September 1957.

[12] Yury Polyanskiy. A perspective on massive random-access. In *Proc. IEEE Int. Symp. Inf. Theory*, June 2017.

[13] Georges Voronoi. Nouvelles applications des paramètres continus à la théorie des formes quadratiques. deuxième mémoire. recherches sur les parallélloèdres primitifs. *Journal für die reine und angewandte Mathematik (Crelles Journal)*, 1908(134):198–287, 1908.

[14] George David Forney Jr. Geometrically uniform codes. *IEEE Transactions on Information Theory*, 37(5):1241–1260, September 1991.

[15] Claude Elwood Shannon. Probability of error for optimal codes in a gaussian channel. *Bell System Technical Journal*, 38(3):611–656, May 1959.

[16] Robert Gray Gallager. *Low-Density Parity-Check Codes*. PhD thesis, Massachusetts Institute of Technology, July 1963.

[17] Claude Berrou, Alain Glavieux, and Punya Thitimajshima. Near Shannon limit error-correcting coding and decoding: Turbo-codes. 1. In *Proc. IEEE International Conference on Communications*, volume 2, pages 1064–1070, May 1993.

[18] Mustafa Cemil Coşkun, Giuseppe Durisi, Thomas Jerkovits, Gianluigi Liva, William Ryan, Brian Stein, and Fabian Steiner. Efficient error-correcting codes in the short blocklength regime. *Physical Communication*, 34:66–79, June 2019.

[19] Peter Elias. Coding for noisy channels. *IRE Wescon Convention Record*, 3:37–46, March 1955.

[20] Lorenzo Gaudio, Tudor Ninacs, Thomas Jerkovits, and Gianluigi Liva. On the performance of short tail-biting convolutional codes for ultra-reliable communications. *Proc. ITG International Conference on Systems, Communications and Coding*, pages 1–6, February 2017.

[21] Erdal Arikan. Channel Polarization: A Method for Constructing Capacity-Achieving Codes for Symmetric Binary-Input Memoryless Channels. *IEEE Transactions on Information Theory*, 55(7):3051–3073, July 2009.

[22] Ido Tal and Alexander Vardy. List Decoding of Polar Codes. *IEEE Transactions on Information Theory*, 61(5):2213–2226, 2015.

[23] Hengjie Yang, Sudarsan VS Ranganathan, and Richard Dale Wesel. Serial list Viterbi decoding with CRC: Managing errors, erasures, and complexity. In *Proc. IEEE Global Communications Conference*, pages 1–6. IEEE, 2018 2018.

[24] Jerrold A. Heller. Short constraint length convolutional codes. *Space Program Summary, Jet Propulsion Laboratory, California Institute of Technology, Pasadena*, 3:171–174, October 1968.

[25] James Hugo Griesmer. A bound for error-correcting codes. *IBM Journal of Research and Development*, 4(5):532–542, November 1960.

[26] Rolf Johannesson and Kamil Sh Zigangirov. *Fundamentals of convolutional coding*. John Wiley & Sons, December 2015.

[27] George David Forney Jr. Review of random tree codes. *NASA Ames Research Center, Moffett Field, CA, USA, Tech. Rep. NASA CR73176*, 1967.

[28] Robert J McEliece. How to compute weight enumerators for convolutional codes. *Communications and Coding*, pages 121–141, January 1998.

[29] Chiara Ravazzi and Fabio Fagnani. On the growth rate of the input-output weight distribution of convolutional encoders. *SIAM Journal on Discrete Mathematics*, 26(3):1310–1345, January 2012.

[30] Mats L. Cedervall and Rolf Johannesson. A fast algorithm for computing distance spectrum of convolutional codes. *IEEE Transactions on Information Theory*, 35(6):1146–1159, November 1989.

[31] Irina E Bocharova, Marc Handlery, Rolf Johannesson, and Boris D Kudryashov. A BEAST for prowling in trees. *IEEE Transactions on Information Theory*, 50(6):1295–1302, June 2004.

[32] Andrew James Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269, 1967.

[33] Jim K. Omura. On the Viterbi decoding algorithm. *IEEE Transactions on Information Theory*, 15(1):177–179, January 1969.

[34] Rose Y Shao, Shu Lin, and Marc P.C. Fossorier. Two decoding algorithms for tailbiting codes. *IEEE Transactions on Communications*, 51(10):1658–1665, October 2003.

[35] W Wesley Peterson. *Error-correcting codes*. Massachusetts Institute of Technology, January 1961.

[36] Richard E Blahut. *Algebraic codes for data transmission*. Cambridge University Press, 2003.

[37] William Wesley Peterson and David T. Brown. Cyclic codes for error detection. *Proceedings of the IRE*, 49(1):228–235, January 1961.

[38] Jack Keil Wolf and Robert D. Blakeney. An exact evaluation of the probability of undetected error for certain shortened binary CRC codes. In *Proc. IEEE Military Communications Conference*, pages 287–292. IEEE, October 1988.

[39] Richard Comroe and Daniel Costello. ARQ schemes for data transmission in mobile radio systems. *IEEE Journal on Selected Areas in Communications*, 2(4):472–481, July 1984.

[40] Chung-Yu Lou, Babak Daneshrad, and Richard Dale Wesel. Convolutional-code-specific CRC code design. *IEEE Transactions on Communications*, 63(10):3459–3470, 2015.

[41] Hengjie Yang, Ethan Liang, and Richard Dale Wesel. Joint design of convolutional code and CRC under serial list Viterbi decoding. *arXiv preprint*, November 2018.

[42] Hengjie Yang, Ethan Liang, Hanwen Yao, Alexander Vardy, Dariush Divsalar, and Richard Dale Wesel. A list-decoding approach to low-complexity soft maximum-likelihood decoding of cyclic codes. In *Proc. IEEE Global Communications Conference*, pages 1–6, December 2019.

[43] Ethan Liang, Hengjie Yang, Dariush Divsalar, and Richard Dale Wesel. List-decoded tail-biting convolutional codes with distance-spectrum optimal CRCs for 5G. In *Proc. IEEE Global Communications Conference*, pages 1–6, December 2019.

[44] Hengjie Yang, Linfang Wang, Vincent Lau, and Richard Dale Wesel. An efficient algorithm for designing optimal CRCs for tail-biting convolutional codes. In *Proc. IEEE International Symposium on Information Theory*, pages 292–297. IEEE, 2020.

[45] Riccardo Schiavone. Channel Coding for Massive IoT Satellite Systems. Master Thesis, Politecnico di Torino, 2021.

[46] Wenhui Sui, Hengjie Yang, Brendan Towell, Ava Asmani, and Richard Dale Wesel. High-rate convolutional codes with CRC-aided list decoding for short blocklengths. In *IEEE International Conference on Communications*, pages 98–103, May 2022.

[47] Jacob King, Alexandra Kwon, Hengjie Yang, William Ryan, and Richard Dale Wesel. CRC-aided list decoding of convolutional and polar codes for short messages in 5G. In *IEEE International Conference on Communications*, pages 92–97, May 2022.

[48] Jacob King, William Ryan, and Richard Dale Wesel. CRC-aided short convolutional codes and RCU bounds for orthogonal signaling. In *IEEE Global Communications Conference*, pages 4256–4261, December 2022.

[49] Linfang Wang, Dan Song, Felipe Areces, and Richard Dale Wesel. Achieving short-blocklength rcu bound via CRC list decoding of tcm with probabilistic shaping. In *IEEE International Conference on Communications*, pages 2906–2911, May 2022.

[50] Dan Song, Felipe Areces, Linfang Wang, and Richard Dale Wesel. Shaped TCM with list decoding that exceeds the RCU bound by optimizing a union bound on FER. In *IEEE Global Communications Conference*, pages 4262–4267, December 2022.

[51] Linfang Wang, Dan Song, Felipe Areces, Thomas Wiegart, and Richard Dale Wesel. Probabilistic shaping for trellis-coded modulation with CRC-aided list decoding. *IEEE Transactions on Communications*, 71(3):1271–1283, January 2023.

[52] Hengjie Yang, Ethan Liang, Minghao Pan, and Richard Dale Wesel. CRC-aided list decoding of convolutional codes in the short blocklength regime. *IEEE Transactions on Information Theory*, 68(6):3744–3766, June 2022.

[53] Riccardo Schiavone, Roberto Garello, and Gianluigi Liva. Application of list Viterbi algorithms to improve the performance in space missions using convolutional codes. In *9th International Workshop on Tracking, Telemetry and Command Systems for Space Applications (TT&C)*, pages 1–8, November 2022.

[54] Nambirajan Seshadri and C-EW Sundberg. List Viterbi decoding algorithms with applications. *IEEE Transactions on Communications*, 42(234):313–323, February 1994.

[55] Frank K Soong and Eng-Fong Huang. A tree-trellis based fast search for finding the N best sentence hypotheses in continuous speech recognition. In *Proc. of the Workshop on Speech and Natural Language*, June 1990.

[56] John William Joseph Williams. Heapsort. *Communications of the ACM*, 7:347–348, June 1964.

[57] Rudolf Bayer. Symmetric binary b-trees: Data structure and maintenance algorithms. *Acta informatica*, 1(4):290–306, January 1972.

[58] CCSDS. TM Synchronization and Channel Coding. *Recommended standard. Blue Book*, (131.0-B-4), April 2022.

[59] CCSDS. TM Space Data Link Protocol. *Recommended standard. Blue Book*, (132.0-B-3), October 2021.

[60] CCSDS. AOS Space Data Link Protocol. *Recommended standard. Blue Book*, (732.0-B-4), October 2021.

[61] George David Forney Jr. Exponential error bounds for erasure, list, and decision feedback schemes. *IEEE Transactions on Information Theory*, 14(2):206–220, March 1968.

[62] Irving Stoy Reed and Gustave Solomon. Polynomial codes over certain finite fields. *Journal of the Society for Industrial and Applied Mathematics*, 8(2):300–304, June 1960.

[63] Bin Li, Hui Shen, and David Tse. An adaptive successive cancellation list decoder for polar codes with cyclic redundancy check. *IEEE Commun. Lett.*, 16(12):2044–2047, December 2012.

[64] Marc P.C. Fossorier, Shu Lin, and Daniel J. Costello Jr. On the weight distribution of terminated convolutional codes. *IEEE Transactions on Information Theory*, 45(5):1646–1648, July 1999.

[65] CCSDS. TM Synchronization and Channel Coding - Summary of Concept and Rationale. *Informational Report. Green Book*, (130.1-G-3), June 2020.

[66] Roberto Garello, Paola Pierleoni, and Sergio Benedetto. Computing the free distance of turbo codes and serially concatenated codes with interleavers: algorithms and applications. *IEEE Journal on Selected Areas in Communications*, 19(5):800–812, May 2001.

[67] ETSI. Lte; evolved universal terrestrial radio access (e-utra); multiplexing and channel coding. *TS 136 212 V17.1.0*, April 2022.

[68] Gregory Poltyrev. Bounds on the decoding error probability of binary linear codes via their spectra. *IEEE Transactions on Information Theory*, 40(4):1284–1292, July 1994.

[69] Ethan Liang. Improved decoding of convolutional and turbo codes via list decoding. Technical report, Stanford University, February 2020.

[70] Wenhui Sui, Brendan Towell, Ava Asmani, Hengjie Yang, Holden Grissett, and Richard D Wesel. CRC-aided high-rate convolutional codes with short blocklengths for list decoding. *IEEE Transactions on Communications*, 72(1):63–74, October 2023.

[71] Jacob King, William Ryan, Chester Hulse, and Richard D Wesel. Efficient maximum-likelihood decoding for TBCC and CRC-TBCC codes via parallel list Viterbi. In *Proc. International Symposium on Topics in Coding*, pages 1–5. IEEE, September 2023.

[72] Oscar Del Rio Herrero and Riccardo De Gaudenzi. A High Efficiency Scheme for Quasi-Real-Time Satellite Mobile Messaging Systems. In *Proc. Int. Workshop Signal Processing for Space Commun.*, October 2008.

[73] Norman Manuel Abramson. VSAT data networks. *Proc. IEEE*, 78(7):1267–1274, July 1990.

[74] Norman Manuel Abramson. The ALOHA System - Another Alternative for Computer Communications. In *Proc. Fall Joint Comp. Conf.*, pages 281–285, November 1970.

[75] 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 Int. Conf. Commun.*, June 2020.

[76] Robert Gray Gallager. Basic Limits on Protocol Information in Data Communication Networks. *IEEE Transactions on Information Theory*, pages 385–398, July 1976.

[77] 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 Transactions on Commun.*, 70(11):7124–7133, November 2022.

[78] Dmitri Truhachev, Murwan Bashir, Alireza Karami, and Ehsan Nassaji. Low-Complexity Coding and Spreading for Unsourced Random Access. *IEEE Commun. Lett.*, 25(3):774–778, March 2021.

[79] 3rd Generation Partnership Project (3GPP). 5GNR: MAC protocol specification. *TS 138.321, Rev. 16.1.0*, July 2020.

[80] Ido Tal and Alexander Vardy. List Decoding of Polar Codes. *IEEE Transactions on Information Theory*, 61(5):2213–2226, May 2015.

[81] V. Anantharam and S. Verdu. Bits Through Queues. *IEEE Transactions on Information Theory*, 42(1):4–18, January 1996.

[82] A. Ephremides and B. Hajek. Information Theory and Communication Networks: An Unconsummated Union. *IEEE Transactions on Information Theory*, 44(6):2416–2434, October 1998.

[83] Laura Galluccio, Giacomo Morabito, and Sergio Palazzo. TC-Aloha: A Novel Access Scheme for Wireless Networks with Transmit-Only Nodes. *IEEE Transactions on Wireless Commun.*, 12(8):3696–3709, August 2013.

[84] Carsten Bockelmann, Nuno Pratas, Hosein Nikopour, Kelvin Au, Tommy Svensson, Cedomir Stefanovic, Petar Popovski, and Armin Dekorsy. Massive Machine-type Communications in 5G: Physical and MAC-layer solutions. *IEEE Commun. Mag.*, 54(9):59–65, September 2016.

[85] Stefano Cioni, Riccardo De Gaudenzi, Oscar Del Rio Herrero, and Nicolas Girault. On the Satellite Role in the Era of 5G Massive Machine Type Communications. *IEEE Netw.*, 32(5):54–61, September 2018.

[86] Evgeny Marshakov, Gleb Balitskiy, Kirill Andreev, and Alexey Frolov. A Polar Code Based Unsourced Random Access for the Gaussian MAC. In *Proc. IEEE Veh. Technol. Conf.*, September 2019.

[87] Alexander Fengler, Peter Jung, and Giuseppe Caire. SPARCs for Unsourced Random Access. *IEEE Trans. Inf. Theory*, (10):6894–6915, October 2021.

[88] Mert Ozates, Mohammad Kazemi, and Tolga M. Duman. Unsourced Random Access Using ODMA and Polar Codes. *IEEE Wireless Communications Letters*, April 2024.

[89] Gianluigi Liva and Yuri Polyanskiy. Unsourced Multiple Access: A Coding Paradigm for Massive Random Access. *Proceedings of the IEEE*, 112(9):1214–1229, September 2024.

[90] David Haccoun and Stéphan Lefrancois. New very low rate nested convolutional codes. Technical report, École Polytechnique de Montréal, November 1995.

[91] Norbert Stolte. *Rekursive Codes mit der Plotkin-Konstruktion und ihre Decodierung.* PhD thesis, Darmstadt University of Technology, Germany, January 2002.

[92] Morris Plotkin. Binary codes with specified minimum distance. *IRE Transactions on Information Theory*, 6(4):445–450, September 1960.

[93] Valerio Bioglio, Carlo Condo, and Ingmar Land. Design of polar codes in 5G new radio. *IEEE Communications Surveys and Tutorials*, 23(1):29–40, January 2020.

[94] Ido Tal and Alexander Vardy. How to construct polar codes. *IEEE Transactions on Information Theory*, 59(10):6562–6582, October 2013.

[95] Peter Trifonov. Efficient design and decoding of polar codes. *IEEE Transactions on Communications*, 60(11):3221–3227, November 2012.

[96] Pal Frenger, Pal Orten, and Tony Ottosson. Code-spread CDMA using maximum free distance low-rate convolutional codes. *IEEE Transactions on Communications*, 48(1):135–144, January 2000.

[97] ETSI. TS 102 721-1 v1.1.1, Satellite Earth Stations and Systems; Air Interface for S-band Mobile Interactive Multimedia (S-MIM), 2011.

[98] Alexander Sauter, Balázs Matuz, and Gianluigi Liva. Error detection strategies for CRC-concatenated polar codes under successive cancellation list decoding. In *57th Annual Conference on Information Sciences and Systems*, pages 1–6. IEEE, March 2023.

[99] Alexander Sauter, Ahmet Oguz Kislal, Giuseppe Durisi, Gianluigi Liva, Balázs Matuz, and Erik G Ström. Undetected error probability in the short blocklength regime: Approaching finite-blocklength bounds with polar codes. *IEEE Transactions on Communications*, February 2025.

[100] Suhas S. Kowshik and Yury Polyanskiy. Fundamental Limits of Many-User MAC With Finite Payloads and Fading. *IEEE Transactions on Information Theory*, 67(9):5853–5884, September 2021.

[101] Zeyu Han, Xiaojun Yuan, Chongbin Xu, Shuchao Jiang, and Xin Wang. Sparse Kronecker-product coding for unsourced multiple access. *IEEE Wireless Communications Letters*, 10(10):2274–2278, October 2021.

[102] Marco Chiani and Maria G Martini. On sequential frame synchronization in AWGN channels. *IEEE Transactions on Communications*, 54(2):339–348, February 2006.

[103] James Massey. Optimum frame synchronization. *IEEE Transactions on Communications*, 20(2):115–119, April 1972.

[104] Patrick Robertson. *Optimal frame synchronization for continuous and packet data transmission*. PhD thesis, UniBw Muenchen, June 1995.

[105] Maria G Martini and Marco Chiani. Optimum metric for frame synchronization with Gaussian noise and unequally distributed data symbols. In *2009 IEEE 10th Workshop on Signal Processing Advances in Wireless Communications*, pages 643–647. IEEE, July 2009.

[106] Marco Chiani. Noncoherent frame synchronization. *IEEE Transactions on Communications*, 58(5):1536–1545, May 2010.

[107] Andrew James Viterbi. *CDMA: principles of spread spectrum communication*. Addison Wesley Longman Publishing Co., Inc., April 1995.

[108] Jerzy Neyman and Egon Sharpe Pearson. IX. on the problem of the most efficient tests of statistical hypotheses. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 231(694-706):289–337, February 1933.

[109] Fred C Leone, Lloyd S Nelson, and RB Nottingham. The folded normal distribution. *Technometrics*, 3(4):543–550, January 1961.

[110] PB Patnaik. The non-central χ^2-and F-distribution and their applications. *Biometrika*, 36(1/2):202–232, June 1949.

[111] William G Cochran. The distribution of quadratic forms in a normal system, with applications to the analysis of covariance. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 30, pages 178–191. Cambridge University Press, April 1934.

[112] Zhiwei Lu, Yudi Chen, Yiwen Jiao, and Kuanfei Sun. An improved PMF-FFT acquisition approach based on frequency segmentation for DSSS signals. In *5th Information Technology, Networking, Electronic and Automation Control Conference*, volume 5, pages 143–147. IEEE, 2021.

[113] Daniele Borio, Cillian O'Driscoll, and Gerard Lachapelle. Coherent, noncoherent, and differentially coherent combining techniques for acquisition of new composite GNSS signals. *IEEE Transactions on Aerospace and Electronic Systems*, 45(3):1227–1240, 2009.

[114] Henry E Daniels. Saddlepoint approximations in statistics. *The Annals of Mathematical Statistics*, pages 631–650, 1954.

[115] Robert Lugannani and Stephen Rice. Saddle point approximation for the distribution of the sum of independent random variables. *Advances in applied probability*, 12(2):475–490, June 1980.

Appendix A

Computation of the Weight Enumerator of Convolutional Codes

.

We describe next, via an example, how the weight enumerating function (WEF) of a ZTCC could be computed by means of its trellis diagram (the extension to the TBCC is trivial). The example is depicted in Figure A.1. Denote by

- $A_{\sigma_i}^{(t)}(X)$ the *partial* WEF at state σ_i in section t
- $a_{\sigma_i \to \sigma_j}(\mathtt{X})$ the monomial $\mathtt{X}^d$, where d is the Hamming weight of the encoder output associated to the branch $\sigma_i \to \sigma_j$.

The WEF is computed recursively, by initializing, at $t = 0$, the *partial* WEF at state σ_0, $A_{\sigma_0}^{(0)}(\mathtt{X})$ to 1. At time t, the partial WEF at state σ_i is computed as

$$A_{\sigma_i}^{(t)}(\mathtt{X}) = A_{\sigma_j}^{(t-1)}(\mathtt{X}) a_{\sigma_j \to \sigma_i}(\mathtt{X}) + A_{\sigma_s}^{(t-1)}(\mathtt{X}) a_{\sigma_s \to \sigma_i}(\mathtt{X}),$$

where σ_j and σ_s are left neighbors of σ_i.

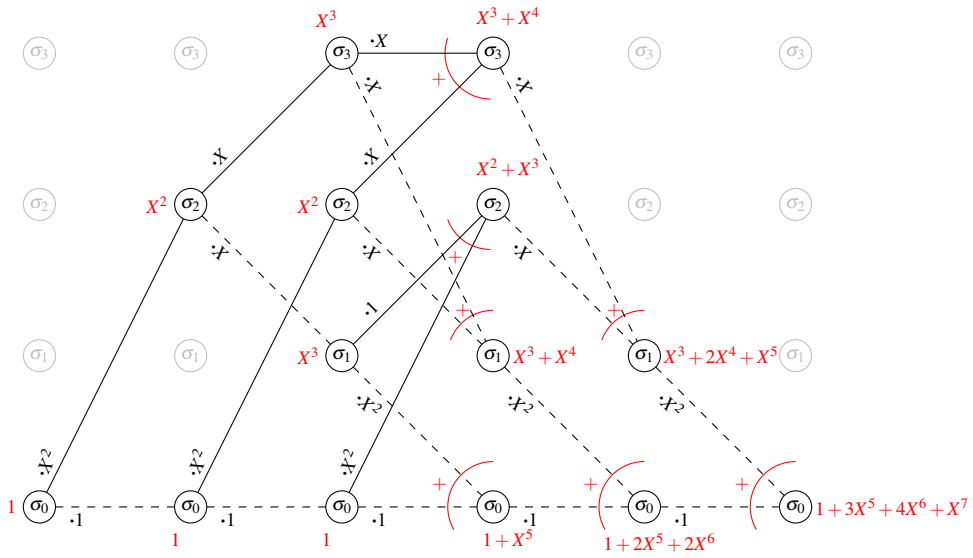


Fig. A.1 A trellis-based algorithm to compute the WEF of the CC with $\mathbf{G}(D)$=$[1+D+D^2, 1+D^2]$.

Appendix B

Numerical Results of CRC-aided List Viterbi Decoding of Convolutional Codes in CCSDS

This section presents additional results concerning the FER analysis of the poly+CC concatenation scheme applied to the CCSDS TM recommendation. The investigation employs Monte Carlo simulations to evaluate the FER for different lengths denoted as K of the uncoded TF. The specific TF lengths examined are $K \in \{1768, 3552, 8904\}$ bits, corresponding to standard CCSDS input lengths of 1784, 3568, and 8920 bits when considering the inclusion of the 16 parity bits of the polynomial code. The simulation assumes a BPSK modulated signal transmitted over an additive white Gaussian noise channel, with ideal frame and carrier synchronization at the receiver. The obtained results are presented in Figure B.1 and Figure B.2 for TF lengths of $K = 3552$ and $K = 8904$, respectively, with a fixed code rate of $R_0 = 1/2$. Moreover, Figure B.3 and Figure B.4 depict the results for a fixed TF length of $K = 1768$, while employing code rates of $R_0 = 3/4$ and $R_0 = 5/6$, respectively.

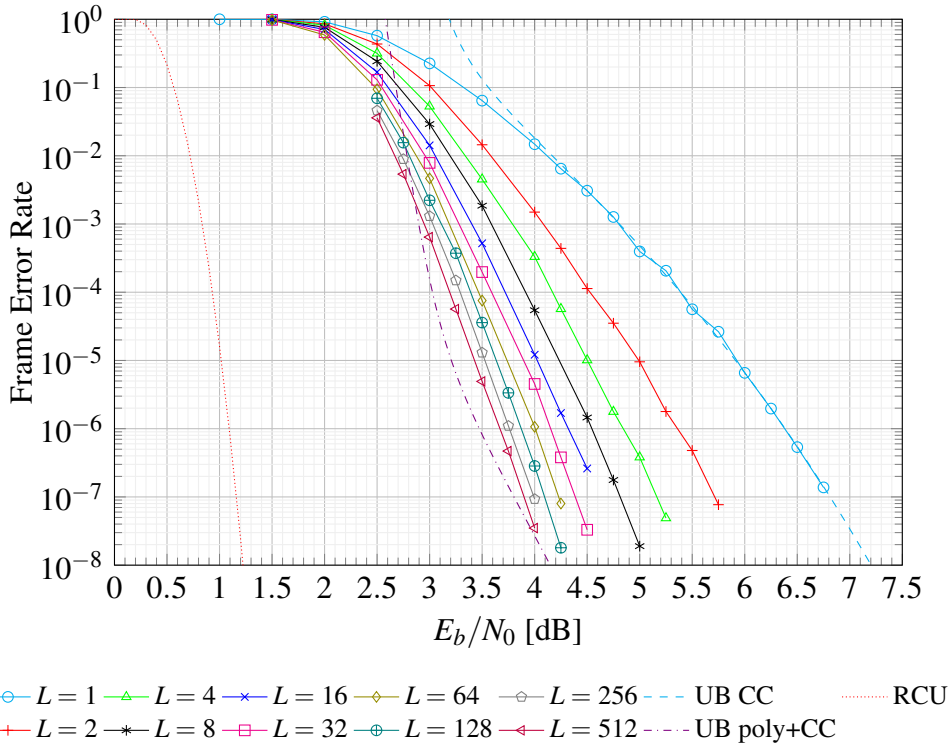


Fig. B.1 Performance comparison of the CCSDS poly+CC using BPSK modulation over an AWGN channel. The graph showcases the FER as a function of the SNR. The evaluation focuses on a TF of length $K = 3552$ bits, employing a CC encoder with a code rate of $R_0 = 1/2$.

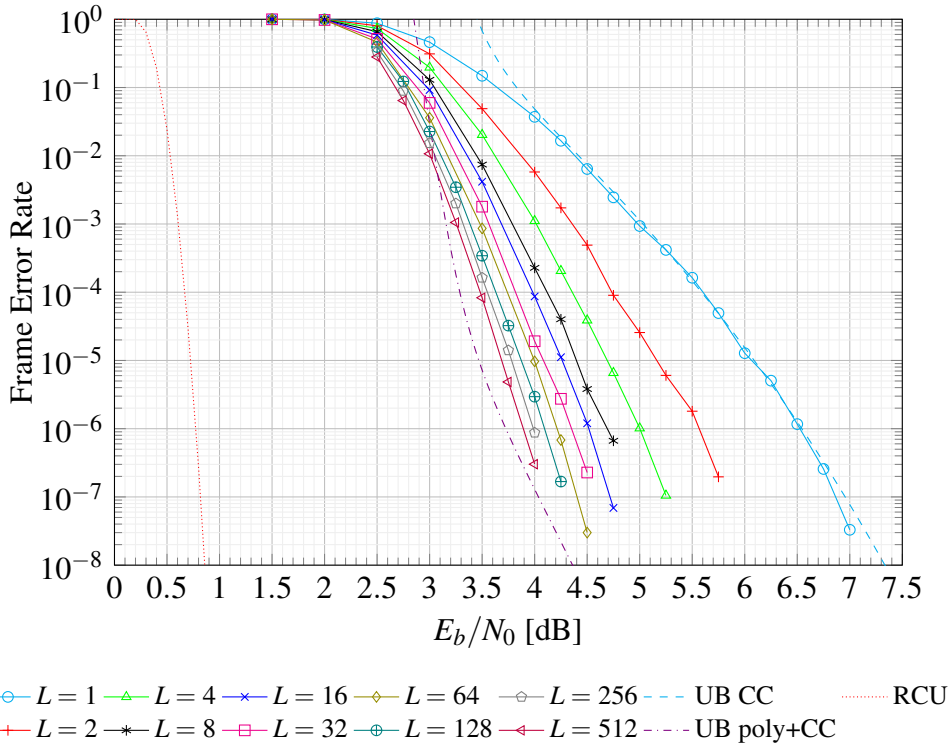


Fig. B.2 Performance comparison of the CCSDS poly+CC using BPSK modulation over an AWGN channel. The graph showcases the FER as a function of the SNR. The evaluation focuses on a TF of length $K = 8904$ bits, employing a CC encoder with a code rate of $R_0 = 1/2$.

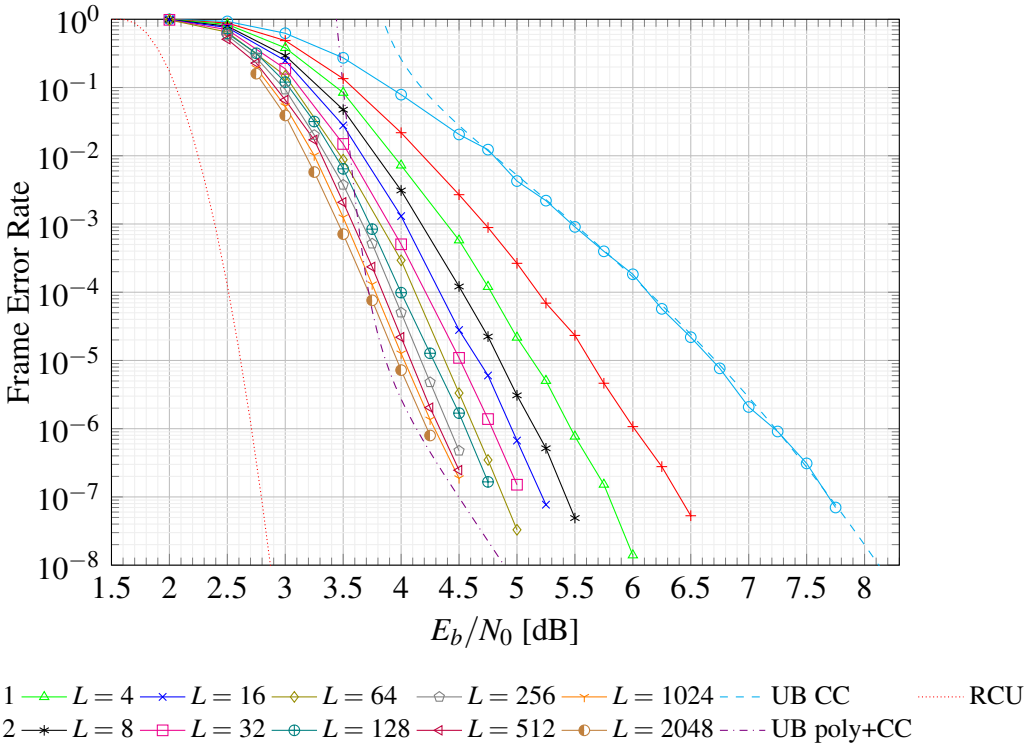


Fig. B.3 Performance comparison of the CCSDS poly+CC using BPSK modulation over an AWGN channel. The graph showcases the FER as a function of the SNR. The evaluation focuses on a TF of length $K = 1768$ bits, employing a CC encoder with a code rate of $R_0 = 3/4$.

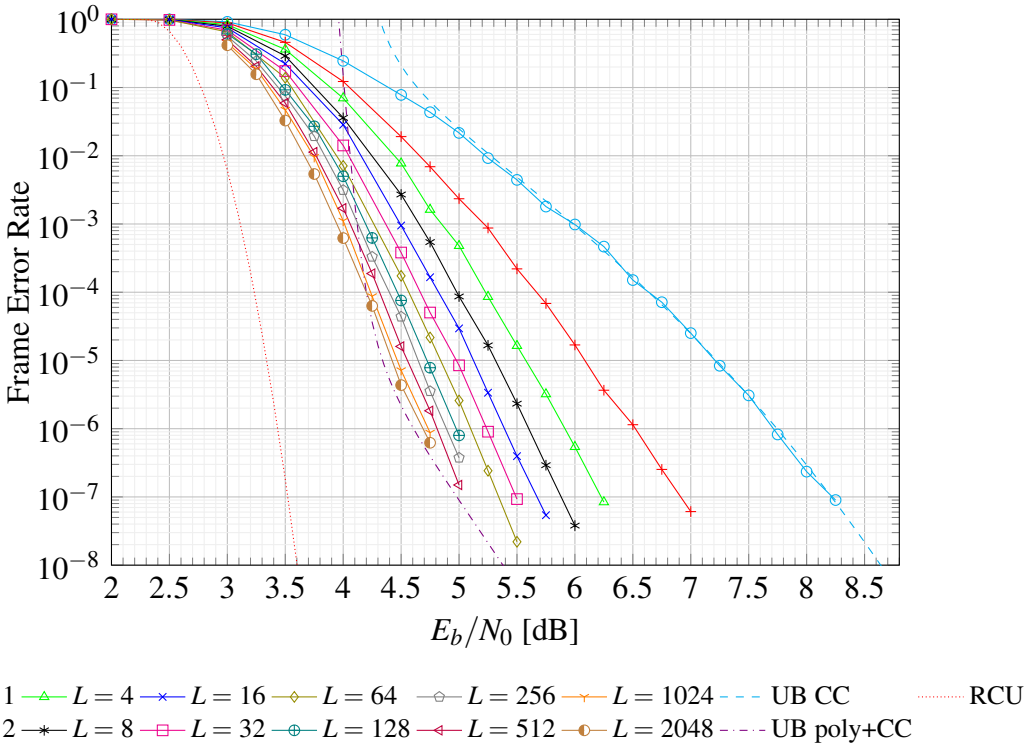


Fig. B.4 Performance comparison of the CCSDS poly+CC using BPSK modulation over an AWGN channel. The graph showcases the FER as a function of the SNR. The evaluation focuses on a TF of length $K = 1768$ bits, employing a CC encoder with a code rate of $R_0 = 5/6$.

Appendix C

Sum of Independent Random Variables

The study of optimum and simplified tests involves the study of the sum of continuous r.v.s. When these r.v.s are independent and the distribution of the sum is not known in closed form, numerical techniques can help in the computation of tight approximations or, even, the exact distributions. It is denoted by A a r.v. obtained as the sum of N independent, not necessarily identically distributed, r.v.s A_i,

$$A = \sum_{i=0}^{N-1} A_i.$$

The probability density function (p.d.f.) of A, denoted by $f_A(a)$, is given by the convolution of $A_0, A_1, \ldots, A_{N-1}$, i.e.,

$$f_A(a) = \circledast_{i=0}^{N-1} f_{A_i}(a_i),$$

where $\circledast_{i=0}^{N-1} b_i = b_0 \circledast b_1 \circledast \ldots \circledast b_{N-1}$, and $\circledast$ stands for convolution. Defining the characteristic function of a random variable as

$$\Phi_A(\nu) \triangleq \int_{-\infty}^{\infty} f_A(a) e^{j2\pi\nu a} \mathrm{d}a$$

following the properties of the Fourier transform,

$$f_A(a) = \mathcal{F}\left\{ \prod_{i=0}^{N-1} \Phi_{A_i}(\nu) \right\},$$

where $\mathcal{F}\{\cdot\}$ indicates the Fourier transform, while the cumulative distribution function (c.d.f.) of A can be obtained as

$$F_A(a) = \int_{-\infty}^{\infty} \prod_{i=0}^{N-1} \Phi_{A_i}(\nu) \left[\frac{1 - e^{-j2\pi\lambda_0\nu}}{j2\pi\nu} \right] \mathrm{d}\nu$$

(see [102]).

An alternative approach to evaluate the p.d.f. and c.d.f. of A relies on the use of the saddlepoint approximation [114, 115], which is a very tight approximation of the distribution of the sum of independent r.v.s. It requires the computation of the cumulant generating function of the random variables involved in the summation, which can also be done via numerical integration when a closed form is not available. The saddlepoint method yields the approximations

$$\hat{f}_A(a) = \frac{e^{K_A(\hat{s})-\hat{s}a}}{\sqrt{2\pi K_A''(\hat{s})}}, \tag{C.1}$$

$$\hat{F}_A(a) = \begin{cases} \frac{1}{2}\operatorname{erfc}\left(-\frac{\hat{w}}{\sqrt{2}}\right) + \phi(\hat{w})\left(\frac{1}{\hat{w}} - \frac{1}{\hat{u}}\right) & \text{for } a \neq \mathsf{E}[A] \\ \frac{1}{2} + \frac{K'''(0)}{6\sqrt{2\pi(K''(0))^3}} & \text{for } a = \mathsf{E}[A] \end{cases} \tag{C.2}$$

where $K_A(t) \triangleq \log \mathsf{E}\left[e^{tA}\right]$ is the cumulant generating function of the random variable A and $K_A'(t)$ and $K_A''(t)$ are its first and second derivatives, respectively, whereas $\hat{s}$ is the solution of $(K_A'(\hat{s}) = a)$, $\hat{w} = \operatorname{sgn}\left(\hat{s}\sqrt{2(\hat{s}x - K(\hat{s}))}\right)$, $\hat{u} = \hat{s}\sqrt{K''(\hat{s})}$, and $\phi(\hat{w})$ is the p.d.f. of a normal distribution, respectively. The approximation can be refined by adding other higher order terms [114]. Note that, due to the independence of the r.v.s A_i, it follows that

$$\log \mathsf{E}\left[e^{tA}\right] = \sum_{i=0}^{N-1} \log \mathsf{E}\left[e^{tA_i}\right].$$

Appendix D

Proof of Lemma 5.1

Consider a random variable $Z \sim \mathcal{N}\left(\mu, \sigma^2\right)$ and a random variable $\Xi = \log\cosh\left(Z\right) = g(Z)$, where

$$g(a) = \begin{cases} g_0(a) & \text{if } a \geq 0 \\ g_0(-a) & \text{if } a < 0, \end{cases} \tag{D.1}$$

with

$$g_0(a) = \begin{cases} \log\cosh(a) & \text{if } a \geq 0 \\ 0 & \text{if } a < 0. \end{cases} \tag{D.2}$$

We can write the c.d.f. of Ξ according to

$$\begin{aligned} F_\Xi(\xi) &= P\left(\Xi \leq \xi\right) \\ &= P\left(g(Z) \leq \xi\right) \\ &= P\left(g_0(Z) \leq \xi\right) + P\left(g_0(-Z) \geq -\xi\right) \\ &= P\left(-g_0^{-1}(\xi) \leq z \leq g_0^{-1}(\xi)\right) \\ &= F_Z(g_0^{-1}(\xi)) - F_Z(-g_0^{-1}(\xi)). \end{aligned} \tag{D.3}$$

Using

$$g_0^{-1}(\xi) \begin{cases} \cosh^{-1}(e^\xi) & \text{if } \xi \geq 0 \\ 0 & \text{if } \xi < 0, \end{cases}$$

in (D.3) we obtain (5.27). In order to obtain its p.d.f. we need to differentiate (D.3) which gives

$$\begin{aligned} f_{\Xi}(\xi) &= \frac{\mathrm{d}}{\mathrm{d}\xi} F_Z(g_0^{-1}(\xi)) - \frac{\mathrm{d}}{\mathrm{d}\xi} F_Z(-g_0^{-1}(\xi)) \\ &= f_Z(g_0^{-1}(\xi)) \frac{\mathrm{d}}{\mathrm{d}\xi} g_0^{-1}(\xi) + f_Z(-g_0^{-1}(\xi)) \frac{\mathrm{d}}{\mathrm{d}\xi} g_0^{-1}(\xi) \\ &= \left[f_Z(g_0^{-1}(\xi)) + f_Z(-g_0^{-1}(\xi)) \right] \frac{\mathrm{d}}{\mathrm{d}\xi} g_0^{-1}(\xi). \end{aligned} \tag{D.4}$$

Using

$$\frac{\mathrm{d}}{\mathrm{d}a} \cosh^{-1}(a) = \frac{1}{\sqrt{a^2 - 1}}$$

we obtain (5.26).