UNIVERSITY OF CALIFORNIA SAN DIEGO

Coding for Distributed Storage and Flash Memories

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Electrical Engineering
(Communication Theory and Systems)

by

Pengfei Huang

Committee in charge:

Professor Paul H. Siegel, Chair
Professor Laurence B. Milstein
Professor Bhaskar D. Rao
Professor Steven J. Swanson
Professor Alexander Vardy

2018

The dissertation of Pengfei Huang is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

_____

_____

_____

_____

_____

Chair

University of California San Diego

2018

DEDICATION

To my parents

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to thank many people for their contributions to my research work during my graduate studies at UCSD.

First, I would like to express my deepest gratitude to my advisor Prof. Paul H. Siegel, for his guidance, encouragement, and kindness. I joined his research group after taking a series of fascinating courses on coding theory taught by him. I was deeply impressed by his immense knowledge and his passion for teaching and research. It is very fortunate that I can have Prof. Siegel as my advisor. He provided numerous insightful suggestions on solving research problems as well as writing and presenting research papers. More importantly, he gave me enough freedom to explore various interesting research topics. Sharing research ideas with him was always full of fun. Prof. Siegel is admired for his brilliance and kindness. I would like to thank him for inviting me to his family Thanksgiving gatherings. His kindness and hospitality truly made me feel at home.

I would like to thank Prof. Laurence Milstein for giving me valuable advice on research topics and serving on my committee. He was the first professor I met after I arrived at UCSD. I am very grateful to him for encouraging me to explore different research directions and talk with many other professors at the beginning of this journey. I want to thank other committee members: Prof. Bhaskar Rao, Prof. Steven Swanson, and Prof. Alexander Vardy, for their time and effort in serving on my committee. Their insightful comments and questions helped improve my work significantly.

I would like to express my sincere gratitude to Prof. Eitan Yaakobi, who led me to the research area of algebraic coding. It was a tremendous pleasure to discuss exciting research problems with Eitan. He gave me many constructive suggestions on research, and collaborating with him was quite fruitful. I am particularly grateful to Eitan for inviting me to visit Technion for three months and for his hospitality during my stay. It was a wonderful experience studying and living at Technion in Haifa. I met new friends, attended several academic and industrial conferences, and visited many amazing places of interest. I would also like to thank Eitan for inviting me to celebrate the traditional Israel holiday with his family.

I am thankful for my mentors Xiaojie Zhang and Ronnie Huang during my internship at CNEX Labs. They provided me with a very comfortable work environment and gave me an opportunity to work

on exciting projects on error-correcting codes for flash memories.

I am grateful to my research collaborators: Erich F. Haratsch, Yi Liu, Hironori Uchikawa, Eitan Yaakobi, and Xiaojie Zhang. It has been a great experience working with them.

I would like to thank the CMRR staff: Jonathan Chae, Ray Descoteaux, Julie Matsuda, Sally Park, Marina Robenko, Gabby Tshamjyan, and Iris Villanueva for their kind help in administrative and technical issues and for providing me with a comfortable and pleasant place for stay and research.

I would like to thank the current and previous STAR members: Aman Bhatia, Sarit Buzaglo, Bing Fan, Seyhan Karakulak, Scott Kayser, Lingjun Kong, Andreas Lenz, Yonglong Li, Yi Liu, Minghai Qin, Veeresh Taranalli, Osamu Torii, Karthik Tunuguntla, Hironori Uchikawa, Wei Wu, Eitan Yaakobi, and Xiaojie Zhang. I have benefited greatly from enlightening conversations with them. Their kind help made my journey easier and more enjoyable. Especially, I want to thank Yi Liu. We shared the same office room and discussed many interesting problems in various areas. I would also like to thank all my friends at UCSD. With their company, my life at San Diego was much more colorful.

I would like to thank my parents for their unconditional love, support, and encouragement. This dissertation is dedicated to them.

This research was supported in part by the NSF under Grants CCF-1116739, CCF-1405119, and CCF-1619053, BSF Grant 2015816, Seagate Technology, Western Digital Corporation, and the Center for Memory and Recording Research at the University of California San Diego.

Chapter 2 is in part a reprint of the material in the paper: Pengfei Huang, Eitan Yaakobi, Hironori Uchikawa, and Paul H. Siegel, "Cyclic linear binary locally repairable codes," in *Proc. IEEE Information Theory Workshop (ITW)*, Jerusalem, Israel, Apr.–May 2015, pp. 1–5. The dissertation author was the primary investigator and author of this paper.

Chapter 3 is in part a reprint of the material in the paper: Pengfei Huang, Eitan Yaakobi, Hironori Uchikawa, and Paul H. Siegel, "Binary linear locally repairable codes," *IEEE Transactions on Information Theory*, vol. 62, no. 11, pp. 6268–6283, Nov. 2016. The dissertation author was the primary investigator and author of this paper.

Chapter 4 is in part a reprint of the material in the paper: Pengfei Huang, Eitan Yaakobi, and Paul H. Siegel, "Multi-erasure locally recoverable codes over small fields," in *Proc. 55th Annual Allerton*

Conference on Communication, Control, and Computing (Allerton), Monticello, IL, USA, Oct. 2017, pp. 1123–1130. The dissertation author was the primary investigator and author of this paper.

Chapter 5 is in part a reprint of the material in the paper: Pengfei Huang, Yi Liu, Xiaojie Zhang, Paul H. Siegel, and Erich F. Haratsch, "Syndrome-coupled rate-compatible error-correcting codes," in *Proc. IEEE Information Theory Workshop (ITW)*, Kaohsiung, Taiwan, Nov. 2017, pp. 454–458. The dissertation author was the primary investigator and author of this paper.

Chapter 6 is in part a reprint of the material in the paper: Pengfei Huang, Eitan Yaakobi, and Paul H. Siegel, "Ladder codes: A class of error-correcting codes with multi-level shared redundancy," to appear in *Proc. IEEE International Conference on Communications (ICC)*, Kansas City, MO, USA, May 2018. The dissertation author was the primary investigator and author of this paper.

Chapter 7 is in part a reprint of the material in the paper: Pengfei Huang, Paul H. Siegel, and Eitan Yaakobi, "Performance of multilevel flash memories with different binary labelings: A multi-user perspective," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 9, pp. 2336–2353, Sept. 2016. The dissertation author was the primary investigator and author of this paper.

VITA

2010	Bachelor of Engineering, Zhejiang University, China
2013	Master of Science in Engineering (Information and Communication Engineering), Shanghai Jiao Tong University, China
2018	Doctor of Philosophy in Electrical Engineering (Communication Theory and Systems), University of California San Diego

PUBLICATIONS

Pengfei Huang, Eitan Yaakobi, and Paul H. Siegel, "Ladder codes: A class of error-correcting codes with multi-level shared redundancy," to appear in *Proc. IEEE International Conference on Communications (ICC)*, Kansas City, MO, USA, May 2018.

Pengfei Huang, Yi Liu, Xiaojie Zhang, Paul H. Siegel, and Erich F. Haratsch, "Syndrome-coupled rate-compatible error-correcting codes," in *Proc. IEEE Information Theory Workshop (ITW)*, Kaohsiung, Taiwan, Nov. 2017, pp. 454–458.

Pengfei Huang, Eitan Yaakobi, and Paul H. Siegel, "Multi-erasure locally recoverable codes over small fields," in *Proc. 55th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, Monticello, IL, USA, Oct. 2017, pp. 1123–1130.

Yi Liu, Pengfei Huang, and Paul H. Siegel, "Performance of optimal data shaping codes," in *Proc. IEEE International Symposium on Information Theory (ISIT)*, Aachen, Germany, June 2017, pp. 1003–1007.

Pengfei Huang, Eitan Yaakobi, Hironori Uchikawa, and Paul H. Siegel, "Binary linear locally repairable codes," *IEEE Transactions on Information Theory*, vol. 62, no. 11, pp. 6268–6283, Nov. 2016.

Pengfei Huang, Paul H. Siegel, and Eitan Yaakobi, "Performance of multilevel flash memories with different binary labelings: A multi-user perspective," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 9, pp. 2336–2353, Sept. 2016.

Pengfei Huang, Paul H. Siegel, and Eitan Yaakobi, "Performance of flash memories with different binary labelings: A multi-user perspective," in *Proc. IEEE International Symposium on Information Theory (ISIT)*, Barcelona, Spain, July 2016, pp. 955–959.

Pengfei Huang, Eitan Yaakobi, Hironori Uchikawa, and Paul H. Siegel, "Linear locally repairable codes with availability," in *Proc. IEEE International Symposium on Information Theory (ISIT)*, Hong Kong, China, June 2015, pp. 1871–1875.

Pengfei Huang, Eitan Yaakobi, Hironori Uchikawa, and Paul H. Siegel, "Cyclic linear binary locally repairable codes," in *Proc. IEEE Information Theory Workshop (ITW)*, Jerusalem, Israel, Apr.–May 2015, pp. 1–5.

Xudong Wang, Aimin Tang, and Pengfei Huang, "Full duplex random access for multi-user OFDMA communication systems," *Elsevier Ad Hoc Networks*, vol. 24, pp. 200–213, Jan. 2015.

Xudong Wang, Pengfei Huang, Jiang Xie, and Mian Li, "OFDMA-based channel-width adaptation in wireless mesh networks," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 8, pp. 4039–4052, Oct. 2014.

Jun Wang, Pengfei Huang, Xudong Wang, and Yang Yang, "Cross-layer scheduling for physical layer secrecy and queue stability in a multi-user system," in *Proc. IEEE Global Communications Conference (GLOBECOM) Workshops*, Atlanta, GA, USA, Dec. 2013, pp. 4513–4518.

Pengfei Huang and Xudong Wang, "Fast secret key generation in static wireless networks: A virtual channel approach," in *Proc. IEEE International Conference on Computer Communications (INFOCOM)*, Turin, Italy, Apr. 2013, pp. 2292–2300.

Pengfei Huang and Xudong Wang, "Secrecy enhancement with artificial noise in decentralized wireless networks: A stochastic geometry perspective," in *Proc. IEEE Wireless Communications and Networking Conference (WCNC)*, Shanghai, China, Apr. 2013, pp. 935–940.

ABSTRACT OF THE DISSERTATION

Coding for Distributed Storage and Flash Memories

by

Pengfei Huang

Doctor of Philosophy in Electrical Engineering
(Communication Theory and Systems)

University of California San Diego, 2018

Professor Paul H. Siegel, Chair

A modern large-scale storage system usually consists of a number of distributed storage nodes, each of which is made up of many storage devices, like flash memory chips. To maintain the data integrity in the system, two independent layers of data protection mechanisms are deployed. At the system level, erasure codes, e.g., maximum distance separable (MDS) codes, are used across a set of storage nodes. At the device level, error-correcting codes (ECCs), e.g., Bose-Chaudhuri-Hocquenghem (BCH) codes, are employed in each flash memory chip. The main research goal of this dissertation is to design new erasure codes for distributed storage and new ECCs for flash memories.

The first part of this dissertation is devoted to studying a new class of erasure codes called locally repairable codes (LRCs) for distributed storage. We focus on LRCs over small fields; in particular, the

binary field. We investigate the locality of classical binary linear codes, e.g., BCH codes and Reed-Muller codes, and their modified versions. Then, we derive bounds for LRCs with availability and present several new code constructions for binary LRCs. In addition, we study erasure codes that can locally correct multiple erasures. Such codes are referred to as multi-erasure locally repairable codes (ME-LRCs). Our constructions based on generalized tensor product codes generate several families of optimal ME-LRCs over small fields.

The second part of this dissertation aims to construct new ECCs and analyze the fundamental performance limits for flash memories. We propose a general framework for constructing rate-compatible ECCs which are capable of adapting different error-correcting capabilities to the corresponding bit error rates at different program/erase (P/E) cycles. Next, we present a new family of shared-redundancy ECCs called ladder codes. Using ladder codes, multiple codewords from good and bad pages in a flash memory block can share some common redundancy. Finally, based on the channel models obtained from empirical data, the performance of multilevel flash memories is studied by using multi-user information theory. The results provide qualitative insight into effective coding solutions.

Chapter 1

Introduction

A modern large-scale storage system consists of a number of distributed storage nodes, each of which is made up of many storage devices, like traditional magnetic hard disk drives or recent flash memory chips. To maintain the data integrity in the system, two independent layers of data protection mechanisms are deployed. At the system level, erasure codes, such as repetition codes and more generally maximum distance separable (MDS) codes, are used across a collection of storage nodes. At the device level, error-correcting codes (ECCs), such as Bose-Chaudhuri-Hocquenghem (BCH) codes and low-density parity-check (LDPC) codes, are employed in each storage device.

Information theory [19, 23, 70] and coding theory [34, 47, 49, 63, 64, 66] are two effective tools to analyze and design wireless communication and data storage systems. This dissertation is focused on studying modern data storage by using these powerful theories. The research goal of this dissertation can be divided into two parts: for the higher system level, we aim to design new erasure codes for distributed storage; with respect to the lower device level, we will construct new ECCs and analyze the fundamental performance limits for flash memories.

1.1 Distributed Storage

Modern large-scale distributed storage systems, such as data centers and peer-to-peer storage systems, are required to tolerate the failure or unavailability of some of the nodes in the system. The

simplest and most commonly used way to accomplish this task is replication, where every node is replicated several times, usually three. This solution has clear advantages due to its simplicity and fast recovery from node failures. However, it entails a large storage overhead which becomes costly in large storage systems.

In order to achieve better storage efficiency, erasure codes, e.g., Reed-Solomon codes, are deployed. Reed-Solomon and more generally MDS codes are attractive since they tolerate the maximum number of node failures for a given redundancy. For example, the $[14, 10]$ Reed-Solomon code used by Facebook has only 40% storage overhead which is much smaller than the 200% overhead associated with the three-replication scheme [67]. However, they suffer from a very slow recovery process, in the case of a single node failure, which is the most common failure scenario. Hence, an important objective in the design of erasure codes is to ensure fast recovery while efficiently supporting a large number of node failures. There are several metrics in the literature to quantify the efficiency of rebuilding failed nodes. Three of the most popular consider the number of communicated bits in the network, the number of read bits, and the number of accessed nodes. In this dissertation, we study erasure codes with respect to the last metric.

Locally repairable codes (LRCs) are a class of erasure codes in which a code symbol can be recovered by accessing at most r other symbols, where r is a predetermined value [30, 55, 75]. More specifically, consider a code of length n, with dimension k. A code symbol has locality r if it can be reconstructed by accessing at most r other symbols in the code. It is said that the code has *all-symbol locality r* if every symbol is recoverable from a set of at most r symbols. If the code is systematic and only its information symbols have this property, then we say that the code has *information locality r*.

Codes with small locality were initially studied in [35, 38, 53]. In [30], Gopalan et al. formally introduced the interesting notion of locality of a code symbol. The trade-off between code minimum distance and information locality was investigated, and a Singleton-like upper bound on the code minimum distance was derived. Following [30], there exist many works on bounds and constructions for LRCs [13, 31, 55, 71, 73, 75, 77, 81, 92]. Furthermore, some erasure codes with small locality were implemented in data storage systems, such as Windows Azure Storage [39] and Facebook clusters [67].

LRCs have been generalized in several directions so far. Besides locality, another important property of LRCs is their symbol availability, meaning the number of disjoint sets of symbols that can be used to recover a given symbol. High availability is a particularly attractive property for so-called

hot data in distributed storage. Bounds and code constructions for LRCs with availability have been studied in [6, 7, 31, 54, 60, 74, 75, 82, 83, 92]. In addition, erasure codes which can locally repair multiple erasures have received considerable attention since simultaneous node failures are becoming common, in view of the increasing trend towards replacing expensive servers with low-cost commodity servers in data centers [7, 10, 11, 17, 27, 58, 59]. In this dissertation, we develop new bounds and constructions for LRCs and their generalizations over small fields.

1.2 Flash Memory

NAND flash memory is a versatile non-volatile data storage medium, and has been widely used in consumer electronics as well as enterprise data centers. It has many advantages over traditional magnetic recording, e.g., higher read throughput and less power consumption [8, 14]. The basic storage unit in a NAND flash memory is a floating-gate transistor referred to as a cell. The voltage levels of a cell can be adjusted by a program operation and are used to represent the stored data. The cells typically have 2, 4, and 8 voltage levels (1, 2, and 3 bits/cell, respectively) and are referred to as single-level cell (SLC), multi-level cell (MLC), and three-level cell (TLC), respectively. Cells are organized into a rectangular array, interconnected by horizontal *wordlines* and vertical *bitlines*, that constitute a *block*. A flash memory chip comprises a collection of such blocks. During program (i.e., write) operations, the voltage level of a cell cannot be decreased. In order to do so, the entire containing block must be erased and reprogrammed. Repeated program/erase (P/E) operations induce wear on the cells in the block, with a detrimental effect on the lifetime of the memory.

In an MLC flash memory, the two bits in the cells connected along a wordline are assigned to two separate *pages*, which represent the basic unit for program and read operations. The most significant bit (MSB) is assigned to the *lower page* while the least significant bit (LSB) is assigned to the *upper page*. We denote the four nominal voltage levels in the MLC as A_0, A_1, A_2, and A_3, in order of increasing voltage. The program operation is not perfect, so the actual cell levels are distributed around the nominal levels, as depicted in Figure 1.1. A particular mapping of 2-bit patterns, namely '11', '10', '00', and '01', to the voltage levels, is also shown in the figure. We refer to this mapping as the *Gray* labeling.

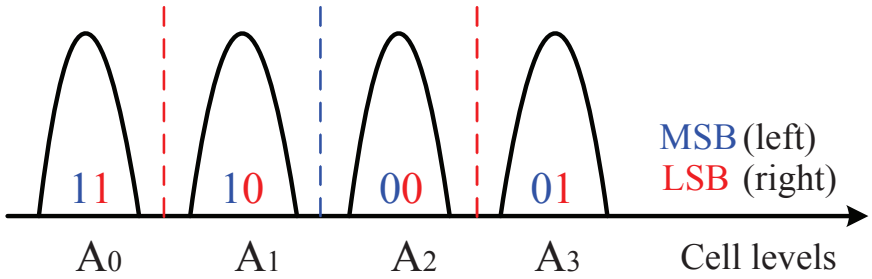


Figure 1.1: The four voltage levels and Gray labeling for a cell in MLC flash memories. A total of three reads are employed for decoding two pages.

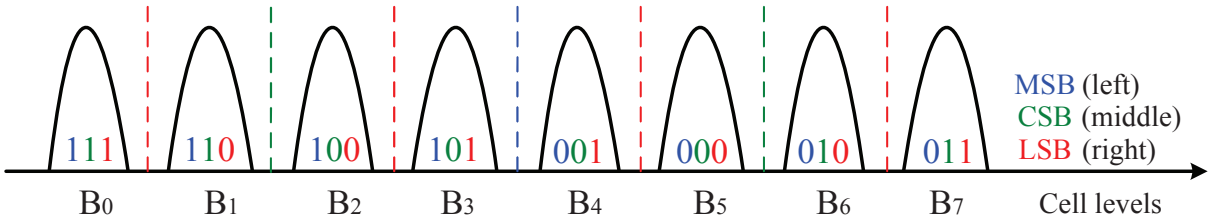


Figure 1.2: The eight voltage levels and Gray labeling for a cell in TLC flash memories. A total of seven reads are employed for decoding three pages.

Similarly, the three bits belonging to a TLC are separately mapped to three pages. We refer to the first bit as the most significant bit (MSB), the second bit as the center significant bit (CSB), and the third bit as the least significant bit (LSB). The corresponding pages are referred to as the *lower page*, *center page*, and *upper page*, respectively. Figure 1.2 depicts the programmed cell level distributions around the eight nominal TLC voltage levels, denoted by B_0, B_1, B_2, B_3, B_4, B_5, B_6, and B_7, along with the corresponding Gray labeling, '111', '110', '100', '101', '001', '000', '010', and '011'.

Several different types of errors can be introduced at any point during the P/E cycling process, e.g., program errors, inter-cell interference (ICI) errors, data retention errors, and read disturb errors [14]. Error characterization of flash memories is important and has been studied extensively [15, 79, 89, 90]. It is well known that the raw bit error rate (BER) increases as the P/E cycle count grows; see [15, 79, 90] for MLC flash memories and [89] for TLC flash memories. Therefore, at a higher P/E cycle count, a stronger ECC is needed to maintain the data integrity. Rate-compatible codes are promising to satisfy this requirement, since they are capable of adapting different error-correcting capabilities to the corresponding bit error rates. It has also been observed that the bit error rates of different pages vary with respect to their locations in a block [15, 89, 90]. To enhance the reliability of all the pages, instead of deploying strong

ECCs for all these pages, a more storage-efficient way is to employ relatively weaker ECCs in these pages and allow them to share some common redundancy. To this end, new ECCs with shared redundancy need to be designed. Moreover, many experiments have shown the asymmetry of bit errors in MLC and TLC flash memories [15, 79, 89, 90]. The threshold voltage distributions of flash memory cells were studied in [14, 16, 56]. Thanks to these flash memory channel models, the fundamental performance limits of flash memories can be analyzed by using information theory. This performance analysis gives insight into the design of practical error-correcting coding schemes.

1.3 Dissertation Overview

In this dissertation, we first study LRCs over small fields for distributed storage. Then, we propose new error-correcting coding schemes for flash memories and also analyze the fundamental performance limits of multilevel flash memories from a multi-user perspective. The dissertation is organized as follows.

In Chapter 2, we study the locality of classical binary linear codes. We first investigate the locality of a variety of well known binary linear cyclic codes, e.g., Hamming codes and simplex codes. Similarly, we study the locality of binary Reed-Muller codes. We then discuss the locality of codes which are obtained by applying the operations of extend, shorten, expurgate, augment, and lengthen to binary linear cyclic codes. Several families of such modified codes are considered and their optimality is addressed.

In Chapter 3, we present an upper bound on the minimum distance of LRCs with availability. Then, we construct LRCs using phantom parity-check symbols and a multi-level tensor product structure, respectively. Finally, availability of LRCs is studied. We investigate the locality and availability properties of several classes of one-step majority-logic decodable codes, including cyclic simplex codes, cyclic difference-set codes, and 4-cycle free regular LDPC codes. We also show the construction of a long LRC with availability from a short one-step majority-logic decodable code.

In Chapter 4, we study multi-erasure locally repairable codes (ME-LRCs). We first develop an upper bound on the minimum distance of ME-LRCs. We then propose a general construction of ME-LRCs based on generalized tensor product codes, and study their erasure-correcting properties. A decoding algorithm tailored for erasure recovery is given, and correctable erasure patterns are identified. Next, we

prove that our construction yields optimal ME-LRCs with a wide range of code parameters, and present some explicit ME-LRCs over small fields. Finally, we show that generalized integrated interleaving (GII) codes can be treated as a subclass of generalized tensor product codes, thus defining the exact relation between these codes.

In Chapter 5, we first study the lower bounds for rate-compatible ECCs, thus proving the existence of good rate-compatible codes. Then, we propose a general framework for constructing rate-compatible ECCs based on cosets and syndromes of a set of nested linear codes. We evaluate our construction from two points of view. From a combinatorial perspective, we show that we can construct rate-compatible codes with increasing minimum distances. From a probabilistic point of view, we prove that we are able to construct capacity-achieving rate-compatible codes. Performance of two-level rate-compatible codes is evaluated for MLC flash memories.

In Chapter 6, we propose a new class of linear error-correcting codes, called ladder codes, whose codeword structure consists of multiple codewords of certain component codes and also their shared redundancy. First, we give a general construction for ladder codes, determine the code length and dimension, and also derive a lower bound on the minimum distance. Then, we study correctable error-erasure patterns of ladder codes and give a corresponding decoding algorithm. Finally, we compare a two-level ladder code with a concatenated code, and show that the former can outperform the latter in many cases. Ladder codes have potential to be used for data protection in flash memories where only a few pages may suffer from severe errors in a block.

In Chapter 7, we study the performance of different decoding schemes for multilevel flash memories where each page in every block is encoded independently. We focus on the MLC flash memory, which is modeled as a two-user multiple-access channel suffering from asymmetric noise. The uniform rate regions and sum rates of treating interference as noise (TIN) decoding and successive cancellation (SC) decoding are investigated for a P/E cycling model and a data retention model. We examine the effect of different binary labelings of the cell levels, as well as the impact of further quantization of the memory output (i.e., additional read thresholds).

Chapter 2

Locality of Classical Binary Linear Codes

2.1 Introduction

Locally repairable codes (LRCs) are a class of codes in which a single code symbol can be recovered by accessing at most r other symbols, where r is a predetermined value [30, 55, 75]. For a length-n code with dimension k, it is said that the code has *all-symbol locality* r if every symbol is recoverable from a set of at most r symbols. If the code is systematic and only its information symbols have this property then the code has *information locality* r. LRCs are well studied in the literature and many works have considered bounds and code constructions for such codes. In [30], an upper bound, which can be seen as a modified version of the Singleton bound, was given on the minimum distance of LRCs. More specifically, if an $[n,k,d]_q$ linear code has information locality r, then

$$d \leq n - k - \left\lceil \frac{k}{r} \right\rceil + 2. \tag{2.1}$$

In [55], it was proved that the bound (2.1) holds also for non-linear codes with all-symbol locality. Code constructions which achieve bound (2.1) were given in [38, 39, 71, 75, 77]. However, for some cases, bound (2.1) is not tight, so several improvements were proposed in [59, 73, 81].

Recently, a new upper bound on the dimension k of LRCs was presented in [13]. This bound takes into account the code length, minimum distance, locality, and field size, and it is applicable to both

non-linear and linear codes. Namely, if an $(n, M, d)_q$ code has all-symbol locality r, then

$$k \leq \min_{x \in \mathbb{Z}^+} \left\{ xr + k_{opt}^{(q)}(n - x(r+1), d) \right\}, \tag{2.2}$$

where M denotes the codebook size, $k = \log_q M$, $\mathbb{Z}^+$ is the set of all positive integers, and $k_{opt}^{(q)}(n', d')$ is the largest possible dimension of a length-n' code with minimum distance d' and a given field size q. There also exist some constructions of LRCs over small fields, e.g., binary field, in [31, 72, 76, 92].

Our main goal in this chapter is to study the locality of classical binary linear codes, in particular, binary linear cyclic codes. The remainder of this chapter is organized as follows. In Section 2.2, we formally define the problem and state some preliminary results. In Section 2.3, we prove the locality of linear cyclic codes and show that such a code has locality that equals the minimum distance of its dual code minus one. We also study similar properties for Reed-Muller codes. In Section 2.4, we study the locality of codes which are obtained by the operations of extend, shorten, expurgate, augment, and lengthen. We conclude the chapter in Section 2.5.

2.2 Definitions and Preliminaries

In this section, we give the basic definitions and preliminaries that will be used in this chapter. We use the notation $[n]$ to define the set $\{1, \ldots, n\}$. For a length-n vector $\boldsymbol{v}$ and a set $\mathcal{I} \subseteq [n]$, the vector $\boldsymbol{v}_{\mathcal{I}}$ denotes the restriction of the vector $\boldsymbol{v}$ to coordinates in the set $\mathcal{I}$, and $w_H(\boldsymbol{v})$ represents the Hamming weight of the vector $\boldsymbol{v}$. A linear code $\mathcal{C}$ over $\mathbb{F}_q$ of length n, dimension k, and minimum distance d is denoted by $[n, k, d]_q$, and a non-linear code is denoted by $(n, M, d)_q$ where M is the number of codewords; the field size q may be omitted if it is clear from the context. The dual code of a linear code $\mathcal{C}$ will be denoted by $\mathcal{C}^\perp$. We use the notation $S(\boldsymbol{c})$ to denote the support of a codeword $\boldsymbol{c}$. We follow the conventional definition of locally repairable codes [55, 58, 75], which is stated as follows.

Definition 2.2.1. *The ith code symbol, $i \in [n]$, is said to have locality r if there exists a repair set R_i of size at most r, such that if it is erased then it can be recovered by reading the symbols from the set R_i. A code $\mathcal{C}$ is said to have* ***all-symbol locality*** *r if all its symbols have locality r. Similarly, a systematic code*

*$\mathcal{C}$ is said to have **information locality** r if all its information symbols have locality r.*

In many papers, a code with all-symbol locality which attains the bound (2.1) is called an *optimal* LRC. Here, we prefer a slightly different definition of optimality.

Definition 2.2.2. *An $[n,k,d]_q$ linear code $\mathcal{C}$ with locality r is said to be d-**optimal**, if there does not exist an $[n,k,d+1]_q$ code with locality r. Similarly, it is called k-**optimal** if there does not exist an $[n,k+1,d]_q$ code with locality r, and it is called r-**optimal** if there does not exist an $[n,k,d]_q$ code with locality $r-1$.*

Example 2.2.1. Consider the binary simplex code $\mathcal{C}$ with parameters $[2^m-1,m,2^{m-1}]$. It was proved in [13] that this code has all-symbol locality $r=2$ and it is r-optimal for these given parameters. Since this code satisfies the Plotkin bound, it is d-optimal and k-optimal as well. □

In the rest of this chapter, we consider only codes with all-symbol locality, and thus when saying that a code has locality r we refer to all-symbol locality.

2.3 Locality of Classical Codes

In this section, we study two classes of classical binary linear codes, namely, cyclic codes and Reed-Muller codes.

2.3.1 Locality of Cyclic Codes

First, we give our main result for cyclic codes, and also present several examples. We start with a simple observation about the locality of code symbols. Even though it has been mentioned before, see e.g., [30, 58, 59], we state and prove it here for completeness.

Claim 2.3.1. *For a binary linear code $\mathcal{C}$, if its ith coordinate, $i \in [n]$, belongs to the support of a codeword in $\mathcal{C}^\perp$ with weight $r+1$, then the ith code symbol has locality r.*

Proof. Assume that there exists a codeword $\boldsymbol{c}' \in \mathcal{C}^\perp$ such that $c'_i = 1$ and $w_H(\boldsymbol{c}') = r+1$. Let $R_i = S(\boldsymbol{c}') \setminus \{i\}$. Then for all $\boldsymbol{c} \in \mathcal{C}$, $c_i = \sum_{j \in R_i} c_j$ and from Definition 2.2.1, the ith symbol has locality r. ■

The next lemma is an immediate consequence of the preceding claim.

Lemma 2.3.2. *Let $\mathcal{C}$ be an $[n,k,d]$ binary linear cyclic code, and let $d^\perp$ be the minimum distance of its dual code $\mathcal{C}^\perp$. Then, the code $\mathcal{C}$ has locality $d^\perp - 1$.*

Proof. The dual code $\mathcal{C}^\perp$ has a codeword of weight $d^\perp$. Since $\mathcal{C}$ is a linear cyclic code, its dual code $\mathcal{C}^\perp$ is also a linear cyclic code. Thus every $i \in [n]$ belongs to the support of some codeword of weight $d^\perp$ in $\mathcal{C}^\perp$. From Claim 2.3.1, every coordinate has locality $d^\perp - 1$. Thus, the code $\mathcal{C}$ has locality $r = d^\perp - 1$. ■

Next, we give several examples to illustrate how the locality of specific codes can be determined from Lemma 2.3.2 and then study their optimality.

Example 2.3.1. Let $\mathcal{C}$ be the $[n = 2^m - 1, k = 2^m - 1 - m, d = 3]$ cyclic binary Hamming code. Its dual code is the $[2^m - 1, m, 2^{m-1}]$ cyclic binary simplex code. Therefore, the Hamming code has locality $r = 2^{m-1} - 1$. Since it is a perfect code, it is both d-optimal and k-optimal. In order to show r-optimality, let us assume on the contrary that there exists an $[n,k,d]$ code with locality $\hat{r} = 2^{m-1} - 2$. According to bound (2.2) for $x = 1$, we have that

$$\begin{aligned} k &\leq x\hat{r} + k_{opt}^{(2)}(n - x(\hat{r}+1), d) = 2^{m-1} - 2 + k_{opt}^{(2)}(2^{m-1}, 3) \\ &\overset{(a)}{<} 2^{m-1} - 2 + 2^{m-1} - (m-1) = 2^m - m - 1, \end{aligned}$$

where step (a) is from the Hamming bound. Thus, we get a contradiction to the value of k. We also get from Lemma 2.3.2 that the simplex code has locality 2. This gives an alternative proof to the one given in [13] in case the code is cyclic. □

Example 2.3.2. Here we consider the $[23,12,7]$ cyclic binary Golay code $\mathcal{C}$. Its dual code $\mathcal{C}^\perp$ is the $[23,11,8]$ cyclic binary code. Hence, we conclude that $\mathcal{C}$ has locality $r = 7$ and the dual code $\mathcal{C}^\perp$ has locality $r^\perp = 6$. The code $\mathcal{C}$ is both d-optimal and k-optimal since it is a perfect code. $\mathcal{C}^\perp$ is d-optimal due to the Hamming bound, and k-optimal according to the online table [68]. The r-optimality of these two codes is proved in a similar way to the optimality proof in Example 2.3.1. □

Example 2.3.3. Let $\mathcal{C}$ be the cyclic double-error-correcting binary primitive BCH (DBCH) code with parameters $[2^m - 1, 2^m - 1 - 2m, 5]$ where $m \geq 4$. Its dual code $\mathcal{C}^\perp$ has parameters $[2^m - 1, 2m, 2^{m-1} - 2^{\lfloor m/2 \rfloor}]$ [47]. Therefore, we conclude that $\mathcal{C}$ has locality $r = 2^{m-1} - 2^{\lfloor m/2 \rfloor} - 1$, and $\mathcal{C}^\perp$ has locality

Table 2.1: Parameters of DBCH codes and their dual codes.

$\mathcal{C}$	n	k	d	r	d-opt	k-opt	r-opt
$m=4$	15	7	5	3	✓	✓	✓
$m=5$	31	21	5	11	✓	✓	?
$m=6$	63	51	5	23	✓	✓	?
$m=7$	127	113	5	55	✓	✓	?
$m=8$	255	239	5	111	✓	✓	?
$\mathcal{C}^{\perp}$	$n^{\perp}$	$k^{\perp}$	$d^{\perp}$	$r^{\perp}$	d-opt	k-opt	r-opt
$m=4$	15	8	4	4	✓	?	?
$m=5$	31	10	12	4	✓	✓	?
$m=6$	63	12	24	4	?	?	?
$m=7$	127	14	56	4	✓	?	?
$m=8$	255	16	112	4	?	?	?

$r^{\perp}=4$. We utilize bound (2.2) and the online table from [68] to check the d-optimality, k-optimality, and r-optimality of the DBCH codes and their dual codes. The results are summarized in Table 2.1 (where ✓ indicates that we could prove optimality while ? means that we could not). □

2.3.2 Locality of Reed-Muller Codes

Reed-Muller (RM) codes form another important class of codes. They are simple to construct and rich in structural properties. This motivates us to study their locality. Recall that a μth-order binary RM code $\mathcal{RM}(\mu,m)$ has code length $n=2^m$, dimension $k=\sum_{i=0}^{\mu}\binom{m}{i}$, and minimum distance $d=2^{m-\mu}$.

In [61], two classes of codes with locality 2 and 3 were constructed based on the non-binary RM codes of first and second orders. Here, we focus on the binary RM codes of any order, and determine their locality as follows.

Lemma 2.3.3. *The μth-order binary RM code $\mathcal{RM}(\mu,m)$ has locality $r=d^{\perp}-1=2^{\mu+1}-1$.*

Proof. It is known that the dual code of $\mathcal{RM}(\mu,m)$ is $\mathcal{RM}(m-\mu-1,m)$, and the minimum weight codewords of an RM code generate all of its codewords [47]. Therefore, every coordinate i, $i\in[n]$, belongs to the support of a certain minimum weight codeword of $\mathcal{RM}(m-\mu-1,m)$. To see that, assume on the contrary that there exists a coordinate j, $j\in[n]$, in which all the minimum weight codewords of $\mathcal{RM}(m-\mu-1,m)$ have value 0. Thus, any linear combinations of the minimum weight codewords

cannot produce the all-one codeword $\mathbf{1}$, which is a valid codeword. Thus, we get a contradiction, which implies that $\mathcal{RM}(\mu,m)$ has locality $r = d^{\perp} - 1 = 2^{\mu+1} - 1$. ■

Finally, we mention that a μth-order cyclic binary RM code $\mathcal{C}$ is a $[2^m - 1, \sum_{i=0}^{\mu} \binom{m}{i}, 2^{m-\mu} - 1]$ punctured binary RM code, represented in a cyclic form [47]. Its dual code $\mathcal{C}^{\perp}$ is also cyclic and is a $[2^m - 1, \sum_{i=\mu+1}^{m} \binom{m}{i} - 1, 2^{\mu+1}]$ binary code. From Lemma 2.3.2, $\mathcal{C}$ has locality $r = 2^{\mu+1} - 1$, and $\mathcal{C}^{\perp}$ has locality $r^{\perp} = 2^{m-\mu} - 2$.

2.4 Locality of Modified Classical Codes

In this section, we show how to find the locality of codes which are obtained by applying the standard code operations of extending, shortening, expurgating, augmenting, and lengthening to existing LRCs. For a binary vector $\boldsymbol{c}$, let $\overline{\boldsymbol{c}}$ represent the complement vector of $\boldsymbol{c}$. For a binary code $\mathcal{C}$, define $\overline{\mathcal{C}} = \{\overline{\boldsymbol{c}} \,:\, \boldsymbol{c} \in \mathcal{C}\}$.

2.4.1 Extend Operation

The extended code of an $[n,k,d]$ binary code $\mathcal{C}$ is an $[n+1,k,d_{ext}]$ code $\mathcal{C}_{ext}$ with an overall parity bit added to each codeword,

$$\mathcal{C}_{ext} = \left\{ (c_1,\ldots,c_n,c_{n+1}) : (c_1,\ldots,c_n) \in \mathcal{C}, c_{n+1} = \sum_{i=1}^{n} c_i \right\},$$

where $d_{ext} = d+1$ for odd d and $d_{ext} = d$ for even d. In the following, we use the notation $\mathcal{C}_{ext}^{\perp}$ to denote the dual code of $\mathcal{C}_{ext}$.

Lemma 2.4.1. *Let $\mathcal{C}$ be an $[n,k,d]$ binary code with locality r. If the maximum Hamming weight of codewords in $\mathcal{C}^{\perp}$ is $n-r$, then the extended code $\mathcal{C}_{ext}$ has locality $r_{ext} = r$.*

Proof. For every $i \in [n]$, there exists a set R_i of size at most r such that the ith symbol is recoverable from the set R_i. Thus, we only need to prove this property for the $(n+1)$st symbol. Since the maximum weight of codewords in $\mathcal{C}^{\perp}$ is $n-r$, there exists a codeword $\boldsymbol{c} \in \mathcal{C}^{\perp}$ such that $w_H(\boldsymbol{c}) = n-r$. Note also that the vectors $(\boldsymbol{c},0)$ and $\mathbf{1}$ are codewords in $\mathcal{C}_{ext}^{\perp}$. Therefore the vector $\boldsymbol{c}' = (\boldsymbol{c},0) + \mathbf{1}$ is a codeword in $\mathcal{C}_{ext}^{\perp}$

and its Hamming weight is $r+1$. Hence, from Claim 2.3.1, we get that the $(n+1)$st symbol can also be recovered by a set of r other symbols. ■

We have the following corollary for binary linear cyclic codes, for which we have already seen that $r = d^{\perp} - 1$.

Corollary 2.4.2. *Let $\mathcal{C}$ be an $[n,k,d]$ binary cyclic code and let $d^{\perp}$ be the minimum distance of its dual code. If the maximum Hamming weight of codewords in $\mathcal{C}^{\perp}$ is $n+1-d^{\perp}$, then the extended code $\mathcal{C}_{ext}$ has locality $r_{ext} = d^{\perp} - 1$.*

Example 2.4.1. Let $\mathcal{C}$ be the $[2^m-1, 2^m-1-m, 3]$ cyclic binary Hamming code. Its extended code $\mathcal{C}_{ext}$ has parameters $[2^m, 2^m-1-m, 4]$. The dual code $\mathcal{C}^{\perp}$ is the simplex code, whose nonzero codewords have constant Hamming weight 2^{m-1}. Hence, the condition from Corollary 2.4.2 holds and we conclude that the extended Hamming code $\mathcal{C}_{ext}$ has locality $r_{ext} = d^{\perp} - 1 = 2^{m-1} - 1$. $\mathcal{C}_{ext}$ is both d-optimal and k-optimal according to the Hamming bound. To show that it is also r-optimal, let us assume on the contrary that there exists a $[2^m, 2^m-1-m, 4]$ binary code with locality $\hat{r} = 2^{m-1} - 2$. According to bound (2.2) for $x = 1$, we have

$$\begin{aligned} k_{ext} \leq & 2^{m-1} - 2 + k_{opt}^{(2)}(2^{m-1}+1, 4) \overset{(a)}{=} 2^{m-1} - 2 + k_{opt}^{(2)}(2^{m-1}, 3) \\ \overset{(b)}{<} & 2^{m-1} - 2 + 2^{m-1} - (m-1) = 2^m - m - 1. \end{aligned}$$

Thus, we get a contradiction to the value of k_{ext}. In the above proof, step (a) follows from the property that $A(n, 2s-1) = A(n+1, 2s)$, where $A(n,d)$ denotes the largest number of codewords M in any binary code (n, M, d) [49]. Step (b) follows from the Hamming bound. □

Next, we determine the locality of the dual of the extension of a cyclic code.

Lemma 2.4.3. *Let $\mathcal{C}$ be an $[n,k,d]$ binary cyclic code with odd minimum distance d. Then, the code $\mathcal{C}_{ext}^{\perp}$ has locality $r_{ext}^{\perp} = d$.*

Proof. Since d is odd, each codeword with weight d in $\mathcal{C}$ generates a parity-check bit 1. Since $\mathcal{C}$ is cyclic, for any $i \in [n]$, i belongs to the support of some codeword $(\boldsymbol{c}, 1) \in \mathcal{C}_{ext}$, where $\boldsymbol{c}$ has weight d. Moreover,

the support of $(\boldsymbol{c},1)$ also contains coordinate $n+1$. Thus, from Claim 2.3.1, every symbol of $\mathcal{C}_{ext}^{\perp}$ has locality d. ■

Example 2.4.2. Let $\mathcal{C}$ be the $[n=2^m-1, k=2^m-1-m, d=3]$ cyclic binary Hamming code. Correspondingly, $\mathcal{C}_{ext}^{\perp}$ is the biorthogonal code $[n_{ext}^{\perp}=2^m, k_{ext}^{\perp}=m+1, d_{ext}^{\perp}=2^{m-1}]$ [41]. From Lemma 2.4.3, $\mathcal{C}_{ext}^{\perp}$ has locality $r_{ext}^{\perp}=d=3$. $\mathcal{C}_{ext}^{\perp}$ is both d-optimal and k-optimal according to the Plotkin bound. To show that $\mathcal{C}_{ext}^{\perp}$ is r-optimal, we utilize bound (2.2) with $x=1$, and have the following constraint on the dimension of the code,

$$\begin{aligned} k_{ext}^{\perp} = m+1 &\leq r_{ext}^{\perp} + k_{opt}^{(2)}(2^m-(r_{ext}^{\perp}+1), 2^{m-1}) \\ &\overset{(a)}{\leq} r_{ext}^{\perp} + \log_2 \frac{2\cdot 2^{m-1}}{2\cdot 2^{m-1}-2^m+(r_{ext}^{\perp}+1)} \\ &= r_{ext}^{\perp} + m - \log_2(r_{ext}^{\perp}+1), \end{aligned}$$

where step (a) is from the Plotkin bound. Therefore, we obtain $r_{ext}^{\perp} \geq \log_2(r_{ext}^{\perp}+1)+1$. Thus, we have $r_{ext}^{\perp} \geq 3$. Therefore, the code is r-optimal. □

2.4.2 Shorten Operation

For an $[n,k,d]$ binary code $\mathcal{C}$, its shortened code $\mathcal{C}_s$ of $\mathcal{C}$ is the set of all codewords in $\mathcal{C}$ that are 0 in a fixed position with that position deleted. Let the last one of the coordinates of $\mathcal{C}$ be the position deleted, then the shortened code $\mathcal{C}_s$ is

$$\mathcal{C}_s = \Big\{ (c_1,\ldots,c_{n-1}) : (c_1,\ldots,c_{n-1},0) \in \mathcal{C} \Big\}.$$

We assume here that there is a codeword $\boldsymbol{c} \in \mathcal{C}$ such that $c_n = 1$. Otherwise, we will remove another coordinate satisfying this condition. The code $\mathcal{C}_s$ has parameters $[n-1, k-1, d_s \geq d]$ and its dual code is denoted by $\mathcal{C}_s^{\perp}$.

Lemma 2.4.4. *Let $\mathcal{C}$ be an $[n,k,d]$ binary code with locality $r \geq 2$. The shortened code $\mathcal{C}_s$ has locality r or $r-1$.*

Proof. Since $\mathcal{C}$ has locality r, for all $i \in [n-1]$, the ith code symbol has a repair set R_i with respect to $\mathcal{C}$ of size at most r. If $n \notin R_i$ then this symbol has the same repair set also with respect to the shortened code $\mathcal{C}_s$. Otherwise, note that if $\boldsymbol{c} \in \mathcal{C}_s$ then $(\boldsymbol{c}, 0) \in \mathcal{C}$, so we conclude that the ith symbol is recoverable also from the set $R_i \setminus \{n\}$. ■

The following is an immediate consequence of Lemma 2.4.4 for binary cyclic codes.

Corollary 2.4.5. *Let $\mathcal{C}$ be an $[n,k,d]$ binary cyclic code whose dual code has minimum distance $d^\perp \geq 3$. Then, the code $\mathcal{C}_s$ has locality either $d^\perp - 2$ or $d^\perp - 1$.*

The next example shows that the shortened code can in fact have locality $r-1$.

Example 2.4.3. Let $\mathcal{C}$ be the $[2^m - 1, 2^m - 1 - m, 3]$ cyclic binary Hamming code. Its shortened code $\mathcal{C}_s$ is a $[2^m - 2, 2^m - 2 - m, 3]$ code and from Corollary 2.4.5 it has locality $d^\perp - 2$ or $d^\perp - 1$, where $d^\perp = 2^{m-1}$. We show that it has locality $d^\perp - 2$. According to the proof of Lemma 2.4.4, it is enough to show that for every $i \in [n-1]$, the ith code symbol has a repair set R_i of size $2^{m-1} - 1$ which contains the nth coordinate. Or, according to Claim 2.3.1, it is enough to show that there exists a codeword $\boldsymbol{c} \in \mathcal{C}^\perp$ such that $c_i = c_n = 1$ and $w_H(\boldsymbol{c}) = 2^{m-1}$. We can omit the last requirement on the weight since all nonzero codewords in $\mathcal{C}^\perp$ have the same weight 2^{m-1}. Let $\boldsymbol{c}_1, \boldsymbol{c}_2 \in \mathcal{C}^\perp$ be two codewords such that $c_{1,i} = c_{2,n} = 1$. If $c_{1,n} = 1$ or $c_{2,i} = 1$ then we are done. Otherwise, the codeword $\boldsymbol{c}_1 + \boldsymbol{c}_2$ satisfies this property. The d-optimality, k-optimality, and r-optimality of $\mathcal{C}_s$ are proved in a similar way to the previous examples. □

2.4.3 Expurgate, Augment, and Lengthen Operations

For an $[n,k,d]$ binary code $\mathcal{C}$ having at least one odd weight codeword, the expurgated code $\mathcal{C}_{exp}$ is a subcode of $\mathcal{C}$ which contains only the codewords of even weight; that is,

$$\mathcal{C}_{exp} = \left\{ \boldsymbol{c} : \boldsymbol{c} \in \mathcal{C}, w_H(\boldsymbol{c}) \text{ is even } \right\}.$$

$\mathcal{C}_{exp}$ is an $[n, k-1, w_e]$ code, where w_e denotes the minimum even weight of nonzero codewords in $\mathcal{C}$. We denote by $\mathcal{C}_{exp}^\perp$ the dual code of $\mathcal{C}_{exp}$ and note that $\mathcal{C}_{exp}^\perp = \mathcal{C}^\perp \cup \overline{\mathcal{C}^\perp}$.

For an $[n,k,d]$ binary code $\mathcal{C}$ which does not contain the all-one codeword $\mathbf{1}$, the augmented code $\mathcal{C}_a$ is the code $\mathcal{C} \cup \overline{\mathcal{C}}$ with parameters $[n,k+1,\min\{d,n-w_{\max}\}]$, where $w_{\max}$ denotes the maximum weight of codewords in $\mathcal{C}$. We use the notation $\mathcal{C}_a^{\perp}$ to denote the dual code of $\mathcal{C}_a$.

According to these definitions, if the code $\mathcal{C}$ is cyclic then the expurgated and augmented codes of $\mathcal{C}$ are cyclic as well. Hence, for an $[n,k,d]$ binary cyclic code $\mathcal{C}$, we have the following two observations:

a) If $\mathcal{C}$ has an odd weight codeword, then $\mathcal{C}_{exp}$ has locality $r_{exp} = \min\{d^{\perp}, n - w_{\max}^{\perp}\} - 1$, where $w_{\max}^{\perp}$ is the maximum weight of codewords in $\mathcal{C}^{\perp}$. (Here, we assume $w_{\max}^{\perp} < n-1$, since $w_{\max}^{\perp} = n-1$ is not an interesting case.)

b) If $\mathcal{C}$ does not contain the all-one codeword $\mathbf{1}$, then $\mathcal{C}_a$ has locality $r_a = w_e^{\perp} - 1$, where $w_e^{\perp}$ is the minimum even weight of nonzero codewords in $\mathcal{C}^{\perp}$.

For an $[n,k,d]$ binary code $\mathcal{C}$ which does not contain the all-one codeword $\mathbf{1}$, the lengthened code $\mathcal{C}_\ell$ is obtained as follows. First, the code $\mathcal{C}$ is augmented to the code $\mathcal{C}_a = \mathcal{C} \cup \overline{\mathcal{C}}$. Then, $\mathcal{C}_a$ is extended. Thus, $\mathcal{C}_\ell = \{(c_1,\ldots,c_n,c_{n+1}) : c_{n+1} = \sum_{i=1}^{n} c_i \text{ and } (c_1,\ldots,c_n) \in \mathcal{C} \cup \overline{\mathcal{C}}\}$. After the lengthen operation, the length and dimension of the code are increased by 1. By leveraging the results from the augment and extend operations, we conclude that if the minimum even weight of nonzero codewords in $\mathcal{C}^{\perp}$ is $w_e^{\perp}$, and the maximum weight of codewords in $\mathcal{C}_a^{\perp}$ is $n+1-w_e^{\perp}$, then the lengthened code $\mathcal{C}_\ell$ has locality $r_\ell = w_e^{\perp} - 1$.

Our results on locality of classical binary codes and their modified versions are summarized in Table 2.2, where ✓ means we can prove the optimality of the given codes, whereas ? means we have not verified their optimality. In Table 2.2, TBCH stands for triple-error-correcting BCH.

2.5 Conclusion

In this chapter, we studied the locality of classical binary linear codes and their modified versions obtained from standard code operations. The optimality of these codes was also investigated. The locality properties of these codes can be used for constructing many other binary LRCs.

Table 2.2: Locality of classical binary codes and their modified versions.

$\mathcal{C}$	n	k	d	r	*d-opt*	*k-opt*	*r-opt*
Hamming code	2^m-1	2^m-1-m	3	$2^{m-1}-1$	✓	✓	✓
Simplex code	2^m-1	m	2^{m-1}	2	✓	✓	✓
Golay code	23	12	7	7	✓	✓	✓
Dual of Golay code	23	11	8	6	✓	✓	✓
DBCH code $(m \geq 4)$	2^m-1	2^m-1-2m	5	$2^{m-1}-2^{\lfloor m/2 \rfloor}-1$	Table 2.1	Table 2.1	Table 2.1
Dual of DBCH code $(m \geq 4)$	2^m-1	$2m$	$2^{m-1}-2^{\lfloor m/2 \rfloor}$	4	Table 2.1	Table 2.1	Table 2.1
Extended Hamming code	2^m	2^m-1-m	4	$2^{m-1}-1$	✓	✓	✓
Extended Golay code	24	12	8	7	✓	✓	✓
Extended DBCH code $(m \geq 4)$	2^m	2^m-1-2m	6	$2^{m-1}-2^{\lfloor m/2 \rfloor}-1$	✓	?	?
Extended TBCH code $(m \geq 5)$	2^m	2^m-1-3m	8	$2^{m-1}-2^{\lfloor m/2+1 \rfloor}-1$	✓	?	?
Biorthogonal code	2^m	$m+1$	2^{m-1}	3	✓	✓	✓
Expurgated Hamming code	2^m-1	2^m-2-m	4	$2^{m-1}-2$	✓	✓	✓
Expurgated DBCH code $(m \geq 4)$	2^m-1	2^m-2-2m	6	$2^{m-1}-2^{\lfloor m/2 \rfloor}-2$	✓	?	?
Expurgated TBCH code $(m \geq 5)$	2^m-1	2^m-2-3m	8	$2^{m-1}-2^{\lfloor m/2+1 \rfloor}-2$	✓	?	?
Augmented simplex code	2^m-1	$m+1$	$2^{m-1}-1$	3	✓	✓	✓
Shortened Hamming code	2^m-2	2^m-2-m	3	$2^{m-1}-2$	✓	✓	✓
Shortened simplex code	2^m-2	$m-1$	2^{m-1}	1	✓	✓	✓
$\mathcal{RM}(\mu,m)$	2^m	$\sum_{i=0}^{\mu}\binom{m}{i}$	$2^{m-\mu}$	$2^{\mu+1}-1$	?	?	?
Cyclic $\mathcal{RM}(\mu,m)$	2^m-1	$\sum_{i=0}^{\mu}\binom{m}{i}$	$2^{m-\mu}-1$	$2^{\mu+1}-1$	?	?	?
Dual of cyclic $\mathcal{RM}(\mu,m)$	2^m-1	$\sum_{i=\mu+1}^{m}\binom{m}{i}-1$	$2^{\mu+1}$	$2^{m-\mu}-2$	?	?	?

Acknowledgement

This chapter is in part a reprint of the material in the paper: Pengfei Huang, Eitan Yaakobi, Hironori Uchikawa, and Paul H. Siegel, "Cyclic linear binary locally repairable codes," in *Proc. IEEE Information Theory Workshop (ITW)*, Jerusalem, Israel, Apr.–May 2015, pp. 1–5. The dissertation author was the primary investigator and author of this paper.

Chapter 3

Binary Linear Codes with Locality and Availability

3.1 Introduction

In Chapter 2, we studied the locality of classical binary linear codes and their modified versions. In addition to symbol locality, another important property of locally repairable codes (LRCs) is their symbol availability, meaning the number of disjoint sets of symbols that can be used to recover a given symbol. High availability is a particularly attractive property for so-called *hot data* in a distributed storage network. More precisely, a code $\mathcal{C}$ has *all-symbol locality r and availability t* if every code symbol can be recovered from t disjoint repair sets of other symbols, each set of size at most r symbols. We refer to such a code as an $(r,t)_a$-LRC. If the code is systematic and these properties apply only to its information symbols, then the code has *information locality r and availability t*, and it is referred to as an $(r,t)_i$-LRC.

Several recent works have considered codes with both locality and availability properties. In [82], it was shown that the minimum distance d of an $[n,k,d]_q$ linear $(r,t)_i$-LRC satisfies the upper bound

$$d \leq n - k - \left\lceil \frac{(k-1)t+1}{(r-1)t+1} \right\rceil + 2. \tag{3.1}$$

In [60], it was proved that bound (3.1) is also applicable to $(n,M,d)_q$ non-linear $(r,t)_i$-LRCs. In the same

paper, it was also shown that if each repair set in a linear $(r,t)_i$-LRC contains only one parity symbol, then the minimum distance d of the code satisfies the following upper bound

$$d \leq n - k - \left\lceil \frac{kt}{r} \right\rceil + t + 1, \tag{3.2}$$

and codes achieving bound (3.2) were constructed using maximum distance separable (MDS) codes and Gabidulin codes [26]. For $(r,t)_a$-LRCs with parameters $(n,M,d)_q$, it was shown in [74] that d satisfies

$$d \leq n - \sum_{i=0}^{t} \left\lfloor \frac{k-1}{r^i} \right\rfloor. \tag{3.3}$$

There are several constructions of LRCs with availability. In [75], two constructions of $(r,2)_a$-LRCs were proposed. One relies on the combinatorial concept of orthogonal partitions, and the other one is based on product codes. In [54], a class of $(r,t)_a$-LRCs was constructed from partial geometries. A family of systematic fountain codes having information locality and strong probabilistic guarantees on availability was introduced in [6]. More recently, in [31, 92], constructions based on the simplex code were proposed. In [83], a family of LRCs with arbitrary availability was constructed, and it outperforms the direct product codes with respect to the information rate.

In this chapter, we study bounds and constructions for linear LRCs over a fixed field size; in particular, we focus on binary linear LRCs. Binary LRCs are of particular interest in practice because of their relatively lower encoding and decoding complexity compared to non-binary LRCs. We first develop field size dependent upper bounds that incorporate the availability t, based on the work by Cadambe and Mazumdar [13]. For constructions, we make contributions in the following two aspects.

Tensor product codes, first proposed by Wolf in [86], are a family of codes defined by a parity-check matrix that is the tensor product of the parity-check matrices of two constituent codes. Later, they were generalized in [40]. As shown in [87], the encoding steps of tensor product codes involve using *phantom* syndrome symbols, which only appear in the encoding procedure and will disappear in the final codewords. Motivated by these ideas, we give three constructions (**Constructions A**, **B**, and **C**) of LRCs that leverage phantom parity-check symbols. These constructions are effective for LRCs with

small minimum distance. To obtain LRCs with higher minimum distance, we present another construction (**Construction D**) based on a multi-level tensor product structure. All our constructions are flexible and generate a variety of high-rate LRCs with different localities. Some of these codes are proved to have optimal minimum distance.

One-step majority-logic decodable codes were first formally studied by Massey [47, 50]. Historically, these codes were introduced for low-complexity error correction. Every symbol of such codes has several disjoint repair sets, and is decoded according to the majority of the values given by all of its repair sets. In this chapter, we make the connection between one-step majority-logic decodable codes and LRCs with availability. We also demonstrate how a long $(r,t)_a$-LRC can be constructed from a short one-step majority-logic decodable code using a multi-level tensor product structure.

The remainder of this chapter is organized as follows. In Section 3.2, we formally define the problem and present field size q dependent bounds on the minimum distance d and the dimension k of $[n,k,d]_q$ linear $(r,t)_i$-LRCs. In Section 3.3, we construct various families of $(r,1)_i$-LRCs and $(r,1)_a$-LRCs using phantom parity-check symbols and a multi-level tensor product structure. In Section 3.4, we review several families of one-step majority-logic decodable codes, and identify the locality and availability of these codes. We conclude the chapter in Section 3.5.

3.2 Definitions and Bounds

We begin with several basic definitions and notational conventions. We use the notation $[n]$ to define the set $\{1,\ldots,n\}$. For a length-n vector $\boldsymbol{v}$ and a set $\mathcal{I} \subseteq [n]$, the vector $\boldsymbol{v}_{\mathcal{I}}$ denotes the restriction of the vector $\boldsymbol{v}$ to coordinates in the set $\mathcal{I}$. A linear code $\mathcal{C}$ over $\mathbb{F}_q$ of length n, dimension k, and minimum distance d will be denoted by $[n,k,d]_q$, where the field size q may be omitted if it is clear from the context, and its generator matrix is $G = (g_1,\ldots,g_n)$, where $g_i \in \mathbb{F}_q^k$ is a column vector for $i \in [n]$. We define $k_{\mathcal{I}}(\mathcal{C}) = \log_q |\{\boldsymbol{c}_{\mathcal{I}} : \boldsymbol{c} \in \mathcal{C}\}|$, and, for simplicity, we write $k_{\mathcal{I}}$ instead of $k_{\mathcal{I}}(\mathcal{C})$ when $\mathcal{C}$ is clear from the context. The dual code of a linear code $\mathcal{C}$ will be denoted by $\mathcal{C}^{\perp}$.

We follow the conventional definitions of linear LRCs with availability, as established in [60,74,82].

Definition 3.2.1. *The ith code symbol of an $[n,k,d]_q$ linear code $\mathcal{C}$ is said to have locality r and availability*

t if there exist t pairwise disjoint repair sets $\mathcal{R}_i^1, \ldots, \mathcal{R}_i^t \subseteq [n] \backslash \{i\}$, such that 1) $|\mathcal{R}_i^j| \leq r$, for $1 \leq j \leq t$, and 2) for each repair set $\mathcal{R}_i^j$, $1 \leq j \leq t$, g_i is a linear combination of the columns g_u, $u \in \mathcal{R}_i^j$.

Definition 3.2.2. *Let $\mathcal{C}$ be an $[n,k,d]_q$ linear code. A set $\mathcal{I} \subseteq [n]$ is said to be an information set if $|\mathcal{I}| = k_{\mathcal{I}} = k$.*

1) The code $\mathcal{C}$ is said to have all-symbol locality r and availability t if every code symbol has locality r and availability t. We refer to $\mathcal{C}$ as a linear $(r,t)_a$-LRC.

2) The code $\mathcal{C}$ is said to have information locality r and availability t if there is an information set $\mathcal{I}$ such that, for any $i \in \mathcal{I}$, the ith code symbol has locality r and availability t. We refer to $\mathcal{C}$ as a linear $(r,t)_i$-LRC.

Note that when $t = 1$, Definition 3.2.2 reduces to the definition of linear LRCs. It is straightforward to verify that the minimum distance d of a linear $(r,t)_a$-LRC satisfies $d \geq t+1$. We now present upper bounds on the minimum distance and the dimension of linear $(r,t)_i$-LRCs, based on the framework established in [13]. The following lemma and theorem are extensions of Lemma 1 and Theorem 1 from [13], respectively.

Let r and x be two positive integers and $\boldsymbol{y} = (y_1, \ldots, y_x) \in ([t])^x$ be a vector of x positive integers. We define the integers $A(r,x,\boldsymbol{y})$ and $B(r,x,\boldsymbol{y})$ as follows,

$$A(r,x,\boldsymbol{y}) = \sum_{j=1}^{x} (r-1)y_j + x,$$
$$B(r,x,\boldsymbol{y}) = \sum_{j=1}^{x} r y_j + x.$$

Lemma 3.2.3. *Let $\mathcal{C}$ be an $[n,k,d]_q$ linear $(r,t)_i$-LRC. Assume that $x \in \mathbb{Z}^+$ and $\boldsymbol{y} = (y_1, \ldots, y_x) \in ([t])^x$ satisfy $1 \leq x \leq \lceil \frac{k}{(r-1)t+1} \rceil$ and $A(r,x,\boldsymbol{y}) < k$. Then, there exists a set $\mathcal{I} \subseteq [n]$ such that $|\mathcal{I}| = B(r,x,\boldsymbol{y})$ and $k_{\mathcal{I}}(\mathcal{C}) \leq A(r,x,\boldsymbol{y})$.*

Proof. See Section 3.6 Appendix A. ■

Now, let $d_{\ell-opt}^{(q)}[n,k]$ denote the largest possible minimum distance of a linear code of length n and dimension k over $\mathbb{F}_q$, and let $k_{\ell-opt}^{(q)}[n,d]$ denote the largest possible dimension of a linear code of length n and minimum distance d over $\mathbb{F}_q$. Applying Lemma 3.2.3, we get the following upper bounds on

d and k for $[n,k,d]_q$ linear $(r,t)_i$-LRCs.

Theorem 3.2.4. *For any $[n,k,d]_q$ linear $(r,t)_i$-LRC, the minimum distance d satisfies*

$$d \leq \min_{\substack{1\leq x\leq \lceil \frac{k}{(r-1)t+1}\rceil,\ x\in\mathbb{Z}^+,\\ \boldsymbol{y}\in([t])^x,\\ A(r,x,\boldsymbol{y})<k}} \left\{ d^{(q)}_{\ell-opt}[n-B(r,x,\boldsymbol{y}),k-A(r,x,\boldsymbol{y})] \right\}, \tag{3.4}$$

and the dimension k satisfies

$$k \leq \min_{\substack{1\leq x\leq \lceil \frac{k}{(r-1)t+1}\rceil,\ x\in\mathbb{Z}^+,\\ \boldsymbol{y}\in([t])^x,\\ A(r,x,\boldsymbol{y})<k}} \left\{ A(r,x,\boldsymbol{y}) + k^{(q)}_{\ell-opt}[n-B(r,x,\boldsymbol{y}),d] \right\}. \tag{3.5}$$

Proof. See Section 3.7 Appendix B. ■

Remark 3.2.1. Since a linear $(r,t)_a$-LRC is also a linear $(r,t)_i$-LRC, bounds (3.4) and (3.5) hold for linear $(r,t)_a$-LRCs as well. □

Remark 3.2.2. We note that for linear codes with availability $t=1$, bound (3.5) in Theorem 3.2.4 becomes

$$k \leq \min_{1\leq x\leq \lceil \frac{k}{r}\rceil -1} \left\{ xr + k^{(q)}_{\ell-opt}[n-x(r+1),d] \right\} = \min_{x\in\mathbb{Z}^+} \left\{ xr + k^{(q)}_{\ell-opt}[n-x(r+1),d] \right\}$$

which coincides with bound (2.2). As was proved in [13], this implies that, for $t=1$, Theorem 3.2.4 is at least as strong as bound (3.1). In fact, the latter statement holds for all values of r and t. For $r=1$, this can be readily verified by evaluating the expression in bound (3.4) at $x=k-1$ and $\boldsymbol{y}=(t,t,\ldots,t)$. For $r,t\geq 2$, we can similarly validate the claim by means of a suitable choice of x and $\boldsymbol{y}$, as we now show.

We consider two cases.

Case 1: Assume that either $k(\bmod((r-1)t+1))=0$ or $k(\bmod((r-1)t+1))>r$. Here we choose $x=\lceil \frac{k}{(r-1)t+1}\rceil$, $y_1=\cdots=y_{x-1}=t$, and $y_x=\lceil \frac{k-(x-1)\left((r-1)t+1\right)-1}{r-1}\rceil -1$. Note that $1\leq y_x\leq t$ and $A(r,x,\boldsymbol{y})<k$. Bound (3.4) implies that

$$d \leq d^{(q)}_{\ell-opt}[n-B(r,x,\boldsymbol{y}),k-A(r,x,\boldsymbol{y})] \leq n-k+1-(x-1)t-y_x,$$

where the latter inequality follows from the Singleton bound. So we only need to show that

$$(x-1)t + y_x \geq \left\lceil \frac{(k-1)t+1}{(r-1)t+1} \right\rceil - 1.$$

If the condition $k(\bmod((r-1)t+1)) = 0$ holds, we verify that

$$(x-1)t + y_x = xt - 1 = \frac{kt}{(r-1)t+1} - 1 \geq \left\lceil \frac{(k-1)t+1}{(r-1)t+1} \right\rceil - 1.$$

On the other hand, if $k(\bmod((r-1)t+1)) > r$, we can write $k = (x-1)\big((r-1)t+1\big) + \gamma$, for some $r < \gamma \leq (r-1)t$, and then we see that

$$(x-1)t + y_x = (x-1)t + \left\lceil \frac{\gamma-1}{r-1} \right\rceil - 1 \geq (x-1)t + \left\lceil \frac{(\gamma-1)t+1}{(r-1)t+1} \right\rceil - 1 = \left\lceil \frac{(k-1)t+1}{(r-1)t+1} \right\rceil - 1.$$

Case 2: Assume that $1 \leq k(\bmod((r-1)t+1)) \leq r$. Here we choose, $x = \lceil \frac{k}{(r-1)t+1} \rceil - 1$ and $\boldsymbol{y} = (t, \cdots, t)$. Note that $A(r, x, \boldsymbol{y}) < k$. Here bound (3.4) leads to

$$d \leq n - k + 1 - xt$$

so we need to verify that

$$xt \geq \left\lceil \frac{(k-1)t+1}{(r-1)t+1} \right\rceil - 1.$$

Setting $k = x((r-1)t+1) + \gamma$, where $1 \leq \gamma \leq r$, and noting that $\frac{(k-1)t+1+(r-\gamma)t}{(r-1)t+1}$ is an integer, we find that

$$\begin{aligned} xt &= \frac{(k-\gamma)t}{(r-1)t+1} \\ &= \frac{(k-1)t+1+(r-\gamma)t}{(r-1)t+1} - 1 \\ &\geq \left\lceil \frac{(k-1)t+1}{(r-1)t+1} \right\rceil - 1. \end{aligned}$$

Hence, we conclude that the bound in Theorem 3.2.4 is at least as strong as bound (3.1) for all $r, t \geq 1$. □

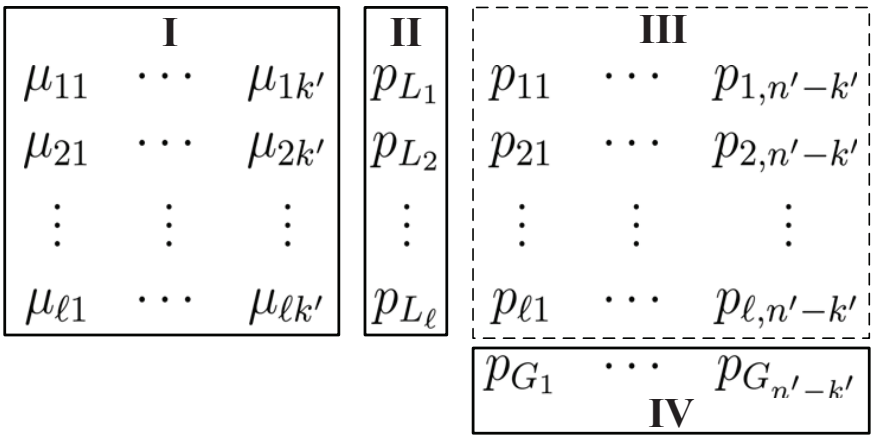


Figure 3.1: An $(r,1)_i$-LRC using Construction A. Information symbols are in block **I**, local parity-check symbols are in block **II**, phantom symbols are in block **III**, and global parity-check symbols are in block **IV**.

3.3 Construction of Binary LRCs

In this section, we focus on constructing binary LRCs. We first present constructions of LRCs with small minimum distance (i.e., $d = 3$, 4, and 5) by using phantom parity-check symbols. Then, in order to obtain LRCs with higher minimum distance, we propose another construction which is based on a multi-level tensor product structure.

3.3.1 Construction Using Phantom Parity-Check Symbols

We first consider constructing binary linear $(r,1)_i$-LRCs with minimum distance 3 and 4. The general framework is depicted in Figure 3.1. We specify an $[n',k',d']$ systematic binary code as a base code, $\mathcal{C}_{base}$. The following construction produces an $(r,1)_i$-LRC of length $n = (k'+1)\ell + n' - k'$, dimension $k = k'\ell$, and information locality $r = k'$.

Construction A

Step 1: Place an $\ell \times k'$ array of information symbols in block **I**.

Step 2: For each row of information symbols, $(\mu_{i1},\ldots,\mu_{ik'})$, $1 \le i \le \ell$, compute *local* parity-check symbols $p_{L_i} = \sum_{j=1}^{k'} \mu_{ij}$, $1 \le i \le \ell$, and place them in the corresponding row of block **II**.

Step 3: Encode each row of information symbols in block **I** using $\mathcal{C}_{base}$, producing parity-check symbols $(p_{i1},\ldots,p_{i,n'-k'})$, $1 \le i \le \ell$. Place these parity-check symbols in block **III**. (These symbols are referred

to as *phantom* symbols because they will not appear in the final codeword.)

Step 4: Compute a row of *global* parity-check symbols, $p_{G_j} = \sum_{i=1}^{\ell} p_{ij}$, $1 \leq j \leq n' - k'$, by summing the rows of phantom symbols in block **III**. Place these symbols in block **IV**.

Step 5: The constructed codeword consists of the symbols in blocks **I**, **II**, and **IV**. ■

Note that for $r|k$, Pyramid codes are optimal $(r,1)_i$-LRCs over sufficiently large field size [38]. A Pyramid code is constructed by splitting a parity-check symbol of a systematic MDS code into k/r local parity-check symbols. However, for the binary case, it is hard to find a good binary code first and then conduct the splitting operation. In contrast, we take a different approach. We first design the local parity-check symbols, and then construct the global parity-check symbols.

If $\mathcal{C}_{base}$ has an *information-sum parity-check symbol*, a parity-check symbol which is the sum of all its information symbols, we can simply modify Step 3 of Construction A to reduce the code redundancy as follows. After encoding each row of information symbols in block **I**, define the corresponding row of phantom symbols to be the computed parity-check symbols with the information-sum parity-check symbol excluded, and store them in block **III**. Then proceed with the remaining steps in Construction A. We refer to this modified construction as **Construction A$'$**. It is easy to verify that the resulting code is an $(r,1)_i$-LRC with length $n = (k'+1)\ell + n' - k' - 1$, dimension $k = k'\ell$, and information locality $r = k'$.

Now, if we use a $\mathcal{C}_{base}$ with minimum distance 3, we have a lower bound on the minimum distance of the constructed LRC, as stated in the following lemma.

Lemma 3.3.1. *If $\mathcal{C}_{base}$ is an $[n', k', d' = 3]$ code, the $(r,1)_i$-LRC produced by Construction A (or Construction A$'$, if appropriate) has minimum distance $d \geq 3$.*

Proof. We prove that the minimum distance of the constructed $(r,1)_i$-LRC from Construction A is at least 3 by verifying that it can correct any two erasures. We consider the following 2-erasure patterns, where we refer to their locations in the blocks in Figure 3.1. We refer to block **I-II** as the union of block **I** and block **II**. 1) Two erasures are in the same row in block **I-II**, e.g., μ_{11} and p_{L_1} are erased in the first row. We can first recover parity-check symbols $(p_{11}, \ldots, p_{1,n'-k'})$, based on which two erased symbols can be recovered. 2) Two erasures in different rows in block **I-II** can be recovered individually from the local parity-check equation. 3) Two erasures in block **IV** can be recovered from all existing information

symbols. 4) One erasure is in block **I-II** and one erasure is in block **IV**. First, the erasure in block **I-II** can be recovered from the local parity-check equation. Then, the erasure in block **IV** can be recovered from all the existing information symbols.

The proof for the constructed $(r,1)_i$-LRC from Construction A$'$ follows the same ideas and is thus omitted. ■

Based on Lemma 3.3.1, we have the following theorem on the construction of $(r,1)_i$-LRCs with optimal minimum distance $d = 3$.

Theorem 3.3.2. *Let $\mathcal{C}_{base}$ be an $[n',k',d'=3]$ binary code with an information-sum parity-check symbol and assume that $d^{(2)}_{\ell-opt}[n',k'] = 3$. The $(r,1)_i$-LRC obtained from Construction A$'$ has parameters $[n = (k'+1)\ell + n' - k' - 1, k = k'\ell, d = 3]$ and $r = k'$. Its minimum distance $d = 3$ is optimal.*

Proof. From Construction A$'$, the length, dimension and locality of the $(r,1)_i$-LRC are determined. From Lemma 3.3.1, the minimum distance satisfies $d \geq 3$. On the other hand, from bound (3.4), with $x = \ell - 1$ and $t = 1$, $d \leq d^{(2)}_{\ell-opt}[n-(k'+1)(\ell-1), k-k'(\ell-1)] = d^{(2)}_{\ell-opt}[n',k'] = 3$. Therefore, $d = 3$ and it is optimal. ■

We give some examples of $(r,1)_i$-LRCs with $d = 3$. First, let $\mathcal{C}_{base}$ be the $[7,4,3]$ systematic binary Hamming code whose parity-check matrix is

$$H_{[7,4,3]} = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}.$$

Using Construction A, we obtain an $(r,1)_i$-LRC with parameters $[5\ell+3, 4\ell, 3]$ with $r = 4$. However, the upper bound on the minimum distance from bound (3.4) is 4. To construct an $(r,1)_i$-LRC whose minimum distance is optimal with respect to bound (3.4), we use a $[6,3,3]$ shortened binary Hamming code as the $\mathcal{C}_{base}$ whose parity-check matrix $H_{[6,3,3]}$ is obtained by deleting the first column of $H_{[7,4,3]}$,

$$H_{[6,3,3]} = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}.$$

Now, the $\mathcal{C}_{base}$ has an information-sum parity-check symbol and $d^{(2)}_{\ell-opt}[6,3] = 3$. From Theorem 3.3.2, the $(r,1)_i$-LRC generated by Construction A$'$ has parameters $[4\ell+2, 3\ell, 3]$ and $r = 3$. Moreover, its minimum distance $d = 3$ is optimal.

The above $[6,3,3]$ base code $\mathcal{C}_{base}$ can be generalized as follows. Let $\mathcal{C}$ be a $[2^m-1, 2^m-1-m, 3]$ systematic binary Hamming code with parity-check matrix

$$H = \begin{bmatrix} h_{1,1} & h_{1,2} & \dots & h_{1,2^m-1-m} & 1 & 0 & \dots & 0 \\ h_{2,1} & h_{2,2} & \dots & h_{2,2^m-1-m} & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ h_{m,1} & h_{m,2} & \dots & h_{m,2^m-1-m} & 0 & 0 & \dots & 1 \end{bmatrix},$$

whose columns range over all the nonzero vectors in $\mathbb{F}_2^m$. The first 2^m-1-m coordinates of $\mathcal{C}$ form the systematic information symbols. The parity-check matrix H_s of the shortened binary Hamming code $\mathcal{C}_s$ is obtained by deleting any ith column of H, if $1 \le i \le 2^m-1-m$ and $h_{1,i} = 0$. As a result, $\mathcal{C}_s$ is systematic and has an information-sum parity-check symbol.

Lemma 3.3.3. *The code $\mathcal{C}_s$ has parameters $[2^{m-1}+m-1, 2^{m-1}-1, 3]$, and its minimum distance is optimal.*

Proof. The first row of H has 2^{m-1} ones and $2^{m-1}-1$ zeros, since it is a nonzero codeword of the $[2^m-1, m, 2^{m-1}]$ binary simplex code. According to the shortening operation, we delete in total $2^{m-1}-m$ columns from H, so the length of $\mathcal{C}_s$ becomes $2^{m-1}+m-1$ and the dimension becomes $2^{m-1}-1$. Since the shortening operation does not decrease the minimum distance, and there always exist three dependent columns in H_s (e.g., $[1,1,0,\dots,0]^T$, $[1,0,0,\dots,0]^T$, and $[0,1,0,\dots,0]^T$), the minimum distance remains 3. Lastly, we have that $d^{(2)}_{\ell-opt}[2^{m-1}+m-1, 2^{m-1}-1] = 3$ from the anticode bound [2]. The anticode bound states that the size of any binary code of length n with minimum distance D is bounded above by $2^n/A(D-1)$, where $A(D-1)$ is the size of the largest anticode of diameter $D-1$. In particular, for $D = 4$ this implies that the largest size of a length-n code with minimum distance 4 is $2^n/(2n)$ and hence there does not exist a code of length $2^{m-1}+m-1$, minimum distance 4, and dimension $2^{m-1}-1$. ■

The following example is a direct result of Theorem 3.3.2 and Lemma 3.3.3.

Example 3.3.1. Let $\mathcal{C}_{base}$ be the shortened binary Hamming code $\mathcal{C}_s$ in Lemma 3.3.3. The $(r,1)_i$-LRC obtained from Construction A$'$ has parameters $[2^{m-1}\ell + m - 1, (2^{m-1}-1)\ell, 3]$ and $r = 2^{m-1} - 1$. Its minimum distance is optimal. □

Next, we use a code $\mathcal{C}_{base}$ with minimum distance 4, and have the following lemma.

Lemma 3.3.4. *If $\mathcal{C}_{base}$ is an $[n', k', d' = 4]$ code, the $(r,1)_i$-LRC produced by Construction A (or Construction A$'$, if appropriate) has minimum distance $d \geq 4$.*

Proof. The proof is similar to the one of Lemma 3.3.1. ■

Based on Lemma 3.3.4, we have the following two theorems on the construction of $(r,1)_i$-LRCs with optimal minimum distance $d = 4$.

Theorem 3.3.5. *Let $\mathcal{C}_{base}$ be an $[n', k', d' = 4]$ binary code with $d^{(2)}_{\ell-opt}[n'+1, k'] = 4$. The $(r,1)_i$-LRC obtained from Construction A has parameters $[n = (k'+1)\ell + n' - k', k = k'\ell, d = 4]$ and $r = k'$. Its minimum distance $d = 4$ is optimal.*

Proof. The proof is similar to the one of Theorem 3.3.2. ■

Theorem 3.3.6. *Let $\mathcal{C}_{base}$ be an $[n', k', d' = 4]$ binary code with an information-sum parity-check symbol and $d^{(2)}_{\ell-opt}[n', k'] = 4$. The $(r,1)_i$-LRC obtained from Construction A$'$ has parameters $[n = (k'+1)\ell + n' - k' - 1, k = k'\ell, d = 4]$ and $r = k'$. Its minimum distance $d = 4$ is optimal.*

Proof. The proof is similar to the one of Theorem 3.3.2. ■

We give examples of $(r,1)_i$-LRCs with $d = 4$ using expurgated or extended binary Hamming code as $\mathcal{C}_{base}$. The following lemma gives properties of expurgated and extended binary Hamming codes.

Lemma 3.3.7. *For $m \geq 4$, the $[2^m - 1, 2^m - 2 - m, 4]$ systematic expurgated binary Hamming code has no information-sum parity-check symbol, and $d^{(2)}_{\ell-opt}[2^m, 2^m - 2 - m] = 4$. For $m \geq 3$, the $[2^m, 2^m - 1 - m, 4]$ systematic extended binary Hamming code has no information-sum parity-check symbol, and $d^{(2)}_{\ell-opt}[2^m + 1, 2^m - 1 - m] = 4$.*

Proof. For the expurgated binary Hamming code, in its dual code, except the all-one codeword with weight $2^m - 1$, there is no codeword with weight larger than 2^{m-1}. If the expurgated binary Hamming code has an

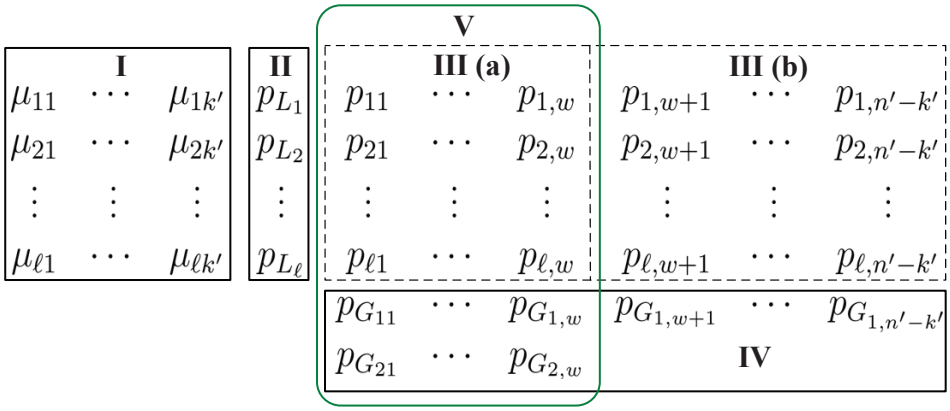


Figure 3.2: An $(r,1)_i$-LRC using Construction B.

information-sum parity-check symbol, then in its dual code there is a codeword with weight 2^m-1-m, which is larger than 2^{m-1} for $m \geq 4$. We have $d^{(2)}_{\ell-opt}[2^m, 2^m-2-m] = 4$ from the Hamming bound. Similarly, for the extended binary Hamming code, in its dual code, except the all-one codeword with weight 2^m, there is no codeword with weight larger than 2^{m-1}. If the extended binary Hamming code has an information-sum parity-check symbol, then in its dual code there is a codeword with weight 2^m-m, which is larger than 2^{m-1} for $m \geq 3$. We have $d^{(2)}_{\ell-opt}[2^m+1, 2^m-1-m] = 4$ from the Hamming bound. ■

The following example presents $(r,1)_i$-LRCs with $d=4$ from Theorem 3.3.5 and Lemma 3.3.7.

Example 3.3.2. Let $\mathcal{C}_{base}$ be the $[2^m-1, 2^m-2-m, 4]$ expurgated binary Hamming code, where $m \geq 4$. The $(r,1)_i$-LRC obtained from Construction A has parameters $[(2^m-1-m)\ell+m+1, (2^m-2-m)\ell, 4]$ and $r = 2^m-2-m$. Its minimum distance 4 is optimal. Similarly, let $\mathcal{C}_{base}$ be the $[2^m, 2^m-1-m, 4]$ extended binary Hamming code, where $m \geq 3$. The $(r,1)_i$-LRC obtained from Construction A has parameters $[(2^m-m)\ell+m+1, (2^m-1-m)\ell, 4]$ and $r = 2^m-1-m$. Its minimum distance 4 is optimal. □

Next, we give a construction of $(r,1)_i$-LRCs for $d=5$. Let $\mathcal{C}_{base}$ be an $[n', k', 5]$ systematic binary code, and let $\mathcal{C}'_{base} = \{\boldsymbol{c}_{[k'+w]} : \boldsymbol{c} \in \mathcal{C}_{base}\}$, i.e., restrict $\mathcal{C}_{base}$ to k' information coordinates and w parity-check coordinates, where w is chosen properly such that $\mathcal{C}'_{base}$ has minimum distance at least 3. The following new construction is based on two rows of global parity-check symbols as shown in Figure 3.2.

Construction B

Step 1: Follow Steps 1, 2, and 3 of Construction A to get *local* parity-check symbols and *phantom* symbols.

Step 2: Divide phantom symbols into two parts: w columns in block **III(a)** and the rest of the columns in block **III(b)**.

Step 3: Compute *global* parity-check symbols in block **IV**: 1) Follow Step 4 of Construction A to get the first row $(p_{G_{11}}, \cdots, p_{G_{1,n'-k'}})$. 2) Use an $[\ell+2, \ell, 3]$ systematic MDS code over $\mathbb{F}_{2^w}$ to encode the phantom symbols in block **III(a)** to get the second row $(p_{G_{21}}, \cdots, p_{G_{2,w}})$, by taking each row in block **III(a)** as a symbol in $\mathbb{F}_{2^w}$. This systematic MDS code should have the property that its first parity-check symbol is the sum of all its information symbols.

Step 4: The constructed codeword consists of the symbols in blocks **I**, **II**, and **IV**. ■

For example, the $[\ell+2, \ell, 3]$ MDS code can be chosen as a doubly-extended Reed-Solomon code. Let α be a primitive element in $\mathbb{F}_{2^w}$, and $\ell \leq 2^w - 1$. Then, the parity-check matrix for the doubly-extended Reed-Solomon code in Construction B is

$$H = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 & 1 & 0 \\ 1 & \alpha & \alpha^2 & \cdots & \alpha^{\ell-1} & 0 & 1 \end{bmatrix}.$$

Note that an alternative to the doubly-extended Reed-Solomon code is an EVENODD code [9].

Theorem 3.3.8. *The $(r,1)_i$-LRC obtained from Construction B has parameters $[n = (k'+1)\ell + n' - k' + w, k = k'\ell, d \geq 5]$, where $\ell \leq 2^w - 1$. It has information locality $r = k'$.*

Proof. From Construction B, the code length, dimension, and locality are determined. As in the proof of Lemma 3.3.1, to prove that the minimum distance of the $(r,1)_i$-LRC is at least 5, we only need to enumerate all possible 4-erasure patterns and then verify they can be corrected. In the following, we show how to recover two typical 4-erasure patterns in Figure 3.2. Other patterns can be verified in a similar way and hence are omitted.

1) There are two erasures in a row in block **I** and two erasures in another row in block **I**. Without loss of generality, assume they appear on the first two rows $(\mu_{11}, \ldots, \mu_{1k'})$ and $(\mu_{21}, \ldots, \mu_{2k'})$ in block **I**. We can recover this 4-erasure pattern as follows. First, with $(p_{G_{11}}, \ldots, p_{G_{1,w}})$ and $(p_{G_{21}}, \ldots, p_{G_{2,w}})$, we can recover $(p_{11}, \ldots, p_{1,w})$ and $(p_{21}, \ldots, p_{2,w})$. Then, two erasures in the first row $(\mu_{11}, \ldots, \mu_{1k'})$ can be recovered since only two erasures appear in the codeword $(\mu_{11}, \ldots, \mu_{1k'}, p_{11}, \ldots, p_{1,w})$ which belongs to a

code with minimum distance at least 3. Similarly, two erasures in the second row $(\mu_{21},\ldots,\mu_{2k'})$ can be recovered since only two erasures appear in the codeword $(\mu_{21},\ldots,\mu_{2k'},p_{21},\ldots,p_{2,w})$.

2) There are two erasures in a row in block **I**, one erasure in $(p_{G_{11}},\ldots,p_{G_{1,w}})$ in block **IV**, and one more erasure in $(p_{G_{21}},\ldots,p_{G_{2,w}})$ in block **IV**. For simplicity, we assume that the erasures are located in positions μ_{11}, μ_{12}, $p_{G_{11}}$, and $p_{G_{21}}$. We can recover this 4-erasure pattern as follows. First, with $(p_{G_{12}},\ldots,p_{G_{1,n'-k'}})$, we can recover $(p_{12},\ldots,p_{1,n'-k'})$. Then, μ_{11} and μ_{12} can be recovered since only three erasures appear in the codeword $(\mu_{11},\ldots,\mu_{1k'},p_{11},\ldots,p_{1,n'-k'})$. Finally, $p_{G_{11}}$ and $p_{G_{21}}$ can be recovered. ■

Example 3.3.3. Let $\mathcal{C}_{base}$ be the $[2^m-1,2^m-1-2m,5]$ binary BCH code where $m\geq 4$. For the case of $m=4$, exhaustive search shows that we can choose w to be 4. For $\ell\leq 15$, the $(r,1)_i$-LRC from Construction B has parameters $[n=8\ell+12,k=7\ell,d=5]$ and $r=7$. An upper bound on d from bound (3.4) is 8. For the case of $m=5$, exhaustive search shows that we can choose w to be 6. For $\ell\leq 63$, the $(r,1)_i$-LRC from Construction B has parameters $[n=22\ell+16,k=21\ell,d=5]$ and $r=21$. An upper bound on d from bound (3.4) is 8. □

We finish this subsection with a construction of $(r,1)_a$-LRCs with minimum distance 4 by using phantom parity-check symbols. We start with an $[n',k',d']$ systematic binary code as a base code, $\mathcal{C}_{base}$. For simplicity, we assume that $k'\geq n'-k'$. We use Figure 3.1 to illustrate our construction of $(r,1)_a$-LRCs as follows.

Construction C

Step 1: Place an $\ell\times k'$ array of symbols in block **I**. The first $\ell-1$ rows are all information symbols. The last row has $2k'-n'$ information symbols (i.e., $\mu_{\ell,1},\ldots,\mu_{\ell,2k'-n'}$) and $n'-k'$ zero symbols (i.e., $\mu_{\ell,2k'-n'+1}=0,\ldots,\mu_{\ell,k'}=0$).

Step 2: Encode each row of symbols in block **I** using the code $\mathcal{C}_{base}$, producing parity-check symbols $(p_{i1},\ldots,p_{i,n'-k'})$, $1\leq i\leq\ell$. Place these parity-check symbols in block **III** as *phantom* symbols.

Step 3: Compute a row of *global* parity-check symbols, $p_{G_j}=\sum_{i=1}^{\ell}p_{ij}$, $1\leq j\leq n'-k'$, by summing the rows of phantom symbols in block **III**. Place these symbols in block **IV**.

Step 4: Let $\mu_{\ell,2k'-n'+j}=p_{G_j}$, $1\leq j\leq n'-k'$. For each row of symbols, $(\mu_{i1},\ldots,\mu_{ik'})$, $1\leq i\leq\ell$,

compute *local* parity-check symbols $p_{L_i} = \sum_{j=1}^{k'} \mu_{ij}$, $1 \leq i \leq \ell$, and place them in the corresponding row of block **II**.

Step 5: The constructed codeword consists of the symbols in blocks **I** and **II**. ■

The resulting code is an $(r,1)_a$-LRC of code length $n = (k'+1)\ell$, dimension $k = k'\ell - (n'-k')$, and all-symbol locality $r = k'$.

We present the following theorem on the construction of $(r,1)_a$-LRCs with optimal minimum distance 4.

Theorem 3.3.9. *Let $\mathcal{C}_{base}$ be an $[n',k',d'=4]$ systematic binary code with $k' \geq n'-k'$ and $d^{(2)}_{\ell-opt}[k'+1, 2k'-n'] \leq 4$. The $(r,1)_a$-LRC obtained from Construction C has parameters $[n = (k'+1)\ell, k = k'\ell - (n'-k'), d=4]$ and all-symbol locality $r = k'$. Its minimum distance $d = 4$ is optimal.*

Proof. From Construction C, the length, dimension, and locality of the $(r,1)_a$-LRC are determined. On the one hand, the minimum distance $d \geq 4$ since the $(r,1)_a$-LRC can correct any 3 erasures (The proof is similar to the one of Lemma 3.3.1, so we omit it here). On the other hand, from bound (3.4), with $x = \ell - 1$ and $t = 1$, $d \leq d^{(2)}_{\ell-opt}[k'+1, 2k'-n'] \leq 4$. ■

We give the following example of $(r,1)_a$-LRCs with $d = 4$.

Example 3.3.4. Let $\mathcal{C}_{base}$ be the $[n' = 2^m - 1, k' = 2^m - 2 - m, d' = 4]$ expurgated binary Hamming code, where $m \geq 4$. Since $d^{(2)}_{\ell-opt}[k'+1, 2k'-n'] = d^{(2)}_{\ell-opt}[2^m - m - 1, 2^m - 2m - 3] \leq 4$ due to the Hamming bound, from Theorem 3.3.9, the $(r,1)_a$-LRC obtained from Construction C has parameters $[(2^m-1-m)\ell, (2^m-2-m)\ell - 1 - m, 4]$ and all-symbol locality $r = 2^m - 2 - m$. Its minimum distance 4 is optimal. Similarly, let $\mathcal{C}_{base}$ be the $[2^m, 2^m - 1 - m, 4]$ extended binary Hamming code, where $m \geq 3$. The $(r,1)_a$-LRC obtained from Construction C has parameters $[(2^m - m)\ell, (2^m - 1 - m)\ell - 1 - m, 4]$ and all-symbol locality $r = 2^m - 1 - m$. Its minimum distance 4 is optimal. □

The binary LRCs constructed in this subsection are summarized in Table 3.1.

3.3.2 Construction Using Multi-Level Tensor Product Structure

In the previous subsection, we presented constructions of binary LRCs with small minimum distance (i.e., $d = 3$, 4, and 5) based on phantom parity-check symbols. Here, we propose a new

Table 3.1: Constructed binary LRCs in Section 3.3.1.

$(r,1)_i$-LRCs	n	k	d	r
Example 3.3.1	$2^{m-1}\ell+m-1$	$(2^{m-1}-1)\ell$	3	$2^{m-1}-1$
Example 3.3.2	$(2^m-1-m)\ell+m+1$	$(2^m-2-m)\ell$	4	2^m-2-m
Example 3.3.2	$(2^m-m)\ell+m+1$	$(2^m-1-m)\ell$	4	2^m-1-m
Example 3.3.3	$8\ell+12$ $(\ell\leq 15)$	7ℓ	5	7
Example 3.3.3	$22\ell+16$ $(\ell\leq 63)$	21ℓ	5	21
$(r,1)_a$-LRCs	n	k	d	r
Example 3.3.4	$(2^m-1-m)\ell$	$(2^m-2-m)\ell-1-m$	4	2^m-2-m
Example 3.3.4	$(2^m-m)\ell$	$(2^m-1-m)\ell-1-m$	4	2^m-1-m

construction by using the multi-level tensor product structure [40], leading to $(r,1)_a$-LRCs with higher minimum distance.

We start by presenting the tensor product operation of two matrices $H^{'}$ and $H^{''}$. Let $H^{'}$ be the parity-check matrix of a binary code with length n' and dimension $n'-v$. $H^{'}$ can be considered as a v (row) by n' (column) matrix over $\mathbb{F}_2$ or as a 1 (row) by n' (column) matrix of elements from $\mathbb{F}_{2^v}$. Let $H^{'}=[h_1^{'}\ h_2^{'}\ \cdots\ h_{n'}^{'}]$, where $h_j^{'}$, $1\leq j\leq n'$, are elements of $\mathbb{F}_{2^v}$. Let $H^{''}$ be the parity-check matrix of a code of length ℓ and dimension $\ell-\lambda$ over $\mathbb{F}_{2^v}$. We denote $H^{''}$ by

$$H^{''}=\begin{bmatrix} h_{11}^{''} & \cdots & h_{1\ell}^{''} \\ \vdots & \ddots & \vdots \\ h_{\lambda 1}^{''} & \cdots & h_{\lambda\ell}^{''} \end{bmatrix},$$

where $h_{ij}^{''}$, $1\leq i\leq\lambda$ and $1\leq j\leq\ell$, are elements of $\mathbb{F}_{2^v}$.

The tensor product of the two matrices $H^{''}$ and $H^{'}$ is defined as

$$H^{''}\bigotimes H^{'}=\begin{bmatrix} h_{11}^{''}H^{'} & \cdots & h_{1\ell}^{''}H^{'} \\ \vdots & \ddots & \vdots \\ h_{\lambda 1}^{''}H^{'} & \cdots & h_{\lambda\ell}^{''}H^{'} \end{bmatrix},$$

where $h_{ij}^{''}H^{'}=[h_{ij}^{''}h_1^{'}\ h_{ij}^{''}h_2^{'}\ \cdots\ h_{ij}^{''}h_{n'}^{'}]$, $1\leq i\leq\lambda$ and $1\leq j\leq\ell$, and the products of elements are calculated according to the rules of multiplication for elements over $\mathbb{F}_{2^v}$.

Our construction of $(r,1)_a$-LRCs is based on the multi-level tensor product structure proposed

in [40]. Define the matrices $H_i^{'}$ and $H_i^{''}$ $(i = 1,2,\ldots,\mu)$ as follows. $H_i^{'}$ is a $v_i \times n'$ matrix over $\mathbb{F}_2$ such that the $(v_1 + v_2 + \cdots + v_i) \times n'$ matrix

$$B_i = \begin{bmatrix} H_1^{'} \\ H_2^{'} \\ \vdots \\ H_i^{'} \end{bmatrix}$$

is a parity-check matrix of an $[n', n' - v_1 - v_2 - \cdots - v_i, d_i^{'}]$ binary code. $H_i^{''}$ is a $\lambda_i \times \ell$ matrix over $\mathbb{F}_{2^{v_i}}$, which is a parity-check matrix of an $[\ell, \ell - \lambda_i, \delta_i]_{2^{v_i}}$ code.

We define a μ-level tensor product code as a binary linear code having a parity-check matrix in the form of the following μ-level tensor product structure

$$H = \begin{bmatrix} H_1^{''} \otimes H_1^{'} \\ H_2^{''} \otimes H_2^{'} \\ \vdots \\ H_\mu^{''} \otimes H_\mu^{'} \end{bmatrix}. \tag{3.6}$$

We denote this code by $\mathcal{C}_{TP}^{\mu}$. Its length is $n = n'\ell$ and the number of parity-check symbols is $n - k = \sum_{i=1}^{\mu} v_i \lambda_i$.

Let us give an example of a 2-level tensor product code $\mathcal{C}_{TP}^{2}$.

Example 3.3.5. Let $H_1^{'} = [1\ 1\ 1\ 1\ 1\ 1\ 1]$ over $\mathbb{F}_2$, and

$$H_2^{'} = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

over $\mathbb{F}_2$. Let $H_1^{''} = [1\ 1\ 1]$ over $\mathbb{F}_2$ and

$$H_2^{''} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

over $\mathbb{F}_8$. Hence, in this construction, we use the following parameters: $n' = 7$, $\ell = 3$, $v_1 = 1$, $v_2 = 3$, $\lambda_1 = 1$, $\lambda_2 = 2$, $\delta_1 = 2$, $\delta_2 = 3$, $d'_1 = 2$ and $d'_2 = 4$. The binary parity-check matrix H of the 2-level tensor product code $\mathcal{C}_{TP}^2$ is

$$H = \begin{bmatrix} H''_1 \otimes H'_1 \\ H''_2 \otimes H'_2 \end{bmatrix} = \begin{bmatrix} 1\,1\,1\,1\,1\,1\,1 & 1\,1\,1\,1\,1\,1\,1 & 1\,1\,1\,1\,1\,1\,1 \\ 0\,0\,0\,1\,1\,1\,1 & 0\,0\,0\,1\,1\,1\,1 & 0\,0\,0\,0\,0\,0\,0 \\ 0\,1\,1\,0\,0\,1\,1 & 0\,1\,1\,0\,0\,1\,1 & 0\,0\,0\,0\,0\,0\,0 \\ 1\,0\,1\,0\,1\,0\,1 & 1\,0\,1\,0\,1\,0\,1 & 0\,0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,1\,1\,1\,1 & 0\,0\,0\,0\,0\,0\,0 & 0\,0\,0\,1\,1\,1\,1 \\ 0\,1\,1\,0\,0\,1\,1 & 0\,0\,0\,0\,0\,0\,0 & 0\,1\,1\,0\,0\,1\,1 \\ 1\,0\,1\,0\,1\,0\,1 & 0\,0\,0\,0\,0\,0\,0 & 1\,0\,1\,0\,1\,0\,1 \end{bmatrix}.$$

The code length is $n = n'\ell = 21$ and the dimension is $k = n - \sum_{i=1}^{2} v_i\lambda_i = 14$. It is possible to verify that every 3 columns of H are linearly independent, but columns 1, 2, 5 and 6 of H are linearly dependent. Therefore, the minimum distance of the code is $d = 4$. □

Next, we give the following lemma on the minimum distance of a μ-level tensor product code $\mathcal{C}_{TP}^{\mu}$.

Lemma 3.3.10. *Assume the following inequalities hold: 1) $d'_\mu \leq \delta_1$, and 2) $d'_\mu \leq \delta_i d'_{i-1}$, for $i = 2,3,\ldots,\mu$. Then, the minimum distance d of the μ-level tensor product code $\mathcal{C}_{TP}^{\mu}$ is d'_μ.*

Proof. First, we show that $d \leq d'_\mu$. For $i = 1,2,\ldots,\mu$, let $H'_i = [h'_1(i), h'_2(i), \ldots, h'_{n'}(i)]$ over $\mathbb{F}_{2^{v_i}}$, and let $[h''_{11}(i), h''_{21}(i), \ldots, h''_{\lambda_i 1}(i)]^T$ over $\mathbb{F}_{2^{v_i}}$ be the first column of H''_i. Since the code with parity-check matrix B_μ has minimum distance d'_μ, there exist d'_μ columns of B_μ, say in the set of positions $J = \{b_1, b_2, \ldots, b_{d'_\mu}\}$, which are linearly dependent. That is $\sum_{j\in J} h'_j(i) = 0$, for $i = 1,2,\ldots,\mu$. Thus, we have $\sum_{j\in J} h''_{p1}(i)h'_j(i) = h''_{p1}(i)\left(\sum_{j\in J} h'_j(i)\right) = 0$, for $p = 1,2,\ldots,\lambda_i$ and $i = 1,2,\ldots,\mu$. That is, the columns in positions $b_1, b_2, \ldots, b_{d'_\mu}$ of H are linearly dependent.

The inequality $d \geq d'_\mu$ is shown in the proof of Theorem 2 in [40]. ■

Remark 3.3.1. Lemma 3.3.10 is a modified version of Theorem 2 in [40], which incorrectly states that the minimum distance of a μ-level tensor product code $\mathcal{C}_{TP}^{\mu}$ is the largest integer d_m satisfying the following

inequalities: 1) $d_m \leq d'_\mu$, 2) $d_m \leq \delta_1$, and 3) $d_m \leq \delta_i d'_{i-1}$, $i = 2,3,\ldots,\mu$. If this were true, the 2-level $\mathcal{C}^2_{TP}$ code in Example 3.3.5, with $\delta_1 = 2$, $\delta_2 = 3$, $d'_1 = 2$ and $d'_2 = 4$, would have minimum distance 2. However, the true minimum distance is 4. Theorem 2 only gives a lower bound on the minimum distance, which we have used in the proof of Lemma 3.3.10. □

We now present a construction of $(r,1)_a$-LRCs based on the multi-level tensor product structure.

Construction D

Step 1: Choose $v_i \times n'$ matrices H'_i over $\mathbb{F}_2$ and $\lambda_i \times \ell$ matrices H''_i over $\mathbb{F}_{2^{v_i}}$, for $i = 1,2,\ldots,\mu$, which satisfy the following two properties:

1) $H'_1 = [1,1,\cdots,1]$, i.e., a length-n' all-one vector, and $H''_1 = \mathbf{I}_{\ell\times\ell}$, i.e., an $\ell \times \ell$ identity matrix.

2) The matrices H'_i and H''_i are chosen such that $d'_\mu \leq \delta_i d'_{i-1}$, for $i = 2,3,\ldots,\mu$.

Step 2: Generate the parity-check matrix H of the $(r,1)_a$-LRC according to (3.6) with the matrices H'_i and H''_i, for $i = 1,2,\ldots,\mu$. ■

Theorem 3.3.11. *The binary $(r,1)_a$-LRC from Construction D has length $n = n'\ell$, dimension $k = n'\ell - \sum_{i=1}^{\mu} v_i\lambda_i$, minimum distance $d = d'_\mu$, and all-symbol locality $r = n' - 1$.*

Proof. According to Construction D, the code length $n = n'\ell$ and dimension $k = n'\ell - \sum_{i=1}^{\mu} v_i\lambda_i$ are determined by the construction of the multi-level tensor product codes. From property 1) in Step 1, the tensor product matrix $H''_1 \bigotimes H'_1$ in H gives all-symbol locality $r = n' - 1$. Since $\delta_1 = \infty$ (H''_1 is the identity matrix), $d'_1 = 2$, and $d'_\mu \leq \delta_i d'_{i-1}$, we conclude from Lemma 3.3.10, that the minimum distance of the constructed $(r,1)_a$-LRC is $d = d'_\mu$. ■

Construction D gives a general method to construct $(r,1)_a$-LRCs, but not an explicit construction. Next, we give a specific code design.

Let $n' = 2^m - 1$ and α be a primitive element of $\mathbb{F}_{2^m}$. In Construction D, for $i = 2,3,\ldots,\mu$, we choose $H'_i = [\beta^0, \beta^1, \cdots, \beta^{n'-1}]$ where $\beta = \alpha^{2i-3}$. Thus, B_i is the parity-check matrix of an expurgated binary BCH code, so we have $d'_i = 2i$. We also choose H''_i to be the parity-check matrix of an $[\ell, \ell - \lambda_i, \delta_i = \lceil \frac{\mu}{i-1} \rceil]_{2^m}$ code, so we have $d'_\mu = 2\mu \leq \delta_i d'_{i-1} = 2(i-1)\lceil \frac{\mu}{i-1} \rceil$. We refer to the $(r,1)_a$-LRC constructed according to the above design as $\mathcal{C}_{LRC}$, and conclude with the following corollary.

Corollary 3.3.12. *The $(r,1)_a$-LRC $\mathcal{C}_{LRC}$ has parameters $[(2^m-1)\ell,(2^m-2)\ell - m\sum_{i=2}^{\mu}\lambda_i, 2\mu]$ and all-symbol locality $r = 2^m-2$.*

In particular, for the construction of the $\mathcal{C}_{LRC}$, in order to minimize the value of λ_i, we can choose $H_i^{''}$ to be the parity-check matrix of an $[\ell, \ell-\delta_i+1, \delta_i = \lceil \frac{\mu}{i-1} \rceil]_{2^m}$ MDS code, where we require that $\ell \leq 2^m+1$ only for the case $\mu > 2$. Thus, the resulting $(r,1)_a$-LRC has parameters $[(2^m-1)\ell, (2^m-2)\ell - m\sum_{i=2}^{\mu}(\lceil \frac{\mu}{i-1} \rceil - 1), 2\mu]$ and all-symbol locality $r = 2^m-2$. We refer to this particular $(r,1)_a$-LRC as $\mathcal{C}_{\mathrm{I}}$. We give some instances of $\mathcal{C}_{\mathrm{I}}$ as follows.

Example 3.3.6. For $\mu = 2$, $\mathcal{C}_{\mathrm{I}}$ is a $[(2^m-1)\ell, (2^m-2)\ell - m, 4]$ LRC with $r = 2^m-2$. It has an optimal minimum distance with respect to bound (3.4). For $\mu = 3$ and $\ell \leq 2^m+1$, $\mathcal{C}_{\mathrm{I}}$ is a $[(2^m-1)\ell, (2^m-2)\ell - 3m, 6]$ LRC with $r = 2^m-2$. For $\mu = 4$ and $\ell \leq 2^m+1$, $\mathcal{C}_{\mathrm{I}}$ is a $[(2^m-1)\ell, (2^m-2)\ell - 5m, 8]$ LRC with $r = 2^m-2$. □

In the design of the code $\mathcal{C}_{LRC}$, we can also choose $H_i^{''}$ to be the parity-check matrix of a non-MDS code to remove the length constraint on ℓ. We illustrate this design with the following example.

Example 3.3.7. For the $\mathcal{C}_{LRC}$ with $\mu = 3$, we choose $H_2^{''}$ to be the parity-check matrix of an $[\ell = \frac{2^{ms}-1}{2^m-1}, \frac{2^{ms}-1}{2^m-1} - s, 3]_{2^m}$ non-binary Hamming code and $H_3^{''} = [1,1,\cdots,1]$. The resulting $(r,1)_a$-LRC has parameters $[2^{ms}-1, \frac{(2^m-2)(2^{ms}-1)}{2^m-1} - (s+1)m, 6]$ and all-symbol locality $r = 2^m-2$. For the $\mathcal{C}_{LRC}$ with $\mu = 4$, we choose $H_2^{''}$ to be the parity-check matrix of an $[\ell = 2^{2m}+1, 2^{2m}-3, 4]_{2^m}$ non-binary code (see problem 3.44 in [64]), $H_3^{''} = [1,1,\cdots,1]$, and $H_4^{''} = [1,1,\cdots,1]$. The resulting $(r,1)_a$-LRC has parameters $[(2^{2m}+1)(2^m-1), (2^{2m}+1)(2^m-2) - 6m, 8]$ and all-symbol locality $r = 2^m-2$. In general, we can choose the matrix $H_i^{''}$ for $i = 2,3,\ldots,\mu$ to be the parity-check matrix of an $[\ell = 2^{ms}-1, \ell - \lambda_i \geq \ell - s(\lceil \frac{\mu}{i-1} \rceil - 1), \delta_i \geq \lceil \frac{\mu}{i-1} \rceil]_{2^m}$ non-binary BCH code [49]. The resulting $(r,1)_a$-LRC has parameters $[n = (2^{ms}-1)(2^m-1), k \geq (2^{ms}-1)(2^m-2) - ms\sum_{i=2}^{\mu}(\lceil \frac{\mu}{i-1} \rceil - 1), d = 2\mu]$ and all-symbol locality $r = 2^m-2$. We refer to this code as $\mathcal{C}_{\mathrm{I}}'$. □

Remark 3.3.2. There exist other choices of the matrices $H_i^{'}$ and $H_i^{''}$ in Construction D. For example, we can choose $H_i^{'}$ so that B_i is the parity-check matrix of an extended binary BCH code, and choose $H_i^{''}$ to be the parity-check matrix of an MDS code. Then, the resulting $(r,1)_a$-LRC has parameters

Table 3.2: Constructed binary $(r,1)_a$-LRCs in Section 3.3.2.

Code	n	k	d	r
$\mathcal{C}_{\rm I}$	$(2^m-1)\ell$	$(2^m-2)\ell - m\sum_{i=2}^{\mu}(\lceil \frac{\mu}{i-1} \rceil - 1)$	2μ	2^m-2
$\mathcal{C}_{\rm I}(\mu=2)$	$(2^m-1)\ell$	$(2^m-2)\ell - m$	4	2^m-2
$\mathcal{C}_{\rm I}(\mu=3)$	$(2^m-1)\ell$	$(2^m-2)\ell - 3m$	6	2^m-2
$\mathcal{C}_{\rm I}(\mu=4)$	$(2^m-1)\ell$	$(2^m-2)\ell - 5m$	8	2^m-2
$\mathcal{C}_{\rm II}$	$2^m\ell$	$(2^m-1)\ell - m\sum_{i=2}^{\mu}(\lceil \frac{\mu}{i-1} \rceil - 1)$	2μ	2^m-1
$\mathcal{C}_{\rm II}(\mu=2)$	$2^m\ell$	$(2^m-1)\ell - m$	4	2^m-1
$\mathcal{C}_{\rm II}(\mu=3)$	$2^m\ell$	$(2^m-1)\ell - 3m$	6	2^m-1
$\mathcal{C}_{\rm II}(\mu=4)$	$2^m\ell$	$(2^m-1)\ell - 5m$	8	2^m-1
Example 3.3.7	$2^{ms}-1$	$\frac{(2^m-2)(2^{ms}-1)}{2^m-1} - (s+1)m$	6	2^m-2
Example 3.3.7	$(2^{2m}+1)(2^m-1)$	$(2^{2m}+1)(2^m-2) - 6m$	8	2^m-2
$\mathcal{C}'_{\rm I}$	$(2^{ms}-1)(2^m-1)$	$(2^{ms}-1)(2^m-2) - ms\sum_{i=2}^{\mu}(\lceil \frac{\mu}{i-1} \rceil - 1)$	2μ	2^m-2

$[2^m\ell, (2^m-1)\ell - m\sum_{i=2}^{\mu}(\lceil \frac{\mu}{i-1} \rceil - 1), 2\mu]$ and all-symbol locality $r = 2^m-1$, where we require that $\ell \leq 2^m+1$ if $\mu > 2$. We refer to this code as $\mathcal{C}_{\rm II}$. □

The $(r,1)_a$-LRCs constructed in this subsection are summarized in Table 3.2, where for $\mathcal{C}_{\rm I}$ and $\mathcal{C}_{\rm II}$, we require $\ell \leq 2^m+1$ when $\mu > 2$.

3.3.3 Comparison to Existing Results

In this subsection, we summarize our constructions of binary LRCs and compare them with previous results.

Our constructions of binary $(r,1)_i$-LRCs and $(r,1)_a$-LRCs have the following features.

1) They provide LRCs with a wide range of values of minimum distance and locality. This diversity is based on the flexible choices of the base code $\mathcal{C}_{base}$ for Construction A, B, C, and of the matrices $H_i^{'}$ and $H_i^{''}$ for Construction D. This feature of our constructions makes it possible to satisfy different design requirements on the code parameters.

2) They produce high-rate LRCs. For example, for the family of code $\mathcal{C}_{\rm I}(\mu=2)$, its code rate asymptotically approaches $\frac{r}{r+1}$ as $\ell \to \infty$. Moreover, for all of the constructed binary LRCs with $d=3$ or $d=4$, the minimum distance is optimal with respect to bound (3.4).

There exist several other constructions of binary $(r,1)_a$-LRCs, which are summarized in Table 3.3. Goparaju and Calderbank [31] and Zeh and Yaakobi [92] focused on constructing high-rate binary $(r,1)_a$-

Table 3.3: Existing constructions of binary $(r,1)_a$-LRCs.

Code	n	k	d	r
[31]	2^m-1 $(2\|m)$	$\frac{2}{3}(2^m-1)-m$	6	2
[31]	2^m-1 $(2\|m)$	$\frac{2}{3}(2^m-1)-2m$	10	2
[92]	2^m+1 $(2\nmid m)$	$\frac{2}{3}(2^m+1)-2m$	10	2
[92]	$(2^r+1)(r+1)$	$(2^r-1)r$	6	r
[72]	$2^m-\binom{s}{2}-1$ $(s\leq m)$	m	$2^{m-1}-\lfloor\frac{s^2}{4}\rfloor$	2
[72]	2^m-2^t+t+1 $(t\leq m)$	m	$2^{m-1}-2^{t-1}+2$	2
[72]	$2^{m-1}-1$	m	$2^{m-2}-1$	3
[72]	$3\cdot 2^{m-2}$	m	$3\cdot 2^{m-3}$	2
[76]	45	30	4	8
[76]	21	12	4	5

LRCs with fixed small locality 2 and small minimum distance. In [92], another construction for LRCs with arbitrary locality and fixed minimum distance 6 was given. In contrast, Silberstein and Zeh [72] proposed constructions of low-rate binary $(r,1)_a$-LRCs with fixed small locality but large minimum distance. In [76], Tamo et al. gave some specific examples of cyclic binary $(r,1)_a$-LRCs from subfield subcodes.

Compared to these previous code constructions, our constructions offer more flexibility with regard to the possible code parameters. First, we compare our results to those in [31, 92]. Roughly speaking, for a given length and minimum distance, our codes generally offer higher rate but at the cost of larger locality. For example, Goparaju et al. give a $[255,162,6]$ LRC with locality $r=2$ and rate 0.6353. Zeh et al. give a $[198,155,6]$ LRC with locality $r=5$ and rate 0.7828. By comparison, referring to Table 3.2, we can use $\mathcal{C}_{\rm I}(\mu=3)$ and parameters $m=4$ and $\ell=16$ to construct a $[240,212,6]$ LRC with locality $r=14$ and rate 0.8833.

We also compare our constructions to some of those examples given in [76]. One example is a $[45,30,4]$ binary $(r,1)_a$-LRC with $r=8$, while we can construct a $[45,35,4]$ binary $(r,1)_a$-LRC with $r=8$ from Construction C using a $[13,8,4]$ binary base code $\mathcal{C}_{base}$. Another example in [76] is a $[21,12,4]$ binary $(r,1)_a$-LRC with $r=5$. In contrast, we can construct a $[20,12,4]$ binary $(r,1)_a$-LRC with $r=4$ from Construction C using an $[8,4,4]$ binary base code $\mathcal{C}_{base}$. In these cases, our codes offer higher rates with the same or smaller locality.

Finally, we apply bound (2.2) to give an upper bound on the dimension of the constructed $(r,1)_a$-LRC $\mathcal{C}_{\rm I}$ from Construction D, which has parameters $[n=(2^m-1)\ell, k=(2^m-2)\ell-m\sum_{i=2}^{\mu}(\lceil\frac{\mu}{i-1}\rceil-$

$1), d = 2\mu]$ and $r = 2^m - 2$. For $x = \ell - 1$, bound (2.2) gives an upper bound k_{ub},

$$\begin{aligned} k_{ub} =& xr + k_{opt}^{(q)}(n - x(r+1), d) \\ =& (2^m - 2)(\ell - 1) + k_{opt}^{(2)}\big((2^m - 1)\ell - (2^m - 1)(\ell - 1), 2\mu\big) \\ =& (2^m - 2)(\ell - 1) + k_{opt}^{(2)}\big(2^m - 1, 2\mu\big) \\ \overset{(a)}{\geq}& (2^m - 2)(\ell - 1) + 2^m - 2 - (\mu - 1)m \\ =& (2^m - 2)\ell - (\mu - 1)m, \end{aligned}$$

where step (a) follows from the existence of an $[n = 2^m - 1, k = 2^m - 2 - (\mu - 1)m, d = 2\mu]$ expurgated BCH code.

For small μ, the gap between k and the upper bound k_{ub} is small, e.g., for $\mu = 3$, $k_{ub} - k = m$, and for $\mu = 4$, $k_{ub} - k = 2m$. For large μ, the gap between k and k_{ub} becomes large.

3.4 Binary LRCs with Availability

In this section, we study binary $(r, t)_a$-LRCs based on one-step majority-logic decodable codes [47].

Definition 3.4.1. *An $[n, k, d]_q$ linear code $\mathcal{C}$ is said to be a one-step majority-logic decodable code with t orthogonal repair sets if the ith symbol, for $i \in [n]$, has t pairwise disjoint repair sets $\mathcal{R}_i^j$, $j \in [t]$, such that for every $j \in [t]$ the ith symbol is a linear combination of all symbols in $\mathcal{R}_i^j$.*

According to Definition 3.4.1, it is evident that if $\mathcal{C}$ is a one-step majority-logic decodable code with t orthogonal repair sets, and if the size of all repair sets is at most r, then $\mathcal{C}$ has all-symbol locality r and availability t. Moreover, referring to a well known result (Theorem 8.1 in [47]), we can see that for an $[n, k, d]_q$ one-step majority-logic decodable code with t orthogonal repair sets, all of the same size r, the availability t satisfies

$$t \leq \left\lfloor \frac{n-1}{r} \right\rfloor. \tag{3.7}$$

Table 3.4: Difference-set codes.

$\mathcal{C}$	n	k	d	r	t	t^u	d^u	d_1^u	d_2^u
$m=2$	21	11	6	4	5	5	6	8	9
$m=3$	73	45	10	8	9	9	12	23	24
$m=4$	273	191	18	16	17	17	31	71	72
$m=5$	1057	813	34	32	33	33	80	219	220

Note that for a cyclic code, once t repair sets are found for one symbol, the repair sets for all other symbols can be determined correspondingly from the cyclic symmetry of the code. Therefore, most of one-step majority-logic decodable codes found so far are cyclic codes. There are several constructions of one-step majority-logic decodable codes, such as doubly transitive invariant (DTI) codes, cyclic simplex codes, cyclic difference-set codes, and 4-cycle free regular linear codes [47]. The following examples present two families of one-step majority-logic decodable cyclic codes, and we give their locality and availability.

Example 3.4.1. Consider a cyclic binary simplex code with parameters $[n = 2^m - 1, k = m, d = 2^{m-1}]$. It is a one-step majority-logic decodable code with $2^{m-1} - 1$ disjoint repair sets [47]. It is easy to verify that every repair set has size 2. Therefore, it has all-symbol locality $r = 2$ and availability $t = 2^{m-1} - 1$. This code has the optimal minimum distance, due to the Plotkin bound. This locality and availability property of the simplex codes was also observed independently in [43]. □

Example 3.4.2. Consider a cyclic binary difference-set code with parameters $[n = 2^{2m} + 2^m + 1, k = 2^{2m} + 2^m - 3^m, d = 2^m + 2]$. It is a one-step majority-logic decodable code with $2^m + 1$ disjoint repair sets [47]. We can verify that every repair set has size 2^m. Thus, this code has all-symbol locality $r = 2^m$ and availability $t = 2^m + 1$. For the codes with $2 \leq m \leq 5$, Table 3.4 gives the upper bound t^u on t from bound (3.7) and the upper bound d^u on d from bound (3.4). The table also gives the upper bounds d_1^u and d_2^u from bounds (3.1) and (3.3), respectively. In all of the examples, we see that d^u is smaller than d_1^u and d_2^u, meaning that bound (3.4) is tighter. □

Another important class of one-step majority-logic decodable codes is 4-cycle free linear codes that have a parity-check matrix H with constant row weight ρ and constant column weight γ. Obviously, such codes have all-symbol locality $r = \rho - 1$ and availability $t = \gamma$. In particular, 4-cycle free (ρ, γ)-regular

Table 3.5: Two-dimensional type-I cyclic $(0,m)$th-order EG-LDPC codes.

$\mathcal{C}$	n	k	d	r	t	t^u	d^u	d_1^u	d_2^u
$m=2$	15	7	5	3	4	4	5	7	7
$m=3$	63	37	9	7	8	8	12	22	22
$m=4$	255	175	17	15	16	16	30	69	70
$m=5$	1023	781	33	31	32	32	80	218	218

low-density parity-check (LDPC) codes have this property. Based upon this observation, a family of codes with all-symbol locality and availability were constructed using partial geometries in [54]. The authors of [54] also derived lower and upper bounds on the code rate; however, the exact dimension and minimum distance of these codes are still not known.

Many 4-cycle free regular LDPC codes have been constructed by leveraging different mathematical tools, e.g., finite geometries, algebraic methods, and block designs [47]. Here we consider a family of such codes based on Euclidean geometry (EG), and we give explicit expressions for their code length, dimension, and minimum distance, as well as their locality and availability.

Example 3.4.3. Consider the class of binary 4-cycle free regular LDPC codes called in [47] the two-dimensional type-I cyclic $(0,m)$th-order EG-LDPC codes, with parameters $[n=2^{2m}-1, k=2^{2m}-3^m, d=2^m+1]$. From the structure of their parity-check matrices, they have all-symbol locality $r=2^m-1$ and availability $t=2^m$. Table 3.5 lists the parameters of these codes for $2 \leq m \leq 5$ and gives the upper bound t^u on t from bound (3.7) and the upper bound d^u on d from bound (3.4). The table also includes the upper bounds d_1^u and d_2^u from bounds (3.1) and (3.3), respectively. We see that, in all of these cases, d^u is smaller than d_1^u and d_2^u. □

Finally, we briefly show how to get a long LRC with availability from a short one-step majority-logic decodable code based on a multi-level tensor product structure. We modify Step 1 in Construction D to provide availability by using the parity-check matrix of a one-step majority-logic decodable code as $H_1^{'}$. We illustrate this modification with the following example where we use for $H_1^{'}$ the parity-check matrix of the $[15,7,5]$ binary BCH code, which is a one-step majority-logic decodable code with all-symbol locality $r=3$ and availability $t=4$ [47].

Example 3.4.4. Let $n' = 15$ and α be a primitive element of $\mathbb{F}_{16}$. Let

$$H_1^{'} = \begin{bmatrix} \alpha^0 & \alpha^1 & \cdots & \alpha^{14} \\ (\alpha^3)^0 & (\alpha^3)^1 & \cdots & (\alpha^3)^{14} \end{bmatrix}$$

and $H_2^{'} = [(\alpha^5)^0, (\alpha^5)^1, \cdots, (\alpha^5)^{14}]$. Let $H_1^{''} = \mathbf{I}_{\ell \times \ell}$ and $H_2^{''} = [1, 1, \cdots, 1]$. The parity-check matrix H of the constructed LRC is

$$H = \begin{bmatrix} H_1^{''} \otimes H_1^{'} \\ H_2^{''} \otimes H_2^{'} \end{bmatrix}.$$

This LRC has parameters $[15\ell, 7\ell - 2, 7]$ with all-symbol locality $r = 3$ and availability $t = 4$. □

3.5 Conclusion

In this chapter, we presented several constructions of binary LRCs by using phantom parity-check symbols and a multi-level tensor product structure. Compared to other recently proposed schemes which produce binary LRCs with fixed minimum distance or locality, our constructions are more flexible and offer wider choices of the code parameters, i.e., code length, dimension, minimum distance, and locality. We also showed that our binary LRCs with minimum distance 3 or 4 are optimal with respect to the minimum distance. Finally, we studied the locality and availability properties of one-step majority-logic decodable codes, and demonstrated a construction of a long binary LRC with availability from a short one-step majority-logic decodable code.

3.6 Appendix A

In this section, we give the proof of Lemma 3.2.3.

Proof. Assume that $x \in \mathbb{Z}^+, \boldsymbol{y} = (y_1, \ldots, y_x) \in ([t])^x$ satisfy the condition $x \leq \lceil \frac{k}{(r-1)t+1} \rceil$ in the lemma. Also, assume without loss of generality that the first k symbols of the code $\mathcal{C}$ form an information set.

The set $\mathcal{I}$ is constructed according to the following procedure.

Procedure A

1) Let $\mathcal{I}_0 = \emptyset$.

2) **For** $j = 1, \ldots, x$

3) Choose an integer $a_j \in [k]$ and $a_j \notin \mathcal{I}_{j-1}$, such that $k_{\mathcal{I}_{j-1} \cup \{a_j\}} = k_{\mathcal{I}_{j-1}} + 1$.

4) $\mathcal{I}_j = \mathcal{I}_{j-1} \cup \{a_j\} \cup \mathcal{R}_{a_j}^1 \cup \cdots \cup \mathcal{R}_{a_j}^{y_j}$.

5) **End**

6) Let $\mathcal{I} = \mathcal{I}_x \cup \mathcal{S}$, where $\mathcal{S} \subseteq [n] \setminus \mathcal{I}_x$ is a set of cardinality $\min\{n, B(r,x,\boldsymbol{y})\} - |\mathcal{I}_x|$. □

This completes the construction of the set $\mathcal{I}$.

First, let us show that the construction of the set $\mathcal{I}$ is well defined.

Claim 3.6.1. *In step 3), it is always possible to find a coordinate $a_j \in [k]$, for $1 \leq j \leq x$, that satisfies the condition in this step.*

Proof. To see this, we show that on the jth loop, for $1 \leq j \leq x$, the value of $k_{\mathcal{I}_{j-1}}$ satisfies $k_{\mathcal{I}_{j-1}} < k$, and thus at least one of the first k coordinates does not belong to the set $\mathcal{I}_{j-1}$. Since the value of $k_{\mathcal{I}_{j-1}}$ increases with j, it is enough to show that $k_{\mathcal{I}_{x-1}} \leq k-1$.

Let $\mathcal{S}_{a_j} = \{a_j\} \cup \mathcal{R}_{a_j}^1 \cup \cdots \cup \mathcal{R}_{a_j}^{y_j}$ for $j \in [x]$. First, we show that $k_{\mathcal{S}_{a_j}} \leq (r-1)t+1$. Let $G = [g_1, \ldots, g_n]$ be a generator matrix of the code $\mathcal{C}$. For the repair set $\mathcal{R}_{a_j}^u$, $u \in [y_j]$, g_{a_j} is a linear combination of the columns g_m, $m \in \mathcal{R}_{a_j}^u$, so there exists a coordinate $b_j^u \in \mathcal{R}_{a_j}^u$ such that $g_{a_j} = \sum_{m \in \mathcal{R}_{a_j}^u \setminus \{b_j^u\}} \alpha_m g_m + \beta_{b_j^u} g_{b_j^u}$, where $\alpha_m, \beta_{b_j^u} \in \mathbb{F}_q$ and $\beta_{b_j^u} \neq 0$. Thus, $k_{\{a_j\} \cup \mathcal{R}_{a_j}^u \setminus \{b_j^u\}} = k_{\{a_j\} \cup \mathcal{R}_{a_j}^u}$. Therefore, we have

$$k_{\mathcal{S}_{a_j}} = k_{\mathcal{S}_{a_j} \setminus \{\cup_{u=1}^{y_j} b_j^u\}} \overset{(a)}{\leq} |\mathcal{S}_{a_j} \setminus \{\cup_{u=1}^{y_j} b_j^u\}| \leq (r-1)y_j + 1 \leq (r-1)t+1,$$

where (a) follows from the fact that $k_{\mathcal{M}} \leq |\mathcal{M}|$ for any set $\mathcal{M} \subseteq [n]$.

From the construction of the set $\mathcal{I}$, we have that $\mathcal{I}_{x-1} = \cup_{j=1}^{x-1} \mathcal{S}_{a_j}$ and therefore

$$\begin{aligned} k_{\mathcal{I}_{x-1}} = k_{\cup_{j=1}^{x-1} \mathcal{S}_{a_j}} &\overset{(a)}{\leq} \sum_{j=1}^{x-1} k_{\mathcal{S}_{a_j}} \overset{(b)}{\leq} (x-1)[(r-1)t+1] \\ &\overset{(c)}{\leq} \left(\left\lceil \frac{k}{(r-1)t+1} \right\rceil - 1 \right) [(r-1)t+1] < \frac{k}{(r-1)t+1} [(r-1)t+1] = k, \end{aligned}$$

where (a) follows from the fact that $k_{\mathcal{M}_1 \cup \mathcal{M}_2} \leq k_{\mathcal{M}_1} + k_{\mathcal{M}_2}$ for any sets $\mathcal{M}_1, \mathcal{M}_2 \subseteq [n]$ and a simple induction. Inequality (b) follows from $k_{\mathcal{S}_{a_j}} \leq (r-1)t+1$, and (c) follows from $x \leq \lceil \frac{k}{(r-1)t+1} \rceil$. ■

It is clear to see that the set $\mathcal{I}$ has size of $|\mathcal{I}| = \min\{n, B(r,x,\boldsymbol{y})\}$.

Next, we show that $k_{\mathcal{I}} \leq A(r,x,\boldsymbol{y})$. To do this, in Procedure A, for each jth iteration, let us add the following coordinate selection steps between step 3) and step 4).

3.1) **For** $\ell = 1, \ldots, y_j$

3.2) Choose an integer $a_j^\ell \in \mathcal{R}_{a_j}^\ell$ and $a_j^\ell \notin \mathcal{I}_{j-1}$, such that $k_{\{a_j\} \cup \mathcal{R}_{a_j}^\ell \setminus \{a_j^\ell\}} = k_{\{a_j\} \cup \mathcal{R}_{a_j}^\ell}$.

3.3) **End**

We next show that the above steps are well defined.

Claim 3.6.2. *In step 3.2), it is always possible to find an integer a_j^ℓ, for $1 \leq j \leq x$ and $1 \leq \ell \leq y_j$, that satisfies the condition in this step.*

Proof. First, assume on the contrary that $\mathcal{R}_{a_j}^\ell \subseteq \mathcal{I}_{j-1}$. Then, we conclude that $k_{\mathcal{I}_{j-1} \cup \{a_j\}} = k_{\mathcal{I}_{j-1}}$, which violates the selection rule in step 3). Second, for the case of $\mathcal{R}_{a_j}^\ell \nsubseteq \mathcal{I}_{j-1}$, since g_{a_j} is a linear combination of g_i, $i \in \mathcal{R}_{a_j}^\ell$, there exists at least one coordinate $a_j^\ell \in \mathcal{R}_{a_j}^\ell$ and $a_j^\ell \notin \mathcal{I}_{j-1}$ such that $g_{a_j} = \sum_{i \in \mathcal{R}_{a_j}^\ell \setminus \{a_j^\ell\}} \alpha_i g_i + \beta_{a_j^\ell} g_{a_j^\ell}$, where $\alpha_i, \beta_{a_j^\ell} \in \mathbb{F}_q$ and $\beta_{a_j^\ell} \neq 0$. Therefore, $g_{a_j^\ell}$ can be expressed as a linear combination of the columns g_i, for $i \in \{a_j\} \cup \mathcal{R}_{a_j}^\ell \setminus \{a_j^\ell\}$, so we have $k_{\{a_j\} \cup \mathcal{R}_{a_j}^\ell \setminus \{a_j^\ell\}} = k_{\{a_j\} \cup \mathcal{R}_{a_j}^\ell}$. ■

Now, let $\mathcal{P}$ be the set of coordinates chosen in steps 3.1) – 3.3): $\mathcal{P} = \{a_1^1, \ldots, a_1^{y_1}, \ldots, a_x^1, \ldots, a_x^{y_x}\}$. From the construction, the integers $a_1^1, \ldots, a_1^{y_1}, \ldots, a_x^1, \ldots, a_x^{y_x}$ are all different, i.e., $|\mathcal{P}| = \sum_{j=1}^x y_j$.

Next, we prove that $k_{\mathcal{I}} \leq A(r,x,\boldsymbol{y})$ by showing that $k_{\mathcal{I} \setminus \mathcal{P}} \leq A(r,x,\boldsymbol{y})$ and $k_{\mathcal{I}} = k_{\mathcal{I} \setminus \mathcal{P}}$.

Claim 3.6.3. $k_{\mathcal{I} \setminus \mathcal{P}} \leq A(r,x,\boldsymbol{y})$.

Proof.

$$k_{\mathcal{I} \setminus \mathcal{P}} \leq |\mathcal{I} \setminus \mathcal{P}| \overset{(a)}{=} \min\{n, B(r,x,\boldsymbol{y})\} - \sum_{j=1}^x y_j \leq B(r,x,\boldsymbol{y}) - \sum_{j=1}^x y_j = A(r,x,\boldsymbol{y}),$$

where (a) follows from $|\mathcal{I}| = \min\{n, B(r,x,\boldsymbol{y})\}$ and $|\mathcal{P}| = \sum_{j=1}^x y_j$. ■

Claim 3.6.4. $k_{\mathcal{I}} = k_{\mathcal{I} \setminus \mathcal{P}}$.

Proof. Showing that $k_{\mathcal{I}} = k_{\mathcal{I}\setminus\mathcal{P}}$ is equivalent to showing that for any two codewords $\boldsymbol{c}$ and $\hat{\boldsymbol{c}}$ in code $\mathcal{C}$, if $\boldsymbol{c}_{\mathcal{I}\setminus\mathcal{P}} = \hat{\boldsymbol{c}}_{\mathcal{I}\setminus\mathcal{P}}$, then $\boldsymbol{c}_{\mathcal{P}} = \hat{\boldsymbol{c}}_{\mathcal{P}}$.

Assume on the contrary that there exist two codewords $\boldsymbol{c} = (c_1, \ldots, c_n)$ and $\hat{\boldsymbol{c}} = (\hat{c}_1, \ldots, \hat{c}_n)$ in code $\mathcal{C}$ that $\boldsymbol{c}_{\mathcal{I}\setminus\mathcal{P}} = \hat{\boldsymbol{c}}_{\mathcal{I}\setminus\mathcal{P}}$, but $\boldsymbol{c}_{\mathcal{P}} \neq \hat{\boldsymbol{c}}_{\mathcal{P}}$. Let $\mathcal{E} = \{i : c_i \neq \hat{c}_i,\ i \in \mathcal{P}\}$. We order the elements in $\mathcal{E}$ according to the lexicographical order $\prec$ defined as follows:

1. If $i < j$, then $a_i^u \prec a_j^v$, for $i, j \in [x]$, $u \in [y_i]$, and $v \in [y_j]$.

2. If $u < v$, then $a_i^u \prec a_i^v$, for $i \in [x]$, $u, v \in [y_i]$.

Suppose that the smallest element with respect to the lexicographical order $\prec$ in $\mathcal{E}$ is a_i^u. According to the construction steps, we have $(\{a_i\} \cup \mathcal{R}_{a_i}^u \setminus \{a_i^u\}) \bigcap \mathcal{E} = \emptyset$ and $(\{a_i\} \cup \mathcal{R}_{a_i}^u \setminus \{a_i^u\}) \subseteq \mathcal{I}$. Since $\boldsymbol{c}_{\mathcal{I}\setminus\mathcal{E}} = \hat{\boldsymbol{c}}_{\mathcal{I}\setminus\mathcal{E}}$, we have $\boldsymbol{c}_{\{a_i\}\cup\mathcal{R}_{a_i}^u\setminus\{a_i^u\}} = \hat{\boldsymbol{c}}_{\{a_i\}\cup\mathcal{R}_{a_i}^u\setminus\{a_i^u\}}$, but $c_{a_i^u} \neq \hat{c}_{a_i^u}$. This violates the selection rule in step 3.2) for a_i^u: $k_{\{a_i\}\cup\mathcal{R}_{a_i}^u\setminus\{a_i^u\}} = k_{\{a_i\}\cup\mathcal{R}_{a_i}^u}$, which indicates that if $\boldsymbol{c}_{\{a_i\}\cup\mathcal{R}_{a_i}^u\setminus\{a_i^u\}} = \hat{\boldsymbol{c}}_{\{a_i\}\cup\mathcal{R}_{a_i}^u\setminus\{a_i^u\}}$ then $c_{a_i^u} = \hat{c}_{a_i^u}$. Thus, we get a contradiction and conclude that there do not exist two codewords $\boldsymbol{c}$ and $\hat{\boldsymbol{c}}$ in code $\mathcal{C}$ that $\boldsymbol{c}_{\mathcal{I}\setminus\mathcal{P}} = \hat{\boldsymbol{c}}_{\mathcal{I}\setminus\mathcal{P}}$, but $\boldsymbol{c}_{\mathcal{P}} \neq \hat{\boldsymbol{c}}_{\mathcal{P}}$. ■

From Claims 3.6.3 and 3.6.4, it is clear to see that we have $k_{\mathcal{I}} \leq A(r, x, \boldsymbol{y})$. Therefore, there exists a set $\mathcal{I} \subseteq [n]$, $|\mathcal{I}| = \min\{n, B(r, x, \boldsymbol{y})\}$, such that $k_{\mathcal{I}} \leq A(r, x, \boldsymbol{y})$. Finally, choose a set $\mathcal{I}$, produced by Procedure A, satisfying $A(r, x, \boldsymbol{y}) < k$. Since $k_{\mathcal{I}} \leq A(r, x, \boldsymbol{y}) < k$ and $k_{[n]} = k$, we conclude that $B(r, x, \boldsymbol{y}) < n$ and $|\mathcal{I}| = \min\{n, B(r, x, \boldsymbol{y})\} = B(r, x, \boldsymbol{y})$.

■

3.7 Appendix B

In this section, we give the proof of Theorem 3.2.4.

Proof. We follow similar steps to the proof in [13] which consists of two parts. First, from Lemma 3.2.3, for any $[n, k, d]_q$ linear code $\mathcal{C}$ with information locality r and availability t, for all $x \in \mathbb{Z}^+$ and $\boldsymbol{y} = (y_1, \ldots, y_x) \in ([t])^x$ satisfying $1 \leq x \leq \lceil \frac{k}{(r-1)t+1} \rceil$ and $A(r, x, \boldsymbol{y}) < k$, there exists a set $\mathcal{I} \subseteq [n]$, $|\mathcal{I}| = B(r, x, \boldsymbol{y})$, such that $k_{\mathcal{I}} \leq A(r, x, \boldsymbol{y})$.

For the second part of the proof, for any $x \in \mathbb{Z}^+$ and $\boldsymbol{y} = (y_1, \ldots, y_x) \in ([t])^x$, the $\mathcal{I} \subseteq [n]$ is constructed as in the first part. Then, we consider the code $\mathcal{C}_{\mathcal{I}}^{\boldsymbol{0}} = \{\boldsymbol{c}_{[n]\setminus\mathcal{I}} : \boldsymbol{c}_{\mathcal{I}} = \boldsymbol{0} \text{ and } \boldsymbol{c} \in \mathcal{C}\}$. Since the

code $\mathcal{C}$ is linear, the size of the code $\mathcal{C}_{\mathcal{I}}^{\mathbf{0}}$ is $q^{k-k_{\mathcal{I}}}$ and it is a linear code as well. Moreover, the minimum distance D of the code $\mathcal{C}_{\mathcal{I}}^{\mathbf{0}}$ is at least d, i.e., $D \geq d$.

Thus, we get an upper bound on the minimum distance d,

$$d \leq D \leq d_{\ell-opt}^{(q)}[n-|\mathcal{I}|,k-k_{\mathcal{I}}] \leq d_{\ell-opt}^{(q)}[n-|\mathcal{I}|,k-A(r,x,\boldsymbol{y})].$$

Therefore, we conclude that

$$d \leq d_{\ell-opt}^{(q)}[n-|\mathcal{I}|,k-A(r,x,\boldsymbol{y})] = d_{\ell-opt}^{(q)}[n-B(r,x,\boldsymbol{y}),k-A(r,x,\boldsymbol{y})].$$

Similarly, we also get an upper bound on the dimension k,

$$k-k_{\mathcal{I}} \leq k_{\ell-opt}^{(q)}[n-|\mathcal{I}|,D] \leq k_{\ell-opt}^{(q)}[n-|\mathcal{I}|,d].$$

Therefore, we conclude that

$$k \leq k_{\ell-opt}^{(q)}[n-|\mathcal{I}|,d]+k_{\mathcal{I}} \leq k_{\ell-opt}^{(q)}[n-B(r,x,\boldsymbol{y}),d]+A(r,x,\boldsymbol{y}).$$

■

Acknowledgement

This chapter is in part a reprint of the material in the paper: Pengfei Huang, Eitan Yaakobi, Hironori Uchikawa, and Paul H. Siegel, "Binary linear locally repairable codes," *IEEE Transactions on Information Theory*, vol. 62, no. 11, pp. 6268–6283, Nov. 2016. The dissertation author was the primary investigator and author of this paper.

Chapter 4

Multi-Erasure Locally Repairable Codes over Small Fields

4.1 Introduction

In this chapter, we extend our previous construction for binary LRCs in Chapter 3 to construct erasure codes that can locally correct multiple erasures. In particular, we consider erasure codes with both local and global erasure-correcting capabilities for a $\rho \times n_0$ storage array [11], where each row contains some local parities, and additional global parities are distributed in the array. The array structure is suitable for many storage applications. For example, a storage array can represent a large-scale distributed storage system consisting of a large number of storage nodes that spread over different geographical locations. The storage nodes that are placed in the same location can form a local storage cluster. Thus, each row of the storage array can stand for such a local storage cluster. Another example is a redundant array of independent disks (RAID) type of architecture for solid-state drives (SSDs) [11, 28]. In this scenario, a $\rho \times n_0$ storage array can represent a total of ρ SSDs, each of which contains n_0 flash memory chips. Within each SSD, an erasure code is applied to these n_0 chips for local protection. In addition, erasure coding is also done across all the SSDs for global protection of all the chips.

More specifically, let us give the formal definition of this class of erasure codes as follows.

Definition 4.1.1. *Consider a code $\mathcal{C}$ over a finite field $\mathbb{F}_q$ consisting of $\rho \times n_0$ arrays such that:*

1. *Each row in each array in $\mathcal{C}$ belongs to a linear local code $\mathcal{C}_0$ with length n_0 and minimum distance d_0 over $\mathbb{F}_q$.*

2. *Reading the symbols of $\mathcal{C}$ row-wise, $\mathcal{C}$ is a linear code with length ρn_0, dimension k, and minimum distance d over $\mathbb{F}_q$.*

*Then, we say that $\mathcal{C}$ is a $(\rho, n_0, k; d_0, d)_q$ Multi-Erasure Locally Repairable Code (**ME-LRC**).*

Thus, a $(\rho, n_0, k; d_0, d)_q$ ME-LRC can locally correct $d_0 - 1$ erasures in each row, and is guaranteed to correct a total of $d - 1$ erasures anywhere in the array.

Our work is motivated by a recent work by Blaum and Hetzler [11]. In their work, the authors studied ME-LRCs where each row is a maximum distance separable (MDS) code, and gave code constructions with field size $q \geq \max\{\rho, n_0\}$ using generalized integrated interleaving (GII) codes [36, 78, 88]. Our Definition 4.1.1 generalizes the definition of the codes in [11] by not requiring each row to be an MDS code. There exist other related works. The ME-LRCs in Definition 4.1.1 can be seen as (r, δ) LRCs with disjoint repair sets. A code $\mathcal{C}$ is called an (r, δ) LRC [58], if for every coordinate, there exists a punctured code (i.e., a repair set) of $\mathcal{C}$ with support containing this coordinate, whose length is at most $r + \delta - 1$, and whose minimum distance is at least δ. Although the existing constructions [58, 75] for (r, δ) LRCs with disjoint repair sets can generate ME-LRCs as in Definition 4.1.1, they use MDS codes as local codes and require a field size that is at least as large as the code length. A recent work [7] gives explicit constructions of (r, δ) LRCs with disjoint repair sets over field $\mathbb{F}_q$ from algebraic curves, whose repair sets have size $r + \delta - 1 = \sqrt{q}$ or $r + \delta - 1 = \sqrt{q} + 1$. Partial MDS (PMDS) codes [10] are also related to but different from ME-LRCs in Definition 4.1.1. In general, PMDS codes need to satisfy stricter requirements than ME-LRCs. A $\rho \times n_0$ array code is called an $(r; s)$ PMDS code if each row is an $[n_0, n_0 - r, r + 1]_q$ MDS code and whenever any r locations in each row are punctured, the resulting code is also an MDS code with minimum distance $s + 1$. The construction of (r, s) PMDS codes for all r and s with field size $O(n_0^{\rho n_0})$ was known [17]. More recently, a family of PMDS codes with field size $O(\max\{\rho, n_0^{r+s}\}^s)$ was constructed [27].

However, the construction of *optimal* ME-LRCs over any small field (e.g., the field size less than the length of the local code, or even the binary field) has not been fully explored and solved. The goal of this chapter is to study ME-LRCs over small fields. We propose a general construction based on generalized tensor product codes [40, 86]. It extends our previous construction in Chapter 3 to the scenario of multi-erasure LRCs over any field. In contrast to [11], our construction does not require field size $q \geq \max\{\rho, n_0\}$, and it can even generate binary ME-LRCs. We derive an upper bound on the minimum distance of ME-LRCs. For $2d_0 \geq d$, we show that our construction can produce optimal ME-LRCs with respect to (w.r.t.) the new upper bound on the minimum distance. We also present an erasure decoding algorithm and its corresponding correctable erasure patterns which include the pattern of any $d-1$ erasures. We show that the ME-LRCs from our construction based on Reed-Solomon (RS) codes are optimal w.r.t. certain correctable erasure patterns. So far the *exact* relation between GII codes [11, 78, 88] and generalized tensor product codes has not been fully investigated. We prove that GII codes are a subclass of generalized tensor product codes. As a result, the parameters of a GII code can be obtained by using the known properties of generalized tensor product codes.

The remainder of this chapter is organized as follows. In Section 4.2, we give notation and derive a field size dependent upper bound for ME-LRCs. In Section 4.3, we propose a general construction of ME-LRCs. The erasure-correcting properties of these codes are studied and an erasure decoding algorithm is presented. In Section 4.4, we study optimal code construction and provide several explicit optimal ME-LRCs over different fields. In Section 4.5, we prove that GII codes are a subclass of generalized tensor product codes. We conclude the chapter in Section 4.6.

4.2 An Upper Bound for ME-LRCs

We begin this section by giving some notation that will be used in this chapter. We use the notation $[n]$ to denote the set $\{1, \ldots, n\}$. For a length-n vector $\boldsymbol{v}$ over $\mathbb{F}_q$ and a set $\mathcal{I} \subseteq [n]$, the vector $\boldsymbol{v}_{\mathcal{I}}$ denotes the restriction of the vector $\boldsymbol{v}$ to coordinates in the set $\mathcal{I}$, and $w_q(\boldsymbol{v})$ represents the Hamming weight of the vector $\boldsymbol{v}$ over $\mathbb{F}_q$. The transpose of a matrix H is written as H^T. For a set $\mathcal{S}$, $|\mathcal{S}|$ represents the cardinality of the set. A linear code $\mathcal{C}$ over $\mathbb{F}_q$ of length n, dimension k, and minimum distance d will be denoted

by $\mathcal{C} = [n,k,d]_q$ or $[n,k,d]_q$ for simplicity. For a code with only one codeword, the minimum distance is defined as ∞.

Now, we give an upper bound on the minimum distance of a $(\rho, n_0, k; d_0, d)_q$ ME-LRC, by extending the shortening bound for LRCs in [13]. The upper bound obtained here will be used to prove the optimality of our construction for ME-LRCs in the following sections.

Let $d_{opt}^{(q)}[n,k]$ denote the largest possible minimum distance of a linear code of length n and dimension k over $\mathbb{F}_q$, and let $k_{opt}^{(q)}[n,d]$ denote the largest possible dimension of a linear code of length n and minimum distance d over $\mathbb{F}_q$.

Lemma 4.2.1. *For any $(\rho, n_0, k; d_0, d)_q$ ME-LRC $\mathcal{C}$, the minimum distance d satisfies*

$$d \leq \min_{0 \leq x \leq \lceil \frac{k}{k^*} \rceil - 1,\ x \in \mathbb{Z}} \left\{ d_{opt}^{(q)}[\rho n_0 - x n_0, k - x k^*] \right\}, \tag{4.1}$$

and the dimension satisfies

$$k \leq \min_{0 \leq x \leq \lceil \frac{k}{k^*} \rceil - 1,\ x \in \mathbb{Z}} \left\{ x k^* + k_{opt}^{(q)}[\rho n_0 - x n_0, d] \right\}, \tag{4.2}$$

where $k^* = k_{opt}^{(q)}[n_0, d_0]$.

Proof. For the case of $x = 0$, it is trivial. For $1 \leq x \leq \lceil \frac{k}{k^*} \rceil - 1$, $x \in \mathbb{Z}^+$, let $\mathcal{I}$ represent the set of the coordinates of the first x rows in the array. Thus, $|\mathcal{I}| = x n_0$. First, consider the code $\mathcal{C}_{\mathcal{I}} = \{\boldsymbol{c}_{\mathcal{I}} : \boldsymbol{c} \in \mathcal{C}\}$ whose dimension is denoted by $k_{\mathcal{I}}$, which satisfies $k_{\mathcal{I}} \leq x k^*$. Then, we consider the code $\mathcal{C}_{\mathcal{I}}^{\mathbf{0}} = \{\boldsymbol{c}_{[\rho n_0] \setminus \mathcal{I}} : \boldsymbol{c}_{\mathcal{I}} = \mathbf{0}$ and $\boldsymbol{c} \in \mathcal{C}\}$. Since the code $\mathcal{C}$ is linear, the size of the code $\mathcal{C}_{\mathcal{I}}^{\mathbf{0}}$ is $q^{k - k_{\mathcal{I}}}$ and it is a linear code as well. Moreover, the minimum distance $\hat{d}$ of the code $\mathcal{C}_{\mathcal{I}}^{\mathbf{0}}$ is at least d, i.e., $\hat{d} \geq d$.

Thus, we get an upper bound on the minimum distance d,

$$d \leq \hat{d} \leq d_{opt}^{(q)}[\rho n_0 - |\mathcal{I}|, k - k_{\mathcal{I}}] \leq d_{opt}^{(q)}[\rho n_0 - x n_0, k - x k^*].$$

Similarly, we also get an upper bound on the dimension k,

$$k - k_{\mathcal{I}} \leq k_{opt}^{(q)}[\rho n_0 - |\mathcal{I}|, \hat{d}] \leq k_{opt}^{(q)}[\rho n_0 - x n_0, d].$$

Therefore, we conclude that

$$k \leq k_{opt}^{(q)}[\rho n_0 - x n_0, d] + k_{\mathcal{I}} \leq k_{opt}^{(q)}[\rho n_0 - x n_0, d] + x k^*.$$

■

4.3 ME-LRCs from Generalized Tensor Product Codes

In this section, we first introduce generalized tensor product codes over a finite field $\mathbb{F}_q$. Then, we give a general construction of ME-LRCs from generalized tensor product codes. The minimum distance of the constructed ME-LRCs is determined, a decoding algorithm tailored for erasure correction is proposed, and corresponding correctable erasure patterns are studied.

4.3.1 Generalized Tensor Product Codes over a Finite Field

We start by presenting the tensor product operation of two matrices $H^{'}$ and $H^{''}$. Let $H^{'}$ be the parity-check matrix of a code with length n' and dimension $n' - v$ over $\mathbb{F}_q$. The matrix $H^{'}$ can be considered as a v (row) by n' (column) matrix over $\mathbb{F}_q$ or as a 1 (row) by n' (column) matrix of elements from $\mathbb{F}_{q^v}$. Let $H^{'}$ be the vector $H^{'} = [h_1^{'}\ h_2^{'}\ \cdots\ h_{n'}^{'}]$, where $h_j^{'}$, $1 \leq j \leq n'$, are elements of $\mathbb{F}_{q^v}$. Let $H^{''}$ be the parity-check matrix of a code of length ℓ and dimension $\ell - \lambda$ over $\mathbb{F}_{q^v}$. We denote $H^{''}$ by

$$H^{''} = \begin{bmatrix} h_{11}^{''} & \cdots & h_{1\ell}^{''} \\ \vdots & \ddots & \vdots \\ h_{\lambda 1}^{''} & \cdots & h_{\lambda\ell}^{''} \end{bmatrix},$$

where $h_{ij}^{''}$, $1 \leq i \leq \lambda$ and $1 \leq j \leq \ell$, are elements of $\mathbb{F}_{q^v}$.

The tensor product of the matrices $H^{''}$ and $H^{'}$ is defined as

$$H_{TP} = H^{''} \bigotimes H^{'} = \begin{bmatrix} h_{11}^{''}H^{'} & \cdots & h_{1\ell}^{''}H^{'} \\ \vdots & \ddots & \vdots \\ h_{\lambda 1}^{''}H^{'} & \cdots & h_{\lambda\ell}^{''}H^{'} \end{bmatrix},$$

where $h''_{ij}H' = [h''_{ij}h'_1\ h''_{ij}h'_2\ \cdots\ h''_{ij}h'_{n'}]$, $1 \le i \le \lambda$ and $1 \le j \le \ell$, and the products of elements are calculated according to the rules of multiplication for elements over $\mathbb{F}_{q^v}$. The matrix H_{TP} will be considered as a $v\lambda \times n'\ell$ matrix of elements from $\mathbb{F}_q$, thus defining a tensor product code over $\mathbb{F}_q$.

Our construction of ME-LRCs is based on generalized tensor product codes [40]. Define the matrices H'_i and H''_i for $i = 1, 2, \ldots, \mu$ as follows. The matrix H'_i is a $v_i \times n'$ matrix over $\mathbb{F}_q$ such that the $(v_1 + v_2 + \cdots + v_i) \times n'$ matrix

$$B_i = \begin{bmatrix} H'_1 \\ H'_2 \\ \vdots \\ H'_i \end{bmatrix}$$

is a parity-check matrix of an $[n', n' - v_1 - v_2 - \cdots - v_i, d'_i]_q$ code $\mathcal{C}'_i$, where $d'_1 \le d'_2 \le \cdots \le d'_i$. The matrix H''_i is a $\lambda_i \times \ell$ matrix over $\mathbb{F}_{q^{v_i}}$, which is a parity-check matrix of an $[\ell, \ell - \lambda_i, \delta_i]_{q^{v_i}}$ code $\mathcal{C}''_i$.

We define a μ-level generalized tensor product code over $\mathbb{F}_q$ as a linear code having a parity-check matrix over $\mathbb{F}_q$ in the form of the following μ-level tensor product structure

$$H = \begin{bmatrix} H''_1 \otimes H'_1 \\ H''_2 \otimes H'_2 \\ \vdots \\ H''_\mu \otimes H'_\mu \end{bmatrix}. \tag{4.3}$$

As the matrix H_{TP}, each level in the matrix H is obtained by operations over $\mathbb{F}_q$ and its extension field. We denote this code by $\mathcal{C}^{\mu}_{GTP}$. Its length is $n_t = n'\ell$ and the dimension is $k_t = n_t - \sum_{i=1}^{\mu} v_i\lambda_i$.

By adapting Theorem 2 in [40] from the field $\mathbb{F}_2$ to $\mathbb{F}_q$, we directly have the following theorem on the minimum distance of $\mathcal{C}^{\mu}_{GTP}$ over $\mathbb{F}_q$.

Theorem 4.3.1. *The minimum distance d_t of a generalized tensor product code $\mathcal{C}^{\mu}_{GTP}$ over $\mathbb{F}_q$ satisfies*

$$d_t \ge \min\{\delta_1, \delta_2 d'_1, \delta_3 d'_2, \ldots, \delta_\mu d'_{\mu-1}, d'_\mu\}.$$

Proof. A codeword $\boldsymbol{x}$ in $\mathcal{C}^{\mu}_{GTP}$ is an $n'\ell$-dimensional vector over $\mathbb{F}_q$, denoted by $\boldsymbol{x} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_\ell)$, where

$\boldsymbol{x}_i$ in $\boldsymbol{x}$ is an n'-dimensional vector, for $i = 1,2,\ldots,\ell$.

Let $\boldsymbol{s}_i^j = \boldsymbol{x}_i H_j^{'T}$, for $i = 1,2,\ldots,\ell$ and $j = 1,2,\ldots,\mu$. Thus, $\boldsymbol{s}_i^j$ is a v_j-dimensional vector over $\mathbb{F}_q$, and is considered as an element in $\mathbb{F}_{q^{v_j}}$. Let $\boldsymbol{s}^j = (\boldsymbol{s}_1^j, \boldsymbol{s}_2^j, \ldots, \boldsymbol{s}_\ell^j)$, an ℓ-dimensional vector over $\mathbb{F}_{q^{v_j}}$, whose components are $\boldsymbol{s}_i^j$, $i = 1,2,\ldots,\ell$. To prove Theorem 4.3.1, we need to show that if $\boldsymbol{x}H^T = \boldsymbol{0}$ and $w_q(\boldsymbol{x}) < d_m = \min\{\delta_1, \delta_2 d'_1, \delta_3 d'_2, \ldots, \delta_\mu d'_{\mu-1}, d'_\mu\}$, then $\boldsymbol{x}$ must be the all-zero vector $\boldsymbol{0}$.

We prove it by contradiction and induction. Assume that there exists a codeword $\boldsymbol{x}$ such that $\boldsymbol{x}H^T = \boldsymbol{0}$, $w_q(\boldsymbol{x}) < d_m$, and $\boldsymbol{x} \neq \boldsymbol{0}$.

We first state a proposition which will be used in the following proof.

Proposition 4.3.2. *If $\boldsymbol{x}H^T = \boldsymbol{0}$ and $\boldsymbol{s}^1 = \boldsymbol{s}^2 = \cdots = \boldsymbol{s}^j = \boldsymbol{0}$, then $w_q(\boldsymbol{x}_i) \geq d'_j$ for $\boldsymbol{x}_i \neq \boldsymbol{0}$, $i = 1,2,\ldots,\ell$.*

Proof. The condition $\boldsymbol{s}^1 = \boldsymbol{s}^2 = \cdots = \boldsymbol{s}^j = \boldsymbol{0}$ means that $\boldsymbol{x}_i B_j^T = \boldsymbol{0}$ for $i = 1,2,\ldots,\ell$; that is, $\boldsymbol{x}_i$ is a codeword in the code defined by the parity-check matrix B_j, whose minimum distance is d'_j. Therefore, we have $w_q(\boldsymbol{x}_i) \geq d'_j$ for $\boldsymbol{x}_i \neq \boldsymbol{0}$, $i = 1,2,\ldots,\ell$. ■

Now, if $\boldsymbol{s}^1 \neq \boldsymbol{0}$, then $w_q(\boldsymbol{x}) \geq w_{q^{v_1}}(\boldsymbol{s}^1) \geq \delta_1 \geq d_m$, which contradicts the assumption. Thus, we have $\boldsymbol{s}^1 = \boldsymbol{0}$.

Then, consider the second level. If $\boldsymbol{s}^2 \neq \boldsymbol{0}$, then $w_q(\boldsymbol{x}) \overset{(a)}{\geq} w_{q^{v_2}}(\boldsymbol{s}^2) d'_1 \geq \delta_2 d'_1 \geq d_m$, where step (a) is from Proposition 4.3.2. This contradicts the assumption, so we have $\boldsymbol{s}^2 = \boldsymbol{0}$. By induction, we must have $\boldsymbol{s}^1 = \boldsymbol{s}^2 = \cdots = \boldsymbol{s}^{\mu-1} = \boldsymbol{0}$.

For the last level, i.e., the μth level, if $\boldsymbol{s}^\mu \neq \boldsymbol{0}$, then $w_q(\boldsymbol{x}) \geq w_{q^{v_\mu}}(\boldsymbol{s}^\mu) d'_{\mu-1} \geq \delta_\mu d'_{\mu-1} \geq d_m$, which contradicts our assumption. Now, if $\boldsymbol{s}^1 = \boldsymbol{s}^2 = \cdots = \boldsymbol{s}^\mu = \boldsymbol{0}$, then $w_q(\boldsymbol{x}) \geq d'_\mu \geq d_m$, which also contradicts our assumption.

Thus, our assumption is violated. ■

4.3.2 Construction of ME-LRCs

Now, we present a general construction of ME-LRCs based on generalized tensor product codes.

Construction A

Step 1: Choose $v_i \times n'$ matrices $H_i^{'}$ over $\mathbb{F}_q$ and $\lambda_i \times \ell$ matrices $H_i^{''}$ over $\mathbb{F}_{q^{v_i}}$, for $i = 1,2,\ldots,\mu$, which satisfy the following two properties:

1) The parity-check matrix $H_1^{''} = \mathbf{I}_{\ell\times\ell}$, i.e., an $\ell \times \ell$ identity matrix.

2) The matrices $H_i^{'}$ (or B_i), $1 \leq i \leq \mu$, and $H_j^{''}$, $2 \leq j \leq \mu$, are chosen such that $d'_\mu \leq \delta_j d'_{j-1}$, for $j = 2,3,\ldots,\mu$.

Step 2: Generate a parity-check matrix H over $\mathbb{F}_q$ according to (4.3) with the matrices $H_i^{'}$ and $H_i^{''}$, for $i = 1,2,\ldots,\mu$. The constructed code corresponding to the parity-check matrix H is referred to as $\mathcal{C}_A$. ■

Theorem 4.3.3. *The code $\mathcal{C}_A$ is a $(\rho, n_0, k; d_0, d)_q$ ME-LRC with parameters $\rho = \ell$, $n_0 = n'$, $k = n'\ell - \sum_{i=1}^{\mu} v_i\lambda_i$, $d_0 = d'_1$, and $d = d'_\mu$.*

Proof. According to Construction A, the code parameters ρ, n_0, k, and d_0 can be easily determined. In the following, we prove that $d = d'_\mu$.

Since $\delta_1 = \infty$ ($H_1^{''}$ is the identity matrix) and $d'_\mu \leq \delta_i d'_{i-1}$ for all $i = 2,3,\ldots,\mu$, from Theorem 4.3.1, $d \geq d'_\mu$.

Now, we show that $d \leq d'_\mu$. For $i = 1,2,\ldots,\mu$, let $H_i^{'} = [h_1^{'}(i),\ldots,h_{n'}^{'}(i)]$ over $\mathbb{F}_{q^{v_i}}$, and let $[h_{11}^{''}(i),\ldots,h_{\lambda_i 1}^{''}(i)]^T$ over $\mathbb{F}_{q^{v_i}}$ be the first column of $H_i^{''}$. Since the code with parity-check matrix B_μ has minimum distance d'_μ, there exist d'_μ columns of B_μ, say in the set of positions $J = \{b_1, b_2, \ldots, b_{d'_\mu}\}$, which are linearly dependent; that is, $\sum_{j\in J} \alpha_j h_j^{'}(i) = 0$, for some $\alpha_j \in \mathbb{F}_q$, for all $i = 1,2,\ldots,\mu$. Thus, we have $\sum_{j\in J} \alpha_j h_{p1}^{''}(i) h_j^{'}(i) = h_{p1}^{''}(i)\left(\sum_{j\in J} \alpha_j h_j^{'}(i)\right) = 0$, for $p = 1,2,\ldots,\lambda_i$ and $i = 1,2,\ldots,\mu$. That is, the columns in positions $b_1, b_2, \ldots, b_{d'_\mu}$ of H are linearly dependent. ■

4.3.3 Erasure Decoding and Correctable Erasure Patterns

We present a decoding algorithm for the ME-LRC $\mathcal{C}_A$ from Construction A, tailored for erasure correction. The decoding algorithm for error correction for generalized tensor product codes can be found in [40].

Let the symbol ? represent an erasure and "e" denote a decoding failure. The erasure decoder $\mathcal{D}_A : (\mathbb{F}_q \cup \{?\})^{n'\ell} \to \mathcal{C}_A \cup \{\text{"e"}\}$ for an ME-LRC $\mathcal{C}_A$ consists of two kinds of component decoders $\mathcal{D}_i^{'}$ and $\mathcal{D}_i^{''}$ for $i = 1,2,\ldots,\mu$ described below.

a) First, the decoder for a coset of the code $\mathcal{C}_i^{'}$ with parity-check matrix B_i, $i = 1,2,\ldots,\mu$, is

denoted by

$$\mathcal{D}_i^{'}:(\mathbb{F}_q \cup \{?\})^{n'} \times (\mathbb{F}_q \cup \{?\})^{\sum_{j=1}^{i} v_j} \rightarrow (\mathbb{F}_q \cup \{?\})^{n'}$$

which uses the following decoding rule: for a length-n' input vector $\boldsymbol{y}'$, and a length-$\sum_{j=1}^{i} v_j$ syndrome vector $\boldsymbol{s}'$ without erasures, if $\boldsymbol{y}'$ agrees with exactly one codeword $\boldsymbol{c}' \in \mathcal{C}_i^{'} + \boldsymbol{e}$ on the entries with values in $\mathbb{F}_q$, where the vector $\boldsymbol{e}$ is a coset leader determined by both the code $\mathcal{C}_i^{'}$ and the syndrome vector $\boldsymbol{s}'$, i.e., $\boldsymbol{s}' = \boldsymbol{e}B_i^T$, then $\mathcal{D}_i^{'}(\boldsymbol{y}', \boldsymbol{s}') = \boldsymbol{c}'$; otherwise, $\mathcal{D}_i^{'}(\boldsymbol{y}', \boldsymbol{s}') = \boldsymbol{y}'$. Therefore, if the length-n' input vector $\boldsymbol{y}'$ is a codeword in $\mathcal{C}_i^{'} + \boldsymbol{e}$ with $d_i' - 1$ or less erasures and the syndrome vector $\boldsymbol{s}'$ is not erased, then the decoder $\mathcal{D}_i^{'}$ can return the correct codeword.

b) Second, the decoder for the code $\mathcal{C}_i^{''}$ with parity-check matrix $H_i^{''}$, $i = 1, 2, \ldots, \mu$, is denoted by

$$\mathcal{D}_i^{''} : (\mathbb{F}_{q^{v_i}} \cup \{?\})^{\ell} \rightarrow (\mathbb{F}_{q^{v_i}} \cup \{?\})^{\ell}$$

which uses the following decoding rule: for a length-ℓ input vector $\boldsymbol{y}''$, if $\boldsymbol{y}''$ agrees with exactly one codeword $\boldsymbol{c}'' \in \mathcal{C}_i^{''}$ on the entries with values in $\mathbb{F}_{q^{v_i}}$, then $\mathcal{D}_i^{''}(\boldsymbol{y}'') = \boldsymbol{c}''$; otherwise, $\mathcal{D}_i^{''}(\boldsymbol{y}'') = \boldsymbol{y}''$. Therefore, if the length-ℓ input vector $\boldsymbol{y}''$ is a codeword in $\mathcal{C}_i^{''}$ with $\delta_i - 1$ or less erasures, then the decoder $\mathcal{D}_i^{''}$ can successfully return the correct codeword.

The erasure decoder $\mathcal{D}_A$ for the code $\mathcal{C}_A$ is summarized in Algorithm 1 below. Let the input word of length $n'\ell$ for the decoder $\mathcal{D}_A$ be $\boldsymbol{y} = (\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_\ell)$, where each component $\boldsymbol{y}_i \in (\mathbb{F}_q \cup \{?\})^{n'}$, $i = 1, \ldots, \ell$. The vector $\boldsymbol{y}$ is an erased version of a codeword $\boldsymbol{c} = (\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_\ell) \in \mathcal{C}_A$.

Algorithm 1: Decoding Procedure of Decoder $\mathcal{D}_A$

Input: received word $\boldsymbol{y} = (\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_\ell)$.

Output: codeword $\boldsymbol{c} \in \mathcal{C}_A$ or a decoding failure "e".

1. Let $\boldsymbol{s}_j^1 = \boldsymbol{0}$, for $j = 1, 2, \ldots, \ell$.

2. $\hat{\boldsymbol{c}} = (\hat{\boldsymbol{c}}_1, \ldots, \hat{\boldsymbol{c}}_\ell) = \left(\mathcal{D}_1^{'}(\boldsymbol{y}_1, \boldsymbol{s}_1^1), \ldots, \mathcal{D}_1^{'}(\boldsymbol{y}_\ell, \boldsymbol{s}_\ell^1)\right)$.

3. Let $\mathcal{F} = \{j \in [\ell] : \hat{\boldsymbol{c}}_j \text{ contains } ?\}$.

4. **For** $i = 2, \ldots, \mu$

- If $\mathcal{F} \neq \emptyset$, do the following steps; otherwise go to step 5.
- $(\boldsymbol{s}_1^i, \ldots, \boldsymbol{s}_\ell^i) = \mathcal{D}_i^{''}\left(\hat{\boldsymbol{c}}_1 H_i^{'T}, \ldots, \hat{\boldsymbol{c}}_\ell H_i^{'T}\right)$.
- $\hat{\boldsymbol{c}}_j = \mathcal{D}_i^{'}\left(\hat{\boldsymbol{c}}_j, (\boldsymbol{s}_j^1, \ldots, \boldsymbol{s}_j^i)\right)$ for $j \in \mathcal{F}$; $\hat{\boldsymbol{c}}_j$ remains the same for $j \in [\ell] \backslash \mathcal{F}$.
- Update $\mathcal{F} = \{j \in [\ell] : \hat{\boldsymbol{c}}_j \text{ contains } ?\}$.

end

5. If $\mathcal{F} = \emptyset$, let $\boldsymbol{c} = \hat{\boldsymbol{c}}$ and output $\boldsymbol{c}$; otherwise return "e".

In Algorithm 1, we use the following rules for operations which involve the symbol ?: 1) Addition $+$: for any element $\gamma \in \mathbb{F}_q \cup \{?\}$, $\gamma + ? = ?$. 2) Multiplication $\times$: for any element $\gamma \in \mathbb{F}_q \cup \{?\} \backslash \{0\}$, $\gamma \times ? = ?$, and $0 \times ? = 0$. 3) If a length-n vector $\boldsymbol{x}$, $\boldsymbol{x} \in (\mathbb{F}_q \cup \{?\})^n$, contains an entry ?, then $\boldsymbol{x}$ is considered as the symbol ? in the set $\mathbb{F}_{q^n} \cup \{?\}$. Similarly, the symbol ? in the set $\mathbb{F}_{q^n} \cup \{?\}$ is treated as a length-n vector whose entries are all ?.

To describe correctable erasure patterns, we use the following notation. Let $w_e(\boldsymbol{v})$ denote the number of erasures ? in the vector $\boldsymbol{v}$. For a received word $\boldsymbol{y} = (\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_\ell)$, let $N_\tau = |\{\boldsymbol{y}_m : w_e(\boldsymbol{y}_m) \geq d'_\tau,\ 1 \leq m \leq \ell\}|$ for $1 \leq \tau \leq \mu$.

Theorem 4.3.4. *The decoder $\mathcal{D}_A$ for a $(\rho, n_0, k; d_0, d)_q$ ME-LRC $\mathcal{C}_A$ can correct any received word $\boldsymbol{y}$ that satisfies the following condition:*

$$N_\tau \leq \delta_{\tau+1} - 1,\ \forall\ 1 \leq \tau \leq \mu, \tag{4.4}$$

where $\delta_{\mu+1}$ is defined to be 1.

Proof. The proof follows from the decoding procedure of decoder $\mathcal{D}_A$. The ME-LRC $\mathcal{C}_A$ has $d_0 = d'_1$ and $d = d'_\mu$. For a received word $\boldsymbol{y} = (\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_\ell)$, each vector $\boldsymbol{y}_i$, $1 \leq i \leq \ell$, corresponds to a row in the array.

For the first level, since $\delta_1 = \infty$, the correct syndrome vector $(\boldsymbol{s}_1^1, \ldots, \boldsymbol{s}_\ell^1)$ is the all-zero vector, i.e., $(\boldsymbol{s}_1^1, \ldots, \boldsymbol{s}_\ell^1) = \boldsymbol{0}$. Thus, the rows with number of erasures less than d'_1 are corrected.

For the second level, the remaining uncorrected row $\hat{\boldsymbol{c}}_j$, $j \in \mathcal{F}$, has at least d'_1 erasures. The total number of such uncorrected rows with indices in $\mathcal{F}$ is less than δ_2, because we require $N_1 \leq \delta_2 - 1$ in the condition. Thus, the correct syndrome vector $(\boldsymbol{s}_1^2, \ldots, \boldsymbol{s}_\ell^2)$ can be obtained. As a result, the rows with number of erasures less than d'_2 are corrected.

Similarly, by induction, if the decoder runs until the μth level, the remaining uncorrected row $\hat{\boldsymbol{c}}_j$, $j \in \mathcal{F}$, has at least $d'_{\mu-1}$ erasures. The total number of such uncorrected rows with indices in $\mathcal{F}$ is less than δ_μ, because we require $N_{\mu-1} \leq \delta_\mu - 1$ in the condition. Therefore, all the correct syndrome vectors $(\boldsymbol{s}_1^i, \ldots, \boldsymbol{s}_\ell^i)$, $i = 1, 2, \ldots, \mu$, are obtained. On the other hand, the remaining uncorrected row $\hat{\boldsymbol{c}}_j$, $j \in \mathcal{F}$, has at most $d'_\mu - 1$ erasures, since we also require $N_\mu \leq 0$ in the condition. Thus, all these uncorrected rows can be corrected in this step with all these correct syndromes. ■

The following corollary follows from Theorem 4.3.4.

Corollary 4.3.5. *The decoder $\mathcal{D}_A$ for a $(\rho, n_0, k; d_0, d)_q$ ME-LRC $\mathcal{C}_A$ can correct any received word $\boldsymbol{y}$ with less than d erasures.*

Proof. The ME-LRC $\mathcal{C}_A$ has $d_0 = d'_1$ and $d = d'_\mu$. We only need to show that the received word $\boldsymbol{y}$ with any $d'_\mu - 1$ erasures satisfies the condition in Theorem 4.3.4. We prove it by contradiction. If the condition is not satisfied, there is at least an integer i, $1 \leq i \leq \mu$, such that $N_i \geq \delta_{i+1}$. Therefore, we have $w_e(\boldsymbol{y}) \geq d'_i \delta_{i+1} \geq d'_\mu$, where the last inequality is from the requirement of Construction A. Thus, we get a contradiction to the assumption that the received word $\boldsymbol{y}$ has $d'_\mu - 1$ erasures. ■

4.4 Optimal Construction and Explicit ME-LRCs over Small Fields

In this section, we study the optimality of Construction A, and also present several explicit ME-LRCs that are optimal over different fields.

4.4.1 Optimal Construction

We show how to construct ME-LRCs which are optimal w.r.t. the bound (4.1) by adding more constraints to Construction A. To this end, we specify the choice of the matrices in Construction A. This specification, referred to as **Design I**, is as follows.

1) $H_1^{'}$ is the parity-check matrix of an $[n',n'-v_1,d_1']_q$ code which satisfies $k_{opt}^{(q)}[n',d_1'] = n'-v_1$.

2) B_μ is the parity-check matrix of an $[n',n'-\sum_{i=1}^{\mu} v_i,d_\mu']_q$ code with $d_{opt}^{(q)}[n',n'-\sum_{i=1}^{\mu} v_i] = d_\mu'$.

3) The minimum distances satisfy $d_\mu' \leq 2d_1'$.

4) $H_i^{''}$ is an all-one vector of length ℓ over $\mathbb{F}_{q^{v_i}}$, i.e., the parity-check matrix of a parity code with minimum distance $\delta_i = 2$, for all $i = 2,\ldots,\mu$. ■

Theorem 4.4.1. *The code $\mathcal{C}_A$ from Construction A with Design I is a $(\rho=\ell, n_0=n', k=n'\ell - v_1\ell - \sum_{i=2}^{\mu} v_i; d_0=d_1', d=d_\mu')_q$ ME-LRC, which is optimal with respect to the bound* (4.1).

Proof. From Theorem 4.3.3, the code parameters ρ, n_0, k, d_0, and d can be determined. We have $k^* = k_{opt}^{(q)}[n',d_1'] = n'-v_1$. Setting $x = \ell - 1$, we get

$$\begin{aligned} d \leq & \min_{0\leq x\leq \lceil \frac{k}{k^*} \rceil -1} \left\{ d_{opt}^{(q)}[\rho n_0 - x n_0, k - x k^*] \right\} \\ \leq & d_{opt}^{(q)}[\ell n' - (\ell-1)n', k-(\ell-1)k^*] \\ = & d_{opt}^{(q)}[n', n' - \sum_{i=1}^{\mu} v_i] = d_\mu'. \end{aligned}$$

This proves that $\mathcal{C}_A$ achieves the bound (4.1). ■

4.4.2 Explicit ME-LRCs

Our construction is very flexible and can generate many ME-LRCs over different fields. In the following, we present several examples.

1) *ME-LRCs with local extended BCH codes over $\mathbb{F}_2$*

From the structure of BCH codes [64], there exists a chain of nested binary extended BCH codes: $\mathcal{C}_3 = [2^m, 2^m-1-3m, 8]_2 \subset \mathcal{C}_2 = [2^m, 2^m-1-2m, 6]_2 \subset \mathcal{C}_1 = [2^m, 2^m-1-m, 4]_2$.

Let the matrices B_1, B_2, and B_3 be the parity-check matrices of $\mathcal{C}_1$, $\mathcal{C}_2$, and $\mathcal{C}_3$, respectively.

Example 4.4.1. For $\mu = 3$, in Construction A, we use the above matrices B_1, B_2, and B_3. We also choose $H_2^{''}$ and $H_3^{''}$ to be the all-one vector of length ℓ over $\mathbb{F}_{2^m}$.

From Theorem 4.3.3, the corresponding $(\rho, n_0, k; d_0, d)_2$ ME-LRC $\mathcal{C}_A$ has parameters $\rho = \ell$, $n_0 = 2^m$, $k = 2^m\ell - (m+1)\ell - 2m$, $d_0 = 4$, and $d = 8$. This code satisfies the requirements of Design I.

Thus, from Theorem 4.4.1, it is optimal w.r.t. the bound (4.1). $\square$

2) *ME-LRCs with local algebraic geometry codes over* $\mathbb{F}_4$

We use a class of algebraic geometry codes called Hermitian codes [91] to construct ME-LRCs.

From the construction of Hermitian codes [91], there exists a chain of nested 4-ary Hermitian codes: $\mathcal{C}_H(1) = [8,1,8]_4 \subset \mathcal{C}_H(2) = [8,2,6]_4 \subset \mathcal{C}_H(3) = [8,3,5]_4 \subset \mathcal{C}_H(4) = [8,4,4]_4 \subset \mathcal{C}_H(5) = [8,5,3]_4 \subset \mathcal{C}_H(6) = [8,6,2]_4 \subset \mathcal{C}_H(7) = [8,7,2]_4$.

Now, let the matrices B_1, B_2, B_3, and B_4 be the parity-check matrices of $\mathcal{C}_H(4)$, $\mathcal{C}_H(3)$, $\mathcal{C}_H(2)$, and $\mathcal{C}_H(1)$, respectively. Let $H_i^{''}$, $i = 2,3,4$, be the all-one vector of length ℓ over $\mathbb{F}_4$.

Example 4.4.2. For $\mu = 2$, in Construction A, we use the above matrices B_1, B_2, and $H_2^{''}$. From Theorem 4.3.3, the corresponding $(\rho, n_0, k; d_0, d)_4$ ME-LRC $\mathcal{C}_A$ has parameters $\rho = \ell$, $n_0 = 8$, $k = 4\ell - 1$, $d_0 = 4$, and $d = 5$.

For $\mu = 3$, in Construction A, we use the above matrices B_1, B_2, B_3, $H_2^{''}$, and $H_3^{''}$. From Theorem 4.3.3, the corresponding $(\rho, n_0, k; d_0, d)_4$ ME-LRC $\mathcal{C}_A$ has parameters $\rho = \ell$, $n_0 = 8$, $k = 4\ell - 2$, $d_0 = 4$, and $d = 6$.

For $\mu = 4$, in Construction A, we use the above matrices B_i, $i = 1, \ldots, 4$, and $H_j^{''}$, $j = 2,3,4$. From Theorem 4.3.3, the corresponding $(\rho, n_0, k; d_0, d)_4$ ME-LRC $\mathcal{C}_A$ has parameters $\rho = \ell$, $n_0 = 8$, $k = 4\ell - 3$, $d_0 = 4$, and $d = 8$.

All of the above three families of ME-LRCs over $\mathbb{F}_4$ are optimal w.r.t. the bound (4.1). $\square$

3) *ME-LRCs with local singly-extended Reed-Solomon codes over* $\mathbb{F}_q$

Let $n' \leq q$ and α be a primitive element of $\mathbb{F}_q$. We choose $H_1^{'}$ to be the parity-check matrix of an $[n', n' - d_1' + 1, d_1']_q$ singly-extended RS code, namely

$$H_1^{'} = \begin{bmatrix} 1 & 1 & \cdots & 1 & 1 \\ 1 & \alpha & \cdots & \alpha^{n'-2} & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & \alpha^{d_1'-2} & \cdots & \alpha^{(n'-2)(d_1'-2)} & 0 \end{bmatrix}.$$

For $i = 2, 3, \ldots, \mu$, we choose H_i' to be

$$H_i^{'} = \begin{bmatrix} 1 & \alpha^{d_{i-1}'-1} & \cdots & \alpha^{(n'-2)(d_{i-1}'-1)} & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & \alpha^{d_i'-2} & \cdots & \alpha^{(n'-2)(d_i'-2)} & 0 \end{bmatrix},$$

where $d_1' < d_2' < \cdots < d_\mu'$. We also require that

$$\delta_i = \lceil \frac{d_\mu'}{d_{i-1}'} \rceil = \lceil \frac{d_\mu'}{d_{i-1}'+1} \rceil = \cdots = \lceil \frac{d_\mu'}{d_i'-1} \rceil, \ \forall i = 2, \ldots, \mu$$

and $\delta_2 > \delta_3 > \cdots > \delta_\mu$.

For $i = 2, 3, \ldots, \mu$, let $H_i^{''}$ be the parity-check matrix of an $[\ell, \ell - \delta_i + 1, \delta_i = \lceil \frac{d_\mu'}{d_{i-1}'} \rceil]_{q^{v_i}}$ MDS code, which exists whenever $\ell \leq q^{v_i}$, where $v_i = d_i' - d_{i-1}'$. Note that for an MDS code with minimum distance 2, the code length can be arbitrarily long.

Example 4.4.3. We use the above chosen matrices H_i' and H_i'' for Construction A. The corresponding $(\rho, n_0, k; d_0, d)_q$ ME-LRC $\mathcal{C}_A$ has parameters $\rho = \ell$, $n_0 = n'$, $k = (n' - d_1' + 1)\ell - \sum_{i=2}^{\mu} (\lceil \frac{d_\mu'}{d_{i-1}'} \rceil - 1)(d_i' - d_{i-1}')$, $d_0 = d_1'$, and $d = d_\mu'$; the field size q satisfies $q \geq \max\{q', n'\}$, where $q' = \max_{i=2,\ldots,\mu} \{ \lceil \ell^{\frac{1}{d_i' - d_{i-1}'}} \rceil \}$.

When $\mu = 2$ and $d_1' < d_2' \leq 2d_1'$, the corresponding $(\rho, n_0, k; d_0, d)_q$ ME-LRC $\mathcal{C}_A$ has parameters $\rho = \ell$, $n_0 = n'$, $k = (n' - d_1' + 1)\ell - (d_2' - d_1')$, $d_0 = d_1'$, and $d = d_2'$; the field size q needs to satisfy $q \geq n'$. Since the code $\mathcal{C}_A$ satisfies the requirements of Design I, from Theorem 4.4.1, it is optimal w.r.t. the bound (4.1). □

The following theorem shows that the μ-level ME-LRC $\mathcal{C}_A$ constructed in Example 4.4.3 is optimal in the sense of possessing the largest possible dimension among all codes with the same erasure-correcting capability.

Theorem 4.4.2. *Let $\mathcal{C}$ be a code of length $\ell n'$ and dimension k over $\mathbb{F}_q$. Each codeword in $\mathcal{C}$ consists of ℓ sub-blocks, each of length n'. Assume that $\mathcal{C}$ corrects all erasure patterns satisfying the condition in (4.4), where $\delta_\tau = \lceil \frac{d_\mu'}{d_{\tau-1}'} \rceil$ for $2 \leq \tau \leq \mu$. Then, we must have dimension $k \leq (n' - d_1' + 1)\ell - \sum_{i=2}^{\mu} (\lceil \frac{d_\mu'}{d_{i-1}'} \rceil - 1)(d_i' - d_{i-1}')$.*

Proof. The proof is based on contradiction.

Let each codeword in $\mathcal{C}$ correspond to an $\ell \times n'$ array. We index the coordinates of the array row by row from number 1 to $\ell n'$. Let $\mathcal{I}_1$ be the set of coordinates defined by $\mathcal{I}_1 = \{(i-1)n' + j : \delta_2 - 1 < i \leq \ell,\ 1 \leq j \leq d'_1 - 1\}$. For $2 \leq \tau \leq \mu$, let $\mathcal{I}_\tau$ be the set of coordinates given by $\mathcal{I}_\tau = \{(i-1)n' + j : \delta_{\tau+1} - 1 < i \leq \delta_\tau - 1,\ 1 \leq j \leq d'_\tau - 1\}$, where $\delta_{\mu+1}$ is defined to be 1. Let $\mathcal{I}$ be the set of all the coordinates of the array.

By calculation, we have $|\mathcal{I} \backslash (\mathcal{I}_1 \cup \mathcal{I}_2 \cup \cdots \cup \mathcal{I}_\mu)| = (n' - d'_1 + 1)\ell - \sum_{i=2}^{\mu} (\lceil \frac{d'_\mu}{d'_{i-1}} \rceil - 1)(d'_i - d'_{i-1})$. Now, assume that $k > (n' - d'_1 + 1)\ell - \sum_{i=2}^{\mu} (\lceil \frac{d'_\mu}{d'_{i-1}} \rceil - 1)(d'_i - d'_{i-1})$. Then, there exist at least two distinct codewords $\boldsymbol{c}'$ and $\boldsymbol{c}''$ in $\mathcal{C}$ that agree on the coordinates in the set $\mathcal{I} \backslash (\mathcal{I}_1 \cup \mathcal{I}_2 \cup \cdots \cup \mathcal{I}_\mu)$. We erase the values on the coordinates in the set $\mathcal{I}_1 \cup \mathcal{I}_2 \cup \cdots \cup \mathcal{I}_\mu$ of both $\boldsymbol{c}'$ and $\boldsymbol{c}''$. This erasure pattern satisfies the condition in (4.4). Since $\boldsymbol{c}'$ and $\boldsymbol{c}''$ are distinct, this erasure pattern is uncorrectable. Thus, our assumption that $k > (n' - d'_1 + 1)\ell - \sum_{i=2}^{\mu} (\lceil \frac{d'_\mu}{d'_{i-1}} \rceil - 1)(d'_i - d'_{i-1})$ is violated. ■

Remark 4.4.1. The construction by Blaum and Hetzler [11] based on GII codes cannot generate ME-LRCs constructed in Examples 4.4.1 and 4.4.2. For the ME-LRC in Example 4.4.3, since the local code is the singly-extended RS code, the construction in [11] can also be used to produce an ME-LRC that has the same code parameters ρ, n_0, k, d_0 and d as those of the ME-LRC $\mathcal{C}_A$ from our construction. However, the construction in [11] requires the field size q to satisfy $q \geq \max\{\ell, n'\}$, which in general is larger than that in our construction. □

4.5 Relation to Generalized Integrated Interleaving Codes

Integrated interleaving (II) codes were first introduced in [36] as a two-level error-correcting scheme for data storage applications, and were then extended in [78] and more recently in [88] as generalized integrated interleaving (GII) codes for multi-level data protection.

The main difference between GII codes and generalized tensor product codes is that a generalized tensor product code over $\mathbb{F}_q$ is defined by operations over the base field $\mathbb{F}_q$ and also its extension field, as shown in (4.3); in contrast, a GII code over $\mathbb{F}_q$ is defined over the same field $\mathbb{F}_q$. As a result, generalized tensor product codes are more flexible than GII codes, and generally GII codes cannot be used to construct

ME-LRCs over very small fields, e.g., the binary field.

The goal of this section is to study the exact relation between generalized tensor product codes and GII codes. We will show that GII codes are in fact a subclass of generalized tensor product codes. The idea is to reformulate the parity-check matrix of a GII code into the form of a parity-check matrix of a generalized tensor product code. Establishing this relation allows some code properties of GII codes to be obtained directly from known results about generalized tensor product codes. We start by considering the II codes, a two-level case of GII codes, to illustrate our idea.

4.5.1 Integrated Interleaving Codes

We follow the definition of II codes in [36]. Let $\mathcal{C}_i$, $i = 1, 2$, be $[n, k_i, d_i]_q$ linear codes over $\mathbb{F}_q$ such that $\mathcal{C}_2 \subset \mathcal{C}_1$ and $d_2 > d_1$. An II code $\mathcal{C}_{II}$ is defined as follows:

$$\mathcal{C}_{II} = \left\{ \boldsymbol{c} = (\boldsymbol{c}_0, \boldsymbol{c}_1, \ldots, \boldsymbol{c}_{m-1}) : \boldsymbol{c}_i \in \mathcal{C}_1, 0 \leq i < m, \text{ and } \sum_{i=0}^{m-1} \alpha^{bi} \boldsymbol{c}_i \in \mathcal{C}_2, \, b = 0, 1, \ldots, \gamma - 1 \right\}, \quad (4.5)$$

where α is a primitive element of $\mathbb{F}_q$ and $\gamma < m \leq q - 1$.

According to the above definition, it is known that the parity-check matrix of $\mathcal{C}_{II}$ is

$$H_{II} = \begin{bmatrix} I & \otimes & H_1 \\ \Gamma_2 & \otimes & H_2 \end{bmatrix}, \quad (4.6)$$

where $\otimes$ denotes the Kronecker product. The matrices H_1 and $\begin{bmatrix} H_1 \\ H_2 \end{bmatrix}$ over $\mathbb{F}_q$ are the parity-check matrices of $\mathcal{C}_1$ and $\mathcal{C}_2$, respectively, the matrix I over $\mathbb{F}_q$ is an $m \times m$ identity matrix, and Γ_2 over $\mathbb{F}_q$ is the parity-check matrix of an $[m, m - \gamma, \gamma + 1]_q$ code in the following form

$$\Gamma_2 = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & \alpha & \cdots & \alpha^{m-1} \\ 1 & \alpha^2 & \cdots & \alpha^{2(m-1)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{(\gamma-1)} & \cdots & \alpha^{(\gamma-1)(m-1)} \end{bmatrix}. \quad (4.7)$$

Remark 4.5.1. The parity-check matrix H_{II} over $\mathbb{F}_q$ in (4.6) of $\mathcal{C}_{II}$ is obtained by operations over the same field $\mathbb{F}_q$. In contrast, the parity-check matrix H over $\mathbb{F}_q$ in (4.3) of a generalized tensor product code is obtained by operations over both the base field $\mathbb{F}_q$ and its extension field. □

Remark 4.5.2. In general, the codes $\mathcal{C}_1$ and $\mathcal{C}_2$ in (4.5) are chosen to be RS codes [36]. If $\mathcal{C}_1$ and $\mathcal{C}_2$ are chosen to be binary codes, then m can only be $m = 1$. □

To see the relation between II codes and generalized tensor product codes, we reformulate H_{II} in (4.6) into the following form, by *splitting* the rows of H_2,

$$H_{II} = \begin{bmatrix} I & \otimes & H_1 \\ \Gamma_2 & \otimes & H_2(1) \\ \Gamma_2 & \otimes & H_2(2) \\ \vdots & \vdots & \vdots \\ \Gamma_2 & \otimes & H_2(k_1 - k_2) \end{bmatrix}, \tag{4.8}$$

where the matrix H_1 over $\mathbb{F}_q$ is the parity-check matrix of $\mathcal{C}_1$, and is treated as a vector over the extension field $\mathbb{F}_{q^{n-k_1}}$ here; correspondingly, the matrix I is treated as an $m \times m$ identity matrix over $\mathbb{F}_{q^{n-k_1}}$. For $1 \leq i \leq k_1 - k_2$, $H_2(i)$ over $\mathbb{F}_q$ represents the ith row of H_2, and Γ_2 over $\mathbb{F}_q$ is the matrix in (4.7).

Now, referring to the matrix in (4.3), the matrix in (4.8) can be interpreted as a parity-check matrix of a $(1 + k_1 - k_2)$-level generalized tensor product code over $\mathbb{F}_q$. Thus, we conclude that an II code is a generalized tensor product code. Using the properties of generalized tensor product codes, we can directly obtain the following result, which was proved in [36] in an alternative way.

Lemma 4.5.1. *The code $\mathcal{C}_{II}$ is a linear code over $\mathbb{F}_q$ of length $N = nm$, dimension $K = (m - \gamma)k_1 + \gamma k_2$, and minimum distance $D \geq \min\{(\gamma + 1)d_1, d_2\}$.*

Proof. For $1 \leq i \leq k_1 - k_2$, let the following parity-check matrix

$$\begin{bmatrix} H_1 \\ H_2(1) \\ \vdots \\ H_2(i) \end{bmatrix}$$

define an $[n, k_1 - i, d_{2,i}]_q$ code. It is clear that $d_1 \leq d_{2,1} \leq d_{2,2} \leq \cdots \leq d_{2,k_1-k_2} = d_2$.

From the properties of generalized tensor product codes, the redundancy is $N - K = nm - K = (n - k_1)m + \gamma(k_1 - k_2)$; that is, the dimension is $K = k_1(m - \gamma) + k_2\gamma$. Using Theorem 4.3.1, the minimum distance is $D \geq \min\left\{d_1(\gamma+1), d_{2,1}(\gamma+1), \ldots, d_{2,k_1-k_2-1}(\gamma+1), d_{2,k_1-k_2}\right\} = \min\left\{(\gamma+1)d_1, d_2\right\}$. ■

4.5.2 Generalized Integrated Interleaving Codes

With the similar idea used in the previous subsection, we continue our proof for GII codes. We use the definition of GII codes from [88] for consistency.

Let $\mathcal{C}_i$, $i = 0, 1, \ldots, \gamma$, be $[n, k_i, d_i]_q$ codes over $\mathbb{F}_q$ such that

$$\mathcal{C}_{i_s} = \cdots = \mathcal{C}_{i_{s-1}+1} \subset \mathcal{C}_{i_{s-1}} = \cdots = \mathcal{C}_{i_{s-2}+1} \subset \cdots \subset \mathcal{C}_{i_1} = \cdots = \mathcal{C}_1 \subset \mathcal{C}_0, \tag{4.9}$$

where $i_0 = 0$ and $i_s = \gamma$. The minimum distances satisfy $d_0 \leq d_1 \leq \cdots \leq d_\gamma$. A GII code $\mathcal{C}_{GII}$ is defined as:

$$\mathcal{C}_{GII} = \left\{\boldsymbol{c} = (\boldsymbol{c}_0, \boldsymbol{c}_1, \ldots, \boldsymbol{c}_{m-1}) : \boldsymbol{c}_i \in \mathcal{C}_0, 0 \leq i < m, \text{and } \sum_{i=0}^{m-1} \alpha^{bi}\boldsymbol{c}_i \in \mathcal{C}_{\gamma-b},\ b = 0, 1, \ldots, \gamma-1\right\}, \tag{4.10}$$

where α is a primitive element of $\mathbb{F}_q$ and $\gamma < m \leq q - 1$.

Let us first define some matrices which will be used below. Let the matrix I over $\mathbb{F}_q$ be an $m \times m$ identity matrix. Let H_0 over $\mathbb{F}_q$ be the parity-check matrix of $\mathcal{C}_0$. For $1 \leq j \leq s$, let the matrix $\begin{bmatrix} H_0 \\ H_{i_j} \end{bmatrix}$ over $\mathbb{F}_q$ represent the parity-check matrix of $\mathcal{C}_{i_j}$, where

$$H_{i_j} = \begin{bmatrix} H_{i_1 \setminus i_0} \\ H_{i_2 \setminus i_1} \\ \vdots \\ H_{i_j \setminus i_{j-1}} \end{bmatrix}.$$

For any $i \leq j$, let matrix $\Gamma(i,j;\alpha)$ over $\mathbb{F}_q$ be the parity-check matrix of an $[m, m-(j-i+1), j-i+2]_q$ code in the following form

$$\Gamma(i,j;\alpha) = \begin{bmatrix} 1 & \alpha^i & \cdots & \alpha^{i(m-1)} \\ 1 & \alpha^{i+1} & \cdots & \alpha^{(i+1)(m-1)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^j & \cdots & \alpha^{j(m-1)} \end{bmatrix}. \tag{4.11}$$

Now, according to the definition in (4.10), using the matrices introduced above, the parity-check matrix of $\mathcal{C}_{GII}$ is

$$H_{GII} = \begin{bmatrix} I & \otimes & H_0 \\ \Gamma(0, i_s - i_{s-1} - 1; \alpha) & \otimes & H_{i_s} \\ \Gamma(i_s - i_{s-1}, i_s - i_{s-2} - 1; \alpha) & \otimes & H_{i_{s-1}} \\ \vdots & \vdots & \vdots \\ \Gamma(i_s - i_2, i_s - i_1 - 1; \alpha) & \otimes & H_{i_2} \\ \Gamma(i_s - i_1, i_s - i_0 - 1; \alpha) & \otimes & H_{i_1} \end{bmatrix}, \tag{4.12}$$

which can be transformed into the form of

$$H_{GII} = \begin{bmatrix} I & \otimes & H_0 \\ \Gamma(0, i_s - i_0 - 1; \alpha) & \otimes & H_{i_1 \setminus i_0} \\ \Gamma(0, i_s - i_1 - 1; \alpha) & \otimes & H_{i_2 \setminus i_1} \\ \vdots & \vdots & \vdots \\ \Gamma(0, i_s - i_{s-2} - 1; \alpha) & \otimes & H_{i_{s-1} \setminus i_{s-2}} \\ \Gamma(0, i_s - i_{s-1} - 1; \alpha) & \otimes & H_{i_s \setminus i_{s-1}} \end{bmatrix}. \tag{4.13}$$

To make a connection between GII codes and generalized tensor product codes, we further

reformulate the matrix H_{GII} in (4.13) as follows,

$$H_{GII} = \left[\begin{array}{lcl}
I & \otimes & H_0 \\ \hline
\Gamma(0, i_s - i_0 - 1; \alpha) & \otimes & H_{i_1 \setminus i_0}(1) \\
\vdots & \vdots & \vdots \\
\Gamma(0, i_s - i_0 - 1; \alpha) & \otimes & H_{i_1 \setminus i_0}(k_{i_0} - k_{i_1}) \\ \hline
\Gamma(0, i_s - i_1 - 1; \alpha) & \otimes & H_{i_2 \setminus i_1}(1) \\
\vdots & \vdots & \vdots \\
\Gamma(0, i_s - i_1 - 1; \alpha) & \otimes & H_{i_2 \setminus i_1}(k_{i_1} - k_{i_2}) \\ \hline
\vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots \\ \hline
\Gamma(0, i_s - i_{s-2} - 1; \alpha) & \otimes & H_{i_{s-1} \setminus i_{s-2}}(1) \\
\vdots & \vdots & \vdots \\
\Gamma(0, i_s - i_{s-2} - 1; \alpha) & \otimes & H_{i_{s-1} \setminus i_{s-2}}(k_{i_{s-2}} - k_{i_{s-1}}) \\ \hline
\Gamma(0, i_s - i_{s-1} - 1; \alpha) & \otimes & H_{i_s \setminus i_{s-1}}(1) \\
\vdots & \vdots & \vdots \\
\Gamma(0, i_s - i_{s-1} - 1; \alpha) & \otimes & H_{i_s \setminus i_{s-1}}(k_{i_{s-1}} - k_{i_s})
\end{array}\right], \tag{4.14}$$

where, in the first level, the matrix H_0 over $\mathbb{F}_q$ is treated as a vector over the extension field $\mathbb{F}_{q^{n-k_0}}$, and correspondingly the matrix I is treated as an $m \times m$ identity matrix over $\mathbb{F}_{q^{n-k_0}}$. For $1 \leq x \leq s$ and $1 \leq y \leq k_{i_{x-1}} - k_{i_x}$, $H_{i_x \setminus i_{x-1}}(y)$ over $\mathbb{F}_q$ represents the yth row of the matrix $H_{i_x \setminus i_{x-1}}$.

Now, referring to the matrix in (4.3), the matrix in (4.14) can be seen as a parity-check matrix of a $(1 + k_0 - k_{i_s})$-level generalized tensor product code over $\mathbb{F}_q$. As a result, we can directly obtain the following lemma, which was also proved in [88] in a different way.

Lemma 4.5.2. *The code $\mathcal{C}_{GII}$ is a linear code over $\mathbb{F}_q$ of length $N = nm$, dimension $K = \sum_{x=1}^{\gamma} k_x + (m - \gamma)k_0 = \sum_{j=1}^{s} (i_j - i_{j-1})k_{i_j} + (m - \gamma)k_0$, and minimum distance $D \geq \min\{(\gamma + 1)d_0, (\gamma - i_1 + 1)d_{i_1}, \ldots, (\gamma - i_{s-1} + 1)d_{i_{s-1}}, d_{i_s}\}$.*

Proof. For $1 \leq x \leq s$ and $1 \leq y \leq k_{i_{x-1}} - k_{i_x}$, let the following parity-check matrix

$$\left[\begin{array}{c} H_0 \\ \hline H_{i_1 \setminus i_0}(1) \\ \vdots \\ H_{i_1 \setminus i_0}(k_{i_0} - k_{i_1}) \\ \hline \vdots \\ \hline H_{i_x \setminus i_{x-1}}(1) \\ \vdots \\ H_{i_x \setminus i_{x-1}}(y) \end{array}\right]$$

define an $[n, k_{i_{x-1}} - y, d_{i_x,y}]_q$ code, so we have $d_{i_{x-1}} \leq d_{i_x,1} \leq d_{i_x,2} \leq \cdots \leq d_{i_x,k_{i_{x-1}}-k_{i_x}} = d_{i_x}$. From the properties of generalized tensor product codes, it is easy to obtain the dimension $K = \sum_{j=1}^{s}(i_j - i_{j-1})k_{i_j} + (m-\gamma)k_0$. From Theorem 4.3.1, the minimum distance satisfies

$$\begin{aligned} D &\geq \min\Big\{(\gamma+1)d_0, (\gamma+1)d_{i_1,1}, \ldots, (\gamma+1)d_{i_1,k_{i_0}-k_{i_1}-1}, (\gamma - i_1 + 1)d_{i_1}, \\ &\qquad \ldots, \ldots, (\gamma - i_{s-1} + 1)d_{i_{s-1}}, (\gamma - i_{s-1} + 1)d_{i_s,1}, \ldots, (\gamma - i_{s-1} + 1)d_{i_s,k_{i_{s-1}}-k_{i_s}-1}, d_{i_s}\Big\} \\ &= \min\Big\{(\gamma+1)d_0, (\gamma - i_1 + 1)d_{i_1}, \ldots, (\gamma - i_{s-1} + 1)d_{i_{s-1}}, d_{i_s}\Big\}. \end{aligned}$$

■

Remark 4.5.3. In some prior works, we find that generalized tensor product codes are called generalized error-location (GEL) codes [12, 51]. Recently, in [88], the similarity between GII codes and GEL codes was observed. However, the exact relation between them was not studied. In [88], the author also proposed a new generalized integrated interleaving scheme over binary BCH codes, called GII-BCH codes. These codes can also be seen as a special case of generalized tensor product codes. □

4.6 Conclusion

In this chapter, we presented a general construction for ME-LRCs over small fields. This construction yields optimal ME-LRCs with respect to an upper bound on the minimum distance for a wide

range of code parameters. Then, an erasure decoder was proposed and corresponding correctable erasure patterns were identified. ME-LRCs based on Reed-Solomon codes were shown to be optimal among all codes having the same erasure-correcting capability. Finally, generalized integrated interleaving codes were proved to be a subclass of generalized tensor product codes, thus giving the exact relation between these two code families.

Acknowledgement

This chapter is in part a reprint of the material in the paper: Pengfei Huang, Eitan Yaakobi, and Paul H. Siegel, "Multi-erasure locally recoverable codes over small fields," in *Proc. 55th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, Monticello, IL, USA, Oct. 2017, pp. 1123–1130. The dissertation author was the primary investigator and author of this paper.

Chapter 5

Syndrome-Coupled Rate-Compatible Error-Correcting Codes

5.1 Introduction

Rate-compatible error-correcting codes (ECCs) consist of a set of extended codes, where all symbols of the higher rate code are part of the lower rate code. The idea of rate-compatible codes dates back to Davida and Reddy [20]. The most commonly used way to construct such codes is by puncturing; that is, to start with a good low-rate code and then successively discard some of the coded symbols (parity-check symbols) to produce higher-rate codes. This approach has been used for convolutional codes [33], turbo codes [48, 65], and low-density parity-check (LDPC) codes [24, 32]. The performance of punctured codes depends on the selected puncturing pattern. However, in general, determining good puncturing patterns is nontrivial, usually done with the aid of computer simulations.

The second approach is by extending; that is, to start with a good high-rate code and then successively add more parity-check symbols to generate lower-rate codes. A two-level extending method called Construction X was introduced in [49] to find new codes with good minimum distance, and later was generalized to Construction XX [3]. Extension-based rate-compatible LDPC codes were designed in [45, 80]. More recently, the extending approach was used to construct capacity-achieving rate-compatible

polar codes [37, 44].

The goal of this chapter is to provide a systematic approach for constructing rate-compatible codes with theoretically guaranteed properties. We use the extending approach and propose a new algebraic construction for rate-compatible codes; the properties of the constructed codes are then analyzed from both combinatorial and probabilistic perspectives. We make contributions in the following aspects: 1) lower bounds for rate-compatible codes, which have not been fully explored before, are derived; 2) a simple and general construction of rate-compatible codes based on cosets and syndromes is proposed, and some examples are given; 3) minimum distances of the constructed codes are determined, decoding algorithms are presented, and correctable error-erasure patterns are studied; 4) a connection to recent capacity-achieving rate-compatible polar codes is made; 5) performance of two-level rate-compatible codes on multi-level cell (MLC) flash memories is evaluated.

The remainder of the chapter is organized as follows. In Section 5.2, we give the formal definition of rate-compatible codes and introduce notation used in this chapter. In Section 5.3, we study lower bounds for rate-compatible codes. In Section 5.4, we present a general construction for M-level rate-compatible codes, whose minimum distances are studied. Correctable patterns of errors and erasures are also investigated. In Section 5.5, we show our construction can generate capacity-achieving rate-compatible codes by choosing the component codes properly. In Section 5.6, we evaluate the performance of two-level BCH-based and LDPC-based rate-compatible codes on MLC flash memories. We conclude the chapter in Section 5.7.

5.2 Definitions and Preliminaries

In this section, we give the basic definitions and preliminaries that will be used in this chapter.

We use the notation $[n]$ to denote the set $\{1,\ldots,n\}$. For a length-n vector $\boldsymbol{v}$ over $\mathbb{F}_q$ and a set $\mathcal{I} \subseteq [n]$, the operation $\pi_{\mathcal{I}}(\boldsymbol{v})$ denotes the restriction of the vector $\boldsymbol{v}$ to coordinates in the set $\mathcal{I}$, and $w_q(\boldsymbol{v})$ represents the Hamming weight of the vector $\boldsymbol{v}$ over $\mathbb{F}_q$. The transpose of a matrix H is written as H^T. A linear code $\mathcal{C}$ over $\mathbb{F}_q$ of length n, dimension k, and minimum distance d will be denoted by $\mathcal{C} = [n,k,d]_q$ or by $[n,k,d]_q$ for simplicity; in some cases, we will use notation $[n,k]_q$ to indicate only length and dimension. For any integers $a > b$, the summation in the form of $\sum_{i=a}^{b} X_i$ is defined to be 0. A binomial

coefficient $\binom{a}{b}$ is defined to be 0 if $a < b$. For a set $\mathcal{C}$, $|\mathcal{C}|$ represents its cardinality. The q-ary entropy function H_q: $[0,1] \to [0,1]$, is defined by $H_q(x) = -x\log_q x - (1-x)\log_q(1-x) + x\log_q(q-1)$.

Now, we present the definition of rate-compatible codes.

Definition 5.2.1. *For $1 \leq i \leq M$, let $\mathcal{C}_i$ be an $[n_i, k, d_i]_q$ linear code, where $n_1 < n_2 < \cdots < n_M$. The encoder of $\mathcal{C}_i$ is denoted by $\mathcal{E}_{\mathcal{C}_i} : \mathbb{F}_q^k \to \mathcal{C}_i$. These M linear codes are said to be M-level rate-compatible, if for each i, $1 \leq i \leq M-1$, the following condition is satisfied for every possible input $\boldsymbol{u} \in \mathbb{F}_q^k$,*

$$\mathcal{E}_{\mathcal{C}_i}(\boldsymbol{u}) = \pi_{[n_i]}\Big(\mathcal{E}_{\mathcal{C}_{i+1}}(\boldsymbol{u})\Big). \tag{5.1}$$

We denote this M-level rate-compatible relation among these codes by $\mathcal{C}_1 \prec \mathcal{C}_2 \prec \cdots \prec \mathcal{C}_M$.

Remark 5.2.1. For $1 \leq i \leq M-1$, the rates satisfy $R_i = \frac{k}{n_i} > R_{i+1} = \frac{k}{n_{i+1}}$, but the minimum distances obey $d_i \leq d_{i+1}$. For systematic codes, the condition in (5.1) indicates that the set of parity-check symbols of a higher rate code is a subset of the parity-check symbols of a lower rate code. □

We will use the memoryless q-ary symmetric channel W with crossover probability p. For every pair of a sent symbol $x \in \mathbb{F}_q$ and a received symbol $y \in \mathbb{F}_q$, the conditional probability is:

$$\Pr\{y|x\} = \begin{cases} 1-p & \text{if } y = x \\ p/(q-1) & \text{if } y \neq x \end{cases}$$

The capacity of this channel is $C(W) = 1 - H_q(p)$ [64].

For a linear code $\mathcal{C} = [n,k,d]_q$ over a q-ary symmetric channel, let $P_e^{(n)}(\boldsymbol{x})$ denote the conditional block probability of error, assuming that $\boldsymbol{x}$ was sent, $\boldsymbol{x} \in \mathcal{C}$. Let $P_e^{(n)}(\mathcal{C})$ denote the average probability of error of this code. Due to symmetry, assuming equiprobable codewords, it is clear that,

$$P_e^{(n)}(\mathcal{C}) = \frac{1}{|\mathcal{C}|}\sum_{\boldsymbol{x}\in\mathcal{C}} P_e^{(n)}(\boldsymbol{x}) = P_e^{(n)}(\boldsymbol{x}).$$

5.3 Lower Bounds for Rate-Compatible Codes

In this section, we derive lower bounds for rate-compatible codes.

5.3.1 A General Lower Bound for M-Level Rate-Compatible Codes

Based on the technique used in the derivation of the Gilbert-Varshamov (GV) bound, we derive a GV-like lower bound for M-level rate-compatible codes.

Theorem 5.3.1. *There exist M-level rate-compatible codes $\mathcal{C}_1 \prec \mathcal{C}_2 \prec \cdots \prec \mathcal{C}_M$, where $\mathcal{C}_i = [n_i = n_1 + \sum_{j=2}^{i} r_j,\ k,\ \geq d_i]_q$ for $1 \leq i \leq M$, if the following inequalities are satisfied for all $1 \leq i \leq M$,*

$$d_i = \max\left\{ d : \sum_{m=0}^{d-2} \binom{n_1 + \sum_{j=2}^{i} r_j - 1}{m} (q-1)^m < \frac{q^{n_1 + \sum_{j=2}^{i} r_j - k}}{M} \right\}. \tag{5.2}$$

Proof. We first define an $(n_M - k) \times n_M$ matrix Φ_M over $\mathbb{F}_q$ in the following block lower triangular form,

$$\Phi_M = \begin{bmatrix} H_{1,1} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ H_{2,1} & H_{2,2} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ H_{M-1,1} & H_{M-1,2} & \cdots & H_{M-1,M-1} & \mathbf{0} \\ H_{M,1} & H_{M,2} & \cdots & H_{M,M-1} & H_{M,M} \end{bmatrix}, \tag{5.3}$$

where $H_{1,1}$ is an $(n_1 - k) \times n_1$ matrix. For $2 \leq i \leq M$, the matrix $H_{i,1}$ has size $r_i \times n_1$. For $2 \leq i \leq M$ and $2 \leq j \leq i$, the matrix $H_{i,j}$ has size $r_i \times r_j$.

For $1 \leq i \leq M$, we assign the upper left $(n_i - k) \times n_i$ submatrix of Φ_M, denoted by Φ_i, to be the parity-check matrix of $\mathcal{C}_i$. For example, the matrices $H_{1,1}$ and Φ_M are parity-check matrices of $\mathcal{C}_1$ and $\mathcal{C}_M$, respectively.

Now, we show how to construct $H_{i,j}$, $1 \leq i \leq M$ and $1 \leq j \leq i$, such that each code $\mathcal{C}_i$ has its desired code parameters.

First, for $2 \leq i \leq M$, we choose $H_{i,i}$ to be an $r_i \times r_i$ identity matrix. For $3 \leq i \leq M$ and $2 \leq j \leq i-1$, we choose matrix $H_{i,j}$ to be an arbitrary matrix in $\mathbb{F}_q^{r_i \times r_j}$. Next, we construct columns of $H_{i,1}$, $1 \leq i \leq M$, iteratively, as the technique used in the proof of the GV bound. We use $\boldsymbol{h}_\ell(i)$, $1 \leq \ell \leq n_1$ and $1 \leq i \leq M$, to denote the ℓth column of the matrix Φ_i which is the parity-check matrix of $\mathcal{C}_i$. Assume that we have already added the leftmost $\ell - 1$ columns of the matrix Φ_M. In order to show that in $\mathbb{F}_q^{n_M - k}$

there is a vector that can be used as the ℓth column $\boldsymbol{h}_\ell(M)$ of the matrix Φ_M, we only need to show that the total number of bad vectors is less than q^{n_M-k}. We count the number of bad vectors as follows.

For the code $\mathcal{C}_1$, it requires that every d_1-1 columns in Φ_1 are linearly independent. A bad vector for the ℓth column $\boldsymbol{h}_\ell(1)$ in Φ_1 is a vector that can be expressed as a linear combination of d_1-2 columns in the preceding $\ell-1$ columns. There are at most a total of $\sum_{m=0}^{d_1-2}\binom{\ell-1}{m}(q-1)^m$ such bad vectors, so we exclude at most $N_1(\ell)=\sum_{m=0}^{d_1-2}\binom{\ell-1}{m}(q-1)^m\times q^{\sum_{j=2}^{M} r_j}$ bad vectors for the column $\boldsymbol{h}_\ell(M)$.

Similarly, for the code $\mathcal{C}_i$, $2\le i\le M$, it requires that every d_i-1 columns in Φ_i are linearly independent. A bad vector for the ℓth column $\boldsymbol{h}_\ell(i)$ in Φ_i is a vector that can be expressed as a linear combination of d_i-2 columns in the preceding $\ell-1+\sum_{j=2}^{i} r_j$ selected columns, so we have at most a total of $\sum_{m=0}^{d_i-2}\binom{\ell-1+\sum_{j=2}^{i} r_j}{m}(q-1)^m$ such bad vectors. Then, we exclude at most $N_i(\ell)=\sum_{m=0}^{d_i-2}\binom{\ell-1+\sum_{j=2}^{i} r_j}{m}(q-1)^m\times q^{\sum_{x=i+1}^{M} r_x}$ bad vectors for the column $\boldsymbol{h}_\ell(M)$.

Since we assume that the inequalities (5.2) are satisfied, we have $N_i(\ell)<\frac{q^{n_M-k}}{M}$ for $1\le i\le M$ and $1\le\ell\le n_1$. Thus, we have $\sum_{i=1}^{M}N_i(\ell)<q^{n_M-k}$, which indicates that a good column $\boldsymbol{h}_\ell(M)$ can be found. ■

The following corollary follows from Theorem 5.3.1, which shows that there exist good rate-compatible codes in the sense that each code can meet the corresponding asymptotic GV bound.

Corollary 5.3.2. *There exist M-level rate-compatible codes $\mathcal{C}_1\prec\mathcal{C}_2\prec\cdots\prec\mathcal{C}_M$, where $\mathcal{C}_i=[n_i,\ k=R_in_i,\ \ge\delta_in_i]_q$ for $1\le i\le M$ and $\delta_M\le 1-(1/q)$, simultaneously meeting the asymptotic GV bound:*

$$R_i\ge 1-H_q(\delta_i). \tag{5.4}$$

Proof. Let $V_q(n,t)=\sum_{m=0}^{t}\binom{n}{m}(q-1)^m$. From Theorem 5.3.1, there exist M-level rate-compatible codes $\mathcal{C}_i=[n_i,\ k=R_in_i,\ \ge\delta_in_i]_q$ for $1\le i\le M$ such that

$$V_q(n_i-1,\delta_in_i-1)\ge\frac{q^{n_i-k}}{M}. \tag{5.5}$$

Since $V_q(n,t)\le q^{nH_q(t/n)}$ for $0\le t/n\le 1-(1/q)$ [64], we have

$$q^{n_iH_q(\delta_i)}\ge V_q(n_i,\delta_in_i)\ge V_q(n_i-1,\delta_in_i-1)\ge\frac{q^{n_i-k}}{M},$$

which gives $R_i \geq 1 - H_q(\delta_i) - \frac{\log_q M}{n_i}$. As n_i goes to infinity, we obtain the result. ■

5.3.2 A Lower Bound for Two-Level Rate-Compatible Codes with Known Weight Enumerator

For two-level rate-compatible codes, if the weight enumerator of the higher rate code is known, we have the following lower bound.

Theorem 5.3.3. *Let $\mathcal{C}_1$ be an $[n_1, k, d_1]_q$ code with weight enumerator $A(s) = \sum_{w=0}^{n_1} A_w s^w$, where A_w is the number of codewords of Hamming weight w. There exist two-level rate-compatible codes $\mathcal{C}_1 \prec \mathcal{C}_2 = [n_2 = n_1 + r_2,\ k,\ \geq d_2]_q$, if*

$$\sum_{w=1}^{d_2-1} B_w < q^{r_2},$$

where $B_w = \frac{1}{q-1} \sum_{m=1}^{w} A_m \binom{r_2}{w-m} (q-1)^{w-m}$, for $1 \leq w \leq n_2$.

Proof. Let an $(n_1 - k) \times n_1$ matrix H_1 represent the parity-check matrix of $\mathcal{C}_1$. Assume that $\mathcal{C}_2$ has a parity-check matrix H_2 in the form

$$H_2 = \begin{bmatrix} H_1 & \mathbf{0} \\ H & I \end{bmatrix}, \tag{5.6}$$

where H is an $r_2 \times n_1$ matrix and the matrix I represents an $r_2 \times r_2$ identity matrix. Construct an ensemble of $(n_2 - k) \times n_2$ matrices $\{H_2\}$ by using all $r_2 \times n_1$ matrices H over $\mathbb{F}_q$. We then assume a uniform distribution over the ensemble $\{H_2\}$.

We say a matrix H_2 is bad, if there exists a vector $\boldsymbol{x} \in \mathbb{F}_q^{n_2}$ such that $\boldsymbol{x} H_2^T = \mathbf{0}$ and $0 < w_q(\boldsymbol{x}) < d_2$. Thus, we only need to prove the probability $\Pr\{H_2 \text{ is bad}\} < 1$, i.e., not all H_2 are bad. Define sets $\mathcal{B}' = \{\boldsymbol{x} \in \mathbb{F}_q^{n_2} : \boldsymbol{x}[H_1, \mathbf{0}]^T = \mathbf{0}\}$, $\mathcal{B}'' = \{\boldsymbol{x} \in \mathcal{B}' : w_q(\boldsymbol{x}) > 0, \text{ and the leading nonzero entry of } \boldsymbol{x} \text{ is } 1\}$, and $\mathcal{B} = \{\boldsymbol{x} \in \mathcal{B}'' : w_q(\pi_{[n_1]}(\boldsymbol{x})) > 0\}$. We also define $B_w = |\{\boldsymbol{x} \in \mathcal{B} : w_q(\boldsymbol{x}) = w\}|$. It is clear that $B_w = \frac{1}{q-1} \sum_{m=1}^{w} A_m \binom{r_2}{w-m} (q-1)^{w-m}$, for $1 \leq w \leq n_2$. Now, we have

$$\begin{aligned}
\Pr\{H_2 \text{ is bad}\} =& \Pr\{\text{For some } \boldsymbol{x} \in \mathcal{B}',\ 0 < w_q(\boldsymbol{x}) < d_2,\ \boldsymbol{x}[H, I]^T = \mathbf{0}\} \\
=& \Pr\{\text{For some } \boldsymbol{x} \in \mathcal{B}'',\ 0 < w_q(\boldsymbol{x}) < d_2,\ \boldsymbol{x}[H, I]^T = \mathbf{0}\} \\
=& \Pr\{\text{For some } \boldsymbol{x} \in \mathcal{B},\ 0 < w_q(\boldsymbol{x}) < d_2,\ \boldsymbol{x}[H, I]^T = \mathbf{0}\}.
\end{aligned}$$

Then, we have

$$\begin{aligned}\Pr\{H_2 \text{ is bad}\} =& \Pr\{\text{For some } \boldsymbol{x} \in \mathcal{B},\ 0 < w_q(\boldsymbol{x}) < d_2,\ \boldsymbol{x}[H,I]^T = \mathbf{0}\} \\ \overset{(a)}{\leq}& \sum_{\boldsymbol{x}\in\mathcal{B} \text{ and } 0<w_q(\boldsymbol{x})<d_2} \Pr\{\boldsymbol{x}[H,I]^T = \mathbf{0}\} \\ =& \frac{\sum_{w=1}^{d_2-1} B_w}{q^{r_2}},\end{aligned}$$

where step (a) follows from the union bound. ■

5.4 A General Construction for M-Level Rate-Compatible Codes

In this section, we present a general construction for M-level rate-compatible codes $\mathcal{C}_1 \prec \mathcal{C}_2 \prec \cdots \prec \mathcal{C}_M$. We then derive their minimum distances. The decoding algorithm and correctable error-erasure patterns are studied. We focus on the combinatorial property here and will leave the discussion on the capacity-achieving property of our construction to the next section.

In our construction for M-level rate-compatible codes, we need a set of component codes which are defined as follows.

1) Choose a set of nested codes $\mathcal{C}_1^M \subset \mathcal{C}_1^{M-1} \subset \cdots \subset \mathcal{C}_1^1 = \mathcal{C}_1 = [n_1, k, d_1]_q$, where $\mathcal{C}_1^i = [n_1, n_1 - \sum_{m=1}^{i} v_m, d_i]_q$ for $1 \leq i \leq M$. We have $k = n_1 - v_1$ and $d_1 \leq d_2 \leq \cdots \leq d_M$. Define $\mathcal{C}_1^0 = \emptyset$ and for $1 \leq \ell \leq i$, let matrix $H_{\mathcal{C}_1^\ell | \mathcal{C}_1^{\ell-1}}$ represent a $v_\ell \times n_1$ matrix over $\mathbb{F}_q$ such that $\mathcal{C}_1^i$ has the following parity-check matrix:

$$H_{\mathcal{C}_1^i} = \begin{bmatrix} H_{\mathcal{C}_1^1} \\ H_{\mathcal{C}_1^2 | \mathcal{C}_1^1} \\ \vdots \\ H_{\mathcal{C}_1^i | \mathcal{C}_1^{i-1}} \end{bmatrix}.$$

The encoder of code $\mathcal{C}_1$ is denoted by $\mathcal{E}_{\mathcal{C}_1} : \mathbb{F}_q^k \to \mathcal{C}_1$. We also use $\mathcal{E}_{\mathcal{C}_1}^{-1}$ as the inverse of the encoding mapping.

2) For ith level, $2 \leq i \leq M$, consider a set of auxiliary nested codes $\mathcal{A}_i^M \subset \mathcal{A}_i^{M-1} \subset \cdots \subset \mathcal{A}_i^{i+1} \subset \mathcal{A}_i^i$, where $\mathcal{A}_i^j = [n_i, v_i + \sum_{m=2}^{i-1} \lambda_m^i - \sum_{\ell=i+1}^{j} \lambda_i^\ell, \delta_i^j]_q$ for $i \leq j \leq M$. Let matrix $H_{\mathcal{A}_i^i}$ represent an

$(n_i - v_i - \sum_{m=2}^{i-1} \lambda_m^i) \times n_i$ matrix over $\mathbb{F}_q$ and matrix $H_{\mathcal{A}_i^\ell | \mathcal{A}_i^{\ell-1}}$, $i+1 \leq \ell \leq j$, represent a $\lambda_i^\ell \times n_i$ matrix over $\mathbb{F}_q$, such that $\mathcal{A}_i^j$ has the following parity-check matrix:

$$H_{\mathcal{A}_i^j} = \begin{bmatrix} H_{\mathcal{A}_i^i} \\ H_{\mathcal{A}_i^{i+1} | \mathcal{A}_i^i} \\ \vdots \\ H_{\mathcal{A}_i^j | \mathcal{A}_i^{j-1}} \end{bmatrix}.$$

For each $2 \leq i \leq M$, the encoder of code $\mathcal{A}_i^i$ is denoted by $\mathcal{E}_{\mathcal{A}_i^i} : \mathbb{F}_q^{v_i + \sum_{m=2}^{i-1} \lambda_m^i} \to \mathcal{A}_i^i$. We also use $\mathcal{E}_{\mathcal{A}_i^i}^{-1}$ as the inverse of the encoding mapping.

Note that we also define $\mathcal{C}_1^{M+1} = \emptyset$ and $\mathcal{A}_i^{M+1} = \emptyset$ for $2 \leq i \leq M$.

5.4.1 Construction and Minimum Distance

Now, we give a general algebraic construction for rate-compatible codes $\mathcal{C}_1 \prec \mathcal{C}_2 \prec \cdots \prec \mathcal{C}_M$ by using the nested component codes introduced above.

Construction 1: Encoding Procedure

Input: A length-k vector $\boldsymbol{u}$ of information symbols over $\mathbb{F}_q$.

Output: A codeword $\boldsymbol{c}_i \in \mathcal{C}_i$ over $\mathbb{F}_q$, for $i = 1, \ldots, M$.

1: $\boldsymbol{c}_1 = \mathcal{E}_{\mathcal{C}_1}(\boldsymbol{u})$.

2: $\boldsymbol{s}_i = \boldsymbol{c}_1 H_{\mathcal{C}_1^i | \mathcal{C}_1^{i-1}}^T$ for $i = 2, 3, \ldots, M$.

3: **for** $i = 2, \ldots, M$ **do**

4: $\quad \boldsymbol{a}_i^i = \mathcal{E}_{\mathcal{A}_i^i}\Big((\boldsymbol{s}_i, \Lambda_2^i, \ldots, \Lambda_{i-1}^i)\Big)$. // Comment: For $i = 2$, we define $(\boldsymbol{s}_i, \Lambda_2^i, \ldots, \Lambda_{i-1}^i) = \boldsymbol{s}_2$. //

5: $\quad \boldsymbol{c}_i = (\boldsymbol{c}_1, \boldsymbol{a}_2^2, \ldots, \boldsymbol{a}_i^i)$.

6: $\quad$ **for** $j = i+1, \ldots, M$ **do**

7: $\quad\quad \Lambda_i^j = \boldsymbol{a}_i^i H_{\mathcal{A}_i^j | \mathcal{A}_i^{j-1}}^T$.

8: $\quad$ **end for**

9: **end for**

Remark 5.4.1. To make Construction 1 clear, consider the case of $M = 3$ as an example. Then a codeword $\boldsymbol{c}_3 \in \mathcal{C}_3$ has the form: $\boldsymbol{c}_3 = \Big(\boldsymbol{c}_1, \mathcal{E}_{\mathcal{A}_2^2}(\boldsymbol{s}_2), \mathcal{E}_{\mathcal{A}_3^3}(\boldsymbol{s}_3, \Lambda_2^3)\Big)$. The main idea of Construction 1 is to extend the base code $\mathcal{C}_1$ by progressively generating and encoding syndromes of component codes in a proper way. Thus, we call it a *syndrome-coupled* construction. □

We have the following theorem on the code parameters of the constructed rate-compatible codes $\mathcal{C}_1 \prec \mathcal{C}_2 \prec \cdots \prec \mathcal{C}_M$.

Theorem 5.4.1. *From Construction 1, the code $\mathcal{C}_i$, $1 \le i \le M$, has length $N_i = \sum_{j=1}^{i} n_j$ and dimension $K_i = k$. Moreover, assume that $\mathcal{A}_i^j$, $2 \le i \le M$ and $i \le j \le M$, has minimum distance $\delta_i^j \ge d_j - d_{i-1}$. Then $\mathcal{C}_i$ has minimum distance $D_i = d_i$.*

Proof. The code length and dimension are obvious. In the following, we prove the minimum distance. Since the proofs for all $\mathcal{C}_i$, $1 \le i \le M$, are similar, we only give a proof for the code $\mathcal{C}_M$.

We first prove $D_M \ge d_M$ by showing that any nonzero codeword $\boldsymbol{c}_M \in \mathcal{C}_M$ has weight at least d_M. To see this, for any nonzero codeword $\boldsymbol{c}_1 \in \mathcal{C}_1$, there exists an integer γ_1, $1 \le \gamma_1 \le M$, such that $\boldsymbol{c}_1 \in \mathcal{C}_1^{\gamma_1}$ and $\boldsymbol{c}_1 \notin \mathcal{C}_1^{\gamma_1+1}$. Let $\boldsymbol{c}_M \in \mathcal{C}_M$ be the codeword derived from $\boldsymbol{c}_1$. Then, we have $w_q(\boldsymbol{c}_M) \ge w_q(\boldsymbol{c}_1) \ge d_{\gamma_1}$. If $\gamma_1 = M$, we are done; otherwise if $1 \le \gamma_1 \le M-1$ we have $\boldsymbol{s}_{\gamma_1+1} \ne \boldsymbol{0}$ and $\boldsymbol{a}_{\gamma_1+1}^{\gamma_1+1} \ne \boldsymbol{0}$.

Now, for $\boldsymbol{a}_{\gamma_1+1}^{\gamma_1+1}$, there exists an integer γ_2, $\gamma_1 + 1 \le \gamma_2 \le M$, such that $\boldsymbol{a}_{\gamma_1+1}^{\gamma_1+1} \in \mathcal{A}_{\gamma_1+1}^{\gamma_2}$ and $\boldsymbol{a}_{\gamma_1+1}^{\gamma_1+1} \notin \mathcal{A}_{\gamma_1+1}^{\gamma_2+1}$. Then, we have $w_q(\boldsymbol{c}_M) \ge w_q(\boldsymbol{c}_1) + w_q(\boldsymbol{a}_{\gamma_1+1}^{\gamma_1+1}) \ge d_{\gamma_1} + d_{\gamma_2} - d_{\gamma_1} = d_{\gamma_2}$. If $\gamma_2 = M$, done; otherwise for $\gamma_1 + 1 \le \gamma_2 \le M-1$, we have $\Lambda_{\gamma_1+1}^{\gamma_2+1} \ne \boldsymbol{0}$ and $\boldsymbol{a}_{\gamma_2+1}^{\gamma_2+1} \ne \boldsymbol{0}$.

Using the same argument as above, it is clear that we can find a sequence of $\gamma_1 < \gamma_2 < \cdots < \gamma_i$, where i is a certain integer $1 \le i \le M$ and $\gamma_i = M$, such that $w_q(\boldsymbol{c}_1) \ge d_{\gamma_1}$, $w_q(\boldsymbol{a}_{\gamma_1+1}^{\gamma_1+1}) \ge d_{\gamma_2} - d_{\gamma_1}$, $w_q(\boldsymbol{a}_{\gamma_2+1}^{\gamma_2+1}) \ge d_{\gamma_3} - d_{\gamma_2}$, $\cdots$, $w_q(\boldsymbol{a}_{\gamma_{i-1}+1}^{\gamma_{i-1}+1}) \ge d_{\gamma_i} - d_{\gamma_{i-1}} = d_M - d_{\gamma_{i-1}}$. Then, we have $w_q(\boldsymbol{c}_M) \ge w_q(\boldsymbol{c}_1) + \sum_{j=1}^{i-1} w_q(\boldsymbol{a}_{\gamma_j+1}^{\gamma_j+1}) \ge d_M$. Thus, we have $D_M \ge d_M$.

There exists a codeword $\boldsymbol{c}_1 \in \mathcal{C}_1^M$ such that $w_q(\boldsymbol{c}_1) = d_M$, so we have $w_q(\boldsymbol{c}_M) = d_M$, implying $D_M \le d_M$. ■

Next, we provide an example of three-level rate-compatible codes to illustrate Construction 1.

Example 5.4.1. Consider a set of nested binary BCH codes $\mathcal{C}_1^3 = [15,5,7]_2 \subset \mathcal{C}_1^2 = [15,7,5]_2 \subset \mathcal{C}_1^1 =$

$[15,11,3]_2$. Choose a set of auxiliary codes $\mathcal{A}_2^3 = [5,1,4]_2 \subset \mathcal{A}_2^2 = [5,4,2]_2$, and $\mathcal{A}_3^3 = [6,5,2]_2$, where the code $\mathcal{A}_2^3$ is obtained by shortening an $[8,4,4]_2$ extended Hamming code by three information bits.

Then, from Construction 1 and Theorem 5.4.1, we obtain three-level rate-compatible codes $\mathcal{C}_1 = [15,11,3]_2 \prec \mathcal{C}_2 = [20,11,5]_2 \prec \mathcal{C}_3 = [26,11,7]_2$. Note that $\mathcal{C}_1$ and $\mathcal{C}_2$ are optimal, achieving the maximum possible dimensions with the given code length and minimum distance. The dimension of $\mathcal{C}_3$ is close to the upper bound 13 according to the online Table [68]. □

5.4.2 Decoding Algorithm and Correctable Error-Erasure Patterns

In the following, we study decoding algorithms and correctable patterns of errors and erasures for rate-compatible codes obtained from Construction 1. For simple notation and concise analysis, we focus on the code $\mathcal{C}_M$. Any results obtained for $\mathcal{C}_M$ can be easily modified for other codes $\mathcal{C}_i$, $1 \leq i \leq M-1$, so details are omitted.

Assume a codeword $\boldsymbol{c}_M \in \mathcal{C}_M$, $\boldsymbol{c}_M = (\boldsymbol{c}_1, \boldsymbol{a}_2^2, \ldots, \boldsymbol{a}_M^M)$, is transmitted. Let the corresponding received word be $\boldsymbol{y} = (\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_M)$ with errors and erasures, i.e., $\boldsymbol{y} \in (\mathbb{F}_q \cup \{?\})^{N_M}$, where the symbol ? represents an erasure. For $1 \leq i \leq M$, let t_i and τ_i denote the number of errors and erasures in the sub-block $\boldsymbol{y}_i$ of the received word $\boldsymbol{y}$.

The code $\mathcal{C}_M$ can correct any combined error and erasure pattern that satisfies the following condition:

$$\begin{aligned} &2t_1 + \tau_1 \leq d_M - 1, \\ &2t_i + \tau_i \leq \delta_i^M - 1, \ \forall\ 2 \leq i \leq M. \end{aligned} \tag{5.7}$$

To see this, we present a decoding algorithm, referred to as Algorithm 1, for $\mathcal{C}_M$. It uses the following component error-erasure decoders:

a) The error-erasure decoder $\mathcal{D}_{\mathcal{C}_1^i}$ for a coset of the code $\mathcal{C}_1^i$, for $1 \leq i \leq M$, is defined by

$$\mathcal{D}_{\mathcal{C}_1^i} : (\mathbb{F}_q \cup \{?\})^{n_1} \times (\mathbb{F}_q \cup \{?\})^{\Sigma_{j=1}^i v_j} \to \mathcal{C}_1^i + \boldsymbol{e} \cup \{\text{“e”}\}$$

The decoder $\mathcal{D}_{\mathcal{C}_1^i}$ either produces a codeword in the coset $\mathcal{C}_1^i + \boldsymbol{e}$ or a decoding failure “e”. For our

purpose, we require that $\mathcal{D}_{\mathcal{C}_1^i}$ have the following error-erasure correcting capability. For a sent codeword $\boldsymbol{c}$ in the coset $\mathcal{C}_1^i + \boldsymbol{e}$, where the vector $\boldsymbol{e}$ is a coset leader, if the inputs of $\mathcal{D}_{\mathcal{C}_1^i}$ are a length-n_1 received word $\boldsymbol{y}$ having t errors and τ erasures, where $2t + \tau \leq d_i - 1$, and a correct length-$\sum_{j=1}^{i} v_j$ syndrome vector $\boldsymbol{s}$, $\boldsymbol{s} = \boldsymbol{e}H_{\mathcal{C}_1^i}^T$, then $\mathcal{D}_{\mathcal{C}_1^i}$ can correct all these errors and erasures. It is well known that such a decoder exists [64].

b) The error-erasure decoder $\mathcal{D}_{\mathcal{A}_i^j}$ for a coset of the code $\mathcal{A}_i^j$, for $2 \leq i \leq M$ and $i \leq j \leq M$, is defined by

$$\mathcal{D}_{\mathcal{A}_i^j} : \ (\mathbb{F}_q \cup \{?\})^{n_i} \times (\mathbb{F}_q \cup \{?\})^{n_i - v_i - \sum_{m=2}^{i-1} \lambda_m^i + \sum_{\ell=i+1}^{j} \lambda_i^\ell} \rightarrow \mathcal{A}_i^j + \boldsymbol{e} \cup \{\text{“e”}\}$$

The decoder $\mathcal{D}_{\mathcal{A}_i^j}$ either produces a codeword in the coset $\mathcal{A}_i^j + \boldsymbol{e}$ or a decoding failure “e”. Similar to $\mathcal{D}_{\mathcal{C}_1^i}$, we assume that $\mathcal{D}_{\mathcal{A}_i^j}$ has the following error-erasure correcting capability. For a sent codeword $\boldsymbol{c}$ in the coset $\mathcal{A}_i^j + \boldsymbol{e}$, where $\boldsymbol{e}$ is a coset leader, if the inputs of $\mathcal{D}_{\mathcal{A}_i^j}$ are a length-n_i received word $\boldsymbol{y}$ having t errors and τ erasures, where $2t + \tau \leq \delta_i^j - 1$, and a correct length-$(n_i - v_i - \sum_{m=2}^{i-1} \lambda_m^i + \sum_{\ell=i+1}^{j} \lambda_i^\ell)$ syndrome vector $\boldsymbol{s}$, $\boldsymbol{s} = \boldsymbol{e}H_{\mathcal{A}_i^j}^T$, then $\mathcal{D}_{\mathcal{A}_i^j}$ can correct all these errors and erasures.

Now, we present the decoding algorithm as follows.

Algorithm 1: Decoding Procedure for $\mathcal{C}_M$

Input: received word $\boldsymbol{y} = (\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_M)$.

Output: A length-k vector $\boldsymbol{u}$ of information symbols over $\mathbb{F}_q$ or a decoding failure “e”.

1: **for** $i = M, M-1, \ldots, 2$ **do**

2: Let the syndrome $\Lambda_i^i = \boldsymbol{0}$.

3: $\hat{\boldsymbol{a}}_i^i = \mathcal{D}_{\mathcal{A}_i^M}\Big(\boldsymbol{y}_i, (\Lambda_i^i, \Lambda_i^{i+1}, \ldots, \Lambda_i^M)\Big)$.

4: $(\boldsymbol{s}_i, \Lambda_2^i, \ldots, \Lambda_{i-1}^i) = \mathcal{E}_{\mathcal{A}_i^i}^{-1}(\hat{\boldsymbol{a}}_i^i)$. // Comment: For $i = 2$, we define $(\boldsymbol{s}_i, \Lambda_2^i, \ldots, \Lambda_{i-1}^i) = \boldsymbol{s}_2$. //

5: **end for**

6: Let the syndrome $\boldsymbol{s}_1 = \boldsymbol{0}$.

7: $\boldsymbol{c}_1 = \mathcal{D}_{\mathcal{C}_1^M}\big(\boldsymbol{y}_1, (\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_M)\big)$.

8: Output $\boldsymbol{u} = \mathcal{E}_{\mathcal{C}_1}^{-1}(\boldsymbol{c}_1)$ if all above steps are successful; otherwise, return "e".

Theorem 5.4.2. *The code $\mathcal{C}_M$ can correct any combined error and erasure pattern that satisfies the condition in (5.7), by using Algorithm 1.*

Proof. The proof follows from Algorithm 1 that decodes the last sub-block $\boldsymbol{y}_M$ to the first sub-block $\boldsymbol{y}_1$ progressively. First, since the code $\mathcal{A}_M^M$ has minimum distance δ_M^M, it can correct $\boldsymbol{y}_M$ under the condition $2t_M + \tau_M \leq \delta_M^M - 1$. Thus, we obtain correct syndromes $\boldsymbol{s}_M, \Lambda_2^M, \ldots, \Lambda_{M-1}^M$.

Next, with the correct syndrome Λ_{M-1}^M, the coset decoder $\mathcal{D}_{\mathcal{A}_{M-1}^M}$ can correct $\boldsymbol{y}_{M-1}$ under the condition $2t_{M-1} + \tau_{M-1} \leq \delta_{M-1}^M - 1$. Thus, we obtain correct syndromes $\boldsymbol{s}_{M-1}, \Lambda_2^{M-1}, \ldots, \Lambda_{M-2}^{M-1}$.

Conduct above decoding procedure progressively. For any i, $2 \leq i \leq M-2$, using the correct syndromes $\Lambda_i^{i+1}, \ldots, \Lambda_i^M$ for coset decoding, the sub-block $\boldsymbol{y}_i$ can be corrected under the condition $2t_i + \tau_i \leq \delta_i^M - 1$.

At the last step, we have obtained correct syndromes $\boldsymbol{s}_2, \ldots, \boldsymbol{s}_M$. Therefore, the sub-block $\boldsymbol{y}_1$ is corrected. ■

Using nested maximum distance separable (MDS) codes as component codes, Construction 1 can generate an *optimal* code $\mathcal{C}_M$ with respect to the capability of correcting certain error-erasure patterns. For simple notation, we present the case of $M = 3$ as an example.

Example 5.4.2. Consider a set of nested MDS codes $\mathcal{C}_1^3 = [n_1, n_1 - d_3 + 1, d_3]_q \subset \mathcal{C}_1^2 = [n_1, n_1 - d_2 + 1, d_2]_q \subset \mathcal{C}_1^1 = [n_1, n_1 - d_1 + 1, d_1]_q$. Choose a set of auxiliary MDS codes $\mathcal{A}_2^3 = [2(d_2 - d_1) - 1, 2d_2 - d_3 - d_1, d_3 - d_1]_q \subset \mathcal{A}_2^2 = [2(d_2 - d_1) - 1, d_2 - d_1, d_2 - d_1]_q$, and $\mathcal{A}_3^3 = [3(d_3 - d_2) - 1, 2(d_3 - d_2), d_3 - d_2]_q$.

Then, from Construction 1 and Theorem 5.4.1, we obtain three-level rate-compatible codes $\mathcal{C}_1 = [n_1, n_1 - d_1 + 1, d_1]_q \prec \mathcal{C}_2 = [n_1 + 2(d_2 - d_1) - 1, n_1 - d_1 + 1, d_2]_q \prec \mathcal{C}_3 = [n_1 + 2(d_2 - d_1) + 3(d_3 - d_2) - 2, n_1 - d_1 + 1, d_3]_q$. □

From the condition in (5.7) and Theorem 5.4.2, the code $\mathcal{C}_3$ can correct any pattern of errors and erasures satisfying

$$2t_i + \tau_i \leq d_3 - d_{i-1} - 1, \ \forall\ 1 \leq i \leq 3, \tag{5.8}$$

where d_0 is defined to be 0.

In general, the dimension of $\mathcal{C}_3$ cannot achieve the upper bounds given by traditional bounds (e.g., Singleton and Hamming bounds). However, $\mathcal{C}_3$ is optimal in the sense of having the largest possible dimension among all codes with the three-level structure and the same error-erasure correcting capability; that is, we have the following lemma.

Lemma 5.4.3. *Let $\mathcal{C}_3$ be a code of length $n_1 + 2(d_2 - d_1) + 3(d_3 - d_2) - 2$ and dimension k_3 over $\mathbb{F}_q$. Each codeword $\boldsymbol{c}_3 \in \mathcal{C}_3$ has three sub-blocks $(\boldsymbol{c}_1, \boldsymbol{a}_2^2, \boldsymbol{a}_3^3)$: 1) $\boldsymbol{c}_1$ of length n_1, 2) $\boldsymbol{a}_2^2$ of length $2(d_2 - d_1) - 1$, and 3) $\boldsymbol{a}_3^3$ of length $3(d_3 - d_2) - 1$. Assume that each sub-block of $\mathcal{C}_3$ can correct all error and erasure patterns satisfying the condition in (5.8). Then, we must have $k_3 \leq n_1 - d_1 + 1$.*

Proof. We prove Lemma 5.4.3 by contradiction.

Let $\mathcal{I}_1$ be the set of any $d_3 - 1$ coordinates of $\boldsymbol{c}_1$, $\mathcal{I}_2$ be the set of any $d_3 - d_1 - 1$ coordinates of $\boldsymbol{a}_2^2$, and $\mathcal{I}_3$ be the set of any $d_3 - d_2 - 1$ coordinates of $\boldsymbol{a}_3^3$. Let $\mathcal{I}$ be the set of all the coordinates of $\boldsymbol{c}_3$.

We have $|\mathcal{I} \backslash (\mathcal{I}_1 \cup \mathcal{I}_2 \cup \mathcal{I}_3)| = n_1 - d_1 + 1$. Now, assume that $k_3 > n_1 - d_1 + 1$. Then, there exist at least two distinct codewords $\boldsymbol{c}_3'$ and $\boldsymbol{c}_3''$ in $\mathcal{C}_3$ that agree on the coordinates in the set $\mathcal{I} \backslash (\mathcal{I}_1 \cup \mathcal{I}_2 \cup \mathcal{I}_3)$. We erase the values on the coordinates in the set $\mathcal{I}_1 \cup \mathcal{I}_2 \cup \mathcal{I}_3$ of both $\boldsymbol{c}_3'$ and $\boldsymbol{c}_3''$. This erasure pattern satisfies the condition in (5.8). Since $\boldsymbol{c}_3'$ and $\boldsymbol{c}_3''$ are distinct, this erasure pattern is uncorrectable. Thus, our assumption that $k_3 > n_1 - d_1 + 1$ is violated. ■

In Algorithm 1, the code $\mathcal{C}_M$ is decoded by M steps, so we can bound the decoding error probability $P_e^{(N_M)}(\mathcal{C}_M)$ of $\mathcal{C}_M$ by the decoding error probability of each step as

$$P_e^{(N_M)}(\mathcal{C}_M) \leq 1 - \left(1 - P_e^{(n_1)}(\mathcal{C}_1^M)\right) \prod_{i=2}^{M} \left(1 - P_e^{(n_i)}(\mathcal{A}_i^M)\right),$$

which provides a fast way to predict the performance of $\mathcal{C}_M$. In particular, if each component code is (shortened) BCH code, then $P_e^{(N_M)}(\mathcal{C}_M)$ can be easily estimated by some calculations. We use a simple example to illustrate this estimation.

Example 5.4.3. Consider two nested binary BCH codes $\mathcal{C}_1^2 = [8191, 7411]_2 \subset \mathcal{C}_1^1 = [8191, 7671]_2$. The codes $\mathcal{C}_1^1$ and $\mathcal{C}_1^2$ can correct 40 and 60 errors, respectively. Choose an auxiliary shortened BCH code

$\mathcal{A}_2^2 = [359, 260]_2$, which can correct 11 errors. Then, from Construction 1, we obtain two-level rate-compatible codes $\mathcal{C}_1 = [8191, 7671]_2 \prec \mathcal{C}_2 = [8550, 7671]_2$. Now, send $\mathcal{C}_2$ over a binary symmetric channel (BSC) with crossover probability p. The error probability of $\mathcal{C}_2$ satisfies

$$\begin{aligned} P_e^{(N_2)}(\mathcal{C}_2) \leq & 1 - \left(1 - P_e^{(n_1)}(\mathcal{C}_1^2)\right)\left(1 - P_e^{(n_2)}(\mathcal{A}_2^2)\right) \\ \leq & 1 - \Big(\sum_{i=0}^{t_1} \binom{n_1}{i} p^i (1-p)^{n_1 - i}\Big)\Big(\sum_{i=0}^{t_2} \binom{n_2}{i} p^i (1-p)^{n_2 - i}\Big), \end{aligned}$$

where $N_2 = 8550$, $n_1 = 8191$, $n_2 = 359$, $t_1 = 60$, and $t_2 = 11$. For instance, for $p = 0.0035$, we compute $P_e^{(N_2)}(\mathcal{C}_2) \leq 1.049 \times 10^{-7}$; for $p = 0.004$, we have $P_e^{(N_2)}(\mathcal{C}_2) \leq 6.374 \times 10^{-6}$. For $p \geq 0.0035$, the performance of $\mathcal{C}_2$ (rate 0.8972) is comparable to, although still worse than, that of a shortened $[8553, 7671]_2$ BCH code $\mathcal{C}_2'$ which has rate 0.8969 and can correct 63 errors. For instance, for $p = 0.0035$ and 0.004, $\mathcal{C}_2'$ has error probabilities 4.035×10^{-8} and 3.315×10^{-6}. □

5.5 Capacity-Achieving Rate-Compatible Codes

In this section, we show that if we choose component codes properly, Construction 1 can generate rate-compatible codes which achieve the capacities of a set of degraded q-ary symmetric channels simultaneously.

More specifically, consider a set of M degraded q-ary symmetric channels $W_1 \succ W_2 \succ \cdots \succ W_M$ with crossover probabilities $p_1 < p_2 < \cdots < p_M$ respectively, where $p_1 > 0$ and $p_M < 1 - (1/q)$. Let $C(W_i)$ denote the capacity of the channel W_i, i.e., $C(W_i) = 1 - H_q(p_i)$. It is clear that $C(W_1) > C(W_2) > \cdots > C(W_M)$. For any rates $R_1 > R_2 > \cdots > R_M$ such that $R_i < C(W_i)$ for all $1 \leq i \leq M$, we will show that Construction 1 can generate rate-compatible codes $\mathcal{C}_1 \prec \mathcal{C}_2 \prec \cdots \prec \mathcal{C}_M$ where $\mathcal{C}_i = [N_i, R_i N_i]_q$ such that the decoding error probability of each $\mathcal{C}_i$ over the channel W_i satisfies $P_e^{(N_i)}(\mathcal{C}_i) \to 0$, as N_i goes to infinity.

To this end, we first present the following lemma on the existence of nested capacity-achieving linear codes.

Lemma 5.5.1. *Consider a set of M degraded q-ary symmetric channels $W_1 \succ W_2 \succ \cdots \succ W_M$ with*

crossover probabilities $p_1 < p_2 < \cdots < p_M$, *where* $p_1 > 0$ *and* $p_M < 1 - (1/q)$. *For any rates* $R_1 > R_2 > \cdots > R_M$ *such that* $R_i < C(W_i) = 1 - H_q(p_i)$, *there exists a sequence of nested linear codes* $\mathcal{C}_1^M = [n, k_M = R_M n]_q \subset \mathcal{C}_1^{M-1} = [n, k_{M-1} = R_{M-1} n]_q \subset \cdots \subset \mathcal{C}_1^1 = [n, k_1 = R_1 n]_q$ *such that the decoding error probability of each* $\mathcal{C}_1^i$ *over the channel* W_i, *under nearest-codeword (maximum-likelihood) decoding, satisfies* $P_e^{(n)}(\mathcal{C}_1^i) \to 0$, *as* n *goes to infinity.*

Proof. To prove the lemma, we will use two known results for the q-ary symmetric channel from Chapter 4 of [64]. We state them as follows.

Lemma 5.5.2. *For the* q*-ary symmetric channel with crossover probability* p, $p \in \big(0, 1 - (1/q)\big)$, *let* n *and* nR *be integers such that* $R < 1 - H_q(p)$. *Let* $\overline{P_e^{(n)}(\mathcal{C})}$ *denote the average of* $P_e^{(n)}(\mathcal{C})$ *over all linear* $[n, nR]_q$ *codes* $\mathcal{C}$ *with nearest-codeword decoding. Then,*

$$\overline{P_e^{(n)}(\mathcal{C})} < 2q^{-nE_q(p,R)},$$

where $E_q(p, R)$ *is a function of* p, R, *and* q, *and* $E_q(p, R) > 0$.

Lemma 5.5.3. *For every* $\rho \in (0, 1]$, *all but a fraction less than* ρ *of the linear* $[n, nR]_q$ *codes* $\mathcal{C}$ *satisfy*

$$P_e^{(n)}(\mathcal{C}) < (1/\rho) 2q^{-nE_q(p,R)}.$$

Now, with the above two lemmas, we are ready to prove Lemma 5.5.1.

Consider an ensemble $\mathcal{G}_1$ of all $k_1 \times n$ full rank matrices over $\mathbb{F}_q$. The size of $\mathcal{G}_1$ is $|\mathcal{G}_1| = (q^n - 1)(q^n - q) \cdots (q^n - q^{k_1 - 1})$. Now, for each matrix $G_i^1 \in \mathcal{G}_1$, $1 \le i \le |\mathcal{G}_1|$, take the lowest k_2 rows to form a new matrix G_i^2. All these new matrices form a new ensemble $\mathcal{G}_2$. It is clear that $|\mathcal{G}_2| = |\mathcal{G}_1|$ and in $\mathcal{G}_2$, each $k_2 \times n$ full rank matrix over $\mathbb{F}_q$ appears $(q^n - q^{k_2})(q^n - q^{k_2+1}) \cdots (q^n - q^{k_1 - 1})$ times. Similarly, for each matrix $G_i^1 \in \mathcal{G}_1$, $1 \le i \le |\mathcal{G}_1|$, take the lowest k_j, $3 \le j \le M$, rows to form a new matrix G_i^j. All these new matrices form a new ensemble $\mathcal{G}_j$. It is clear that $|\mathcal{G}_j| = |\mathcal{G}_1|$ and in $\mathcal{G}_j$, each $k_j \times n$ full rank matrix over $\mathbb{F}_q$ appears $(q^n - q^{k_j})(q^n - q^{k_j+1}) \cdots (q^n - q^{k_1 - 1})$ times.

Note that the number of generator matrices of a linear $[n, k]_q$ code is the same for all such codes.

Therefore, from Lemma 5.5.3, in each ensemble $\mathcal{G}_j$ for $1 \leq j \leq M$, at least a fraction x of all matrices in this ensemble will generate linear codes $\mathcal{C}$ such that the error probability $P_e^{(n)}(\mathcal{C}) < (\frac{1}{1-x})2q^{-nE_q(p_j,R_j)}$.

Now, with the basic set operations, it is not hard to see that for any x satisfying $\frac{M-1}{M} < x < 1$, in the ensemble $\mathcal{G}_1$, we can find a subset $\overline{\mathcal{G}}_1 \subseteq \mathcal{G}_1$ such that: 1) $\overline{\mathcal{G}}_1$ has at least a fraction $Mx-(M-1)$ of all the matrices in $\mathcal{G}_1$, and 2) for each matrix $\overline{G}_1$ in $\overline{\mathcal{G}}_1$, for each j, $1 \leq j \leq M$, the lowest k_j rows of $\overline{G}_1$ will generate a linear code $\mathcal{C}_1^j$ with the error probability $P_e^{(n)}(\mathcal{C}_1^j) < (\frac{1}{1-x})2q^{-nE_q(p_j,R_j)}$.

Thus, there exists a sequence of nested linear codes $\mathcal{C}_1^M = [n, k_M = R_M n]_q \subset \mathcal{C}_1^{M-1} = [n, k_{M-1} = R_{M-1}n]_q \subset \cdots \subset \mathcal{C}_1^1 = [n, k_1 = R_1 n]_q$ such that for all $1 \leq i \leq M$, the error probability $P_e^{(n)}(\mathcal{C}_1^i) \to 0$, as n goes to infinity. ■

Now, we are ready to construct capacity-achieving rate-compatible codes from Construction 1. To do so, we choose a set of nested capacity-achieving codes to be the component codes, which exist according to Lemma 5.5.1.

1) Choose a set of nested capacity-achieving codes $\mathcal{C}_1^M \subset \mathcal{C}_1^{M-1} \subset \cdots \subset \mathcal{C}_1^1 = \mathcal{C}_1 = [n_1, k]_q$, where $\mathcal{C}_1^i = [n_1, n_1 - \sum_{m=1}^{i} v_m]_q$ for $1 \leq i \leq M$. Let $\mathcal{C}_1^i$ have the required rate $R_i < C(W_i)$, and for $\mathcal{C}_1^i$ over the channel W_i, its error probability satisfies $P_e^{(n_1)}(\mathcal{C}_1^i) \to 0$, as n_1 goes to infinity.

2) For ith level, $2 \leq i \leq M$, choose a set of auxiliary nested capacity-achieving codes $\mathcal{A}_i^M \subset \mathcal{A}_i^{M-1} \subset \cdots \subset \mathcal{A}_i^{i+1} \subset \mathcal{A}_i^i$, where $\mathcal{A}_i^j = [n_i, v_i + \sum_{m=2}^{i-1} \lambda_m^i - \sum_{\ell=i+1}^{j} \lambda_i^\ell]_q$ for $i \leq j \leq M$. Let $\mathcal{A}_i^j$ have the required rate $R_j < C(W_j)$, and for $\mathcal{A}_i^j$ over the channel W_j, the decoding error probability satisfies $P_e^{(n_i)}(\mathcal{A}_i^j) \to 0$, as n_i goes to infinity.

Note that compared to Section 5.4, here we care about rate and capacity-achieving property, instead of minimum distance, of each component code.

Theorem 5.5.4. *With the above component codes, from Construction 1, we obtain a sequence of rate-compatible codes $\mathcal{C}_1 \prec \mathcal{C}_2 \prec \cdots \prec \mathcal{C}_M$, where $\mathcal{C}_i$, $1 \leq i \leq M$, has length $N_i = \sum_{j=1}^{i} n_j$, dimension $K_i = k$, and rate R_i. Moreover, for each $\mathcal{C}_i$ over the channel W_i, it is capacity-achieving, i.e., the error probability $P_e^{(N_i)}(\mathcal{C}_i) \to 0$, as N_i goes to infinity.*

Proof. The code length and dimension of $\mathcal{C}_i$ are obvious. In the following, we first prove the rate of $\mathcal{C}_i$; that is, to show $\frac{k}{N_i} = \frac{k}{\sum_{j=1}^{i} n_j} = R_i$. For $i = 1$, it is trivial, since the rate of $\mathcal{C}_1^1$ is R_1. For $i = 2$, observe that

the rate of $\mathcal{C}_1^2$ is $R_2 = \frac{k-v_2}{n_1}$ and the rate of $\mathcal{A}_2^2$ is $R_2 = \frac{v_2}{n_2}$, so we have $(n_1+n_2)R_2 = k$. Similarly, for $3 \le i \le M$, from the rates of codes $\mathcal{C}_1^i$, $\mathcal{A}_2^i$, ..., $\mathcal{A}_i^i$, we have $(n_1+n_2+\cdots+n_i)R_i = k$. Thus, we prove the rates.

Second, we prove the decoding error probability of $\mathcal{C}_M$, since the proof also works for any $\mathcal{C}_i$, $1 \le i \le M-1$. For $\mathcal{C}_M$ over the channel W_M, we use Algorithm 1 for decoding, where each component decoder is chosen to be a nearest-codeword (maximum-likelihood) decoder defined as follows:

a) The nearest-codeword decoder $\mathcal{D}_{\mathcal{C}_1^i}$ for a coset of the code $\mathcal{C}_1^i$, for $1 \le i \le M$, is defined by

$$\mathcal{D}_{\mathcal{C}_1^i} : \mathbb{F}_q^{n_1} \times \mathbb{F}_q^{\sum_{j=1}^{i} v_j} \to \mathcal{C}_1^i + \boldsymbol{e}$$

according to the following decoding rules: for a length-n_1 input vector $\boldsymbol{y}$, and a length-$\sum_{j=1}^{i} v_j$ syndrome vector $\boldsymbol{s}$, if $\boldsymbol{c}$ is a closest codeword to $\boldsymbol{y}$ in the coset $\mathcal{C}_1^i + \boldsymbol{e}$, where the vector $\boldsymbol{e}$ is a coset leader determined by both the code $\mathcal{C}_1^i$ and the syndrome vector $\boldsymbol{s}$, i.e., $\boldsymbol{s} = \boldsymbol{e}H_{\mathcal{C}_1^i}^T$, then $\mathcal{D}_{\mathcal{C}_1^i}(\boldsymbol{y}, \boldsymbol{s}) = \boldsymbol{c}$.

b) The nearest-codeword decoder $\mathcal{D}_{\mathcal{A}_i^j}$ for a coset of the code $\mathcal{A}_i^j$, for $2 \le i \le M$ and $i \le j \le M$, is defined by

$$\mathcal{D}_{\mathcal{A}_i^j} : \mathbb{F}_q^{n_i} \times \mathbb{F}_q^{n_i - v_i - \sum_{m=2}^{i-1} \lambda_m^i + \sum_{\ell=i+1}^{j} \lambda_i^\ell} \to \mathcal{A}_i^j + \boldsymbol{e}$$

according to the following decoding rules: for a length-n_i input vector $\boldsymbol{y}$, and a length-$(n_i - v_i - \sum_{m=2}^{i-1} \lambda_m^i + \sum_{\ell=i+1}^{j} \lambda_i^\ell)$ syndrome vector $\boldsymbol{s}$, if $\boldsymbol{c}$ is a closest codeword to $\boldsymbol{y}$ in the coset $\mathcal{A}_i^j + \boldsymbol{e}$, where the vector $\boldsymbol{e}$ is a coset leader determined by both the code $\mathcal{A}_i^j$ and the syndrome vector $\boldsymbol{s}$, i.e., $\boldsymbol{s} = \boldsymbol{e}H_{\mathcal{A}_i^j}^T$, then $\mathcal{D}_{\mathcal{A}_i^j}(\boldsymbol{y}, \boldsymbol{s}) = \boldsymbol{c}$.

In Algorithm 1, with the above component decoders, the decoding for $\mathcal{C}_M$ consists of M steps, so it will succeed if each step is successful. Thus, we can bound the decoding error probability $P_e^{(N_M)}(\mathcal{C}_M)$ by the decoding error probability of each step as

$$\begin{aligned} P_e^{(N_M)}(\mathcal{C}_M) \le & 1 - \left(1 - P_e^{(n_1)}(\mathcal{C}_1^M)\right) \prod_{i=2}^{M} \left(1 - P_e^{(n_i)}(\mathcal{A}_i^M)\right) \\ = & 1 - \left(1 - P_e^{(\phi_1 N_M)}(\mathcal{C}_1^M)\right) \prod_{i=2}^{M} \left(1 - P_e^{(\phi_i N_M)}(\mathcal{A}_i^M)\right) \end{aligned} \tag{5.9}$$

where constants $\phi_1 = \frac{R_M}{R_1}$ and $\phi_i = \frac{(R_{i-1}-R_i)R_M}{R_i R_{i-1}}$ for $2 \leq i \leq M$. From the chosen capacity-achieving component codes, we already have $P_e^{(\phi_1 N_M)}(\mathcal{C}_1^M) \to 0$ and $P_e^{(\phi_i N_M)}(\mathcal{A}_i^M) \to 0$ as N_M goes to infinity, so in (5.9), $P_e^{(N_M)}(\mathcal{C}_M) \to 0$ as N_M goes to infinity. Thus, we conclude that $\mathcal{C}_M$ can achieve the capacity of the channel W_M. ■

Remark 5.5.1. Polar codes are a family of linear codes that provably achieve the capacity of memoryless symmetric channels using low-complexity encoding and decoding algorithms [4]. Moreover, polar codes were proved to have the nested capacity-achieving property for a set of degraded channels [42]. Thus, they can be used as the component codes in Construction 1 to construct capacity-achieving rate-compatible codes. □

There exist recent independent works on capacity-achieving rateless and rate-compatible codes based on polar codes [37, 44]. By investigating the construction in [37] carefully, we find our construction with polar codes as component codes is equivalent to theirs by mapping the *syndrome* in our construction to the *frozen bits* in their construction using a full rank matrix. In the following, we show the equivalence for the case of two-level rate-compatible codes. Extension to the M-level case can be done in a similar way, so it is omitted.

First, let us consider the construction in [37] for generating two-level rate-compatible codes $\mathcal{C}_1 \prec \mathcal{C}_2$. For simplicity, we refer to the construction in [37] as HHM construction. Consider two nested binary polar codes $\mathcal{C}_1^2 = [n_1, n_1 - v_1 - v_2]_2 \subset \mathcal{C}_1^1 = [n_1, k = n_1 - v_1]_2$. The set of frozen bit indices of $\mathcal{C}_1^i$ is denoted by $\mathcal{F}_1^i$ for $i = 1, 2$. It is clear that $|\mathcal{F}_1^1| = v_1$ and $|\mathcal{F}_1^2| = v_1 + v_2$. The nested property of polar codes gives $\mathcal{F}_1^1 \subset \mathcal{F}_1^2$ [42]. The HHM construction has the following two steps.

For the first step, let a length-n_1 vector $\bar{\boldsymbol{u}}$ have k information bits $\boldsymbol{u}$ on the coordinates in $[n_1] \backslash \mathcal{F}_1^1$ and value 0 on the coordinates in $\mathcal{F}_1^1$. A codeword $\boldsymbol{c}_1 \in \mathcal{C}_1$ is obtained by $\boldsymbol{c}_1 = \bar{\boldsymbol{u}} G_{n_1}$. Here, the code length n_1 is $n_1 = 2^m$ and the matrix G_{n_1} is $G_{n_1} = \mathrm{B}_{n_1} G_2^{\otimes m}$, where $G_2 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$ and B_{n_1} is a bit-reversal permutation matrix defined in [4]. It is known that $G_{n_1} = G_{n_1}^{-1}$, i.e., $G_{n_1} G_{n_1} = I$ [29].

For the second step, to obtain a codeword $\boldsymbol{c}_2 \in \mathcal{C}_2$, the HHM construction uses an auxiliary code $\mathcal{A}_2^2$, which is a capacity-achieving (punctured) polar code, to encode the bits on the coordinates in $\mathcal{F}_1^2 \backslash \mathcal{F}_1^1$

of $\bar{\boldsymbol{u}}$; these bits are denoted by $\pi_{\mathcal{F}_1^2\backslash\mathcal{F}_1^1}(\bar{\boldsymbol{u}})$, which will be treated as the *frozen bits* during the last step of decoding the code $\mathcal{C}_2$. Let $\boldsymbol{a}_2^2$ denote the codeword obtained by encoding $\pi_{\mathcal{F}_1^2\backslash\mathcal{F}_1^1}(\bar{\boldsymbol{u}})$ using $\mathcal{A}_2^2$. Then, the resulting codeword $\boldsymbol{c}_2 \in \mathcal{C}_2$ is $\boldsymbol{c}_2 = (\boldsymbol{c}_1, \boldsymbol{a}_2^2)$.

Now, let us consider our syndrome-coupled construction, i.e., Construction 1. Let us first denote the parity-check matrices of $\mathcal{C}_1^1$ and $\mathcal{C}_1^2$ by $H_{\mathcal{C}_1^1}$ and $H_{\mathcal{C}_1^2} = \begin{bmatrix} H_{\mathcal{C}_1^1} \\ H_{\mathcal{C}_1^2|\mathcal{C}_1^1} \end{bmatrix}$, respectively. Based on Lemma 1 in [29], we have $H_{\mathcal{C}_1^2|\mathcal{C}_1^1} = EH'_{\mathcal{C}_1^2|\mathcal{C}_1^1}$, where the matrix $H'_{\mathcal{C}_1^2|\mathcal{C}_1^1}$ is formed by the columns of G_{n_1} with indices in $\mathcal{F}_1^2\backslash\mathcal{F}_1^1$ and E is a full rank matrix which represents a series of elementary row operations.

The first step of our construction is the same as that of the HHM construction introduced above. In the second step, we use the same auxiliary code $\mathcal{A}_2^2$ to encode the syndrome $\boldsymbol{s}_2$ which is $\boldsymbol{s}_2 = \boldsymbol{c}_1 H^T_{\mathcal{C}_1^2|\mathcal{C}_1^1}$. Let $\boldsymbol{a}_2^2$ denote the codeword obtained by encoding $\boldsymbol{s}_2$ using $\mathcal{A}_2^2$. Then, the codeword $\boldsymbol{c}_2 \in \mathcal{C}_2$ is $\boldsymbol{c}_2 = (\boldsymbol{c}_1, \boldsymbol{a}_2^2)$.

By comparing the HHM construction with our construction, the only difference is that in the second step, we use $\mathcal{A}_2^2$ to encode the syndrome $\boldsymbol{s}_2$, instead of $\pi_{\mathcal{F}_1^2\backslash\mathcal{F}_1^1}(\bar{\boldsymbol{u}})$. In the following, we will prove the equivalence between $\pi_{\mathcal{F}_1^2\backslash\mathcal{F}_1^1}(\bar{\boldsymbol{u}})$ and $\boldsymbol{s}_2$ by showing that $\pi_{\mathcal{F}_1^2\backslash\mathcal{F}_1^1}(\bar{\boldsymbol{u}})$ can be one-to-one mapped to $\boldsymbol{s}_2$. Specifically, we will show that $\boldsymbol{s}_2 = \pi_{\mathcal{F}_1^2\backslash\mathcal{F}_1^1}(\bar{\boldsymbol{u}})E^T$. To see this, we have the following equations,

$$\begin{aligned}
\boldsymbol{s}_2 &= \boldsymbol{c}_1 H^T_{\mathcal{C}_1^2|\mathcal{C}_1^1} \\
&= \bar{\boldsymbol{u}} G_{n_1} H^T_{\mathcal{C}_1^2|\mathcal{C}_1^1} \\
&= \pi_{\mathcal{F}_1^2\backslash\mathcal{F}_1^1}(\bar{\boldsymbol{u}}) G'_{n_1} H'^T_{\mathcal{C}_1^2|\mathcal{C}_1^1} E^T \\
&= \pi_{\mathcal{F}_1^2\backslash\mathcal{F}_1^1}(\bar{\boldsymbol{u}}) E^T,
\end{aligned}$$

where G'_{n_1} is the submatrix of G_{n_1} obtained by taking the rows of G_{n_1} with indices in $\mathcal{F}_1^2\backslash\mathcal{F}_1^1$. The product $G'_{n_1} H'^T_{\mathcal{C}_1^2|\mathcal{C}_1^1}$ is an identity matrix, because $H'_{\mathcal{C}_1^2|\mathcal{C}_1^1}$ is formed by the columns of G_{n_1} with indices in the set $\mathcal{F}_1^2\backslash\mathcal{F}_1^1$ and also we have the property $G_{n_1} G_{n_1} = I$ [29]. In particular, if we choose $E = I$, then $\boldsymbol{s}_2 = \pi_{\mathcal{F}_1^2\backslash\mathcal{F}_1^1}(\bar{\boldsymbol{u}})$. Thus, we prove the equivalence between the HHM construction and our construction.

Since the HHM construction [37] is based on the generator matrix, our construction can be seen as another interpretation of the HHM construction from a parity-check matrix perspective.

5.6 Performance of Two-Level Rate-Compatible Codes for MLC Flash Memories

In this section, we briefly investigate an application of rate-compatible codes to flash memories. Specifically, we construct two-level rate-compatible codes based on BCH and LDPC codes, respectively. Then, we evaluate the performance of these codes for MLC flash memories.

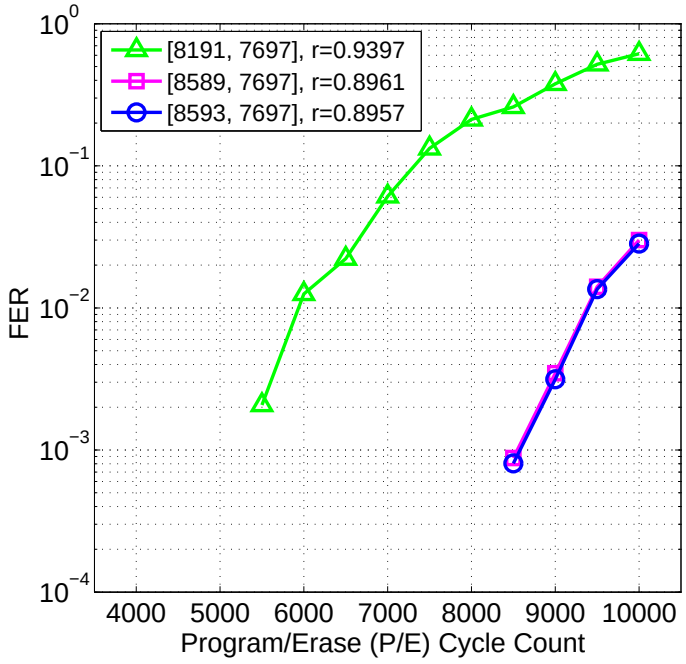
5.6.1 Rate-Compatible Codes Based on BCH Codes

Let us construct two-level rate-compatible codes based on binary BCH codes. We choose two nested binary BCH codes $\mathcal{C}_1^2 = [8191, 7398]_2 \subset \mathcal{C}_1^1 = [8191, 7697]_2$ as our component codes; the codes $\mathcal{C}_1^1$ and $\mathcal{C}_1^2$ can correct 38 and 61 errors, respectively. We also choose an auxiliary shortened BCH code $\mathcal{A}_2^2 = [398, 299]_2$, which can correct 11 errors. Then, from Construction 1, we obtain two-level rate-compatible codes $\mathcal{C}_1 = [8191, 7697]_2 \prec \mathcal{C}_2 = [8589, 7697]_2$, whose code rates are 0.9397 and 0.8961, respectively. We apply $\mathcal{C}_1$ and $\mathcal{C}_2$ to an MLC flash memory and evaluate their performance. In addition, we evaluate a shortened BCH code $\mathcal{C}_3 = [8593, 7697]_2$ with rate 0.8957, whose code length and rate are similar to those of the code $\mathcal{C}_2$. The code $\mathcal{C}_3$ can correct 64 errors.

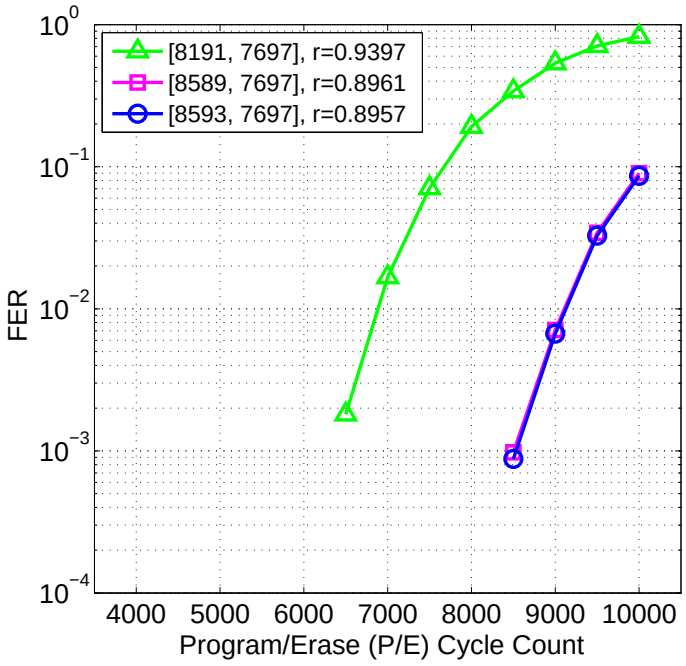
For the performance evaluation of BCH-based two-level rate-compatible codes for an MLC flash memory, we assume that the all-zero codeword is stored and that the memory introduces errors in the locations indicated by our empirical measurements. For the BCH decoder, we assume that if the BCH code (or its coset) could correct t errors, then it would correct any error vector with at most t errors. If the number of errors exceeds t, we assume that the BCH decoder would fail. The constructed rate-compatible codes $\mathcal{C}_1 \prec \mathcal{C}_2$ are evaluated over a total of 20 blocks, i.e., 40960 codewords.

The frame error rate (FER) performance of the constructed codes $\mathcal{C}_1$ and $\mathcal{C}_2$ for the lower page and upper page of an MLC flash memory is shown in Figure 5.1(a) and Figure 5.1(b), respectively. Compared to $\mathcal{C}_1$, the code $\mathcal{C}_2$ extends the lifetime around 3500 program/erase (P/E) cycles for the lower page and around 2000 P/E cycles for the upper page.

In addition, we evaluate the shortened BCH code $\mathcal{C}_3$. The FER performance results for the lower page and upper page are shown in Figure 5.1(a) and Figure 5.1(b), respectively. It can be seen that the FER



(a)



(b)

Figure 5.1: FER performance of two-level rate-compatible codes based on BCH codes for an MLC flash memory: (a) lower page and (b) upper page.

of $\mathcal{C}_2$ is comparable to that of $\mathcal{C}_3$, which indicates the effectiveness of our construction.

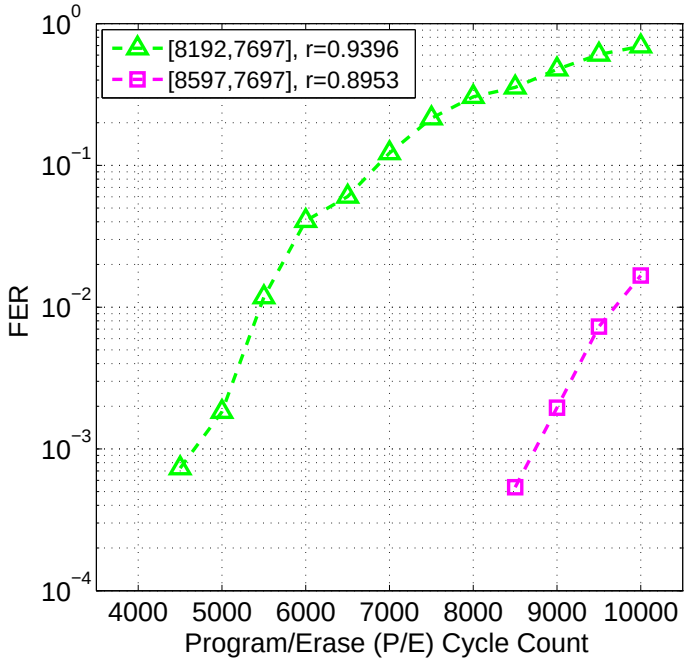
5.6.2 Rate-Compatible Codes Based on LDPC Codes

Let us also construct two-level rate-compatible codes based on binary LDPC codes. We use a Reed-Solomon (RS) codes based construction for regular LDPC codes, since this construction provides a nested and 4-cycle free structure [66]. We can obtain two nested binary LDPC codes $\mathcal{C}_1^2 \subset \mathcal{C}_1^1$, where $\mathcal{C}_1^1$ is a $(4,64)$-regular $[8192,7697]_2$ LDPC code with rate 0.9396 and $\mathcal{C}_1^2$ is a $(7,64)$-regular $[8192,7400]_2$ LDPC code with rate 0.9033. We also choose an auxiliary (4,15)-regular $[405,300]_2$ LDPC code $\mathcal{A}_2^2$. Then, in the second step of Construction 1, we obtain the syndrome $\boldsymbol{s}_2$ of length 297. We add three zeros to the end of $\boldsymbol{s}_2$ to form a new vector which is encoded by $\mathcal{A}_2^2$ to generate the vector $\boldsymbol{a}_2^2$. Thus, from Construction 1, we obtain two-level rate-compatible codes $\mathcal{C}_1 = [8192,7697]_2 \prec \mathcal{C}_2 = [8597,7697]_2$, whose code rates are 0.9396 and 0.8953, respectively.

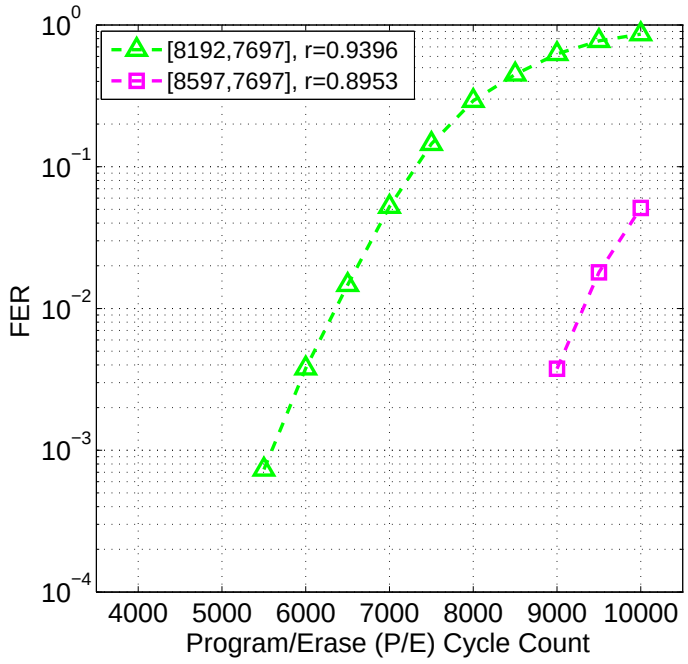
For the performance evaluation of LDPC-based two-level rate-compatible codes for an MLC flash memory, we assume that the all-zero codeword is stored and that the memory introduces errors in the locations indicated by our empirical measurements. We treat the channel as a binary symmetric channel with crossover error probability p equal to the average probability of error reflected in the measured error data. The decoder is based upon belief-propagation (BP) decoding, implemented in software as the floating-point sum-product algorithm (SPA). The maximum number of iterations is set to be 100 and early termination is used. The constructed rate-compatible codes $\mathcal{C}_1 \prec \mathcal{C}_2$ are evaluated over a total of 20 blocks, i.e., 40960 codewords.

The FER performance of the constructed codes $\mathcal{C}_1$ and $\mathcal{C}_2$ for the lower page and upper page of an MLC flash memory is shown in Figure 5.2(a) and Figure 5.2(b), respectively. Compared to $\mathcal{C}_1$, the code $\mathcal{C}_2$ extends the lifetime around 4000 P/E cycles for the lower page and around 3000 P/E cycles for the upper page.

We combine the results from Figure 5.1(a) and Figure 5.2(a), as shown in Figure 5.3(a). We also merge the curves from Figure 5.1(b) and Figure 5.2(b), as shown in Figure 5.3(b). From Figure 5.3(a) and Figure 5.3(b), it is shown that at a higher rate, i.e., 0.94, the $[8191,7697]_2$ BCH code outperforms the $[8192,7697]_2$ LDPC code. However, at a lower rate of 0.90, the $[8597,7697]_2$ LDPC-based code is

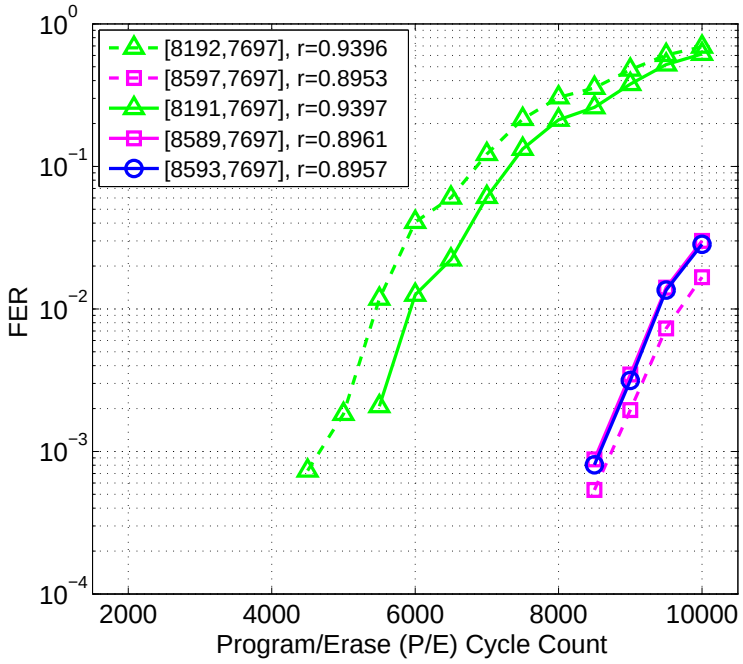


(a)

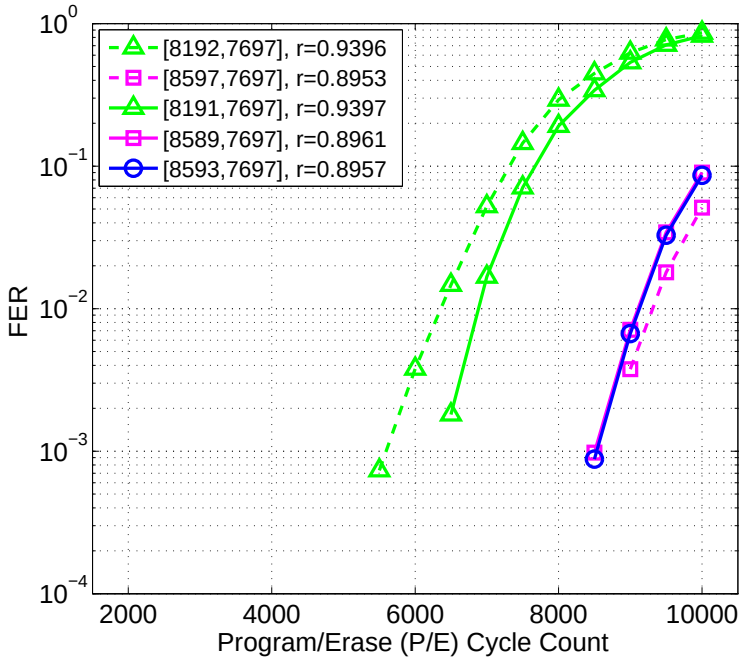


(b)

Figure 5.2: FER performance of two-level rate-compatible codes based on LDPC codes for an MLC flash memory: (a) lower page and (b) upper page.



(a)



(b)

Figure 5.3: FER performance of two-level rate-compatible codes based on BCH and LDPC codes for an MLC flash memory: (a) lower page and (b) upper page.

better than the $[8589, 7697]_2$ BCH-based code.

5.7 Conclusion

In this chapter, we proposed a new algebraic construction for generating rate-compatible codes with increasing minimum distances. We also proved that our construction can generate capacity-achieving rate-compatible codes by using proper component codes, validating the optimality of the construction. With polar codes as component codes, the equivalence between our construction and the one in [37] was identified.

Our construction is very general. Many linear codes (e.g., BCH, RS, and LDPC codes) can be used as its component codes, and some of them were shown as examples. Our parity-check matrix based approach enables us to conveniently obtain the combinatorial properties (e.g., minimum distance) of the constructed rate-compatible codes, as well as their decoders.

Finally, we constructed two-level rate-compatible codes from BCH and LDPC codes, respectively, and evaluated the performance of these codes on MLC flash memories.

Acknowledgement

This chapter is in part a reprint of the material in the paper: Pengfei Huang, Yi Liu, Xiaojie Zhang, Paul H. Siegel, and Erich F. Haratsch, "Syndrome-coupled rate-compatible error-correcting codes," in *Proc. IEEE Information Theory Workshop (ITW)*, Kaohsiung, Taiwan, Nov. 2017, pp. 454–458. The dissertation author was the primary investigator and author of this paper.

Chapter 6

A Class of Error-Correcting Codes with Multi-Level Shared Redundancy

6.1 Introduction

As the volume of data continues to explode, error-correcting codes (ECCs) with multi-level redundancy become increasingly important in data storage, since they can balance the reliability and the total redundancy cost. The idea of using multi-level redundancy dates back to Patel [57]. In [57], a two-level coding scheme was used for a data *block*, which consists of several *sub-blocks* and also extra parity-check symbols shared by all these sub-blocks. The scheme in [57] was later extended in [1]. In [36], integrated interleaving codes with two-level protection were proposed for data storage. A codeword (block) of a two-level integrated interleaving code comprises several component codewords (sub-blocks) of a Reed-Solomon (RS) code $\mathcal{C}$, satisfying the constraints that some linear combinations of these component codewords are codewords of a subcode of $\mathcal{C}$. More recently, generalized integrated interleaving codes were studied in [78, 88].

In this chapter, we present a new class of ECCs with multi-level shared redundancy. We call them ladder codes, since the decoding procedure, which uses multi-level redundancy successively from the lowest level to the highest level, mimics climbing up a ladder.

Our construction is motivated by the construction of tensor product codes, first proposed by Wolf in [86], and later generalized in [40]. A tensor product code is defined by a parity-check matrix that is the tensor product of the parity-check matrices of component codes. Tensor product codes and integrated interleaving codes are similar, and integrated interleaving codes can be treated as a subclass of tensor product codes. A codeword of a tensor product code consists of multiple component codewords (sub-blocks) of equal length. As shown by Example 1 in [87], the encoding steps of a tensor product code involve using *phantom* syndrome symbols, which only appear in the encoding procedure but are not stored in the encoded codeword. By imposing constraints on these phantom syndrome symbols (this step is done over an extension field [40, 86]), some of the information symbols of some sub-blocks are turned into parity-check symbols commonly shared by all the sub-blocks. However, in our ladder codes, these shared parity-check symbols do not reside in sub-blocks; instead, they are protected by other levels of coding. Thus, in some sense, ladder codes can be considered as an *external* version of tensor product codes. As a result, to provide extra protection for sub-blocks, unlike tensor product codes, the encoder for each sub-block in ladder codes can be kept intact and we only need to generate the extra shared redundancy part, which seems an attractive feature for some data storage applications.

Aiming at the specific code structure consisting of multiple sub-blocks and their external shared redundancy, ladder codes provide a systematic way to generate multi-level shared redundancy successively. However, due to this particular structure embedded in the code, the performance of a three-level (or higher) ladder code might be worse than that of a corresponding generalized tensor product code, if one directly compares their rates and minimum distances.

One possible application of ladder codes could be for flash memories [8, 14]. It is well known that in a flash memory block, there exist a few bad pages that have high bit error rates [15, 89, 90]. Using ladder codes, the codewords from good and bad pages are able to share some common redundancy, which can be stored in some spare pages in the flash. However, in this chapter, we only focus on the *theoretical* aspects of ladder codes, leaving their applications as a future work. Our contributions are as follows: 1) We propose a new class of ECCs with multi-level shared redundancy. Specifically, we present a general construction of an m-level ladder code, and determine the code length and dimension; in addition, we derive a lower bound d_L^* on the minimum distance. We also provide explicit examples of ladder codes,

some of which turn out to be optimal with respect to the minimum distance. 2) We present a general result on the correctable error-erasure patterns for ladder codes and give a corresponding decoding algorithm. With respect to erasure correction, it is shown that ladder codes can correct at least $d_L^* - 1$ erasures; as for error correction, ladder codes can correct at least $\lfloor \frac{d_L^*-1}{2} \rfloor$ errors. 3) We compare two-level ladder codes with concatenated codes [64]. Our first code design results in a ladder code possessing the same code parameters as those of a corresponding concatenated code. The second design shows that a ladder code can even outperform a concatenated code in some cases.

The remainder of this chapter is organized as follows. In Section 6.2, we present a general construction of ladder codes, and determine the corresponding code parameters. In Section 6.3, we study the correctable error-erasure patterns of ladder codes and give a corresponding decoding algorithm. In Section 6.4, we compare two-level ladder codes with concatenated codes. We conclude the chapter in Section 6.5.

Throughout this chapter, we use the following notation. The transpose of a matrix H is written as H^T. The cardinality of a set A is denoted by $|A|$. For a vector $\boldsymbol{v}$ over $\mathbb{F}_q$, we use $w_q(\boldsymbol{v})$ to represent its Hamming weight. For two vectors $\boldsymbol{v}$ and $\boldsymbol{u}$ over $\mathbb{F}_q$, we use $\mathrm{d}_q(\boldsymbol{v}, \boldsymbol{u})$ to denote their Hamming distance. A linear code over $\mathbb{F}_q$ of length n, dimension k, and minimum distance d is denoted by $[n, k, d]_q$, where q may be omitted if the field is clear from the context. A linear code which consists of all length-n vectors over $\mathbb{F}_q$ is denoted by $[n, n, 1]_q$, and its dual code only has the all-zero codeword, so it is denoted by $[n, 0, \infty]_q$.

6.2 Ladder Codes: Construction and Minimum Distance

In this section, we present a general construction for ladder codes that have multi-level shared redundancy. We then give the code parameters of a ladder code; in particular, we derive a lower bound on its minimum distance.

6.2.1 Construction of Ladder Codes

An m-level ladder code $\mathcal{C}_L$ over $\mathbb{F}_q$ is based on the following component codes.

1) A collection of m nested $[n, k_i, d_i]_q$ codes $\mathcal{C}_i$, $1 \leq i \leq m$, over $\mathbb{F}_q$, such that

$$\mathcal{C}_m \subset \mathcal{C}_{m-1} \subset \cdots \subset \mathcal{C}_1.$$

The corresponding dimensions satisfy $k_m < k_{m-1} < \cdots < k_1$ and the minimum distances satisfy $d_m \geq d_{m-1} \geq \cdots \geq d_1$. We denote the parity-check matrix of $\mathcal{C}_i$ by

$$H_{\mathcal{C}_i} = \begin{bmatrix} H_1 \\ H_2 \\ \vdots \\ H_i \end{bmatrix},$$

where H_i, $1 \leq i \leq m$, is a matrix of size $(k_{i-1} - k_i) \times n$, by defining $k_0 = n$. The encoder of $\mathcal{C}_1$ is denoted by $\mathcal{E}_{\mathcal{C}_1} : \mathbb{F}_q^{k_1} \to \mathcal{C}_1$. We also use $\mathcal{E}_{\mathcal{C}_1}^{-1}$ as the inverse of the encoding mapping.

2) A collection of $m-1$ $[n_i', k' = \ell, \delta_i]_{q^{v_i}}$ codes $\mathcal{C}_i'$, $2 \leq i \leq m$, over $\mathbb{F}_{q^{v_i}}$, where $v_i = k_{i-1} - k_i$. Without loss of generality, we assume that $n_2' > n_3' > \cdots > n_m'$ and $\delta_2 > \delta_3 > \cdots > \delta_m$. The encoder of $\mathcal{C}_i'$ is systematic and is denoted by $\mathcal{E}_{\mathcal{C}_i'} : \mathbb{F}_{q^{v_i}}^{\ell} \to \mathcal{C}_i'$. We also use $\mathcal{E}_{\mathcal{C}_i'}^{-1}$ as the inverse of the encoding mapping.

3) A collection of $m-1$ $[n_i'', k_i'', d_i'']_q$ codes $\mathcal{C}_i''$, $2 \leq i \leq m$, over $\mathbb{F}_q$. The encoder of $\mathcal{C}_i''$ is denoted by $\mathcal{E}_{\mathcal{C}_i''} : \mathbb{F}_q^{v_i} \to \mathcal{C}_i''$. We also use $\mathcal{E}_{\mathcal{C}_i''}^{-1}$ as the inverse of the encoding mapping.

With the component codes introduced above, the construction of an m-level ladder code $\mathcal{C}_L$ is outlined in the following procedure.

Construction 1: Encoding Procedure for Ladder Codes

Input: ℓ information vectors $\boldsymbol{u}_i \in \mathbb{F}_q^{k_1}$, $1 \leq i \leq \ell$.

Output: a codeword $\boldsymbol{c}_L = (\boldsymbol{c}_1, \ldots, \boldsymbol{c}_\ell, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_m)$ of the ladder code $\mathcal{C}_L$ over $\mathbb{F}_q$, where

- $\boldsymbol{c}_i \in \mathcal{C}_1$, $1 \leq i \leq \ell$, from step 1.
- $\boldsymbol{r}_i = (\boldsymbol{g}_1^i, \ldots, \boldsymbol{g}_{n_i'-\ell}^i)$, $2 \leq i \leq m$, from step 5.

1: For $1 \leq i \leq \ell$, encode $\boldsymbol{u}_i$ according to the code $\mathcal{C}_1$ to obtain a codeword $\boldsymbol{c}_i = (c_{i,1}, c_{i,2}, \ldots, c_{i,n})$, i.e.,

$$\boldsymbol{c}_i = \mathcal{E}_{\mathcal{C}_1}(\boldsymbol{u}_i),\ \forall 1 \le i \le \ell.$$

2: For $2 \le i \le m$ and $1 \le j \le \ell$, compute the ith level syndromes by

$$\boldsymbol{s}_j^i = (s_{j,1}^i, s_{j,2}^i, \ldots, s_{j,v_i}^i) = \boldsymbol{c}_j H_i^T.$$

Since $\boldsymbol{s}_j^i$ is a v_i-dimensional vector over $\mathbb{F}_q$, we treat it as a symbol in $\mathbb{F}_{q^{v_i}}$. See Figure 6.1.

3: For $2 \le i \le m$ and $1 \le j \le n_i' - \ell$, calculate the ith level parity-check symbols $\boldsymbol{p}_j^i$ in $\mathbb{F}_{q^{v_i}}$, by encoding syndromes $(\boldsymbol{s}_1^i, \ldots, \boldsymbol{s}_\ell^i)$ with the code $\mathcal{C}_i'$, i.e.,

$$(\boldsymbol{s}_1^i, \ldots, \boldsymbol{s}_\ell^i, \boldsymbol{p}_1^i, \ldots, \boldsymbol{p}_{n_i'-\ell}^i) = \mathcal{E}_{\mathcal{C}_i'}(\boldsymbol{s}_1^i, \ldots, \boldsymbol{s}_\ell^i).$$

See Figure 6.2.

4: Encode the parity-check symbols obtained in step 3. For $2 \le i \le m$ and $1 \le j \le n_i' - \ell$, apply the code $\mathcal{C}_i''$ to encode $\boldsymbol{p}_j^i$ to obtain $\boldsymbol{g}_j^i$, i.e.,

$$\boldsymbol{g}_j^i = \mathcal{E}_{\mathcal{C}_i''}(\boldsymbol{p}_j^i),$$

where $\boldsymbol{p}_j^i$ is treated as a vector of size v_i over $\mathbb{F}_q$. See Figure 6.2.

5: For $2 \le i \le m$, represent the ith level shared redundancy $\boldsymbol{r}_i$ in the form of $\boldsymbol{r}_i = (\boldsymbol{g}_1^i, \ldots, \boldsymbol{g}_{n_i'-\ell}^i)$.

Remark 6.2.1. For an m-level ladder code $\mathcal{C}_L$ which is obtained in Construction 1, its codeword $\boldsymbol{c}_L = (\boldsymbol{c}_1, \ldots, \boldsymbol{c}_\ell, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_m)$ consists of two ingredients: 1) the ℓ sub-blocks $\boldsymbol{c}_i$, $1 \le i \le \ell$, each representing a codeword in $\mathcal{C}_1$, and 2) a total of $m-1$ parts of shared redundancy denoted by $\boldsymbol{r}_i$, $2 \le i \le m$.

Referring to Figure 6.1 and Figure 6.2, a codeword $\boldsymbol{c}_L$ is comprised of symbols in the regions formed by the solid lines. □

Remark 6.2.2. As in the construction of tensor product codes [40, 86], Construction 1 for ladder codes is based on operations over the base field $\mathbb{F}_q$ as well as its extension fields; see step 3 in Construction 1. One

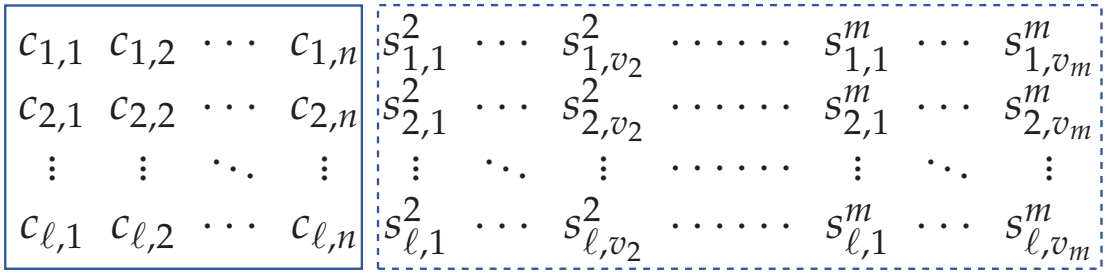


Figure 6.1: Step 2 of the encoding procedure in Construction 1.

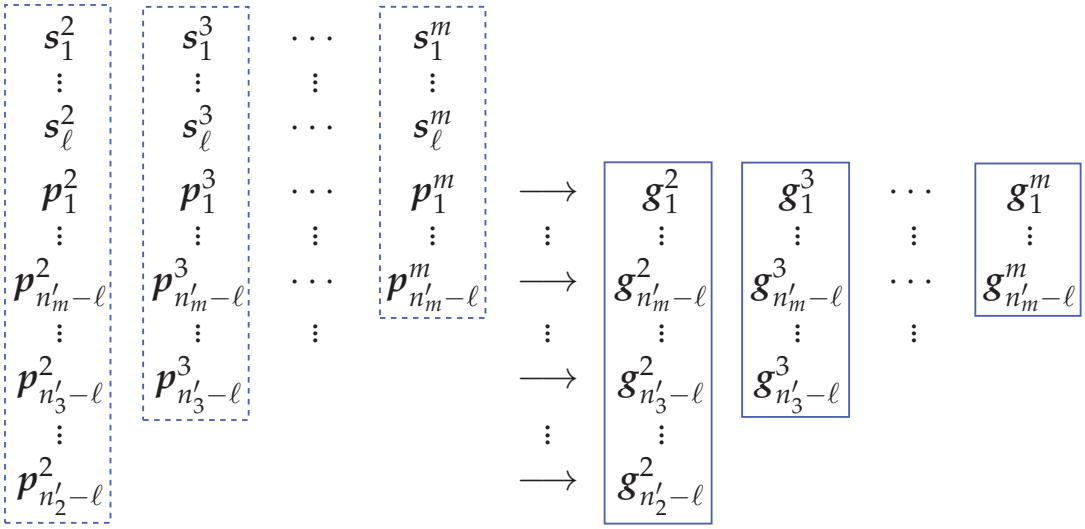


Figure 6.2: Steps 3 and 4 of the encoding procedure in Construction 1.

possible variation of Construction 1 is to modify step 3 by using different component codes so that it is also carried out over the same base field $\mathbb{F}_q$. □

6.2.2 Minimum Distance of Ladder Codes

The following theorem gives the code parameters of a ladder code $\mathcal{C}_L$ generated by Construction 1.

Theorem 6.2.1. *An m-level ladder code $\mathcal{C}_L$ from Construction 1 is a linear code over $\mathbb{F}_q$ of length $n_L = n\ell + \sum_{i=2}^{m} n_i''(n_i' - \ell)$ and dimension $k_L = k_1\ell$. Its minimum distance d_L is lower bounded by d_L^* as*

$$d_L \geq d_L^* = \min\left\{\delta_2 \tilde{d}_1, \delta_3 \tilde{d}_2, \ldots, \delta_m \tilde{d}_{m-1}, d_m\right\},$$

where $\tilde{d}_i = \min\{d_i, d''_{i+1}\}$ *for* $1 \leq i \leq m-1$.

Proof. From the code construction procedure, the code length and dimension can be easily determined. In the following, we derive a lower bound on the minimum distance.

Let $\boldsymbol{c}_L$ be a nonzero codeword of the ladder code $\mathcal{C}_L$. Then, $\boldsymbol{c}_L$ contains nonzero vectors $\boldsymbol{c}_i \in \mathcal{C}_1$, $1 \leq i \leq \ell$. Let the subscripts of all these nonzero vectors form a set $\Phi \subseteq \{1, 2, \ldots, \ell\}$.

Consider the first case that there exists a nonzero vector $\boldsymbol{c}_\lambda \notin \mathcal{C}_2$, $\lambda \in \Phi$. Then, the syndrome $\boldsymbol{s}_\lambda^2 \neq \boldsymbol{0}$, so there are at least δ_2 nonzero symbols in $(\boldsymbol{s}_1^2, \ldots, \boldsymbol{s}_\ell^2, \boldsymbol{p}_1^2, \ldots, \boldsymbol{p}_{n'_2-\ell}^2)$. For any $1 \leq j \leq \ell$, $\boldsymbol{s}_j^2 \neq \boldsymbol{0}$, we have $w_q(\boldsymbol{c}_j) \geq d_1$. For any $1 \leq j \leq n'_2 - \ell$, $\boldsymbol{p}_j^2 \neq \boldsymbol{0}$, we have $w_q(\boldsymbol{g}_j^2) \geq d''_2$. Thus, in total, $w_q(\boldsymbol{c}_L) \geq \delta_2 \min\{d_1, d''_2\}$.

For the second case, if $\boldsymbol{c}_i \in \mathcal{C}_2$, for all $i \in \Phi$, and there exists a nonzero vector $\boldsymbol{c}_\lambda \notin \mathcal{C}_3$, $\lambda \in \Phi$. Then, the syndrome $\boldsymbol{s}_\lambda^3 \neq \boldsymbol{0}$, so there are at least δ_3 nonzero symbols in $(\boldsymbol{s}_1^3, \ldots, \boldsymbol{s}_\ell^3, \boldsymbol{p}_1^3, \ldots, \boldsymbol{p}_{n'_3-\ell}^3)$. For any $1 \leq j \leq \ell$, $\boldsymbol{s}_j^3 \neq \boldsymbol{0}$, we have $w_q(\boldsymbol{c}_j) \geq d_2$. For any $1 \leq j \leq n'_3 - \ell$, $\boldsymbol{p}_j^3 \neq \boldsymbol{0}$, we have $w_q(\boldsymbol{g}_j^3) \geq d''_3$. Thus, in total, $w_q(\boldsymbol{c}_L) \geq \delta_3 \min\{d_2, d''_3\}$.

Similarly, for $3 \leq i \leq m-1$, if $\boldsymbol{c}_j \in \mathcal{C}_i$, for all $j \in \Phi$, and there exists a nonzero vector $\boldsymbol{c}_\lambda \notin \mathcal{C}_{i+1}$, $\lambda \in \Phi$. It can be shown that $w_q(\boldsymbol{c}_L) \geq \delta_{i+1} \min\{d_i, d''_{i+1}\}$.

For the last case, if $\boldsymbol{c}_i \in \mathcal{C}_m$, for all $i \in \Phi$, then it is clear that $w_q(\boldsymbol{c}_L) \geq d_m$. ■

The following corollary follows from Theorem 6.2.1. We give a condition under which the exact minimum distance can be determined.

Corollary 6.2.2. *For an m-level ladder code $\mathcal{C}_L$ generated by Construction 1,*

1) if $d''_i = d_{i-1}$ for all $2 \leq i \leq m$, then

$$d_L \geq d_L^* = \min\left\{\delta_2 d_1, \delta_3 d_2, \ldots, \delta_m d_{m-1}, d_m\right\};$$

2) if $\delta_i \min\{d_{i-1}, d''_i\} \geq d_m$ for all $2 \leq i \leq m$, then

$$d_L = d_m.$$

Proof. The first claim is evident, by applying Theorem 6.2.1. Here, we prove the second claim. On the one hand, since $\delta_i \min\{d_{i-1}, d_i''\} \geq d_m$ for all $2 \leq i \leq m$, we have $d_L \geq d_L^* = d_m$. On the other hand, there exists a codeword $\boldsymbol{c}_L \in \mathcal{C}_L$ with weight d_m. To see this, let $\boldsymbol{c}_1 \in \mathcal{C}_1$ be a codeword with weight d_m and $\boldsymbol{c}_i \in \mathcal{C}_1$, $2 \leq i \leq \ell$, be the all-zero codeword. It is not hard to verify that the corresponding codeword $\boldsymbol{c}_L \in \mathcal{C}_L$ has weight d_m. Thus, we have $d_L \leq d_m$. ■

Now, we present an example of a two-level ladder code to illustrate the encoding procedure of Construction 1.

Example 6.2.1. Let $\mathcal{C}_1$ be the $[8,7,2]_2$ single parity code, with parity-check matrix $H_1 = [1\ 1\ 1\ 1\ 1\ 1\ 1\ 1]$. Let $\mathcal{C}_2 \subset \mathcal{C}_1$ be the $[8,4,4]_2$ extended Hamming code with parity-check matrix $H_{\mathcal{C}_2}$,

$$H_{\mathcal{C}_2} = \left[\begin{array}{c} H_1 \\ \hline H_2 \end{array}\right] = \left[\begin{array}{cccccccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \hline 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{array}\right].$$

Let $\mathcal{C}_2'$ be the $[\ell+1, \ell, 2]_{2^3}$ systematic single parity code. Let $\mathcal{C}_2''$ be the $[4,3,2]_2$ systematic single parity code. Thus, the two-level ladder code $\mathcal{C}_L$ is an $[n_L = 8\ell + 4, k_L = 7\ell, d_L = 4]_2$ code. Note that, for $2 \leq \ell \leq 7$, from the online table [68], $\mathcal{C}_L$ achieves the *optimal* minimum distance.

Suppose that $\ell = 2$ and the two input information vectors are: $\boldsymbol{u}_1 = (1\ 0\ 1\ 0\ 0\ 0\ 0)$ and $\boldsymbol{u}_2 = (1\ 1\ 1\ 1\ 0\ 0\ 0)$. From Construction 1, we have $\boldsymbol{c}_1 = (1\ 0\ 1\ 0\ 0\ 0\ 0\ 0)$ and $\boldsymbol{c}_2 = (1\ 1\ 1\ 1\ 0\ 0\ 0\ 0)$, so $\boldsymbol{s}_1^2 = (0\ 1\ 0)$ and $\boldsymbol{s}_2^2 = (1\ 0\ 0)$. Then, we obtain $\boldsymbol{r}_2 = \boldsymbol{g}_1^2 = (1\ 1\ 0\ 0)$. Thus, the output codeword $\boldsymbol{c}_L = (\boldsymbol{c}_1, \boldsymbol{c}_2, \boldsymbol{r}_2) = (1\,0\,1\,0\,0\,0\,0\,0, 1\,1\,1\,1\,0\,0\,0\,0, 1\,1\,0\,0)$. □

6.3 Correctable Error-Erasure Pattern and Decoding Algorithm

In this section, we study the correctable error-erasure patterns for ladder codes. A decoding algorithm that can correct those patterns is proposed. Explicit results on correctable erasure patterns and correctable error patterns are presented.

6.3.1 Correction Capability of a Linear Code and Its Cosets

To study the error-erasure correcting capability of ladder codes, we start by investigating the correction capability of a linear code and its cosets.

Let us introduce the erasure symbol and related operations. Let ? represent an erasure. We extend the addition operation over $\mathbb{F}_q$ to $\mathbb{F}_q \cup \{?\}$ by defining $x+? =? + x =?$ for $x \in \mathbb{F}_q \cup \{?\}$.

Consider a code $\mathcal{C}$ of length n over $\mathbb{F}_q$. The set of its correctable error-erasure patterns is defined as follows.

Definition 6.3.1. *Let $\mathcal{T}(\mathcal{C})$ be a set of vectors $\boldsymbol{e}$ of length n over $\mathbb{F}_q \cup \{?\}$. We say that $\mathcal{T}(\mathcal{C})$ is a set of correctable error-erasure patterns for the code $\mathcal{C}$ if for any given $\boldsymbol{e} \in \mathcal{T}(\mathcal{C})$, it satisfies the following condition: for every $\boldsymbol{c} \in \mathcal{C}$, the equation $\boldsymbol{c} + \boldsymbol{e} = \boldsymbol{c}' + \boldsymbol{e}'$, where $\boldsymbol{c}' \in \mathcal{C}$ and $\boldsymbol{e}' \in \mathcal{T}(\mathcal{C})$, implies that $\boldsymbol{c}' = \boldsymbol{c}$.*

Based on $\mathcal{T}(\mathcal{C})$, we define the detectable but uncorrectable error-erasure patterns below.

Definition 6.3.2. *A vector $\boldsymbol{e}$ of length n over $\mathbb{F}_q \cup \{?\}$ is a detectable but uncorrectable error-erasure pattern for the code $\mathcal{C}$ if it satisfies the following condition: for every $\boldsymbol{c} \in \mathcal{C}$, $\boldsymbol{y} = \boldsymbol{c} + \boldsymbol{e}$ cannot be expressed as $\boldsymbol{y} = \boldsymbol{c}' + \boldsymbol{e}'$, where $\boldsymbol{c}' \in \mathcal{C}$ and $\boldsymbol{e}' \in \mathcal{T}(\mathcal{C})$. We denote the set of all such detectable but uncorrectable error-erasure patterns by $\Delta(\mathcal{C})$.*

Remark 6.3.1. It is clear that the two sets $\mathcal{T}(\mathcal{C})$ and $\Delta(\mathcal{C})$ are disjoint; that is, if an error-erasure pattern $\boldsymbol{e} \in \mathcal{T}(\mathcal{C})$, then we have $\boldsymbol{e} \notin \Delta(\mathcal{C})$. □

Based on $\mathcal{T}(\mathcal{C})$, we can also define a decoder $\mathcal{D}_{\mathcal{C}}$ for $\mathcal{C}$:

$$\mathcal{D}_{\mathcal{C}} : (\mathbb{F}_q \cup \{?\})^n \to \mathcal{C} \cup \{\text{“e”}\},$$

where “e” is a decoding failure indicator. For a received word $\boldsymbol{y} = \boldsymbol{c} + \boldsymbol{e}$, where $\boldsymbol{c} \in \mathcal{C}$ and $\boldsymbol{e} \in (\mathbb{F}_q \cup \{?\})^n$, the decoder $\mathcal{D}_{\mathcal{C}}$ of $\mathcal{C}$ searches for a codeword $\hat{\boldsymbol{c}} \in \mathcal{C}$ and an error-erasure pattern $\hat{\boldsymbol{e}} \in \mathcal{T}(\mathcal{C})$ such that $\boldsymbol{y} = \hat{\boldsymbol{c}} + \hat{\boldsymbol{e}}$:

1) If such $\hat{\boldsymbol{c}}$ and $\hat{\boldsymbol{e}}$ exist, then they are unique and the decoder $\mathcal{D}_{\mathcal{C}}$ outputs $\hat{\boldsymbol{c}}$.

2) If such $\hat{\boldsymbol{c}}$ and $\hat{\boldsymbol{e}}$ do not exist, then the decoder $\mathcal{D}_{\mathcal{C}}$ outputs a decoding failure indicator “e”.

Remark 6.3.2. If $\boldsymbol{e} \in \mathcal{T}(\mathcal{C})$, the decoder $\mathcal{D}_{\mathcal{C}}$ will output the correct codeword; if $\boldsymbol{e} \in \Delta(\mathcal{C})$, the decoder $\mathcal{D}_{\mathcal{C}}$ will detect the error and output a decoding failure "e". □

We will use the cosets of a linear code. For an $[n,k,d]_q$ linear code $\mathcal{C}$ with a parity-check matrix H, let $\mathcal{C}(\boldsymbol{s})$ be the set of all vectors in $\mathbb{F}_q^n$ that have syndrome $\boldsymbol{s}$, namely,

$$\mathcal{C}(\boldsymbol{s}) = \{\boldsymbol{x} \in \mathbb{F}_q^n : \boldsymbol{x}H^T = \boldsymbol{s}\}.$$

The code $\mathcal{C}(\boldsymbol{s})$ is a coset of $\mathcal{C}$. For $\boldsymbol{s} = \boldsymbol{0}$, we have $\mathcal{C}(\boldsymbol{s}) = \mathcal{C}$. It is easy to verify the following two claims:
1) A set of correctable error-erasure patterns for $\mathcal{C}$ is also a set of correctable error-erasure patterns for its coset $\mathcal{C}(\boldsymbol{s})$, i.e., $\mathcal{T}(\mathcal{C}) = \mathcal{T}\big(\mathcal{C}(\boldsymbol{s})\big)$.
2) A set of detectable but uncorrectable error-erasure patterns for $\mathcal{C}$ is also a set of detectable but uncorrectable error-erasure patterns for $\mathcal{C}(\boldsymbol{s})$, i.e., $\Delta(\mathcal{C}) = \Delta\big(\mathcal{C}(\boldsymbol{s})\big)$.

6.3.2 A General Result on Correctable Error-Erasure Patterns

Now we are ready to investigate the correctable error-erasure patterns of a ladder code $\mathcal{C}_L$ which is generated by Construction 1.

Suppose a codeword $\boldsymbol{c}_L \in \mathcal{C}_L$ is transmitted, and the corresponding received word is $\boldsymbol{y} = \boldsymbol{c}_L + \boldsymbol{e}$, $\boldsymbol{e} \in (\mathbb{F}_q \cup \{?\})^{n_L}$. More specifically, we use the following notation:
1) the transmitted codeword is $\boldsymbol{c}_L = (\boldsymbol{c}_1, \ldots, \boldsymbol{c}_\ell, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_m)$, where $\boldsymbol{r}_i = (\boldsymbol{g}_1^i, \ldots, \boldsymbol{g}_{n_i'-\ell}^i)$ for $2 \le i \le m$;
2) the error-erasure vector is $\boldsymbol{e} = (\boldsymbol{e}_1, \ldots, \boldsymbol{e}_\ell, \boldsymbol{e}_2'', \ldots, \boldsymbol{e}_m'')$, where $\boldsymbol{e}_i'' = (\boldsymbol{e}_1^i, \ldots, \boldsymbol{e}_{n_i'-\ell}^i)$ for $2 \le i \le m$;
3) the received word is $\boldsymbol{y} = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_\ell, \boldsymbol{y}_2'', \ldots, \boldsymbol{y}_m'')$, where $\boldsymbol{y}_i'' = (\boldsymbol{y}_1^i, \ldots, \boldsymbol{y}_{n_i'-\ell}^i)$ for $2 \le i \le m$.

Given the error-erasure vector $\boldsymbol{e}$, we define the following new error-erasure vector denoted by $\boldsymbol{e}_i' = (e_{i,1}', e_{i,2}', \ldots, e_{i,n_i'}')$, $2 \le i \le m$. For $1 \le j \le \ell$,

$$e_{i,j}' = \begin{cases} 0 & \boldsymbol{e}_j \in \mathcal{T}(\mathcal{C}_{i-1}) \\ ? & \boldsymbol{e}_j \in \Delta(\mathcal{C}_{i-1}) \\ w & \boldsymbol{e}_j \notin \mathcal{T}(\mathcal{C}_{i-1}) \cup \Delta(\mathcal{C}_{i-1}); \end{cases} \tag{6.1}$$

for $1 \leq j \leq n_i' - \ell$,

$$e'_{i,j+\ell} = \begin{cases} 0 & \boldsymbol{e}_j^i \in \mathcal{T}(\mathcal{C}_i'') \\ ? & \boldsymbol{e}_j^i \in \Delta(\mathcal{C}_i'') \\ w & \boldsymbol{e}_j^i \notin \mathcal{T}(\mathcal{C}_i'') \cup \Delta(\mathcal{C}_i''), \end{cases} \tag{6.2}$$

where w is an indeterminate symbol. The above assignment for $e'_{i,j}$, $1 \leq j \leq \ell$, can be interpreted as follows: $e'_{i,j} = 0$ if $\boldsymbol{e}_j$ is correctable, $e'_{i,j} =?$ if $\boldsymbol{e}_j$ is detectable but uncorrectable, and $e'_{i,j} = w$ if $\boldsymbol{e}_j$ is miscorrected. The same interpretation holds for $e'_{i,j+\ell}$, $1 \leq j \leq n_i' - \ell$.

With the error-erasure vector $\boldsymbol{e}_i'$ defined above in (6.1) and (6.2), we say that $\boldsymbol{e}_i'$ is correctable if $\boldsymbol{e}_i' \in \mathcal{T}(\mathcal{C}_i')$ for all $w \in \mathbb{F}_{q^{v_i}}$. In other words, the set of vectors obtained by replacing each w in $\boldsymbol{e}_i'$ by all possible elements in $\mathbb{F}_{q^{v_i}}$ are in $\mathcal{T}(\mathcal{C}_i')$.

The following theorem describes the correctable error-erasure patterns for a ladder code $\mathcal{C}_L$.

Theorem 6.3.3. *An m-level ladder code $\mathcal{C}_L$ from Construction 1 corrects any error-erasure pattern $\boldsymbol{e} = (\boldsymbol{e}_1, \ldots, \boldsymbol{e}_\ell, \boldsymbol{e}_2'', \ldots, \boldsymbol{e}_m'')$, $\boldsymbol{e} \in (\mathbb{F}_q \cup \{?\})^{n_L}$, that satisfies the following two conditions:*
1) for $1 \leq i \leq \ell$, the error-erasure pattern $\boldsymbol{e}_i$ is correctable by $\mathcal{C}_m$, i.e., $\boldsymbol{e}_i \in \mathcal{T}(\mathcal{C}_m)$;
2) for $2 \leq i \leq m$, the ith level error-erasure pattern $\boldsymbol{e}_i' = (e'_{i,1}, e'_{i,2}, \ldots, e'_{i,n_i'})$, defined in (6.1) and (6.2), is correctable by $\mathcal{C}_i'$, i.e., $\boldsymbol{e}_i' \in \mathcal{T}(\mathcal{C}_i')$ for all $w \in \mathbb{F}_{q^{v_i}}$.

6.3.3 A Decoding Algorithm for Ladder Codes

To prove Theorem 6.3.3, we present a decoding algorithm, referred to as Algorithm 1, for a ladder code $\mathcal{C}_L$. It employs the following decoders for different component codes used in Construction 1:

a) The decoder $\mathcal{D}_{\mathcal{C}_i}$ for a coset of the code $\mathcal{C}_i$ with syndrome $\boldsymbol{s}$, for $1 \leq i \leq m$, is defined by

$$\mathcal{D}_{\mathcal{C}_i} : (\mathbb{F}_q \cup \{?\})^n \times (\mathbb{F}_q \cup \{?\})^{n-k_i} \to \mathcal{C}_i(\boldsymbol{s}) \cup \{\text{“e”}\}.$$

For a length-n input vector $\boldsymbol{y}$ and a length-$(n-k_i)$ syndrome $\boldsymbol{s}$ without erasures, the decoder $\mathcal{D}_{\mathcal{C}_i}(\boldsymbol{y}, \boldsymbol{s})$ searches for a codeword $\hat{\boldsymbol{c}} \in \mathcal{C}_i(\boldsymbol{s})$ and an error-erasure pattern $\hat{\boldsymbol{e}} \in \mathcal{T}(\mathcal{C}_i)$ (Here, we use $\mathcal{T}(\mathcal{C}_i)$, since we have $\mathcal{T}\big(\mathcal{C}_i(\boldsymbol{s})\big) = \mathcal{T}(\mathcal{C}_i)$) such that $\boldsymbol{y} = \hat{\boldsymbol{c}} + \hat{\boldsymbol{e}}$. If such $\hat{\boldsymbol{c}}$ and $\hat{\boldsymbol{e}}$ exist, the decoder outputs $\hat{\boldsymbol{c}}$; otherwise, the decoder returns a decoding failure “e”.

b) The decoder $\mathcal{D}_{\mathcal{C}_i''}$ for the code $\mathcal{C}_i''$, for $2 \leq i \leq m$, is defined by

$$\mathcal{D}_{\mathcal{C}_i''} : (\mathbb{F}_q \cup \{?\})^{n_i''} \rightarrow \mathcal{C}_i'' \cup \{\text{“e”}\}.$$

For a length-n_i'' input vector $\boldsymbol{y}$, the decoder $\mathcal{D}_{\mathcal{C}_i''}(\boldsymbol{y})$ searches for a codeword $\hat{\boldsymbol{c}} \in \mathcal{C}_i''$ and an error-erasure pattern $\hat{\boldsymbol{e}} \in \mathcal{T}(\mathcal{C}_i'')$ such that $\boldsymbol{y} = \hat{\boldsymbol{c}} + \hat{\boldsymbol{e}}$. If such $\hat{\boldsymbol{c}}$ and $\hat{\boldsymbol{e}}$ exist, the decoder outputs $\hat{\boldsymbol{c}}$; otherwise, the decoder returns a decoding failure “e”.

c) The decoder $\mathcal{D}_{\mathcal{C}_i'}$ for the code $\mathcal{C}_i'$, for $2 \leq i \leq m$, is defined by

$$\mathcal{D}_{\mathcal{C}_i'} : (\mathbb{F}_{q^{v_i}} \cup \{?\})^{n_i'} \rightarrow \mathcal{C}_i' \cup \{\text{“e”}\}.$$

For a length-n_i' input vector $\boldsymbol{y}$, the decoder $\mathcal{D}_{\mathcal{C}_i'}(\boldsymbol{y})$ searches for a codeword $\hat{\boldsymbol{c}} \in \mathcal{C}_i'$ and an error-erasure pattern $\hat{\boldsymbol{e}} \in \mathcal{T}(\mathcal{C}_i')$ such that $\boldsymbol{y} = \hat{\boldsymbol{c}} + \hat{\boldsymbol{e}}$. If such $\hat{\boldsymbol{c}}$ and $\hat{\boldsymbol{e}}$ exist, the decoder outputs $\hat{\boldsymbol{c}}$; otherwise, the decoder returns a decoding failure “e”.

Note that each decoder defined above is merely based on the correctable set of the corresponding code.

The decoding algorithm for $\mathcal{C}_L$ is outlined as follows.

Algorithm 1: Decoding Procedure for $\mathcal{C}_L$

Input: received word $\boldsymbol{y} = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_\ell, \boldsymbol{y}_2'', \ldots, \boldsymbol{y}_m'')$.

Output: information vectors $\boldsymbol{u}_i$, $1 \leq i \leq \ell$, or a decoding failure indicator “e”.

// Level 1:

1: Let the 1st level syndrome $\hat{\mathbf{s}}_i^1 = \mathbf{0}$, $1 \leq i \leq \ell$.

2: Let $\mathcal{F} = \{i : \mathcal{D}_{\mathcal{C}_1}(\boldsymbol{y}_i, \hat{\mathbf{s}}_i^1) == \text{“e”}, 1 \leq i \leq \ell\}$.

3: **for** $1 \leq i \leq \ell$ and $i \notin \mathcal{F}$ **do**

4: $\quad \hat{\boldsymbol{c}}_i \leftarrow \mathcal{D}_{\mathcal{C}_1}(\boldsymbol{y}_i, \hat{\mathbf{s}}_i^1)$.

5: **end for**

// Level $2-$ Level m:

1: **for** $\mu = 2,3,\ldots,m$ **do**

2: Let $\mathcal{F}_\mu = \{i : \mathcal{D}_{\mathcal{C}''_\mu}(\boldsymbol{y}_i^\mu) == \text{“e”}, 1 \le i \le n'_\mu - \ell\}$.

3: **for** $1 \le i \le n'_\mu - \ell$ and $i \notin \mathcal{F}_\mu$ **do**

4: $\hat{\boldsymbol{g}}_i^\mu \leftarrow \mathcal{D}_{\mathcal{C}''_\mu}(\boldsymbol{y}_i^\mu)$, and $\hat{\boldsymbol{p}}_i^\mu \leftarrow \mathcal{E}^{-1}_{\mathcal{C}''_\mu}(\hat{\boldsymbol{g}}_i^\mu)$.

5: **end for**

6: Let $\mathbf{X}_\mu = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{n'_\mu})$ be the word over $\mathbb{F}_{q^{v_\mu}} \cup \{?\}$ that is defined as follows:

for $1 \le i \le \ell$,

$$\boldsymbol{x}_i = \begin{cases} \hat{\boldsymbol{c}}_i H_\mu^T & \text{if } i \notin \mathcal{F} \\ ? & \text{otherwise;} \end{cases}$$

for $1 \le j \le n'_\mu - \ell$,

$$\boldsymbol{x}_{j+\ell} = \begin{cases} \hat{\boldsymbol{p}}_j^\mu & \text{if } j \notin \mathcal{F}_\mu \\ ? & \text{otherwise.} \end{cases}$$

7: **if** $\mathcal{D}_{\mathcal{C}'_\mu}(\mathbf{X}_\mu) == \text{“e”}$ **then**

8: Go to step 17.

9: **else**

10: Get syndromes $(\hat{\mathbf{s}}_1^\mu, \ldots, \hat{\mathbf{s}}_\ell^\mu)$ by: $(\hat{\mathbf{s}}_1^\mu, \ldots, \hat{\mathbf{s}}_\ell^\mu, \hat{\boldsymbol{p}}_1^\mu, \ldots, \hat{\boldsymbol{p}}_{n'_\mu - \ell}^\mu) \leftarrow \mathcal{D}_{\mathcal{C}'_\mu}(\mathbf{X}_\mu)$.

11: **end if**

12: Update the index list $\mathcal{F}$: $\mathcal{F} = \{i : \mathcal{D}_{\mathcal{C}_\mu}\big(\boldsymbol{y}_i, (\hat{\mathbf{s}}_i^1, \hat{\mathbf{s}}_i^2, \ldots, \hat{\mathbf{s}}_i^\mu)\big) == \text{“e”}, 1 \le i \le \ell\}$.

13: **for** $1 \le i \le \ell$ and $i \notin \mathcal{F}$ **do**

14: Update $\hat{\boldsymbol{c}}_i$: $\hat{\boldsymbol{c}}_i \leftarrow \mathcal{D}_{\mathcal{C}_\mu}\big(\boldsymbol{y}_i, (\hat{\mathbf{s}}_i^1, \hat{\mathbf{s}}_i^2, \ldots, \hat{\mathbf{s}}_i^\mu)\big)$.

15: **end for**

16: **end for**

// Decoding Output:

17: **if** $\mathcal{F} == \emptyset$ **then**

18: **for** $1 \le i \le \ell$ **do**

19: $\boldsymbol{u}_i \leftarrow \mathcal{E}^{-1}_{\mathcal{C}_1}(\hat{\boldsymbol{c}}_i)$, and output $\boldsymbol{u}_i$.

20: **end for**

21: **else**

22: Output a decoding failure "e".

23: **end if**

Claim 6.3.4. *For a ladder code $\mathcal{C}_L$, Algorithm 1 corrects any error-erasure pattern $\boldsymbol{e} = (\boldsymbol{e}_1, \ldots, \boldsymbol{e}_\ell, \boldsymbol{e}''_2, \ldots, \boldsymbol{e}''_m)$ that satisfies the two conditions in Theorem* 6.3.3.

Proof. The proof follows from the decoding procedure of Algorithm 1. At level 1, we obtain the correct syndromes $\hat{\boldsymbol{s}}_i^1 = \boldsymbol{0}$, for $1 \leq i \leq \ell$. Then, these syndromes are used in decoding for the received words $\boldsymbol{y}_i$, $1 \leq i \leq \ell$.

In the loop $\mu = 2$, since the error-erasure pattern $\boldsymbol{e}'_2$ satisfies condition 2) in Theorem 6.3.3, the vector $\mathbf{X}_2$ obtained in step 6 can be decoded successfully. Thus, we obtain the correct syndromes $\hat{\boldsymbol{s}}_i^2$, $1 \leq i \leq \ell$. Then, the syndromes $\hat{\boldsymbol{s}}_i^1$ and $\hat{\boldsymbol{s}}_i^2$, $1 \leq i \leq \ell$, will be used to help decode the received words $\boldsymbol{y}_i$, $1 \leq i \leq \ell$.

Similarly, for each loop $3 \leq \mu \leq m$, since the error-erasure pattern $\boldsymbol{e}'_\mu$ satisfies condition 2) in Theorem 6.3.3, we can obtain the correct syndromes $\hat{\boldsymbol{s}}_i^\mu$, $1 \leq i \leq \ell$.

Therefore, when the decoding runs until the last loop, i.e., $\mu = m$, we have obtained all the correct syndromes $\hat{\boldsymbol{s}}_i^1, \hat{\boldsymbol{s}}_i^2, \ldots, \hat{\boldsymbol{s}}_i^m$, $1 \leq i \leq \ell$. Since the error-erasure patterns $\boldsymbol{e}_i$, $1 \leq i \leq \ell$, satisfy condition 1) in Theorem 6.3.3, using all these correct syndromes for coset decoding, the error-erasure patterns $\boldsymbol{e}_i$, $1 \leq i \leq \ell$, can be corrected. Thus, in this last loop, the decoder is guaranteed to return the correct information vectors $\boldsymbol{u}_i$, $1 \leq i \leq \ell$. ■

Remark 6.3.3. The decoding procedure of Algorithm 1 that utilizes the multi-level shared redundancy successively from the lowest level to the highest level mimics climbing up a ladder, which suggests the name of ladder codes. □

6.3.4 More Explicit Results on Correctable Patterns

In the previous Section 6.3.2, Theorem 6.3.3 gives a very general result on correctable error-erasure patterns for ladder codes. Now, we present more explicit results on erasure patterns and error patterns separately.

The following notation will be used. For a length-n vector $\boldsymbol{x} \in (\mathbb{F}_q \cup \{?\})^n$, let $\mathcal{N}_\sigma(\boldsymbol{x})$ denote the number of erasures ? in $\boldsymbol{x}$, and let $\mathcal{N}_\tau(\boldsymbol{x})$ denote the number of nonzero elements that belong to $\mathbb{F}_q/\{0\}$ in $\boldsymbol{x}$.

a) *Correctable Erasure Patterns*

Let us consider the case when only erasures occur.

We first choose the following correctable sets for the component codes in Construction 1.

1) For $1 \leq i \leq m$, choose the correctable set for $\mathcal{C}_i$ as

$$\mathcal{T}(\mathcal{C}_i) = \big\{\boldsymbol{x} : \boldsymbol{x} \in \{0,?\}^n \text{ and } \mathcal{N}_\sigma(\boldsymbol{x}) \leq d_i - 1\big\}.$$

Based on $\mathcal{T}(\mathcal{C}_i)$, we have

$$\Delta(\mathcal{C}_i) = \{\boldsymbol{x} : \boldsymbol{x} \in \{0,?\}^n \text{ and } \mathcal{N}_\sigma(\boldsymbol{x}) \geq d_i\}.$$

2) For $2 \leq i \leq m$, choose the correctable set for $\mathcal{C}_i''$ as

$$\mathcal{T}(\mathcal{C}_i'') = \{\boldsymbol{x} : \boldsymbol{x} \in \{0,?\}^{n_i''} \text{ and } \mathcal{N}_\sigma(\boldsymbol{x}) \leq d_i'' - 1\}.$$

Based on $\mathcal{T}(\mathcal{C}_i'')$, we have

$$\Delta(\mathcal{C}_i'') = \{\boldsymbol{x} : \boldsymbol{x} \in \{0,?\}^{n_i''} \text{ and } \mathcal{N}_\sigma(\boldsymbol{x}) \geq d_i''\}.$$

3) For $2 \leq i \leq m$, choose the correctable set for $\mathcal{C}_i'$ as

$$\mathcal{T}(\mathcal{C}_i') = \{\boldsymbol{x} : \boldsymbol{x} \in \{0,?\}^{n_i'} \text{ and } \mathcal{N}_\sigma(\boldsymbol{x}) \leq \delta_i - 1\}.$$

By applying Theorem 6.3.3, we directly obtain the following explicit result on correctable erasure patterns.

Lemma 6.3.5. *An m-level ladder code $\mathcal{C}_L$ obtained from Construction 1 corrects any erasure pattern $\boldsymbol{e} = (\boldsymbol{e}_1, \ldots, \boldsymbol{e}_\ell, \boldsymbol{e}_2'', \ldots, \boldsymbol{e}_m'')$, $\boldsymbol{e} \in \{0,?\}^{n_L}$, that satisfies the following two conditions:*

1) for $1 \leq i \leq \ell$, $\mathcal{N}_\sigma(\boldsymbol{e}_i) \leq d_m - 1$;

2) for $2 \leq i \leq m$, $a_i^1 + a_i^2 \leq \delta_i - 1$, *where*

$$a_i^1 = |\{j : 1 \leq j \leq \ell, \mathcal{N}_\sigma(\boldsymbol{e}_j) \geq d_{i-1}\}|,$$
$$a_i^2 = |\{j : 1 \leq j \leq n_i' - \ell, \mathcal{N}_\sigma(\boldsymbol{e}_j^i) \geq d_i''\}|.$$

Recall that d_L^* is a lower bound on the minimum distance of $\mathcal{C}_L$. The following lemma shows that $\mathcal{C}_L$ corrects any $d_L^* - 1$ erasures.

Lemma 6.3.6. *An m-level ladder code $\mathcal{C}_L$ from Construction 1 corrects any erasure pattern of less than d_L^* erasures.*

Proof. We only need to show that any erasure pattern of $d_L^* - 1$ erasures satisfies the two conditions in Lemma 6.3.5, so it can be corrected. To see this, first let us assume that condition 1) in Lemma 6.3.5 is violated. Then, for some integer j, $1 \leq j \leq \ell$, we have $\mathcal{N}_\sigma(\boldsymbol{e}_j) \geq d_m \geq d_L^*$, which violates the assumption that there are only $d_L^* - 1$ erasures. Second, let us assume that condition 2) is violated; that is, for some integer i, $2 \leq i \leq m$, we have $a_i^1 + a_i^2 \geq \delta_i$. It means that $\sum_{j=1}^{\ell} \mathcal{N}_\sigma(\boldsymbol{e}_j) + \sum_{j=1}^{n_i'-\ell} \mathcal{N}_\sigma(\boldsymbol{e}_j^i) \geq \delta_i \min\{d_{i-1}, d_i''\} \geq d_L^*$, which violates the assumption that there are only $d_L^* - 1$ erasures. ■

Let us give a simple example on correctable erasure patterns. It shows that ladder codes can correct more than $d_L^* - 1$ erasures in some cases.

Example 6.3.1. Consider the $[n_L = 8\ell + 4, k_L = 7\ell, d_L = 4]_2$ ladder code $\mathcal{C}_L$ constructed in Example 6.2.1. According to Lemma 6.3.6, any erasure pattern of 3 erasures can be corrected. In addition, using Lemma 6.3.5, it is easy to verify that some erasure patterns with more than 3 erasures can be corrected, but some are not. For instance, assume that the codeword $\boldsymbol{c}_L = (1\,0\,1\,0\,0\,0\,0\,0, 1\,1\,1\,1\,0\,0\,0\,0, 1\,1\,0\,0)$ is sent. The received word $\boldsymbol{y} = (1\,?\,1\,?\,0\,0\,0\,0, 1\,?\,1\,?\,0\,0\,0\,0, 1\,1\,0\,0)$ with 4 erasures cannot be corrected. In contrast, the received word $\boldsymbol{y} = (?\,0\,?\,?\,0\,0\,0\,0, 1\,?\,1\,1\,0\,0\,0\,0, ?\,1\,0\,0)$ with 5 erasures can be decoded. □

b) *Correctable Error Patterns*

Now, let us consider the case when the transmitted codeword $\boldsymbol{c}_L \in \mathcal{C}_L$ only suffers from errors.

We choose the following correctable sets for the component codes in Construction 1.

1) For $1 \leq i \leq m$, choose the correctable set for $\mathcal{C}_i$ as

$$\mathcal{T}(\mathcal{C}_i) = \big\{\boldsymbol{x} : \boldsymbol{x} \in \mathbb{F}_q^n \text{ and } \mathcal{N}_\tau(\boldsymbol{x}) \leq \rho_i\big\},$$

where ρ_i is an integer that satisfies $0 \leq \rho_i \leq \lfloor \frac{d_i - 1}{2} \rfloor$. The value of ρ_i can be interpreted as the decoding radius of the bounded-distance decoding. Based on $\mathcal{T}(\mathcal{C}_i)$, we can express the detectable but uncorrectable set as

$$\Delta(\mathcal{C}_i) = \left\{\boldsymbol{x} : \boldsymbol{x} \in \mathbb{F}_q^n \text{ and } \boldsymbol{x} \notin \bigcup_{\boldsymbol{y} \in \mathcal{T}(\mathcal{C}_i)} \{\mathcal{C}_i + \boldsymbol{y}\}\right\}.$$

2) For $2 \leq i \leq m$, choose the correctable set for $\mathcal{C}_i''$ as

$$\mathcal{T}(\mathcal{C}_i'') = \{\boldsymbol{x} : \boldsymbol{x} \in \mathbb{F}_q^{n_i''} \text{ and } \mathcal{N}_\tau(\boldsymbol{x}) \leq \rho_i''\},$$

where ρ_i'' is an integer satisfying $0 \leq \rho_i'' \leq \lfloor \frac{d_i'' - 1}{2} \rfloor$. Based on $\mathcal{T}(\mathcal{C}_i'')$, we have

$$\Delta(\mathcal{C}_i'') = \left\{\boldsymbol{x} : \boldsymbol{x} \in \mathbb{F}_q^{n_i''} \text{ and } \boldsymbol{x} \notin \bigcup_{\boldsymbol{y} \in \mathcal{T}(\mathcal{C}_i'')} \{\mathcal{C}_i'' + \boldsymbol{y}\}\right\}.$$

3) For $2 \leq i \leq m$, choose the correctable set for $\mathcal{C}_i'$ as

$$\mathcal{T}(\mathcal{C}_i') = \{\boldsymbol{x} : \ \boldsymbol{x} \in (\mathbb{F}_{q^{v_i}} \cup \{?\})^{n_i'} \text{ and } 2\mathcal{N}_\tau(\boldsymbol{x}) + \mathcal{N}_\sigma(\boldsymbol{x}) \leq \delta_i - 1\}.$$

Now, using Theorem 6.3.3, it is not hard to obtain the following lemma.

Lemma 6.3.7. *An m-level ladder code $\mathcal{C}_L$ obtained from Construction 1 corrects any error pattern $\boldsymbol{e} = (\boldsymbol{e}_1, \ldots, \boldsymbol{e}_\ell, \boldsymbol{e}_2'', \ldots, \boldsymbol{e}_m'')$, $\boldsymbol{e} \in \mathbb{F}_q^{n_L}$, that satisfies the following two conditions:*

1) for $1 \leq i \leq \ell$, $\mathcal{N}_\tau(\boldsymbol{e}_i) \leq \rho_m$;

2) for $2 \leq i \leq m$, $2\big(a_i^1(\rho_{i-1}) + a_i^2(\rho_i'')\big) + \big(b_i^1(\rho_{i-1}) + b_i^2(\rho_i'')\big) \leq \delta_i - 1$, where

$$a_i^1(\rho_{i-1}) = |\{j : 1 \leq j \leq \ell, \boldsymbol{e}_j \notin \mathcal{T}(\mathcal{C}_{i-1}) \cup \Delta(\mathcal{C}_{i-1})\}|,$$

$$b_i^1(\rho_{i-1}) = |\{j : 1 \leq j \leq \ell, \boldsymbol{e}_j \in \Delta(\mathcal{C}_{i-1})\}|,$$

$$a_i^2(\rho_i'') = |\{j : 1 \leq j \leq n_i' - \ell, \boldsymbol{e}_j^i \notin \mathcal{T}(\mathcal{C}_i'') \cup \Delta(\mathcal{C}_i'')\}|,$$

$$b_i^2(\rho_i'') = |\{j : 1 \leq j \leq n_i' - \ell, \boldsymbol{e}_j^i \in \Delta(\mathcal{C}_i'')\}|.$$

From Lemma 6.3.7, it is clear that the error correcting capability of $\mathcal{C}_L$ depends on the choices of the integers ρ_i and ρ_i''. Note that in Lemma 6.3.7, we use the notation such as $a_i^1(\rho_{i-1})$ to explicitly indicate that $a_i^1(\rho_{i-1})$ depends on ρ_{i-1}.

Based on Lemma 6.3.7, we have the following theorem.

Theorem 6.3.8. *For any length-n_L error pattern $\boldsymbol{e}$ whose Hamming weight is less than $d_L^*/2$, there exist ρ_i, $1 \le i \le m$, and ρ_j'', $2 \le j \le m$, such that the two conditions in Lemma 6.3.7 are satisfied.*

Proof. See Section 6.6 Appendix A. ■

Remark 6.3.4. Theorem 6.3.8 indicates that any received word $\boldsymbol{y}$ with number of errors less than $d_L^*/2$ can be corrected. □

6.4 Two-Level Ladder Codes versus Concatenated Codes

In this section, we study the similarity and difference between two-level ladder codes and concatenated codes [25, 64]. It will be shown that compared to a concatenated code, a two-level ladder code can achieve a higher rate for a given minimum distance.

In the following, we denote a two-level (i.e., $m = 2$) ladder code by $\mathcal{C}_L^2$. We also assume that $\mathcal{C}_i''$, $2 \le i \le m$, has minimum distance $d_i'' = d_{i-1}$.

The following corollary on the code parameters of $\mathcal{C}_L^2$ is directly concluded from Corollary 6.2.2.

Corollary 6.4.1. *A two-level ladder code $\mathcal{C}_L^2$ is a linear code over $\mathbb{F}_q$ of length $n_L = n\ell + n_2''(n_2' - \ell)$, dimension $k_L = k_1\ell$, and minimum distance $d_L \ge \min\{\delta_2 d_1, d_2\}$.*

A concatenated code $\mathcal{C}_{cont}$ over $\mathbb{F}_q$ is formed from an inner code $\mathcal{C}_{in}$ and an outer code $\mathcal{C}_{out}$ [25,64]. Here, let the inner code be an $[n,k,d]_q$ code and the outer code be an $[N,K,D = N-K+1]_{q^k}$ MDS code, which exists whenever $N \le q^k$. Thus, the corresponding $\mathcal{C}_{cont}$ is an $[nN,kK,\ge dD]_q$ code.

First, we show that a two-level ladder code $\mathcal{C}_L^2$ can have the same code parameters as those of a corresponding concatenated code $\mathcal{C}_{cont}$. To this end, we choose the following component codes in Construction 1 for constructing $\mathcal{C}_L^2$.

Design I:

1) Let $\mathcal{C}_1$ be an $[n, k_1 = k, d_1 = d]_q$ code, and $\mathcal{C}_2$ be the $[n, 0, \infty]_q$ code with only an all-zero codeword.

2) Let $\mathcal{C}_2'$ be an $[n_2' = N, \ell = K, \delta_2 = D]_{q^k}$ MDS code, where $N \leq q^k$.

3) Let $\mathcal{C}_2''$ be an $[n_2'' = n, k_2'' = k, d_2'' = d]_q$ code.

Lemma 6.4.2. *From Construction 1 with Design I, the corresponding two-level ladder code $\mathcal{C}_L^2$ over $\mathbb{F}_q$ has code length $n_L = nN$, dimension $k_L = kK$, and minimum distance $d_L \geq dD$.*

Proof. From Design I and Corollary 6.4.1, the code length, dimension, and minimum distance are obtained. ■

Second, for some cases, a two-level ladder code $\mathcal{C}_L^2$ can even outperform a concatenated code $\mathcal{C}_{cont}$ in the sense of possessing a higher rate but the same minimum distance. To see this, in Construction 1, we choose the following component codes to construct $\mathcal{C}_L^2$.

Design II:

1) Let $\mathcal{C}_1$ be an $[n, k_1 = k, d_1 = d]_q$ code, and $\mathcal{C}_2 \subset \mathcal{C}_1$ be an $[n, k_2, d_2 = dD]_q$ code (here, we assume that $\mathcal{C}_2$ exists with positive dimension $k_2 > 0$ and finite minimum distance $d_2 = dD < \infty$).

2) Let $\mathcal{C}_2'$ be an $[n_2' = N, \ell = K, \delta_2 = D]_{q^{k-k_2}}$ MDS code, which exists whenever $N \leq q^{k-k_2}$.

3) Let $\mathcal{C}_2''$ be an $[n_2'' \leq n, k_2'' = k - k_2, d_2'' = d]_q$ code. Note that here we can choose $n_2'' \leq n$, since an $[n, k, d]_q$ inner code $\mathcal{C}_{in}$ of $\mathcal{C}_{cont}$ exists and $k_2'' < k$.

Lemma 6.4.3. *From Construction 1 with Design II, the corresponding two-level ladder code $\mathcal{C}_L^2$ over $\mathbb{F}_q$ has code length $n_L = nK + (N - K)n_2''$, dimension $k_L = kK$, and minimum distance $d_L = dD$.*

Proof. The code parameters are obtained directly from Design II, Corollary 6.2.2, and Corollary 6.4.1. ■

From Lemma 6.4.3, the rate of $\mathcal{C}_L^2$ is $R_L = \frac{kK}{nK+(N-K)n_2''}$. Denote the rate of $\mathcal{C}_{cont}$ by $R_{cont} = \frac{kK}{nN}$. Since $n_2'' \leq n$, we have $R_L \geq R_{cont}$, where the inequality is strict for many cases, one simple example of which is as follows.

Example 6.4.1. Consider a concatenated code $\mathcal{C}_{cont}$ with an $[n = 8, k = 7, d = 2]_2$ inner code $\mathcal{C}_{in}$ and an $[N = \ell + 1, K = \ell, D = 2]_{2^7}$ outer code $\mathcal{C}_{out}$. Thus, $\mathcal{C}_{cont}$ is an $[8(\ell + 1), 7\ell, 4]_2$ code with rate $R_{cont} = \frac{7\ell}{8(\ell+1)}$. For comparison, we choose the corresponding ladder code $\mathcal{C}_L^2$ from Design II as the code

constructed in Example 6.2.1. It is an $[n_L = 8\ell + 4, k_L = 7\ell, d_L = 4]_2$ code with rate $R_L = \frac{7\ell}{8\ell+4}$. Thus, in this case, $\mathcal{C}_L^2$ has a higher rate than that of $\mathcal{C}_{cont}$, while their minimum distances are the same. □

In addition, we briefly compare two-level ladder codes with two-level generalized tensor product codes [40, 88] by using the following example.

Example 6.4.2. Let $\mathcal{C}_1$ be the $[16, 15, 2]_2$ single parity code and $\mathcal{C}_2 \subset \mathcal{C}_1$ be the $[16, 11, 4]_2$ extended Hamming code. Choose $\mathcal{C}_2'$ to be the $[\ell + 1, \ell, 2]_{2^4}$ single parity code and $\mathcal{C}_2''$ to be the $[5, 4, 2]_2$ single parity code. From Construction 1, the resulting two-level ladder code $\mathcal{C}_L^2$ is an $[n_L = 16\ell + 5, k_L = 15\ell, d_L = 4]_2$ code.

In the construction of generalized tensor product codes, we use the same component codes $\mathcal{C}_1$ and $\mathcal{C}_2$. From [40, 88], we can construct a two-level generalized tensor product code $\mathcal{C}_{GTP}$ with parameters $[16\ell + 16, 15\ell + 11, 4]_2$. By shortening $\mathcal{C}_{GTP}$ by 11 information symbols, we obtain a $[16\ell + 5, 15\ell, 4]_2$ code, which has the same code parameters as the above ladder code $\mathcal{C}_L^2$. □

6.5 Conclusion

In this chapter, we proposed a new family of shared-redundancy codes, called ladder codes, and studied their basic code properties. We derived the code length, dimension, and a lower bound d_L^* on the minimum distance. Then, we analyzed correctable error-erasure patterns and presented a corresponding decoding algorithm. Finally, it was also shown that in some cases a two-level ladder code can have a higher rate but the same minimum distance, compared to a corresponding concatenated code.

6.6 Appendix A

In this section, we give the proof of Theorem 6.3.8.

Proof. First, let us choose $\rho_m = \lfloor \frac{d_m - 1}{2} \rfloor$, then condition 1) in Lemma 6.3.7 is satisfied, since $\mathcal{N}_\tau(\boldsymbol{e}_i) \leq \mathcal{N}_\tau(\boldsymbol{e}) \leq \lfloor \frac{d_L^* - 1}{2} \rfloor \leq \lfloor \frac{d_m - 1}{2} \rfloor = \rho_m$.

Now, we prove that condition 2) in Lemma 6.3.7 can be satisfied. For notation simplicity yet without loss of generality, we give a proof for the μth level; that is, for $i = \mu$, we will prove that there exist

$\rho_{\mu-1}$ and ρ''_μ such that $2\big(a^1_\mu(\rho_{\mu-1}) + a^2_\mu(\rho''_\mu)\big) + \big(b^1_\mu(\rho_{\mu-1}) + b^2_\mu(\rho''_\mu)\big) \le \delta_\mu - 1$. For other values of i, the proof is similar, so it is omitted.

More specifically, let $\rho_{\mu-1} = \rho''_\mu = \theta$, we will show that there exists a threshold value $\theta \in \{0,1,\ldots,\lfloor\frac{\tilde{d}_{\mu-1}-1}{2}\rfloor\}$ where $\tilde{d}_{\mu-1} = \min\{d_{\mu-1}, d''_\mu\}$ such that

$$2\Big(a^1_\mu(\theta) + a^2_\mu(\theta)\Big) + \Big(b^1_\mu(\theta) + b^2_\mu(\theta)\Big) \le \delta_\mu - 1.$$

The existence of such a threshold θ will be established by following the similar proof for the generalized minimum distance (GMD) decoder [25, 64]; that is, to prove the average of $2\sum_{j=1}^2 a^j_\mu(\theta) + \sum_{j=1}^2 b^j_\mu(\theta)$, when θ ranges over $\{0,1,\ldots,\lfloor\frac{\tilde{d}_{\mu-1}-1}{2}\rfloor\}$ with respect to a certain probability measure, is less than δ_μ.

In the proof, we will use the following notation. First, recall that $\boldsymbol{y} = \boldsymbol{c}_L + \boldsymbol{e}$, $\boldsymbol{e} \in \mathbb{F}_q^{n_L}$, where $\boldsymbol{y} = (\boldsymbol{y}_1,\ldots,\boldsymbol{y}_\ell,\boldsymbol{y}''_2,\ldots,\boldsymbol{y}''_m)$, $\boldsymbol{c}_L = (\boldsymbol{c}_1,\ldots,\boldsymbol{c}_\ell,\boldsymbol{r}_2,\ldots,\boldsymbol{r}_m)$, and $\boldsymbol{e} = (\boldsymbol{e}_1,\ldots,\boldsymbol{e}_\ell,\boldsymbol{e}''_2,\ldots,\boldsymbol{e}''_m)$.

- For $j = 1,2,\ldots,\ell$, let $\overline{\mathcal{C}}_{\mu-1}(j)$ denote a coset of the code $\mathcal{C}_{\mu-1}$ such that $\overline{\mathcal{C}}_{\mu-1}(j) = \mathcal{C}_{\mu-1} + \boldsymbol{c}_j$. Let $\hat{\boldsymbol{c}}_j$ be the nearest codeword to the received vector $\boldsymbol{y}_j$ in the coset $\overline{\mathcal{C}}_{\mu-1}(j)$, i.e., $\mathrm{d}_q(\boldsymbol{y}_j,\hat{\boldsymbol{c}}_j) \le \mathrm{d}_q(\boldsymbol{y}_j,\boldsymbol{x})$ for all $\boldsymbol{x} \in \overline{\mathcal{C}}_{\mu-1}(j)$. Let $\eta_j = \mathrm{d}_q(\boldsymbol{y}_j,\hat{\boldsymbol{c}}_j)$, and

$$\pi_j(\theta) = \begin{cases} 0 & \text{if } \hat{\boldsymbol{c}}_j = \boldsymbol{c}_j \text{ and } \eta_j \le \theta \\ 2 & \text{if } \hat{\boldsymbol{c}}_j \ne \boldsymbol{c}_j \text{ and } \eta_j \le \theta \\ 1 & \text{if } \eta_j > \theta \end{cases}.$$

- For $j = \ell+1, \ell+2,\ldots,n'_\mu$, let $\hat{\boldsymbol{g}}^\mu_{j-\ell}$ be the nearest codeword to the received vector $\boldsymbol{y}^\mu_{j-\ell}$ in the code $\mathcal{C}''_\mu$, i.e., $\mathrm{d}_q(\boldsymbol{y}^\mu_{j-\ell},\hat{\boldsymbol{g}}^\mu_{j-\ell}) \le \mathrm{d}_q(\boldsymbol{y}^\mu_{j-\ell},\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathcal{C}''_\mu$. Let $\eta_j = \mathrm{d}_q(\boldsymbol{y}^\mu_{j-\ell},\hat{\boldsymbol{g}}^\mu_{j-\ell})$, and

$$\pi_j(\theta) = \begin{cases} 0 & \text{if } \hat{\boldsymbol{g}}^\mu_{j-\ell} = \boldsymbol{g}^\mu_{j-\ell} \text{ and } \eta_j \le \theta \\ 2 & \text{if } \hat{\boldsymbol{g}}^\mu_{j-\ell} \ne \boldsymbol{g}^\mu_{j-\ell} \text{ and } \eta_j \le \theta \\ 1 & \text{if } \eta_j > \theta \end{cases}.$$

It is not hard to verify that

$$2\sum_{j=1}^{2} a^j_\mu(\theta) + \sum_{j=1}^{2} b^j_\mu(\theta) = \sum_{j=1}^{n'_\mu} \pi_j(\theta).$$

We treat θ as a random variable which takes values in the set $\{0,1,\ldots,\lfloor\frac{\tilde{d}_{\mu-1}-1}{2}\rfloor\}$ and assign the following probability over $\{0,1,\ldots,\lfloor\frac{\tilde{d}_{\mu-1}-1}{2}\rfloor\}$:

$$\mathrm{P}_\theta\{\theta=x\}=\begin{cases}\frac{2}{\tilde{d}_{\mu-1}} & \text{if } x\in\{0,1,\ldots,\lfloor\frac{\tilde{d}_{\mu-1}}{2}\rfloor-1\}\\ \frac{1}{\tilde{d}_{\mu-1}} & \text{if } \tilde{d}_{\mu-1} \text{ is odd and } x=\lfloor\frac{\tilde{d}_{\mu-1}-1}{2}\rfloor\end{cases}.$$

We use $\mathbb{E}_\theta\{\cdot\}$ for the expected value with respect to the probability P_θ.

To show there is a threshold value $\theta\in\{0,\ldots,\lfloor\frac{\tilde{d}_{\mu-1}-1}{2}\rfloor\}$ such that $2\sum_{j=1}^{2}a_\mu^j(\theta)+\sum_{j=1}^{2}b_\mu^j(\theta)<\delta_\mu$ is equivalent to showing that $\mathbb{E}_\theta\{2\sum_{j=1}^{2}a_\mu^j(\theta)+\sum_{j=1}^{2}b_\mu^j(\theta)\}<\delta_\mu$. Thus, we only need to prove

$$\mathbb{E}_\theta\{\sum_{j=1}^{n'_\mu}\pi_j(\theta)\}<\delta_\mu. \tag{6.3}$$

To this end, we need the following two lemmas.

Lemma 6.6.1. *For every $j\in\{1,\ldots,\ell\}$,*

$$\mathbb{E}_\theta\{\pi_j(\theta)\}\le\frac{2\mathrm{d}_q(\boldsymbol{y}_j,\boldsymbol{c}_j)}{\tilde{d}_{\mu-1}}. \tag{6.4}$$

Proof. For every $j\in\{1,\ldots,\ell\}$, to prove inequality (6.4), we consider the following two cases:

Case I: $\hat{\boldsymbol{c}}_j=\boldsymbol{c}_j$.

$$\begin{aligned}\mathbb{E}_\theta\{\pi_j(\theta)\}&=0\mathrm{P}(\theta\ge\eta_j)+1\mathrm{P}(\theta<\eta_j)\\&=\mathrm{P}(\theta<\eta_j)\\&\le\frac{2}{\tilde{d}_{\mu-1}}\eta_j=\frac{2}{\tilde{d}_{\mu-1}}\mathrm{d}_q(\boldsymbol{y}_j,\hat{\boldsymbol{c}}_j)=\frac{2}{\tilde{d}_{\mu-1}}\mathrm{d}_q(\boldsymbol{y}_j,\boldsymbol{c}_j).\end{aligned}$$

Case II: $\hat{\boldsymbol{c}}_j\neq\boldsymbol{c}_j$.

$$\mathbb{E}_\theta\{\pi_j(\theta)\}=2\mathrm{P}(\theta\ge\eta_j)+1\mathrm{P}(\theta<\eta_j).$$

(i) If $\eta_j > \lfloor \frac{\tilde{d}_{\mu-1}-1}{2} \rfloor$, then since $\theta \in \{0,\ldots,\lfloor \frac{\tilde{d}_{\mu-1}-1}{2} \rfloor\}$, we have

$$\begin{aligned}\mathbb{E}_\theta\{\pi_j(\theta)\} &= 2\mathrm{P}(\theta \geq \eta_j) + 1\mathrm{P}(\theta < \eta_j) \\ &= \mathrm{P}(\theta < \eta_j) \\ &\leq \frac{2}{\tilde{d}_{\mu-1}}\eta_j = \frac{2}{\tilde{d}_{\mu-1}}\mathrm{d}_q(\boldsymbol{y}_j, \hat{\boldsymbol{c}}_j) \leq \frac{2}{\tilde{d}_{\mu-1}}\mathrm{d}_q(\boldsymbol{y}_j, \boldsymbol{c}_j),\end{aligned}$$

where the last step follows from the assumption that $\hat{\boldsymbol{c}}_j$ is the nearest codeword to the received word $\boldsymbol{y}_j$.

(ii) If $\eta_j \leq \lfloor \frac{\tilde{d}_{\mu-1}-1}{2} \rfloor$, then since $\theta \in \{0,\ldots,\lfloor \frac{\tilde{d}_{\mu-1}-1}{2} \rfloor\}$, we have

$$\begin{aligned}\mathbb{E}_\theta\{\pi_j(\theta)\} &= 2\mathrm{P}(\theta \geq \eta_j) + 1\mathrm{P}(\theta < \eta_j) \\ &= 2\big(1 - \mathrm{P}(\theta < \eta_j)\big) + \mathrm{P}(\theta < \eta_j) \\ &= 2 - \mathrm{P}(\theta < \eta_j) = 2 - \frac{2}{\tilde{d}_{\mu-1}}\eta_j \\ &= 2\frac{\tilde{d}_{\mu-1} - \mathrm{d}_q(\boldsymbol{y}_j, \hat{\boldsymbol{c}}_j)}{\tilde{d}_{\mu-1}} \leq 2\frac{\mathrm{d}_q(\boldsymbol{y}_j, \boldsymbol{c}_j)}{\tilde{d}_{\mu-1}},\end{aligned}$$

where the last step follows from the triangle inequality: $\tilde{d}_{\mu-1} \leq d_{\mu-1} \leq \mathrm{d}_q(\hat{\boldsymbol{c}}_j, \boldsymbol{c}_j) \leq \mathrm{d}_q(\boldsymbol{y}_j, \hat{\boldsymbol{c}}_j) + \mathrm{d}_q(\boldsymbol{y}_j, \boldsymbol{c}_j)$. ■

Lemma 6.6.2. *For every $j \in \{\ell+1,\ldots,n'_\mu\}$,*

$$\mathbb{E}_\theta\{\pi_j(\theta)\} \leq \frac{2\mathrm{d}_q(\boldsymbol{y}^\mu_{j-\ell}, \boldsymbol{g}^\mu_{j-\ell})}{\tilde{d}_{\mu-1}}. \tag{6.5}$$

Proof. The proof is similar to that of Lemma 6.6.1, so it is omitted. ■

Now, we prove inequality (6.3) from Lemma 6.6.1 and Lemma 6.6.2 as follows.

$$\begin{aligned}\mathbb{E}_\theta\{\sum_{j=1}^{n'_\mu} \pi_j(\theta)\} &= \sum_{j=1}^{n'_\mu} \mathbb{E}_\theta\{\pi_j(\theta)\} \\ &\leq \frac{2\sum_{j=1}^{\ell} \mathrm{d}_q(\boldsymbol{y}_j, \boldsymbol{c}_j) + 2\sum_{j=1}^{n'_\mu-\ell} \mathrm{d}_q(\boldsymbol{y}^\mu_j, \boldsymbol{g}^\mu_j)}{\tilde{d}_{\mu-1}} \\ &\leq \frac{2\mathrm{d}_q(\boldsymbol{y}, \boldsymbol{c}_L)}{\tilde{d}_{\mu-1}} \overset{(a)}{<} \frac{d^*_L}{\tilde{d}_{\mu-1}} \leq \delta_\mu,\end{aligned}$$

where step (a) follows from the assumption $\mathrm{d}_q(\boldsymbol{y}, \boldsymbol{c}_L) < d_L^*/2$.

■

Acknowledgement

This chapter is in part a reprint of the material in the paper: Pengfei Huang, Eitan Yaakobi, and Paul H. Siegel, "Ladder codes: A class of error-correcting codes with multi-level shared redundancy," to appear in *Proc. IEEE International Conference on Communications (ICC)*, Kansas City, MO, USA, May 2018. The dissertation author was the primary investigator and author of this paper.

Chapter 7

Performance of Multilevel Flash Memories with Different Binary Labelings

7.1 Introduction

The channel characterization of flash memories is important for understanding fundamental limits on storage density, as well as for designing effective signal processing algorithms and error-correcting codes (ECCs) [5, 21, 22]. Many experiments have shown that the distribution of the readback signal (i.e., the voltage level of a cell) in flash memories is asymmetric [16, 18, 52]. In [56], a mixed normal-Laplace distribution model was proposed and shown to accurately capture this asymmetry.

Several papers have recently studied the capacity of multilevel flash memory using a variety of channel models [46, 69, 79, 84, 85]. For example, in [85], the capacity of multi-level cell (MLC) flash memory was analyzed by modeling MLC flash memory as a 4-ary input point-to-point channel with additive white Gaussian noise. The performance improvement provided by soft information obtained with multiple read thresholds was also evaluated. In a similar vein, the capacity of three-level cell (TLC) flash memory was recently studied in [69] by considering TLC flash memory as an 8-ary input point-to-point channel with asymmetric mixed normal-Laplace noise. In [79], empirical error measurements at the cell and bit levels were used to estimate the capacity of an MLC flash memory as a function of program/erase

(P/E) cycle count. Results obtained using 4-ary discrete memoryless channel (DMC) model were compared to those based upon a union of two independent binary symmetric channel (BSC) models corresponding to the lower page and upper page.

A common feature of these prior studies of flash memory capacity is the use of point-to-point channel models. In this chapter, we take a different approach, and model the flash memory as a multi-user system, where the pages correspond to independently encoded users of a shared multiple-access channel. To the best of our knowledge, this is the first time that flash memories have been examined from this perspective.

Our goal is to study the fundamental performance limits of page-oriented multilevel flash memories using various decoding schemes. Specifically, we consider both low-complexity Treating Interference as Noise (TIN) decoding and relatively high-complexity Successive Cancellation (SC) decoding for the MLC case. We first examine a general discrete memoryless multiple-access channel model with two binary inputs and a four-level output. We derive elementary conditions such that the sum rate of TIN decoding equals that of SC decoding. Then, we determine achievable rate regions and sum rates of both decoding schemes for several specific flash memory channel models, represented by channel transition matrices from cell voltage levels to quantized readback outputs. The effect of different binary labelings of the cell levels is also studied, and the optimal labelings for each decoding scheme and channel model are identified. It is shown that TIN and SC decodings both outperform Default Setting (DS) decoding, a model of current flash memory technology, which uses Gray labeling of cell levels, along with separate quantization and decoding of each page. We also study the impact of further quantization of the memory output (i.e., additional read thresholds), and the resulting effect on performance is evaluated by means of computer simulation. Although the focus of this chapter is on information-theoretic analysis, some of the results provide qualitative insight into effective coding solutions.

The remainder of the chapter is organized as follows. In Section 7.2, we introduce the discrete memoryless multiple-access channel model for multilevel flash memories. We then review information-theoretic characterizations of the uniform rate regions and sum rates for the three decoding schemes, namely, TIN, SC, and DS. We also derive some elementary but useful characterizations of those channels for which TIN and SC decoders yield the same rate regions, as well as an elementary general bound on the

difference between the respective sum rates of TIN and SC decoders. In Section 7.3, we analyze properties and relationships among the rate regions for various MLC flash memory channel models. We also study the effect of using different binary labelings of cell voltage levels. In Section 7.4, we investigate the impact of employing additional read thresholds to obtain refined soft information. We conclude the chapter in Section 7.5.

Throughout this chapter, we follow the notation in [23]. Random variables are denoted with upper case letters (e.g., X) and their realizations with lower case (e.g., x). Calligraphic letters (e.g., $\mathcal{X}$) are used for finite sets. $\mathbb{R}$ is the set of real numbers. The discrete interval $[i:j]$ is defined as the set $\{i, i+1, \ldots, j\}$. For a length-n vector v, $v(i)$ represents the value of its ith coordinate, $i = 1, 2, \ldots, n$. We use the notation $p(x)$ to abbreviate the probability $P(X = x)$, and likewise for conditional and joint probabilities of both scalar and vector random variables, e.g., $p(y|x) = P(Y = y|X = x)$. For a probability vector $(\frac{p_1}{p_s}, \frac{p_2}{p_s}, \ldots, \frac{p_n}{p_s})$ where $p_s = \sum_{i=1}^{n} p_i$ and $p_i \geq 0$, $i = 1, 2, \ldots, n$, the entropy function is defined by $H(\frac{p_1}{p_s}, \frac{p_2}{p_s}, \ldots, \frac{p_n}{p_s}) = -\sum_{i=1}^{n} \frac{p_i}{p_s} \log_2 \frac{p_i}{p_s}$. We will also use the function $f(x) = x \log_2 x$. Therefore, we can express the entropy function by $H(\frac{p_1}{p_s}, \frac{p_2}{p_s}, \ldots, \frac{p_n}{p_s}) = -\frac{1}{p_s} \sum_{i=1}^{n} f(p_i) + \log_2 p_s$.

7.2 Multiple-Access Channel Model for Flash Memories

In this section, we introduce the multiple-access channel model for multilevel flash memories, define the decoding schemes to be considered, and present some of their basic information-theoretic properties.

7.2.1 System Model

We model a multilevel flash memory as a k-user multiple-access channel with k independent inputs $X_1, \ldots, X_k$, and one output Y ($k = 2$ for MLC flash, and $k = 3$ for TLC flash).

Specifically, the readback signal $\tilde{Y} \in \mathbb{R}$ in a flash memory is expressed as

$$\tilde{Y} = \sigma(X_1, \ldots, X_k) + Z, \tag{7.1}$$

where $X_1, \ldots, X_k \in \{0, 1\}$ represent data from k independent pages, $Z \in \mathbb{R}$ stands for the asymmetric

noise (see [56] for more details on the normal-Laplace distribution model), and σ maps an input $(x_1,\ldots,x_k)$ to a voltage level v. More specifically, σ is a bijective mapping from the set $\mathcal{T}$ which consists of all length-k binary strings to the set $\mathcal{V}$ which consists of 2^k voltage level values. For $k=2$ (MLC flash), $\mathcal{T}_{MLC}=\{11,10,01,00\}$ and $\mathcal{V}_{MLC}=\{A_0,A_1,A_2,A_3\}$; see Figure 1.1. By a slight abuse of notation, we write the mapping σ as a vector $\sigma=(w_0,w_1,w_2,w_3)$ (where w_i, $i=0,1,2,3$, represent the full set of possible 2-tuples) to represent the mapping $\sigma(w_i)=A_i$ for $i=0,1,2,3$. For example, the vector $\sigma=(11,10,00,01)$ corresponds to $\sigma(11)=A_0$, $\sigma(10)=A_1$, $\sigma(00)=A_2$, and $\sigma(01)=A_3$. Similarly, for $k=3$ (TLC flash), $\mathcal{T}_{TLC}=\{111,110,101,100,011,010,001,000\}$ and $\mathcal{V}_{TLC}=\{B_0,B_1,\ldots,B_7\}$; see Figure 1.2. We write $\sigma=(w_0,w_1,\ldots,w_7)$ (where w_i, $i=0,1,\ldots,7$, represent the full set of possible 3-tuples) to represent the mapping $\sigma(w_i)=B_i$ for $i=0,1,\ldots,7$. We will refer to a *mapping* σ as a *labeling*.

During the readback process, a quantizer Q is used to quantize $\tilde{Y}$ to obtain an output Y, i.e., $Y=Q(\tilde{Y})$, where the function $Q(\cdot)$ is a mapping from $\mathbb{R}$ to a finite alphabet set $\mathcal{Y}=\{s_0,s_1,\ldots,s_{q-1}\}$ of cardinality q. Usually $q=2^k$, but this is not necessary. The cardinality q can correspond to a large number by applying multiple reads. From an information-theoretic point of view, this means that more soft information is obtained for decoding.

7.2.2 Decoding Schemes for MLC Flash Memories

In this subsection, we investigate three decoding schemes for MLC flash memories.

Given a labeling σ and a quantizer Q, the MLC flash memory channel can be modeled as a 2-user discrete memoryless multiple-access channel $\mathcal{W}_{MLC}$: $(\mathcal{X}\times\mathcal{X},p(y|x_1,x_2),\mathcal{Y})$, where $\mathcal{X}=\{0,1\}$, $\mathcal{Y}=\{s_0,s_1,\ldots,s_{q-1}\}$, and $p(y|x_1,x_2)$ is the transition probability for any $x_1,x_2\in\mathcal{X}$ and $y\in\mathcal{Y}$. For simplicity, denote the conditional probabilities by $p_{BD(x_1,x_2),y}\stackrel{\text{def}}{=}P(Y=y|X_1=x_1,X_2=x_2)$, where $BD(\cdot)$ is a function that converts a binary string into its decimal value; e.g., $p_{2,s_0}=P(Y=s_0|X_1=1,X_2=0)$.

Users $j=1,2$ independently encode their messages M_j into the corresponding length-n codewords x_j^n and send (write) them over the shared channel (i.e., a set of cells) to the receiver (reader). Following the notation in [23], we define a $(2^{nR_1},2^{nR_2},n)$ code by two encoders $x_1^n(m_1)$ and $x_2^n(m_2)$ for messages

m_1 and m_2 from message sets $[1:2^{nR_1}]$ and $[1:2^{nR_2}]$ respectively, and a decoder that assigns an estimate $(\hat{m}_1, \hat{m}_2)$ based on the received sequence y^n. We assume that the message pair (M_1, M_2) is uniform over $[1:2^{nR_1}] \times [1:2^{nR_2}]$. The average probability of error is defined as $\mathrm{P}_e^{(n)} = P\{(M_1, M_2) \neq (\hat{M}_1, \hat{M}_2)\}$. A rate pair (R_1, R_2) is said to be achievable if there exists a sequence of $(2^{nR_1}, 2^{nR_2}, n)$ codes such that $\lim_{n\to\infty} \mathrm{P}_e^{(n)} = 0$. The capacity region is the closure of the set of achievable rate pairs (R_1, R_2).

The capacity region of this multiple-access channel is fully characterized [19, 23]. However, to make the analysis simple and yet representative, we focus on the *uniform* rate region for different decoding schemes. This represents the achievable region corresponding to the case that the input distributions are uniform. For other input distributions, the analysis is similar.

For a channel $\mathcal{W}_{MLC}$, the **Treating Interference as Noise (TIN)** decoding scheme decodes X_1 and X_2 independently based on Y [19, 23]. Its uniform rate region $\mathscr{R}^{TIN}$ for lower page X_1 and upper page X_2 is the set of all pairs (R_1, R_2) such that $R_1 \leq I(X_1;Y)$ and $R_2 \leq I(X_2;Y)$. In $\mathscr{R}^{TIN}$, the sum rate is $r_s^{TIN} = \max\{R_1 + R_2 : (R_1, R_2) \in \mathscr{R}^{TIN}\} = I(X_1;Y) + I(X_2;Y)$; here and in the following, for the sake of brevity, we use the term "sum rate" to represent the maximum sum rate in the corresponding rate region.

For a channel $\mathcal{W}_{MLC}$, the **Successive Cancellation (SC)** decoding scheme decodes X_1 and X_2 in some order based on Y [19, 23]. Its uniform rate region $\mathscr{R}^{SC}$ for lower page X_1 and upper page X_2 is the set of all pairs (R_1, R_2) such that $R_1 \leq I(X_1;Y|X_2)$, $R_2 \leq I(X_2;Y|X_1)$, and $R_1 + R_2 \leq I(X_1, X_2;Y)$. In $\mathscr{R}^{SC}$, the sum rate is $r_s^{SC} = \max\{R_1 + R_2 : (R_1, R_2) \in \mathscr{R}^{SC}\} = I(X_1, X_2;Y)$.

Remark 7.2.1. For TIN decoding, X_1 and X_2 are decoded independently and can be implemented in parallel. However, for SC decoding, X_1 and X_2 are decoded in a certain order. In general, TIN decoding is preferred for its low decoding complexity, but there may be a cost in performance relative to SC decoding, as reflected in the uniform rate region containment $\mathscr{R}^{TIN} \subseteq \mathscr{R}^{SC}$ and the sum rate relationship $r_s^{TIN} \leq r_s^{SC}$. □

The following theorem identifies the channels for which the sum rates of TIN decoding and SC decoding are the same.

Theorem 7.2.1. *For a channel $\mathcal{W}_{MLC}$, the sum rates satisfy $r_s^{TIN} \leq r_s^{SC}$ with equality if and only if*

$p_{3,s_j}p_{0,s_j} = p_{2,s_j}p_{1,s_j}$ *for all* $j = 0, 1, \ldots, q-1$. *If* $r_s^{TIN} = r_s^{SC}$, *then* $\mathscr{R}^{TIN} = \mathscr{R}^{SC}$ *and the rate region is a rectangle.*

Proof. We bound the value $r_s^{SC} - r_s^{TIN}$ as follows

$$\begin{aligned} r_s^{SC} - r_s^{TIN} =& I(X_1, X_2; Y) - I(X_1; Y) - I(X_2; Y) \\ =& I(X_2; Y|X_1) - I(X_2; Y) \\ =& H(X_2|X_1) - H(X_2|X_1, Y) - \Big(H(X_2) - H(X_2|Y)\Big) \\ \overset{(a)}{=}& I(X_1; X_2|Y) \\ =& \sum_{j=0}^{q-1} (\sum_{i=0}^{3} \frac{p_{i,s_j}}{4}) I(X_1; X_2|Y = s_j) \geq 0, \end{aligned}$$

where in step (a) we use $H(X_2|X_1) = H(X_2)$ which follows from the fact that X_1 and X_2 are independent.

Now, $I(X_1; X_2|Y) \geq 0$ with equality if and only if X_1 and X_2 are conditionally independent given $Y = s_j$, i.e., $P(X_1, X_2|Y = s_j) = P(X_1|Y = s_j)P(X_2|Y = s_j)$. Thus, we need to check four cases:

1) $P(X_1 = 1, X_2 = 1|Y = s_j) = P(X_1 = 1|Y = s_j)P(X_2 = 1|Y = s_j)$, i.e., $\frac{p_{3,s_j}}{\sum_{i=0}^{3} p_{i,s_j}} = \frac{p_{3,s_j}+p_{2,s_j}}{\sum_{i=0}^{3} p_{i,s_j}} \frac{p_{3,s_j}+p_{1,s_j}}{\sum_{i=0}^{3} p_{i,s_j}}$.

2) $P(X_1 = 1, X_2 = 0|Y = s_j) = P(X_1 = 1|Y = s_j)P(X_2 = 0|Y = s_j)$, i.e., $\frac{p_{2,s_j}}{\sum_{i=0}^{3} p_{i,s_j}} = \frac{p_{3,s_j}+p_{2,s_j}}{\sum_{i=0}^{3} p_{i,s_j}} \frac{p_{2,s_j}+p_{0,s_j}}{\sum_{i=0}^{3} p_{i,s_j}}$.

3) $P(X_1 = 0, X_2 = 1|Y = s_j) = P(X_1 = 0|Y = s_j)P(X_2 = 1|Y = s_j)$, i.e., $\frac{p_{1,s_j}}{\sum_{i=0}^{3} p_{i,s_j}} = \frac{p_{1,s_j}+p_{0,s_j}}{\sum_{i=0}^{3} p_{i,s_j}} \frac{p_{3,s_j}+p_{1,s_j}}{\sum_{i=0}^{3} p_{i,s_j}}$.

4) $P(X_1 = 0, X_2 = 0|Y = s_j) = P(X_1 = 0|Y = s_j)P(X_2 = 0|Y = s_j)$, i.e., $\frac{p_{0,s_j}}{\sum_{i=0}^{3} p_{i,s_j}} = \frac{p_{1,s_j}+p_{0,s_j}}{\sum_{i=0}^{3} p_{i,s_j}} \frac{p_{2,s_j}+p_{0,s_j}}{\sum_{i=0}^{3} p_{i,s_j}}$.

To satisfy conditions 1) – 4), we have $p_{3,s_j}p_{0,s_j} = p_{2,s_j}p_{1,s_j}$ for all $j = 0, 1, \ldots, q-1$.

Finally, assuming $r_s^{TIN} = r_s^{SC}$, i.e., $I(X_1; Y) + I(X_2; Y) = I(X_1, X_2; Y)$, since $I(X_1, X_2; Y) = I(X_1; Y) + I(X_2; Y|X_1) = I(X_2; Y) + I(X_1; Y|X_2)$, we have $I(X_1; Y) = I(X_1; Y|X_2)$ and $I(X_2; Y) = I(X_2; Y|X_1)$, which means $\mathscr{R}^{TIN} = \mathscr{R}^{SC}$. ■

An upper bound on the difference between r_s^{SC} and r_s^{TIN} is given by the following theorem.

Theorem 7.2.2. *For a channel* $\mathcal{W}_{MLC}$, *the rate difference* $r_s^{SC} - r_s^{TIN} \leq 1$ *with equality if and only if* $p_{3,s_j} + p_{2,s_j} = p_{1,s_j} + p_{0,s_j}$ *and* $p_{3,s_j}p_{1,s_j} = p_{2,s_j}p_{0,s_j} = 0$ *for all* $j = 0, 1, \ldots, q-1$.

Proof. We bound the value $r_s^{SC} - r_s^{TIN}$ as

$$\begin{aligned} r_s^{SC} - r_s^{TIN} =& I(X_1, X_2; Y) - I(X_1; Y) - I(X_2; Y) \\ =& I(X_1; X_2|Y) = H(X_1|Y) - H(X_1|X_2, Y) \\ \overset{(a)}{\leq}& H(X_1) - H(X_1|X_2, Y) \\ \overset{(b)}{\leq}& H(X_1) = 1, \end{aligned}$$

where step (a) follows from $H(X_1|Y) \leq H(X_1)$, and step (b) is due to $H(X_1|X_2, Y) \geq 0$. Thus, $r_s^{SC} - r_s^{TIN} = 1$ if and only if 1) $H(X_1|Y) = H(X_1) = 1$, and 2) $H(X_1|X_2, Y) = 0$.

The condition $H(X_1|Y) = \sum_{j=0}^{q-1}(\sum_{i=0}^{3} \frac{p_{i,s_j}}{4})H(X_1|Y = s_j) = 1$ holds if and only if $p_{3,s_j} + p_{2,s_j} = p_{1,s_j} + p_{0,s_j}$ for all $j = 0, 1, \ldots, q-1$. It means that even if Y is given, X_1 is still completely random to the observer. Similarly, $H(X_1|X_2, Y) = 0$ requires that $p_{3,s_j} p_{1,s_j} = p_{2,s_j} p_{0,s_j} = 0$ for all $j = 0, 1, \ldots, q-1$. It indicates that if X_2 and Y are obtained, then X_1 can be determined. ■

From the proof of Theorem 7.2.2, it is easy to see that $r_s^{SC} - r_s^{TIN} = 1$ is satisfied if and only if $I(X_1; Y) = I(X_2; Y) = 0$ and $I(X_1, X_2; Y) = 1$. Thus, in this case, if we use TIN decoding, no information can be reliably stored in the memory (i.e., $I(X_1; Y) + I(X_2; Y) = 0$), whereas if we use SC decoding, one bit of information can be reliably stored in each cell (i.e., $I(X_1, X_2; Y) = 1$).

The third decoding scheme we consider is modeled upon current MLC flash memory technology. For this scheme, the Gray labeling $\sigma = (11, 10, 00, 01)$ is used to map binary inputs (X_1, X_2) to cell levels V. The lower page X_1 and upper page X_2 are decoded independently according to different quantization rules and a total of three reads are employed. To decode X_1, $\tilde{Y}$ is quantized by one read between voltage levels A_1 and A_2 (see Figure 1.1), and the corresponding output is Y_1. To decode X_2, $\tilde{Y}$ is quantized by two reads between voltage levels A_0 and A_1, and between A_2 and A_3, respectively (see Figure 1.1), and the corresponding output is Y_2. We call this **Default Setting (DS)** decoding, and it is used as our baseline decoding scheme. Its uniform rate region $\mathscr{R}^{DS}$ for the lower page X_1 and the upper page X_2 is the set of all pairs (R_1, R_2) such that $R_1 \leq I(X_1; Y_1)$ and $R_2 \leq I(X_2; Y_2)$. In $\mathscr{R}^{DS}$, the sum rate is $r_s^{DS} = \max\{R_1 + R_2 : (R_1, R_2) \in \mathscr{R}^{DS}\} = I(X_1; Y_1) + I(X_2; Y_2)$.

Table 7.1: Channel transition matrix $p^{E}_{MLC}(y|v)$ at early-stage of P/E cycling for MLC flash memories.

V	Inputs: (X_1, X_2)			Output: Y			
Levels	Gray	NO	EO	s_0	s_1	s_2	s_3
A_0	(11)	(11)	(11)	a_1	$1-a_1$	0	0
A_1	(10)	(10)	(00)	0	b_1	$1-b_1$	0
A_2	(00)	(01)	(01)	0	0	c_1	$1-c_1$
A_3	(01)	(00)	(10)	0	0	0	1

7.3 Performance of MLC Flash Memory with Different Decoding Schemes and Labelings

In this section, we study the uniform rate region and sum rate of several MLC flash memory channel models with different decoding schemes and labelings. The channel models, inspired by empirical observation of flash memory behavior, are defined by channel transition matrices relating voltage levels to quantized outputs. Specifically, we consider *program/erase (P/E) cycling* models (early-stage and late-stage) and a *data retention* model. The P/E cycling model is used to characterize errors caused by inter-cell interference and the wear induced by repeated program/erase operations. The data retention model is used to characterize retention errors resulting from charge leakage over time from programmed cells. Note that although we concentrate on these particular models here, similar analysis can be applied to other relevant models.

Among the $4! = 24$ possible binary labelings of the 4 nominal cell voltage levels, we initially consider 3 representative examples.

Definition 7.3.1. *For MLC flash memory, the mapping σ_G=(11, 10, 00, 01) is called Gray labeling, σ_{NO}=(11, 10, 01, 00) is called Natural Order (NO) labeling, and σ_{EO}=(11, 00, 01, 10) is called Even Odd (EO) labeling.*

For each of these three labelings, the mapping between inputs $(X_1, X_2) \in \mathcal{T}_{MLC}$ and voltage levels $V \in \mathcal{V}_{MLC}$ is shown in Table 7.1.

Remark 7.3.1. It is a standard practice in current MLC flash technology to program the lower page and upper page sequentially in a 2-step procedure [56]. The cell voltage level is initially set to reflect the value

of the lower bit, with a '1' corresponding to the lowest level and a '0' corresponding to an intermediate level. When programming the upper bit, this voltage level is increased to reach the desired level corresponding to the binary labeling. Recall that a programming operation cannot decrease the cell level. Clearly this procedure is only compatible with labelings in which two lower voltage levels share the same lower bit value, and likewise for the two upper voltage levels. The Gray labeling, which is used in practice, and the NO labeling satisfy this property, but the EO labeling does not.

The 2-step programming process also influences the behavior of the flash memory channel. The early-stage P/E cycling model and data retention model are largely independent of the programming method, but the late-stage P/E cycling model includes the effect of an incorrectly programmed lower bit on the programming of the upper bit. Hence, when analyzing the performance of any MLC flash memory labeling, we will assume that the 2-bit labels of the labeling under consideration are mapped to the corresponding labels in the Gray labeling, to which the 2-step programming process is then applied. This approach can be extended to higher density flash memory, such as TLC, as well. □

We assume the quantizer Q uses three reads, placed between every pair of adjacent voltage levels, as shown in Figure 1.1. Hence, the output alphabet $\mathcal{Y}_{MLC} = \{s_0, s_1, s_2, s_3\}$. For DS decoding, we assume that the output alphabet for the lower page X_1 is $\mathcal{Y}^1_{MLC} = \{s_{0\cup1}, s_{2\cup3}\}$ of cardinality two, and the output alphabet for the upper page X_2 is $\mathcal{Y}^2_{MLC} = \{s_0, s_{1\cup2}, s_3\}$ of cardinality three; here, we use the notation $s_{u\cup v}$ to represent an output obtained by merging two outputs s_u and s_v in $\mathcal{Y}_{MLC}$, i.e., $P(Y = s_{u\cup v}|X_1 = x_1, X_2 = x_2) = \sum_{i\in\{u,v\}} P(Y = s_i|X_1 = x_1, X_2 = x_2)$ for any $x_1, x_2 \in \{0,1\}$. Strictly speaking, for the upper page decoding, current MLC flash memories use an output alphabet $\mathcal{Y}^2_{MLC} = \{s_{1\cup2}, s_{0\cup3}\}$. The resulting performance cannot exceed that obtained with the output alphabet $\{s_0, s_{1\cup2}, s_3\}$ used in this chapter. Thus, DS decoding also requires a total of three reads.

7.3.1 Performance of Gray, NO, and EO Labelings

We study the performance of MLC flash memories using the P/E cycling model, which has different channel characteristics for early and late stages of the memory lifetime.

a) Early-Stage P/E Cycling Model

The early-stage P/E cycling channel transition matrix $p^E_{MLC}(y|v)$, for output $y \in \mathcal{Y}_{MLC}$ and

voltage level $v \in \mathcal{V}_{MLC}$, reflects empirical results in [79] and is shown in Table 7.1, where a_1, $1-a_1$, b_1, $1-b_1$, c_1, and $1-c_1$ represent nonzero probabilities.

As shown in Table 7.1, note that the transition probability from inputs (X_1, X_2) to output Y depends upon the labeling which maps inputs to voltage levels, as well as the channel transition matrix from voltage levels to output.

Lemma 7.3.2. *For channel transition matrix $p^E_{MLC}(y|v)$, using Gray labeling, we have $r_s^{TIN} = r_s^{SC}$ and $\mathscr{R}^{TIN}$=$\mathscr{R}^{SC}$. Using either NO labeling or EO labeling, we have $r_s^{TIN} < r_s^{SC}$.*

Proof. With Gray labeling, in Table 7.1, for the column $Y = s_0$, we have $p_{0,s_0} = 0$, $p_{1,s_0} = 0$, $p_{2,s_0} = 0$, and $p_{3,s_0} = a_1$. Thus, $p_{3,s_0}p_{0,s_0} = p_{2,s_0}p_{1,s_0}$. We can also verify $p_{3,s_i}p_{0,s_i} = p_{2,s_i}p_{1,s_i}$ for $i = 1,2,3$. Thus, from Theorem 7.2.1, we conclude $r_s^{TIN} = r_s^{SC}$ and $\mathscr{R}^{TIN} = \mathscr{R}^{SC}$. On the other hand, under NO labeling $p_{3,s_2}p_{0,s_2} \neq p_{2,s_2}p_{1,s_2}$, and under EO labeling $p_{3,s_3}p_{0,s_3} \neq p_{2,s_3}p_{1,s_3}$. Thus, from Theorem 7.2.1, for these two labelings, $r_s^{TIN} < r_s^{SC}$. ■

Next, we calculate and compare the uniform rate regions and sum rates for the three decoding schemes under the Gray, NO, and EO labelings. The results are shown in Table 7.2, where λ_1, λ_2, λ_3, λ_4, and λ_5 are given by

$$
\begin{aligned}
\lambda_1 =& \frac{f(1-b_1) - f(3-b_1)}{4} + \frac{3}{2},\\
\lambda_2 =& 1 + \frac{1}{4}\Big(f(1-a_1) + f(1+c_1) + f(1-c_1)\Big) - \frac{1}{4}\Big(f(2-c_1) + f(2-a_1+c_1)\Big),\\
\lambda_3 =& 1 + \frac{1}{4}\Big(f(1-b_1) + f(c_1) - f(1-b_1+c_1)\Big),\\
\lambda_4 =& 1 + \frac{1}{4}\Big(f(1-a_1) + f(b_1) + f(1-c_1)\Big) - \frac{1}{4}\Big(f(1-a_1+b_1) + f(2-c_1)\Big),\\
\lambda_5 =& 1 - \frac{1}{4}\Big(f(1-a_1+b_1) + f(2-c_1) + f(1-b_1+c_1)\Big)\\
&+ \frac{1}{4}\Big(f(1-a_1) + f(c_1) + f(1-c_1) + f(b_1) + f(1-b_1)\Big).
\end{aligned}
$$

As an example, we show how to calculate λ_1; the quantities λ_2, λ_3, λ_4, and λ_5 can be obtained in a similar manner. We see that λ_1 corresponds to $I(X_1;Y_1)$ under DS decoding. Referring to Table 7.1, we see that

Table 7.2: Uniform rate regions and sum rates of DS, TIN, and SC decodings at early-stage of P/E cycling for MLC flash memories.

Gray	DS	$\mathscr{R}_G^{DS}$	$0 \le R_1 \le \lambda_1, 0 \le R_2 \le \lambda_2$	$r_{s(G)}^{DS} = \lambda_1 + \lambda_2$
	TIN	$\mathscr{R}_G^{TIN}$	$0 \le R_1 \le \lambda_3, 0 \le R_2 \le \lambda_4$	$r_{s(G)}^{TIN} = \lambda_3 + \lambda_4$
	SC	$\mathscr{R}_G^{SC}$	$0 \le R_1 \le \lambda_3, 0 \le R_2 \le \lambda_4$	$r_{s(G)}^{SC} = \lambda_3 + \lambda_4$
NO	TIN	$\mathscr{R}_{NO}^{TIN}$	$0 \le R_1 \le \lambda_3, 0 \le R_2 \le \lambda_5$	$r_{s(NO)}^{TIN} = \lambda_3 + \lambda_5$
	SC	$\mathscr{R}_{NO}^{SC}$	$0 \le R_1 \le 1, 0 \le R_2 \le \lambda_4, R_1 + R_2 \le 1 + \lambda_5$	$r_{s(NO)}^{SC} = 1 + \lambda_5$
EO	TIN	$\mathscr{R}_{EO}^{TIN}$	$0 \le R_1 \le \lambda_4, 0 \le R_2 \le \lambda_5$	$r_{s(EO)}^{TIN} = \lambda_4 + \lambda_5$
	SC	$\mathscr{R}_{EO}^{SC}$	$0 \le R_1 \le 1, 0 \le R_2 \le \lambda_3, R_1 + R_2 \le 1 + \lambda_5$	$r_{s(EO)}^{SC} = 1 + \lambda_5$

$$\begin{aligned}
P(Y_1 = s_{0\cup 1}|X_1 = 1) =& \frac{P(Y_1 = s_{0\cup 1}, X_1 = 1)}{P(X_1 = 1)} \\
=& \frac{\sum_{i=0}^{1} P(Y_1 = s_{0\cup 1}, X_1 = 1, X_2 = i)}{P(X_1 = 1)} \\
=& \frac{\sum_{i=0}^{1} P(Y_1 = s_{0\cup 1}|X_1 = 1, X_2 = i) P(X_1 = 1, X_2 = i)}{P(X_1 = 1)} = \frac{1 + b_1}{2}.
\end{aligned}$$

Similarly, we calculate $P(Y_1 = s_{2\cup 3}|X_1 = 1) = \frac{1-b_1}{2}$, $P(Y_1 = s_{0\cup 1}|X_1 = 0) = 0$, and $P(Y_1 = s_{2\cup 3}|X_1 = 0) = 1$. Noting that $P(Y_1 = s_{0\cup 1}) = \frac{1+b_1}{4}$ and $P(Y_1 = s_{2\cup 3}) = \frac{3-b_1}{4}$, we get

$$\begin{aligned}
\lambda_1 =& I(X_1; Y_1) \\
=& H(Y_1) - H(Y_1|X_1) \\
=& H(\frac{1+b_1}{4}, \frac{3-b_1}{4}) - \frac{1}{2} H(\frac{1+b_1}{2}, \frac{1-b_1}{2}) \\
=& -\frac{1}{4}\Big(f(1+b_1) + f(3-b_1)\Big) + 2 - \frac{1}{2}\Big(-\frac{1}{2}\big(f(1+b_1) + f(1-b_1)\big) + 1\Big) \\
=& \frac{f(1-b_1) - f(3-b_1)}{4} + \frac{3}{2}.
\end{aligned}$$

From Table 7.2, we have the following rate region and sum rate comparisons for the Gray, NO, and EO labelings. We denote the uniform rate region under DS decoding and Gray labeling by the term $\mathscr{R}_G^{DS}$, where the superscript and subscript represent decoding scheme and labeling, respectively. The same holds for other terms in Table 7.2.

Theorem 7.3.3. *With channel transition matrix $p^E_{MLC}(y|v)$, the rate regions satisfy $\mathscr{R}^{DS}_G \subset \mathscr{R}^{TIN}_G$, $\mathscr{R}^{TIN}_{NO} \subset \mathscr{R}^{TIN}_G$, and $\mathscr{R}^{SC}_G \subset \mathscr{R}^{SC}_{NO}$. For the sum rates, we have $r^{TIN}_{s(G)} > r^{DS}_{s(G)}$, $r^{TIN}_{s(G)} > r^{TIN}_{s(NO)}$, $r^{TIN}_{s(G)} > r^{TIN}_{s(EO)}$, and $r^{SC}_{s(G)} = r^{SC}_{s(NO)} = r^{SC}_{s(EO)}$.*

Proof. Referring to Table 7.2, we only need to show that $\lambda_3 > \lambda_1$, $\lambda_4 > \lambda_2$, $\lambda_4 > \lambda_5$, $\lambda_3 > \lambda_5$, $1 > \lambda_3$, and $\lambda_3 + \lambda_4 = 1 + \lambda_5$. Here we only give proofs of $\lambda_3 > \lambda_1$ and $\lambda_4 > \lambda_5$. Other relationships can be proved in a similar way.

We have $\lambda_3 - \lambda_1 = \frac{1}{4}\Big(f(c_1) + f(3-b_1) - f(1-b_1+c_1)\Big) - \frac{1}{2}$. Let $h_1(b_1,c_1) = f(c_1) + f(3-b_1) - f(1-b_1+c_1)$. For $0 < b_1 < 1$ and $0 < c_1 < 1$, we have

$$\frac{\partial h_1(b_1,c_1)}{\partial b_1} = \log_2(1-b_1+c_1) - \log_2(3-b_1) < 0.$$

Thus, for $0 < b_1 < 1$ and $0 < c_1 < 1$, we have $h_1(b_1,c_1) > h_1(b_1 = 1,c_1) = 2$, so $\lambda_3 > \lambda_1$.

For $0 < b_1 < 1$ and $0 < c_1 < 1$, we have

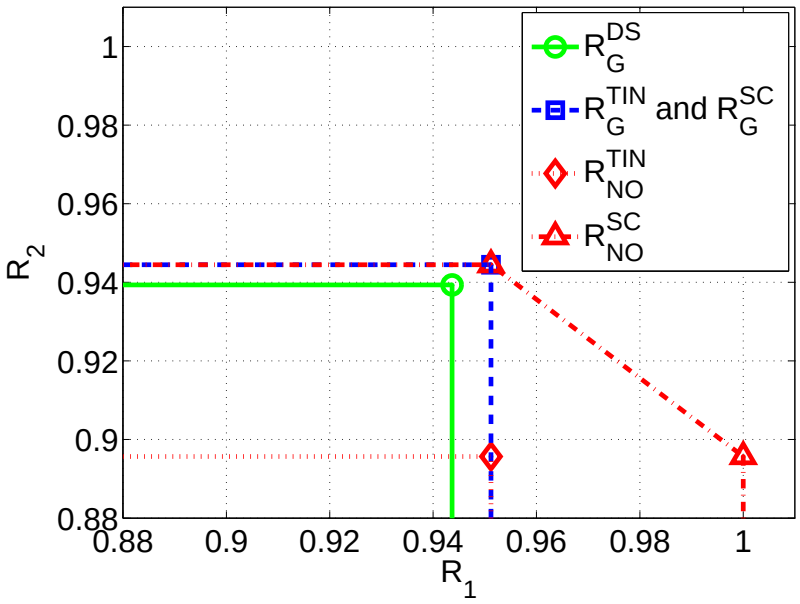
$$f(1-b_1) + f(c_1) - f(1-b_1+c_1) = (1-b_1)\log_2\frac{1-b_1}{1-b_1+c_1} + c_1\log_2\frac{c_1}{1-b_1+c_1} < 0,$$

so $\lambda_4 - \lambda_5 = -\frac{1}{4}\Big(f(1-b_1) + f(c_1) - f(1-b_1+c_1)\Big) > 0$. ■

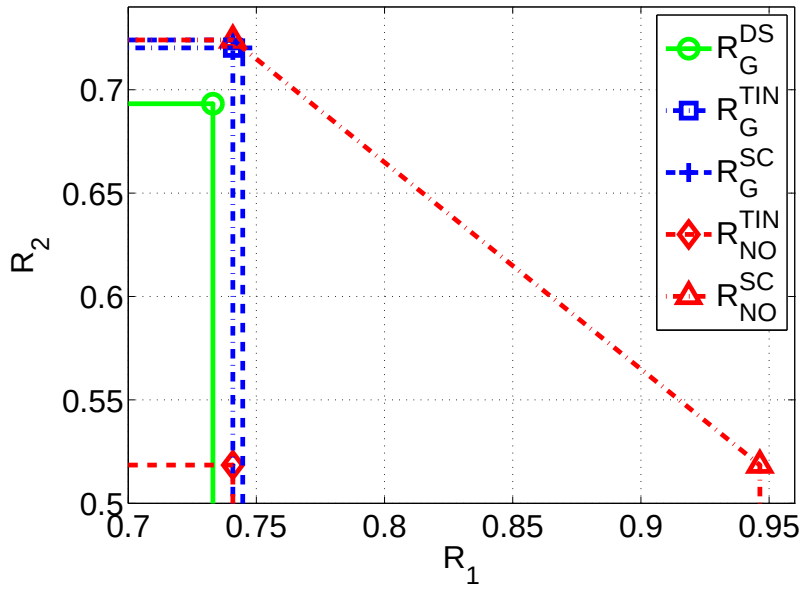
Example 7.3.1. For the early-stage P/E cycling model in Table 7.1, let $a_1 = 0.98$, $b_1 = 0.97$, and $c_1 = 0.99$. The uniform rate regions under Gray and NO labelings are plotted in Figure 7.1(a). It can be seen that $\mathscr{R}^{TIN}_{NO} \subset \mathscr{R}^{TIN}_G = \mathscr{R}^{SC}_G \subset \mathscr{R}^{SC}_{NO}$. The SC decoding with NO labeling gives the largest rate region. Note that since the channel transition matrix varies from different flash chip vendors, the channel parameters are chosen to help visualize the relationship among the rate regions. For other choices of channel parameters, the relative positions of the rate regions and qualitative conclusions stay the same. □

b) Late-Stage P/E Cycling Model

The late-stage P/E cycling channel transition matrix $p^L_{MLC}(y|v)$, for output $y \in \mathcal{Y}_{MLC}$ and voltage level $v \in \mathcal{V}_{MLC}$, reflects measurements in [79] and its structure is shown in Table 7.3 where $\hat{a}_1$, $\hat{a}_2$, $1-\hat{a}_1-\hat{a}_2$, $\hat{b}_1$, $1-\hat{b}_1$, $\hat{c}_1$, and $1-\hat{c}_1$ represent nonzero probabilities.



(a)



(b)

Figure 7.1: (a) Uniform rate regions under Gray and NO labelings with $a_1 = 0.98$, $b_1 = 0.97$, and $c_1 = 0.99$ for the early-stage P/E cycling model. (b) Uniform rate regions under Gray and NO labelings with $\hat{a}_1 = 0.82$, $\hat{a}_2 = 0.1$, $\hat{b}_1 = 0.85$, and $\hat{c}_1 = 0.85$ for the late-stage P/E cycling model.

Lemma 7.3.4. *For channel transition matrix $p^L_{MLC}(y|v)$, we have $r_s^{TIN} < r_s^{SC}$ with Gray labeling, NO labeling or EO labeling.*

Table 7.3: Channel transition matrix $p^L_{MLC}(y|v)$ at late-stage of P/E cycling for MLC flash memories.

V	Inputs: (X_1, X_2)			Output: Y			
Levels	Gray	NO	EO	s_0	s_1	s_2	s_3
A_0	(11)	(11)	(11)	$\hat{a}_1$	$\hat{a}_2$	0	$1-\hat{a}_1-\hat{a}_2$
A_1	(10)	(10)	(00)	0	$\hat{b}_1$	$1-\hat{b}_1$	0
A_2	(00)	(01)	(01)	0	0	$\hat{c}_1$	$1-\hat{c}_1$
A_3	(01)	(00)	(10)	0	0	0	1

Table 7.4: Uniform rate regions and sum rates of DS, TIN, and SC decodings at late-stage of P/E cycling for MLC flash memories.

Gray	DS	$\mathscr{R}_G^{DS}$	$0 \le R_1 \le \tau_1, 0 \le R_2 \le \tau_2$	$r^{DS}_{s(G)} = \tau_1 + \tau_2$
	TIN	$\mathscr{R}_G^{TIN}$	$0 \le R_1 \le \tau_3, 0 \le R_2 \le \tau_4$	$r^{TIN}_{s(G)} = \tau_3 + \tau_4$
	SC	$\mathscr{R}_G^{SC}$	$0 \le R_1 \le \tau_5, 0 \le R_2 \le \tau_6, R_1 + R_2 \le \tau_4 + \tau_5$	$r^{SC}_{s(G)} = \tau_4 + \tau_5$
NO	TIN	$\mathscr{R}_{NO}^{TIN}$	$0 \le R_1 \le \tau_3, 0 \le R_2 \le \tau_7$	$r^{TIN}_{s(NO)} = \tau_3 + \tau_7$
	SC	$\mathscr{R}_{NO}^{SC}$	$0 \le R_1 \le \tau_8, 0 \le R_2 \le \tau_6, R_1 + R_2 \le \tau_7 + \tau_8$	$r^{SC}_{s(NO)} = \tau_7 + \tau_8$
EO	TIN	$\mathscr{R}_{EO}^{TIN}$	$0 \le R_1 \le \tau_4, 0 \le R_2 \le \tau_7$	$r^{TIN}_{s(EO)} = \tau_4 + \tau_7$
	SC	$\mathscr{R}_{EO}^{SC}$	$0 \le R_1 \le \tau_8, 0 \le R_2 \le \tau_5, R_1 + R_2 \le \tau_7 + \tau_8$	$r^{SC}_{s(EO)} = \tau_7 + \tau_8$

Proof. For any of the three labelings, consider the column $Y = s_3$ in Table 7.3. Since three of p_{0,s_3}, p_{1,s_3}, p_{2,s_3}, and p_{3,s_3} are positive, it is impossible to satisfy $p_{3,s_3}p_{0,s_3} = p_{2,s_3}p_{1,s_3} = 0$. Thus, from Theorem 7.2.1, we conclude $r_s^{TIN} < r_s^{SC}$. ■

Next, we calculate the uniform rate regions and the sum rates of the three decoding schemes under different labelings. The results are shown in Table 7.4, where τ_i, $i = 1, \ldots, 8$, are given by

$$\begin{aligned}
\tau_1 =& \frac{f(2-\hat{a}_1-\hat{a}_2-\hat{b}_1) - f(4-\hat{a}_1-\hat{a}_2-\hat{b}_1)}{4} + \frac{3}{2},\\
\tau_2 =& 1 + \frac{1}{4}\Big(f(\hat{a}_2) + f(2-\hat{a}_1-\hat{a}_2) + f(1+\hat{c}_1) + f(1-\hat{c}_1)\Big)\\
&- \frac{1}{4}\Big(f(3-\hat{a}_1-\hat{a}_2-\hat{c}_1) + f(\hat{a}_2+\hat{c}_1+1)\Big),\\
\tau_3 =& 1 + \frac{1}{4}\Big(f(1-\hat{b}_1) + f(\hat{c}_1) + f(1-\hat{a}_1-\hat{a}_2) + f(2-\hat{c}_1)\Big)\\
&- \frac{1}{4}\Big(f(1-\hat{b}_1+\hat{c}_1) + f(3-\hat{a}_1-\hat{a}_2-\hat{c}_1)\Big),\\
\tau_4 =& 1 + \frac{1}{4}\Big(f(\hat{a}_2) + f(2-\hat{a}_1-\hat{a}_2) + f(\hat{b}_1) + f(1-\hat{c}_1)\Big)\\
&- \frac{1}{4}\Big(f(\hat{a}_2+\hat{b}_1) + f(3-\hat{a}_1-\hat{a}_2-\hat{c}_1)\Big),
\end{aligned}$$

$$
\begin{aligned}
\tau_5 =& 1+\frac{1}{4}\Big(f(1-\hat{a}_1-\hat{a}_2)+f(1-\hat{b}_1)+f(\hat{c}_1)\Big)\\
&-\frac{1}{4}\Big(f(2-\hat{a}_1-\hat{a}_2)+f(1-\hat{b}_1+\hat{c}_1)\Big),\\
\tau_6 =& 1+\frac{1}{4}\Big(f(\hat{a}_2)+f(\hat{b}_1)+f(1-\hat{c}_1)\Big)-\frac{1}{4}\Big(f(\hat{a}_2+\hat{b}_1)+f(2-\hat{c}_1)\Big),\\
\tau_7 =& 1-\frac{1}{4}\Big(f(\hat{a}_2+\hat{b}_1)+f(1-\hat{b}_1+\hat{c}_1)+f(3-\hat{a}_1-\hat{a}_2-\hat{c}_1)\Big)\\
&+\frac{1}{4}\Big(f(\hat{a}_2)+f(\hat{c}_1)+f(2-\hat{a}_1-\hat{a}_2-\hat{c}_1)+f(\hat{b}_1)+f(1-\hat{b}_1)\Big),\\
\tau_8 =& 1+\frac{1}{4}\Big(f(1-\hat{c}_1)+f(1-\hat{a}_1-\hat{a}_2)-f(2-\hat{a}_1-\hat{a}_2-\hat{c}_1)\Big).
\end{aligned}
$$

From Table 7.4, we can infer the following rate region and sum rate relationships.

Theorem 7.3.5. *With channel transition matrix $p^L_{MLC}(y|v)$, the rate regions satisfy $\mathscr{R}^{DS}_G \subset \mathscr{R}^{TIN}_G$, $\mathscr{R}^{TIN}_{NO} \subset \mathscr{R}^{TIN}_G$, and $\mathscr{R}^{SC}_G \subset \mathscr{R}^{SC}_{NO}$. For the sum rates, we have $r^{TIN}_{s(G)} > r^{DS}_{s(G)}$, $r^{TIN}_{s(G)} > r^{TIN}_{s(NO)}$, $r^{TIN}_{s(G)} > r^{TIN}_{s(EO)}$, and $r^{SC}_{s(G)} = r^{SC}_{s(NO)} = r^{SC}_{s(EO)}$.*

Proof. Using Table 7.4, we only need to show that $\tau_3 \geq \tau_1$, $\tau_4 > \tau_2$, $\tau_4 > \tau_7$, $\tau_8 > \tau_5$, $\tau_3 > \tau_7$, and $\tau_4 + \tau_5 = \tau_7 + \tau_8$. Here, we give proofs of $\tau_3 \geq \tau_1$ and $\tau_4 > \tau_2$. The other relationships can be proved in a similar way. First, we compute

$$
\begin{aligned}
&4(\tau_3-\tau_1)\\
=&f(1-\hat{b}_1)+f(\hat{c}_1)+f(1-\hat{a}_1-\hat{a}_2)+f(2-\hat{c}_1)+f(4-\hat{a}_1-\hat{a}_2-\hat{b}_1)-f(1-\hat{b}_1+\hat{c}_1)\\
&-f(3-\hat{a}_1-\hat{a}_2-\hat{c}_1)-f(2-\hat{a}_1-\hat{a}_2-\hat{b}_1)-f(2)\\
=&(1-\hat{b}_1+\hat{c}_1)\log_2(1+\frac{3-\hat{a}_1-\hat{a}_2-\hat{c}_1}{1-\hat{b}_1+\hat{c}_1})-(1-\hat{b}_1)\log_2(1+\frac{1-\hat{a}_1-\hat{a}_2}{1-\hat{b}_1})-\hat{c}_1\log_2(1+\frac{2-\hat{c}_1}{\hat{c}_1})\\
&+(3-\hat{a}_1-\hat{a}_2-\hat{c}_1)\log_2(1+\frac{1-\hat{b}_1+\hat{c}_1}{3-\hat{a}_1-\hat{a}_2-\hat{c}_1})-(1-\hat{a}_1-\hat{a}_2)\log_2(1+\frac{1-\hat{b}_1}{1-\hat{a}_1-\hat{a}_2})\\
&-(2-\hat{c}_1)\log_2(1+\frac{\hat{c}_1}{2-\hat{c}_1}).
\end{aligned}
$$

Note that the function $t\log_2(1+1/t)$ is concave. Define $t_1 = \frac{1-\hat{b}_1}{1-\hat{a}_1-\hat{a}_2}$, $t_2 = \frac{\hat{c}_1}{2-\hat{c}_1}$, $r_1 = \frac{1-\hat{a}_1-\hat{a}_2}{3-\hat{a}_1-\hat{a}_2-\hat{c}_1}$, and $r_2 = \frac{2-\hat{c}_1}{3-\hat{a}_1-\hat{a}_2-\hat{c}_1}$. Then we have

$$
(r_1t_1+r_2t_2)\log_2\big(1+1/(r_1t_1+r_2t_2)\big) \geq r_1t_1\log_2(1+1/t_1)+r_2t_2\log_2(1+1/t_2).
$$

That is,

$$(1-\hat{b}_1+\hat{c}_1)\log_2(1+\frac{3-\hat{a}_1-\hat{a}_2-\hat{c}_1}{1-\hat{b}_1+\hat{c}_1}) \geq (1-\hat{b}_1)\log_2(1+\frac{1-\hat{a}_1-\hat{a}_2}{1-\hat{b}_1})+\hat{c}_1\log_2(1+\frac{2-\hat{c}_1}{\hat{c}_1}).$$

Similarly, we have

$$\begin{aligned}&(3-\hat{a}_1-\hat{a}_2-\hat{c}_1)\log_2(1+\frac{1-\hat{b}_1+\hat{c}_1}{3-\hat{a}_1-\hat{a}_2-\hat{c}_1})\\ \geq&(1-\hat{a}_1-\hat{a}_2)\log_2(1+\frac{1-\hat{b}_1}{1-\hat{a}_1-\hat{a}_2})+(2-\hat{c}_1)\log_2(1+\frac{\hat{c}_1}{2-\hat{c}_1}).\end{aligned}$$

Therefore, $\tau_3-\tau_1\geq 0$.

Next, we compute $\tau_4-\tau_2=\frac{1}{4}\Big(f(\hat{b}_1)+f(\hat{a}_2+\hat{c}_1+1)-f(\hat{a}_2+\hat{b}_1)-f(1+\hat{c}_1)\Big)$. Let $\hat{h}_1(\hat{a}_2,\hat{b}_1,\hat{c}_1)=f(\hat{b}_1)+f(\hat{a}_2+\hat{c}_1+1)-f(\hat{a}_2+\hat{b}_1)-f(1+\hat{c}_1)$. For $0<\hat{a}_2<1$, $0<\hat{b}_1<1$, and $0<\hat{c}_1<1$, we have $\frac{\partial\hat{h}_1(\hat{a}_2,\hat{b}_1,\hat{c}_1)}{\partial\hat{a}_2}=\log_2(\hat{a}_2+\hat{c}_1+1)-\log_2(\hat{a}_2+\hat{b}_1)>0$. Thus, for $0<\hat{a}_2<1$, $0<\hat{b}_1<1$, and $0<\hat{c}_1<1$, $\hat{h}_1(\hat{a}_2,\hat{b}_1,\hat{c}_1)>\hat{h}_1(\hat{a}_2=0,\hat{b}_1,\hat{c}_1)=0$, so $\tau_4>\tau_2$. ■

Example 7.3.2. For the late-stage P/E cycling model in Table 7.3, let $\hat{a}_1=0.82$, $\hat{a}_2=0.1$, $\hat{b}_1=0.85$, and $\hat{c}_1=0.85$. The uniform rate regions under Gray and NO labelings are plotted in Figure 7.1(b). We see that $\mathscr{R}_{NO}^{TIN}\subset\mathscr{R}_{G}^{TIN}\subset\mathscr{R}_{G}^{SC}\subset\mathscr{R}_{NO}^{SC}$. Note that unlike the early-stage P/E cycling model in Example 7.3.1, here the region $\mathscr{R}_{G}^{TIN}$ is strictly included in $\mathscr{R}_{G}^{SC}$. □

Remark 7.3.2. For both P/E cycling models, Theorems 7.3.3 and 7.3.5 imply that, for TIN decoding, among the 3 labelings, Gray labeling gives the largest sum rate, which is also larger than the sum rate of DS decoding. Moreover, compared to NO labeling, Gray labeling generates a larger uniform rate region for TIN decoding, but a smaller one for SC decoding.

For the early-stage P/E cycling model, we can also draw several conclusions from Lemma 7.3.2, Theorem 7.3.3, and Table 7.2 that provide insight into efficient coding schemes.

First, the sum rate of TIN decoding under Gray labeling is the same as that of SC decoding under any of Gray, NO, and EO labelings. This provides a symmetric capacity-achieving coding solution based on good codes for the point-to-point channel. With Gray labeling, we only need to use two point-to-point

symmetric capacity-achieving codes, e.g., polar codes [4], for the lower and upper pages, to achieve the rates $I(X_1;Y)$ and $I(X_2;Y)$, respectively. The two pages can be decoded independently.

Second, for NO labeling or EO labeling, with SC decoding, the rate pair $(R_1 = 1, R_2 = \lambda_5)$ can be achieved, which implies that no coding is required for the lower page X_1. This also suggests us a very simple coding solution. We only need to apply a symmetric capacity-achieving, point-to-point code to the upper page X_2 to achieve rate $I(X_2;Y)$, and no coding is needed for the lower page X_1. To recover the data, we first decode the upper page. Then, we can determine the lower page using the decoded data from the upper page, the binary labeling, the channel transition matrix, and the output Y. For example, referring to the NO labeling in Table 7.1, we see that if the correctly decoded bit of the upper page is $X_2 = 1$ and the output is $Y = s_2$, then the lower page bit must be $X_1 = 0$.

For the late-stage P/E cycling model, from Lemma 7.3.4, Theorem 7.3.5, and Table 7.4, we see that the sum rate of TIN decoding under Gray labeling is strictly less than that of SC decoding. The gap Δ between the two sum rates is

$$\begin{aligned}\Delta &= r_{s(G)}^{SC} - r_{s(G)}^{TIN} = \tau_5 - \tau_3 \\ &= \frac{1}{4}\Big(f(3-\hat{a}_1-\hat{a}_2-\hat{c}_1) - f(2-\hat{a}_1-\hat{a}_2) - f(2-\hat{c}_1)\Big).\end{aligned}$$

To bound the gap Δ, let $\hat{a} = \hat{a}_1 + \hat{a}_2$ and $\hat{h}(\hat{a},\hat{c}_1) = f(3-\hat{a}-\hat{c}_1) - f(2-\hat{a}) - f(2-\hat{c}_1)$. For $0 < \hat{a} < 1$ and $0 < \hat{c}_1 < 1$, we have $\frac{\partial \hat{h}(\hat{a},\hat{c}_1)}{\partial \hat{a}} = \log_2(2-\hat{a}) - \log_2(3-\hat{a}-\hat{c}_1) < 0$, and $\frac{\partial \hat{h}(\hat{a},\hat{c}_1)}{\partial \hat{c}_1} = \log_2(2-\hat{c}_1) - \log_2(3-\hat{a}-\hat{c}_1) < 0$. Therefore, $\frac{\hat{h}(\hat{a}=1,\hat{c}_1=1)}{4} < \Delta < \frac{\hat{h}(\hat{a}=0,\hat{c}_1=0)}{4}$; that is, $0 < \Delta < \frac{3\log_2 3 - 4}{4} = 0.1887$. If we impose constraints $\eta_{\hat{a}} \le \hat{a}_1 + \hat{a}_2 < 1$ and $\eta_{\hat{c}_1} \le \hat{c}_1 < 1$, we have $0 < \Delta \le \frac{1}{4}\Big(f(3-\eta_{\hat{a}}-\eta_{\hat{c}_1}) - f(2-\eta_{\hat{a}}) - f(2-\eta_{\hat{c}_1})\Big)$. For example, for $\eta_{\hat{a}} = 0.95$ and $\eta_{\hat{c}_1} = 0.85$, we get $0 < \Delta \le 0.00246$. In general, the gap Δ is very small. □

c) Data Retention Model

For the data retention model, the structure of the channel transition matrix $p_{MLC}^{DR}(y|v)$, for output $y \in \mathcal{Y}_{MLC}$ and voltage level $v \in \mathcal{V}_{MLC}$, is shown in Table 7.5, where $\tilde{a}_1$, $1-\tilde{a}_1$, $\tilde{b}_1$, $1-\tilde{b}_1$, $\tilde{c}_1$, and $1-\tilde{c}_1$ represent nonzero probabilities. In contrast to the early-stage P/E cycling model, where errors are caused by upward drift of cell voltages, in the data retention model, errors arise from downward drift of the

Table 7.5: Channel transition matrix $p_{MLC}^{DR}(y|v)$ of data retention for MLC flash memories.

V	Inputs: (X_1, X_2)			Output: Y			
Levels	Gray	NO	EO	s_0	s_1	s_2	s_3
A_0	(11)	(11)	(11)	1	0	0	0
A_1	(10)	(10)	(00)	$1-\tilde{a}_1$	$\tilde{a}_1$	0	0
A_2	(00)	(01)	(01)	0	$1-\tilde{b}_1$	$\tilde{b}_1$	0
A_3	(01)	(00)	(10)	0	0	$1-\tilde{c}_1$	$\tilde{c}_1$

cell voltages.

Analysis and results for the data retention model are very similar to those of the early-stage P/E cycling model. We state only one representative result here, without a detailed proof.

Lemma 7.3.6. *For channel transition matrix $p_{MLC}^{DR}(y|v)$, using Gray labeling, we have $r_s^{TIN} = r_s^{SC}$ and $\mathscr{R}^{TIN} = \mathscr{R}^{SC}$. Using either NO labeling or EO labeling, we have $r_s^{TIN} < r_s^{SC}$.*

7.3.2 Extension to All Labelings

In this subsection, we extend the analysis to the entire set of labelings. There exist a total of $4! = 24$ labelings. In order to categorize and analyze these 24 labelings, we take advantage of the algebraic structure of permutation groups, and consider a labeling σ as a permutation π in the symmetric group $\mathcal{S}_4$. This is the group whose elements are all the permutation operations that can be performed on the 4 distinct elements in $\mathcal{T}_{MLC}$, and whose group operation, denoted as $*$, is the composition of such permutation operations. A labeling $\sigma = (w_0, w_1, w_2, w_3)$ corresponds to the permutation $\pi = (w_0, w_1, w_2, w_3)$ in $\mathcal{S}_4$, where the permutation vector $\pi = (w_0, w_1, w_2, w_3)$ is defined to represent $\pi(11) = w_0$, $\pi(10) = w_1$, $\pi(01) = w_2$, and $\pi(00) = w_3$, e.g., $\pi = (11, 10, 01, 00)$ is the identity permutation in $\mathcal{S}_4$. The group operation $*$ of two permutations π_1 and π_2 is defined as their composition and results in another permutation $\pi_3 = \pi_1 * \pi_2$. In other words, $\pi_1 * \pi_2$ is the function that maps any element $w \in \mathcal{T}_{MLC}$ to $\pi_1\big(\pi_2(w)\big)$. Note that the rightmost permutation is applied first. For example, $(10, 11, 00, 01) * (11, 10, 00, 01) = (10, 11, 01, 00)$.

Lemma 7.3.7. *In the symmetric group $\mathcal{S}_4$, $G_0 = \{(11, 10, 01, 00),\ (10, 11, 00, 01),\ (01, 00, 11, 10),\ (00, 01, 10, 11)\}$ forms a normal subgroup (the Klein four-group).*

Proof. The element $(11,10,01,00)$ in G_0 is the identity element in $\mathcal{S}_4$. We can verify that the 4 elements in G_0 have the following properties: 1) the composition of the identity element and any element is that element itself; 2) the composition of any non-identity element with itself is the identity element; 3) the composition of two distinct non-identity elements is the third non-identity element. Thus, G_0 is the Klein four-group. ■

With the subgroup G_0 in Lemma 7.3.7, we partition $\mathcal{S}_4$ into G_0 and its 5 cosets, each of size 4: $\mathcal{S}_4 = G_0 \cup G_1 \cup G_2 \cup \bar{G}_0 \cup \bar{G}_1 \cup \bar{G}_2$, where $G_1 = G_0 * (11,10,00,01)$; $G_2 = G_0 * (11,00,01,10)$; $\bar{G}_0 = G_0 * (11,01,10,00)$; $\bar{G}_1 = G_0 * (11,01,00,10)$; $\bar{G}_2 = G_0 * (11,00,10,01)$.

In the following, we will treat each vector in every coset as a labeling. For example, G_0 includes $\sigma_{NO} = (11,10,01,00)$, G_1 includes $\sigma_G = (11,10,00,01)$, and G_2 includes $\sigma_{EO} = (11,00,01,10)$. The following two lemmas give properties of the uniform rate regions for different labelings. We assume an *arbitrary* channel transition matrix $p_{MLC}(y|v)$, for output $y \in \mathcal{Y}_{MLC}$ and voltage level $v \in \mathcal{V}_{MLC}$, is given. The first lemma leverages the symmetries within the Klein four-group and its cosets to deduce the relationship of the rate regions of different labelings.

Lemma 7.3.8. *With an arbitrary channel transition matrix $p_{MLC}(y|v)$, for TIN decoding, the 4 labelings in each of G_0, G_1, G_2, $\bar{G}_0$, $\bar{G}_1$, and $\bar{G}_2$ give the same uniform rate region $\mathscr{R}^{TIN}$ and sum rate r_s^{TIN}. For SC decoding, the 4 labelings in each of G_0, G_1, G_2, $\bar{G}_0$, $\bar{G}_1$, and $\bar{G}_2$ give the same uniform rate region $\mathscr{R}^{SC}$, and all 24 labelings in $\mathcal{S}_4$ give the same sum rate r_s^{SC}.*

Proof. This is based on the fact that the 4 labelings in each of G_0, G_1, G_2, $\bar{G}_0$, $\bar{G}_1$, and $\bar{G}_2$ are interchangeable by one of the following three operations: 1) in position X_1, change 0 to 1 and 1 to 0; 2) in position X_2, change 0 to 1 and 1 to 0; 3) in both positions X_1 and X_2, change 0 to 1 and 1 to 0. For example, in G_0, $(11,10,01,00)$ is transformed to $(01,00,11,10)$ by changing 0 to 1 and 1 to 0 in position X_1, is transformed to $(10,11,00,01)$ by changing 0 to 1 and 1 to 0 in position X_2, and is transformed to $(00,01,10,11)$ by changing 0 to 1 and 1 to 0 in both positions X_1 and X_2. Since the distributions for X_1 and X_2 are uniform, the values of $I(X_1;Y)$, $I(X_2;Y)$, $I(X_1;Y|X_2)$, and $I(X_2;Y|X_1)$ under a labeling σ_1 are the same as those under a labeling σ_2 which is obtained by one of the above three operations on the labeling σ_1. Thus, for a fixed decoding scheme (TIN or SC), the uniform rate region and sum rate under

the labeling σ_1 are the same as those under the labeling σ_2. Therefore, for a fixed decoding scheme (TIN or SC), the 4 labelings in each coset give the same uniform rate region and sum rate. For the sum rate of SC decoding, for all 24 labelings, $I(X_1, X_2; Y)$ is the same due to the uniform distributions for X_1 and X_2. ■

Lemma 7.3.9. *With an arbitrary channel transition matrix $p_{MLC}(y|v)$, for TIN decoding, if the labelings in G_i, $i = 0, 1, 2$, give a uniform rate region: $R_1 \leq \varphi_1$ and $R_2 \leq \varphi_2$, then the labelings in $\bar{G}_i$ give a uniform rate region: $R_1 \leq \varphi_2$ and $R_2 \leq \varphi_1$. For SC decoding, if the labelings in G_i give a uniform rate region: $R_1 \leq \psi_1$, $R_2 \leq \psi_2$, and $R_1 + R_2 \leq \psi_3$, then the labelings in $\bar{G}_i$ give a uniform rate region: $R_1 \leq \psi_2$, $R_2 \leq \psi_1$, and $R_1 + R_2 \leq \psi_3$.*

Proof. This is based on the fact that the 4 labelings in G_i, i=0, 1, 2, are transformed (one-to-one) to the 4 labelings in $\bar{G}_i$ by swapping the values in positions X_1 and X_2. For example, $(11, 10, 01, 00)$ in G_0 is transformed to $(11, 01, 10, 00)$ in $\bar{G}_0$. With the uniform distributions for X_1 and X_2, X_1 (or X_2) with labeling $(11, 10, 01, 00)$ is equivalent to X_2 (or X_1) with labeling $(11, 01, 10, 00)$. Thus, for a fixed decoding scheme (TIN or SC), the uniform rate region under labeling $(11, 10, 01, 00)$ will become the one under labeling $(11, 01, 10, 00)$ by swapping the constraints on R_1 and R_2. From Lemma 7.3.8, it follows that the uniform rate region under labelings in G_0 will become the one under labelings in $\bar{G}_0$ by swapping the constraints on R_1 and R_2. The same conclusion holds for the labelings in G_i and $\bar{G}_i$, $i = 1, 2$. ■

Remark 7.3.3. Theorems 7.3.3 and 7.3.5, and Lemmas 7.3.8 and 7.3.9, imply that for both P/E cycling models, with TIN decoding, the 8 labelings in G_1 (including Gray labeling) and $\bar{G}_1$ produce the largest sum rate among all 24 labelings. With SC decoding, all of the 24 labelings give the same sum rate. □

Finally, we examine the uniform rate region that can be achieved by using multiple labelings in $\mathcal{S}_4$ within a codeword in a time-sharing fashion. Define $\mathscr{R}_{\mathcal{S}_4}^{TIN} = Conv\Big(\bigcup_{\sigma \in \mathcal{S}_4} \mathscr{R}_{\sigma}^{TIN}\Big)$, the convex hull of uniform rate regions of all 24 labelings for TIN decoding. Define $\mathscr{R}_{\mathcal{S}_4}^{SC} = Conv\Big(\bigcup_{\sigma \in \mathcal{S}_4} \mathscr{R}_{\sigma}^{SC}\Big)$, the convex hull of uniform rate regions of all 24 labelings for SC decoding. Through time-sharing of different labelings, we obtain the following lemma.

Lemma 7.3.10. *For TIN decoding, any point $(R_1, R_2) \in \mathscr{R}_{\mathcal{S}_4}^{TIN}$ can be achieved. For SC decoding, any point $(R_1, R_2) \in \mathscr{R}_{\mathcal{S}_4}^{SC}$ can be achieved.*

Proof. We first show that any point $(R_1, R_2) \in \mathscr{R}_{\mathcal{S}_4}^{TIN}$ can be achieved. From Carathéodory's Theorem [19], any point (R_1, R_2) in $\mathscr{R}_{\mathcal{S}_4}^{TIN}$ can be represented as a convex combination of 3 points in $\bigcup_{\sigma \in \mathcal{S}_4} \mathscr{R}_{\sigma}^{TIN}$. Without loss of generality, we assume $(R_1, R_2) = \alpha_1(R_1^1, R_2^1) + \alpha_2(R_1^2, R_2^2) + \alpha_3(R_1^3, R_2^3)$, where $\alpha_1, \alpha_2, \alpha_3 \geq 0$ and $\sum_{i=1}^{3} \alpha_i = 1$. Points (R_1^1, R_2^1), (R_1^2, R_2^2), and (R_1^3, R_2^3) in $\bigcup_{\sigma \in \mathcal{S}_4} \mathscr{R}_{\sigma}^{TIN}$ are achievable under some labelings. Consider three sequences of codes, achieving (R_1^1, R_2^1), (R_1^2, R_2^2), and (R_1^3, R_2^3), respectively. For each block length n, consider the $(2^{\alpha_1 n R_1^1}, 2^{\alpha_1 n R_2^1}, \alpha_1 n)$, $(2^{\alpha_2 n R_1^2}, 2^{\alpha_2 n R_2^2}, \alpha_2 n)$, and $(2^{\alpha_3 n R_1^3}, 2^{\alpha_3 n R_2^3}, \alpha_3 n)$ codes from the given three sequences of codes, respectively. By a standard time-sharing argument [19], a fourth $(2^{nR_1}, 2^{nR_2}, n)$ code can be constructed from the above three codes. Thus, any point $(R_1, R_2) \in \mathscr{R}_{\mathcal{S}_4}^{TIN}$ can be achieved. In a similar manner, one can show that any point $(R_1, R_2) \in \mathscr{R}_{\mathcal{S}_4}^{SC}$ is achievable. ■

Moreover, for the early-stage P/E cycling model in Table 7.1, the rate region $\mathscr{R}_{\mathcal{S}_4}^{SC}$ can be determined explicitly.

Theorem 7.3.11. *For the early-stage P/E cycling model, $\mathscr{R}_{\mathcal{S}_4}^{SC}$ is the set of all pairs (R_1, R_2) such that $R_1 \leq 1$, $R_2 \leq 1$, and $R_1 + R_2 \leq I(X_1, X_2; Y) = 1 + \lambda_5$.*

Proof. Using Table 7.2, Lemma 7.3.8, and Lemma 7.3.9, we can calculate the convex hull of the uniform rate regions of all 24 labelings. We can also see that the convex hull of the uniform rate regions of two labelings $(11,10,01,00)$ and $(11,01,10,00)$ are enough to achieve $\mathscr{R}_{\mathcal{S}_4}^{SC}$. ■

Example 7.3.3. For the early-stage of P/E cycling, let $a_1 = 0.98$, $b_1 = 0.97$, and $c_1 = 0.99$. The uniform rate regions $\mathscr{R}_{\mathcal{S}_4}^{TIN}$, $\mathscr{R}_{\mathcal{S}_4}^{SC}$, and $\mathscr{R}_{G}^{DS}$ are plotted in Figure 7.2. It can be seen that $\mathscr{R}_{G}^{DS} \subset \mathscr{R}_{\mathcal{S}_4}^{TIN} \subset \mathscr{R}_{\mathcal{S}_4}^{SC}$, and the line connecting the two corner points in $\mathscr{R}_{\mathcal{S}_4}^{TIN}$ is on the line connecting the two corner points in $\mathscr{R}_{\mathcal{S}_4}^{SC}$. Moreover, either R_1 or R_2 can achieve rate 1 with SC decoding. For the late-stage P/E cycling model, let $\hat{a}_1 = 0.82$, $\hat{a}_2 = 0.1$, $\hat{b}_1 = 0.85$, and $\hat{c}_1 = 0.85$. The uniform rate regions $\mathscr{R}_{\mathcal{S}_4}^{TIN}$, $\mathscr{R}_{\mathcal{S}_4}^{SC}$, and $\mathscr{R}_{G}^{DS}$ are plotted in Figure 7.3. For this case, there is a gap between the line connecting the two corner points in $\mathscr{R}_{\mathcal{S}_4}^{TIN}$ and the one connecting the two corner points in $\mathscr{R}_{\mathcal{S}_4}^{SC}$. This property implies that SC decoding will give a larger sum rate than TIN decoding. □

Remark 7.3.4. Although we focus on the P/E cycling and data retention models, our analysis can be

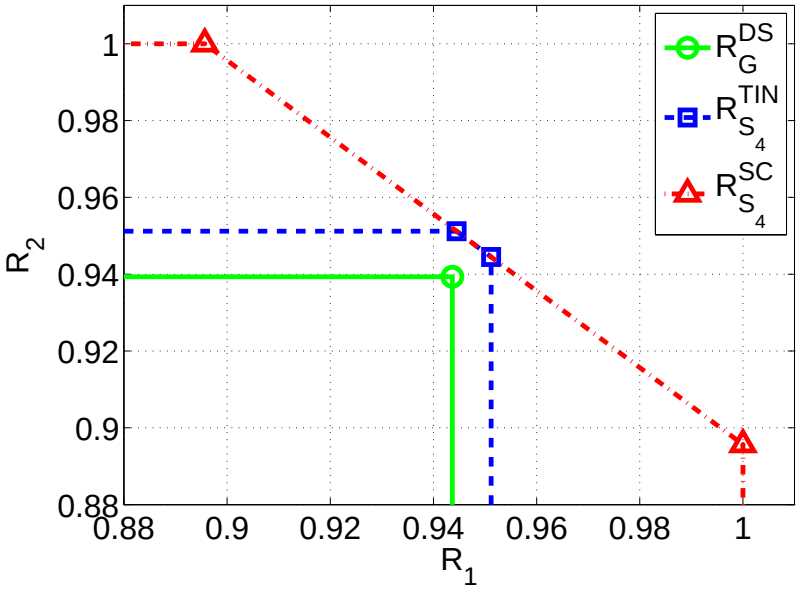
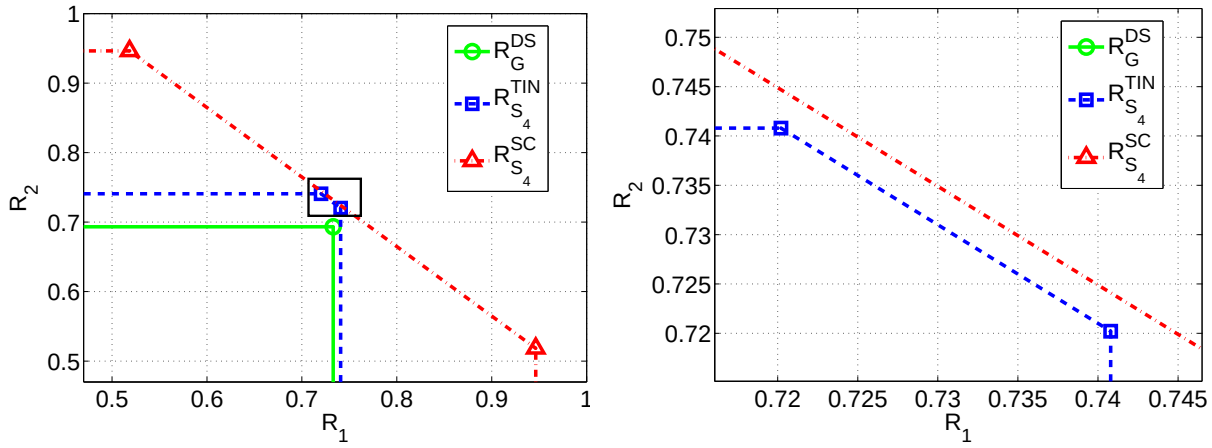


Figure 7.2: Uniform rate regions $\mathscr{R}_{\mathcal{S}_4}^{TIN}$, $\mathscr{R}_{\mathcal{S}_4}^{SC}$, and $\mathscr{R}_{G}^{DS}$ with $a_1 = 0.98$, $b_1 = 0.97$, and $c_1 = 0.99$ for the early-stage P/E cycling model.



(a) (b)

Figure 7.3: (a) Uniform rate regions $\mathscr{R}_{\mathcal{S}_4}^{TIN}$, $\mathscr{R}_{\mathcal{S}_4}^{SC}$, and $\mathscr{R}_{G}^{DS}$ with $\hat{a}_1 = 0.82$, $\hat{a}_2 = 0.1$, $\hat{b}_1 = 0.85$, and $\hat{c}_1 = 0.85$ for the late-stage P/E cycling model, where the two curves (blue and red) in the black rectangle are enlarged and shown in (b).

extended to other channel models. As a simple example, we consider a channel model whose channel transition matrix $p_{MLC}^{C}(y|v)$ reflects both upward and downward drift of voltage levels due to the combined effects of P/E cycling and data retention. The structure of the transition matrix is shown in Table 7.6, where $\bar{a}_1$, $1-\bar{a}_1$, $\bar{b}_1$, $\bar{b}_2$, $1-\bar{b}_1-\bar{b}_2$, $\bar{c}_1$, $\bar{c}_2$, $1-\bar{c}_1-\bar{c}_2$, $\bar{d}_1$, and $1-\bar{d}_1$ represent nonzero probabilities. From

Table 7.6: Channel transition matrix $p^C_{MLC}(y|v)$ for combined effects of P/E cycling and data retention for MLC flash memories.

V	Inputs: (X_1, X_2)			Output: Y			
Levels	Gray	NO	EO	s_0	s_1	s_2	s_3
A_0	(11)	(11)	(11)	$\bar{a}_1$	$1-\bar{a}_1$	0	0
A_1	(10)	(10)	(00)	$\bar{b}_2$	$\bar{b}_1$	$1-\bar{b}_1-\bar{b}_2$	0
A_2	(00)	(01)	(01)	0	$\bar{c}_2$	$\bar{c}_1$	$1-\bar{c}_1-\bar{c}_2$
A_3	(01)	(00)	(10)	0	0	$1-\bar{d}_1$	$\bar{d}_1$

Theorem 7.2.1, we readily conclude that $r_s^{TIN} < r_s^{SC}$ with any of the 24 possible labelings since, for any labeling, three of the probabilities p_{0,s_1}, p_{1,s_1}, p_{2,s_1}, and p_{3,s_1} are positive. □

7.4 Performance Improvement with Increased Number of Reads

In the previous analysis, we used a quantizer with three read thresholds. We now investigate the improvement in performance that can be obtained by applying additional reads to obtain more refined soft information.

For a channel $\mathcal{W}_{MLC}$, assume there is an output set $\mathcal{Y}^q_{MLC} = \{s_0, s_1, \ldots, s_{q-1}\}$ obtained by a set of $q-1$ reads. Denote the corresponding uniform rate regions for TIN and SC decodings by $\mathscr{R}^{TIN}$ and $\mathscr{R}^{SC}$, and the sum rates for TIN and SC decodings by r_s^{TIN} and r_s^{SC}.

Now, introduce one more read threshold to split one of the outputs $s_0, s_1, \ldots, s_{q-1}$. Without loss of generality, we split s_0 into s_0^1 and s_0^2 to obtain a new output set $\hat{\mathcal{Y}}^{q+1}_{MLC} = \{s_0^1, s_0^2, s_1, s_2, \ldots, s_{q-1}\}$. The resulting uniform rate regions for TIN and SC decodings are denoted by $\hat{\mathscr{R}}^{TIN}$ and $\hat{\mathscr{R}}^{SC}$, respectively, and the corresponding sum rates by $\hat{r}_s^{TIN}$ and $\hat{r}_s^{SC}$, respectively. The following lemma shows that for both TIN and SC decodings, one-step progressive quantization produces rate regions that contain the original rate regions. Thus the sum rates are not decreased, and in fact they become strictly larger except when the transition probabilities satisfy very specific conditions after quantization.

Lemma 7.4.1. *For uniform rate regions, under TIN and SC decodings, $\mathscr{R}^{TIN} \subseteq \hat{\mathscr{R}}^{TIN}$ and $\mathscr{R}^{SC} \subseteq \hat{\mathscr{R}}^{SC}$. For sum rates, under TIN and SC decodings, $r_s^{TIN} \leq \hat{r}_s^{TIN}$ and $r_s^{SC} \leq \hat{r}_s^{SC}$.*

Proof. We first prove that the sum rate $\hat{r}_s^{SC} \geq r_s^{SC}$, i.e., $I(X_1, X_2; \hat{Y}) \geq I(X_1, X_2; Y)$, and give the condition

when the equality holds. The value $I(X_1, X_2; Y)$ can be expressed as follows:

$$\begin{aligned} I(X_1, X_2; Y) =& H(Y) - H(Y|X_1, X_2) \\ =& H\Big(\frac{\sum_{i=0}^{3} p_{i,s_0}}{4}, \frac{\sum_{i=0}^{3} p_{i,s_1}}{4}, \ldots, \frac{\sum_{i=0}^{3} p_{i,s_{q-1}}}{4}\Big) - \frac{1}{4}\sum_{i=0}^{3} H(p_{i,s_0}, p_{i,s_1}, \ldots, p_{i,s_{q-1}}). \end{aligned}$$

Similarly, the value $I(X_1, X_2; \hat{Y})$ is

$$\begin{aligned} & I(X_1, X_2; \hat{Y}) \\ =& H(\hat{Y}) - H(\hat{Y}|X_1, X_2) \\ =& H\Big(\frac{\sum_{i=0}^{3} p_{i,s_0^1}}{4}, \frac{\sum_{i=0}^{3} p_{i,s_0^2}}{4}, \frac{\sum_{i=0}^{3} p_{i,s_1}}{4}, \ldots, \frac{\sum_{i=0}^{3} p_{i,s_{q-1}}}{4}\Big) - \frac{1}{4}\sum_{i=0}^{3} H(p_{i,s_0^1}, p_{i,s_0^2}, p_{i,s_1}, \ldots, p_{i,s_{q-1}}) \\ \overset{(a)}{=}& H\Big(\frac{\sum_{i=0}^{3} p_{i,s_0}}{4}, \frac{\sum_{i=0}^{3} p_{i,s_1}}{4}, \ldots, \frac{\sum_{i=0}^{3} p_{i,s_{q-1}}}{4}\Big) + \frac{\sum_{i=0}^{3} p_{i,s_0}}{4} H\Big(\frac{\sum_{i=0}^{3} p_{i,s_0^1}}{\sum_{i=0}^{3} p_{i,s_0}}, \frac{\sum_{i=0}^{3} p_{i,s_0^2}}{\sum_{i=0}^{3} p_{i,s_0}}\Big) \\ & - \frac{1}{4}\sum_{i=0}^{3} H(p_{i,s_0}, p_{i,s_1}, \ldots, p_{i,s_{q-1}}) - \frac{1}{4}\sum_{i=0}^{3} p_{i,s_0} H\Big(\frac{p_{i,s_0^1}}{p_{i,s_0}}, \frac{p_{i,s_0^2}}{p_{i,s_0}}\Big), \end{aligned}$$

where step (a) is from the grouping property of entropy and $p_{i,s_0} = p_{i,s_0^1} + p_{i,s_0^2}$ for $i = 0, 1, 2, 3$.

The difference between $I(X_1, X_2; \hat{Y})$ and $I(X_1, X_2; Y)$ is

$$\begin{aligned} & I(X_1, X_2; \hat{Y}) - I(X_1, X_2; Y) \\ =& \frac{\sum_{i=0}^{3} p_{i,s_0}}{4} H\Big(\frac{\sum_{i=0}^{3} p_{i,s_0^1}}{\sum_{i=0}^{3} p_{i,s_0}}, \frac{\sum_{i=0}^{3} p_{i,s_0^2}}{\sum_{i=0}^{3} p_{i,s_0}}\Big) - \frac{1}{4}\sum_{i=0}^{3} p_{i,s_0} H\Big(\frac{p_{i,s_0^1}}{p_{i,s_0}}, \frac{p_{i,s_0^2}}{p_{i,s_0}}\Big) \\ =& \frac{\sum_{i=0}^{3} p_{i,s_0}}{4}\Big(- \frac{1}{\sum_{i=0}^{3} p_{i,s_0}}\big(f(\sum_{i=0}^{3} p_{i,s_0^1}) + f(\sum_{i=0}^{3} p_{i,s_0^2})\big) + \log_2(\sum_{i=0}^{3} p_{i,s_0})\Big) \\ & - \frac{1}{4}\sum_{i=0}^{3} p_{i,s_0}\Big(- \frac{1}{p_{i,s_0}}\big(f(p_{i,s_0^1}) + f(p_{i,s_0^2})\big) + \log_2(p_{i,s_0})\Big) \\ =& \frac{1}{4}\Big(f(\sum_{i=0}^{3} p_{i,s_0}) - f(\sum_{i=0}^{3} p_{i,s_0^1}) - f(\sum_{i=0}^{3} p_{i,s_0^2})\Big) - \frac{1}{4}\sum_{i=0}^{3}\Big(f(p_{i,s_0}) - f(p_{i,s_0^1}) - f(p_{i,s_0^2})\Big). \end{aligned}$$

Note that the difference is only related to the probabilities p_{i,s_0}, p_{i,s_0^1}, and p_{i,s_0^2} for $i = 0, 1, 2, 3$.

To prove that $I(X_1, X_2; \hat{Y}) \geq I(X_1, X_2; Y)$, we define a new function $g(u_1, u_2) = f(u_1 + u_2) - f(u_1) - f(u_2)$ and utilize its properties. We first prove $g(u_1 + v_1, u_2 + v_2) \geq g(u_1, u_2) + g(v_1, v_2)$ as

follows:

$$
\begin{aligned}
&g(u_1+v_1,u_2+v_2)-\Big(g(u_1,u_2)+g(v_1,v_2)\Big)\\
=&(u_1+u_2)\log_2(1+\frac{v_1+v_2}{u_1+u_2})+(v_1+v_2)\log_2(1+\frac{u_1+u_2}{v_1+v_2})\\
&-\Big(u_1\log_2(1+\frac{v_1}{u_1})+v_1\log_2(1+\frac{u_1}{v_1})\Big)-\Big(u_2\log_2(1+\frac{v_2}{u_2})+v_2\log_2(1+\frac{u_2}{v_2})\Big).
\end{aligned}
$$

The function $t\log_2(1+1/t)$ is concave. Let $t_1=\frac{u_1}{v_1}$, $t_2=\frac{u_2}{v_2}$, $r_1=\frac{v_1}{v_1+v_2}$, and $r_2=\frac{v_2}{v_1+v_2}$. We have $(r_1t_1+r_2t_2)\log_2\Big(1+1/(r_1t_1+r_2t_2)\Big)\geq r_1t_1\log_2(1+1/t_1)+r_2t_2\log_2(1+1/t_2)$; that is, $(u_1+u_2)\log_2(1+\frac{v_1+v_2}{u_1+u_2})\geq u_1\log_2(1+\frac{v_1}{u_1})+u_2\log_2(1+\frac{v_2}{u_2})$. Similarly, $(v_1+v_2)\log_2(1+\frac{u_1+u_2}{v_1+v_2})\geq v_1\log_2(1+\frac{u_1}{v_1})+v_2\log_2(1+\frac{u_2}{v_2})$. Therefore, $g(u_1+v_1,u_2+v_2)\geq g(u_1,u_2)+g(v_1,v_2)$, where equality holds if and only if $\frac{u_1}{u_2}=\frac{v_1}{v_2}$.

Now, we apply $g(u_1+v_1,u_2+v_2)\geq g(u_1,u_2)+g(v_1,v_2)$ twice and have

$$
\begin{aligned}
&f(\sum_{i=0}^{3}p_{i,s_0})-f(\sum_{i=0}^{3}p_{i,s_0^1})-f(\sum_{i=0}^{3}p_{i,s_0^2})\\
=&g(p_{0,s_0^1}+p_{1,s_0^1}+p_{2,s_0^1}+p_{3,s_0^1},p_{0,s_0^2}+p_{1,s_0^2}+p_{2,s_0^2}+p_{3,s_0^2})\\
\geq&g(p_{0,s_0^1}+p_{1,s_0^1},p_{0,s_0^2}+p_{1,s_0^2})+g(p_{2,s_0^1}+p_{3,s_0^1},p_{2,s_0^2}+p_{3,s_0^2})\\
\geq&g(p_{0,s_0^1},p_{0,s_0^2})+g(p_{1,s_0^1},p_{1,s_0^2})+g(p_{2,s_0^1},p_{2,s_0^2})+g(p_{3,s_0^1},p_{3,s_0^2})\\
=&\sum_{i=0}^{3}\Big(f(p_{i,s_0})-f(p_{i,s_0^1})-f(p_{i,s_0^2})\Big),
\end{aligned}
$$

where equality holds if and only if $\frac{p_{0,s_0^1}}{p_{0,s_0^2}}=\frac{p_{1,s_0^1}}{p_{1,s_0^2}}=\frac{p_{2,s_0^1}}{p_{2,s_0^2}}=\frac{p_{3,s_0^1}}{p_{3,s_0^2}}$. Thus, we have proved $I(X_1,X_2;\hat{Y})\geq I(X_1,X_2;Y)$.

With the same proof technique, we can prove inequalities $I(X_1;\hat{Y})\geq I(X_1;Y)$, $I(X_2;\hat{Y})\geq I(X_2;Y)$, $I(X_1;\hat{Y}|X_2)\geq I(X_1;Y|X_2)$, and $I(X_2;\hat{Y}|X_1)\geq I(X_2;Y|X_1)$. These inequalities lead to $r_s^{TIN}\leq\hat{r}_s^{TIN}$, $\mathscr{R}^{TIN}\subseteq\hat{\mathscr{R}}^{TIN}$, and $\mathscr{R}^{SC}\subseteq\hat{\mathscr{R}}^{SC}$. ■

In the following example, we show by means of computer simulation that the performance of MLC flash can be improved through the use of additional reads.

Example 7.4.1. Following [56], we assume that the readback cell voltage has the normal-Laplace distribu-

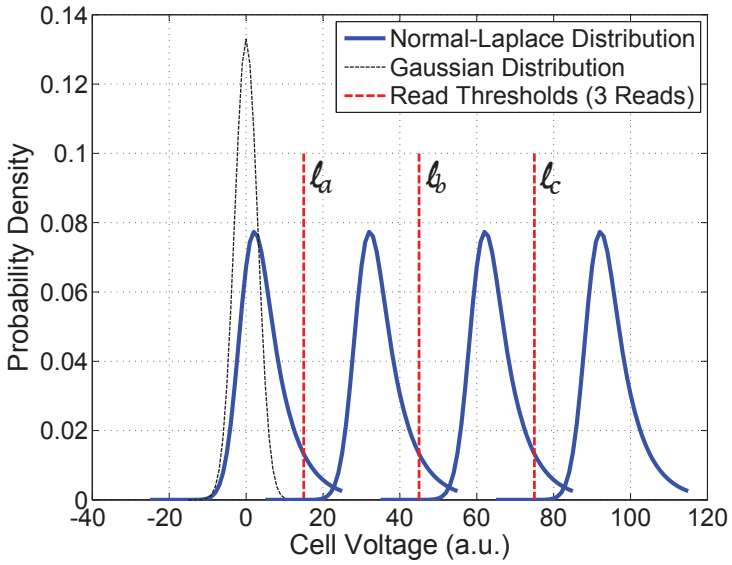


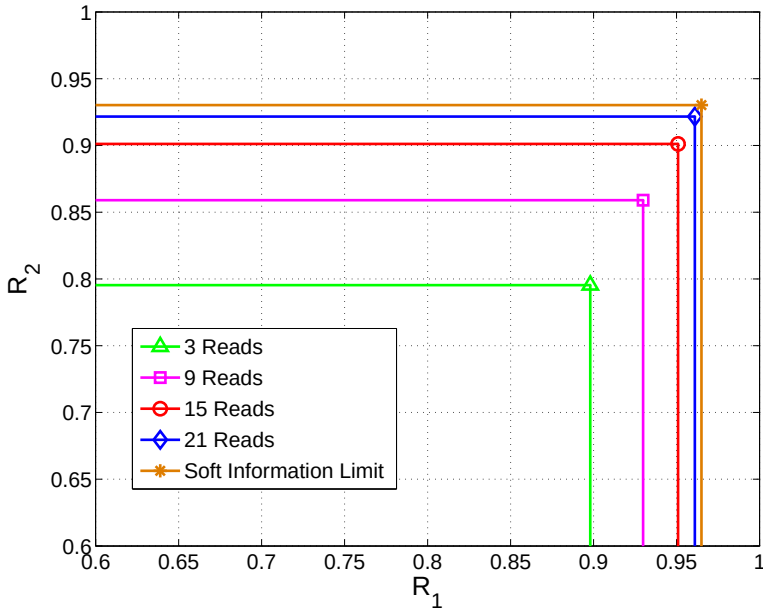
Figure 7.4: Channel model for MLC flash memories with cell voltage modeled as the normal-Laplace distribution.

tion $\mathcal{NL}(\mu,\nu,\alpha,\beta)$. The corresponding cumulative distribution function (cdf) for all real y is

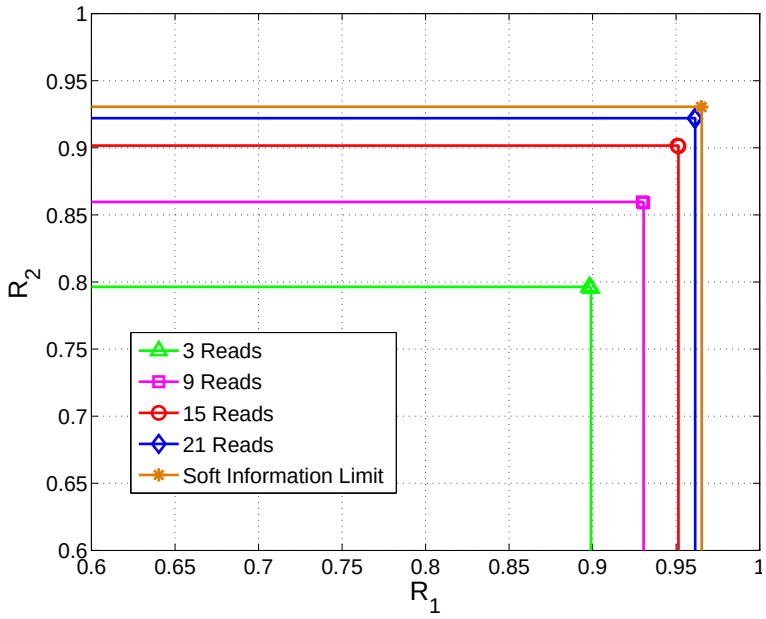
$$F(y) = \Phi\left(\frac{y-\mu}{\nu}\right) - \left(\phi\left(\frac{y-\mu}{\nu}\right) \cdot \frac{\beta\mathfrak{R}(\alpha\nu - (y-\mu)/\nu) - \alpha\mathfrak{R}(\beta\nu + (y-\mu)/\nu)}{\alpha+\beta}\right),$$

where Φ and ϕ are the cdf and probability density function (pdf) of a standard normal random variable and $\mathfrak{R}$ is Mills' ratio $\mathfrak{R}(z) = \frac{1-\Phi(z)}{\phi(z)}$ [62]. The readback cell voltage distributions for inputs A_0, A_1, A_2, and A_3 are defined to be $\mathcal{NL}(0,3,1/6,1)$, $\mathcal{NL}(30,3,1/6,1)$, $\mathcal{NL}(60,3,1/6,1)$, and $\mathcal{NL}(90,3,1/6,1)$, respectively, as shown in Figure 7.4. Here, the parameters of the normal-Laplace distributions are chosen to qualitatively reflect the distributions reported in the literature and to illustrate the effect of output quantization using multiple reads. The cell voltage is in arbitrary units (a.u.), chosen merely for convenience.

In the standard 3-read setting, the read thresholds are placed at positions $\ell_a = 15$, $\ell_b = 45$, and $\ell_c = 75$. We use a vector to represent these positions, $\vec{L}_3 = (15,45,75)$. On either side of each of the positions ℓ_a, ℓ_b, and ℓ_c, we define additional t read positions, regularly spaced at intervals of d units. For example, setting $t = 1$ and $d = 3$, we specify a total of 9 reads centered at ℓ_a, ℓ_b, and ℓ_c, with resulting



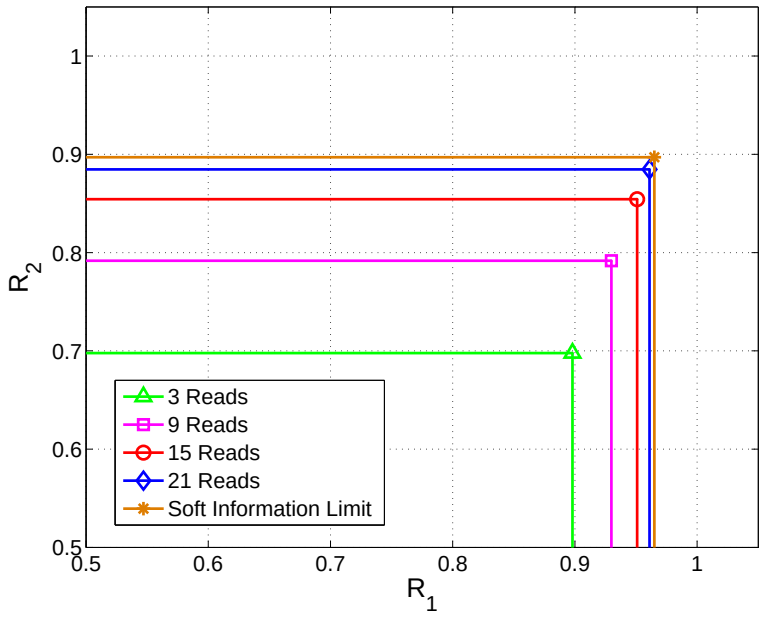
(a)



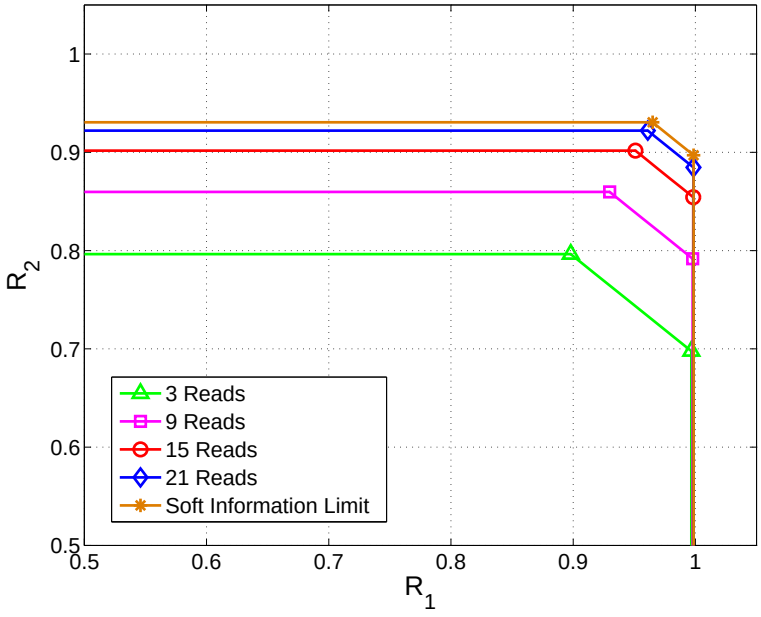
(b)

Figure 7.5: Uniform rate regions under Gray labeling with different number of reads: (a) using TIN decoding, and (b) using SC decoding.

read position vector $\vec{L}_9 = (12, 15, 18, 42, 45, 48, 72, 75, 78)$. Similarly, for a total of 15 reads, we set $t = 2$ and $d = 3$, and for a total of 21 reads, we set $t = 3$ and $d = 3$.



(a)



(b)

Figure 7.6: Uniform rate regions under NO labeling with different number of reads: (a) using TIN decoding, and (b) using SC decoding.

For the Gray labeling, the uniform rate regions under TIN and SC decodings are plotted in Figure 7.5. As expected, the rate regions of TIN decoding and SC decoding are similar. Additional reads

can significantly improve the rates of both lower and upper pages. For the NO labeling, the uniform rate regions under TIN and SC decodings are plotted in Figure 7.6. With either TIN decoding or SC decoding, additional reads effectively enhance the rate of the upper page. However, with SC decoding, the rate improvement of the lower page is very limited. □

7.5 Conclusion

We analyzed the performance of MLC flash memories with different decoding schemes and cell voltage labelings from a multi-user perspective. We showed that both TIN and SC decodings outperform the current default decoding scheme in terms of both uniform rate region and sum rate. For the P/E cycling model, with TIN decoding, we found that 8 labelings, including the standard Gray labeling, offer the largest sum rate among all 24 possible labelings. The sum rate of TIN decoding under Gray labeling equals that of SC decoding at the early-stage of P/E cycling, and is smaller than but close to that of SC decoding at the late-stage of P/E cycling. It was also shown that additional read thresholds can effectively enhance the rate region and sum rate.

Acknowledgement

This chapter is in part a reprint of the material in the paper: Pengfei Huang, Paul H. Siegel, and Eitan Yaakobi, "Performance of multilevel flash memories with different binary labelings: A multi-user perspective," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 9, pp. 2336–2353, Sept. 2016. The dissertation author was the primary investigator and author of this paper.

Bibliography

[1] K. A. Abdel-Ghaffar and M. Hassner, "Multilevel error-control codes for data storage channels," *IEEE Trans. Inf. Theory*, vol. 37, no. 3, pp. 735–741, May 1991.

[2] R. Ahlswede, H. K. Aydinian, and L. H. Khachatrian, "On perfect codes and related concepts," *Designs, Codes and Cryptography*, vol. 22, no. 3, pp. 221–237, Jan. 2001.

[3] W. Alltop, "A method for extending binary linear codes," *IEEE Trans. Inf. Theory*, vol. 30, no. 6, pp. 871–872, Nov. 1984.

[4] E. Arikan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Trans. Inf. Theory*, vol. 55, no. 7, pp. 3051–3073, July 2009.

[5] M. Asadi, X. Huang, A. Kavcic, and N. P. Santhanam, "Optimal detector for multilevel NAND flash memory channels with intercell interference," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 5, pp. 825–835, May 2014.

[6] M. Asteris and A. Dimakis, "Repairable fountain codes," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 5, pp. 1037–1047, May 2014.

[7] A. Barg, I. Tamo, and S. Vlăduţ, "Locally recoverable codes on algebraic curves," *IEEE Trans. Inf. Theory*, vol. 63, no. 8, pp. 4928–4939, Aug. 2017.

[8] R. Bez, E. Camerlenghi, A. Modelli, and A. Visconti, "Introduction to flash memory," *Proceedings of the IEEE*, vol. 91, no. 4, pp. 489–502, Apr. 2003.

[9] M. Blaum, J. Brady, J. Bruck, and J. Menon, "EVENODD: An efficient scheme for tolerating double disk failures in RAID architectures," *IEEE Trans. Comput.*, vol. 44, no. 2, pp. 192–202, Feb. 1995.

[10] M. Blaum, J. L. Hafner, and S. Hetzler, "Partial-MDS codes and their application to RAID type of architectures," *IEEE Trans. Inf. Theory*, vol. 59, no. 7, pp. 4510–4519, July 2013.

[11] M. Blaum and S. R. Hetzler, "Integrated interleaved codes as locally recoverable codes: Properties and performance," *International Journal of Information and Coding Theory*, vol. 3, no. 4, pp. 324–344, 2016.

[12] M. Bossert, H. Grießer, J. Maucher, and V. V. Zyablov, "Some results on generalized concatenation of block codes," in *Proc. Springer International Symposium on Applied Algebra, Algebraic Algorithms, and Error-Correcting Codes*, 1999, pp. 181–190.

[13] V. Cadambe and A. Mazumdar, "An upper bound on the size of locally recoverable codes," in *Proc. IEEE Int. Symp. Netw. Coding (NetCod)*, June 2013, pp. 1–5.

[14] Y. Cai, S. Ghose, E. F. Haratsch, Y. Luo, and O. Mutlu, "Error characterization, mitigation, and recovery in flash-memory-based solid-state drives," *Proceedings of the IEEE*, vol. 105, no. 9, pp. 1666–1704, Sept. 2017.

[15] Y. Cai, E. F. Haratsch, O. Mutlu, and K. Mai, "Error patterns in MLC NAND flash memory: Measurement, characterization, and analysis," in *Proc. Design, Automation & Test in Europe Conference & Exhibition (DATE)*, Mar. 2012, pp. 521–526.

[16] ——, "Threshold voltage distribution in MLC NAND flash memory: Characterization, analysis, and modeling," in *Proc. Design, Automation & Test in Europe Conference & Exhibition (DATE)*, Mar. 2013, pp. 1285–1290.

[17] G. Calis and O. O. Koyluoglu, "A general construction for PMDS codes," *IEEE Communications Letters*, vol. 21, no. 3, pp. 452–455, Mar. 2017.

[18] C. M. Compagnoni, M. Ghidotti, A. L. Lacaita, A. S. Spinelli, and A. Visconti, "Random telegraph noise effect on the programmed threshold-voltage distribution of flash memories," *IEEE Electron Device Letters*, vol. 30, no. 9, pp. 984–986, Sept. 2009.

[19] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Hoboken, New Jersey: John Wiley & Sons, 2006.

[20] G. I. Davida and S. M. Reddy, "Forward-error correction with decision feedback," *Elsevier Information and Control*, vol. 21, no. 2, pp. 117–133, Sept. 1972.

[21] G. Dong, S. Li, and T. Zhang, "Using data postcompensation and predistortion to tolerate cell-to-cell interference in MLC NAND flash memory," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 57, no. 10, pp. 2718–2728, Oct. 2010.

[22] G. Dong, N. Xie, and T. Zhang, "On the use of soft-decision error-correction codes in NAND flash memory," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 58, no. 2, pp. 429–439, Feb. 2011.

[23] A. El Gamal and Y.-H. Kim, *Network Information Theory*. New York: Cambridge University Press, 2011.

[24] M. El-Khamy, J. Hou, and N. Bhushan, "Design of rate-compatible structured LDPC codes for hybrid ARQ applications," *IEEE J. Sel. Areas Commun.*, vol. 27, no. 6, pp. 965–973, Aug. 2009.

[25] G. D. Forney, *Concatenated Codes*. Cambridge, MA: MIT Press, 1966.

[26] E. M. Gabidulin, "Theory of codes with maximum rank distance," *Problems of Information Transmission*, vol. 21, no. 1, pp. 1–12, 1985.

[27] R. Gabrys, E. Yaakobi, M. Blaum, and P. H. Siegel, "Constructions of partial MDS codes over small fields," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, June 2017, pp. 1–5.

[28] G. A. Gibson, *Redundant Disk Arrays: Reliable, Parallel Secondary Storage*. Cambridge, MA: MIT Press, 1992.

[29] N. Goela, S. B. Korada, and M. Gastpar, "On LP decoding of polar codes," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Aug.–Sept. 2010, pp. 1–5.

[30] P. Gopalan, C. Huang, H. Simitci, and S. Yekhanin, "On the locality of codeword symbols," *IEEE Trans. Inf. Theory*, vol. 58, no. 11, pp. 6925–6934, Nov. 2012.

[31] S. Goparaju and R. Calderbank, "Binary cyclic codes that are locally repairable," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, June–July 2014, pp. 676–680.

[32] J. Ha, J. Kim, and S. W. McLaughlin, "Rate-compatible puncturing of low-density parity-check codes," *IEEE Trans. Inf. Theory*, vol. 50, no. 11, pp. 2824–2836, Nov. 2004.

[33] J. Hagenauer, "Rate-compatible punctured convolutional codes (RCPC codes) and their applications," *IEEE Trans. Commun.*, vol. 36, no. 4, pp. 389–400, Apr. 1988.

[34] R. W. Hamming, "Error detecting and error correcting codes," *Bell Syst. Tech. J.*, vol. 29, pp. 147–160, 1950.

[35] J. Han and L. A. Lastras-Montano, "Reliable memories with subline accesses," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, June 2007, pp. 2531–2535.

[36] M. Hassner, K. Abdel-Ghaffar, A. Patel, R. Koetter, and B. Trager, "Integrated interleaving – a novel ECC architecture," *IEEE Trans. Magn.*, vol. 37, no. 2, pp. 773–775, Mar. 2001.

[37] S.-N. Hong, D. Hui, and I. Marić, "Capacity-achieving rate-compatible polar codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, July 2016, pp. 41–45.

[38] C. Huang, M. Chen, and J. Li, "Pyramid codes: Flexible schemes to trade space for access efficiency in reliable data storage systems," in *Proc. 6th IEEE Int. Symp. Netw. Comput. Appl.*, July 2007, pp. 79–86.

[39] C. Huang, H. Simitci, Y. Xu, A. Ogus, B. Calder, P. Gopalan, J. Li, and S. Yekhanin, "Erasure coding in Windows Azure Storage," in *Proc. USENIX Annu. Tech. Conf.*, June 2012, pp. 15–26.

[40] H. Imai and H. Fujiya, "Generalized tensor product codes," *IEEE Trans. Inf. Theory*, vol. 27, no. 2, pp. 181–187, Mar. 1981.

[41] J. Justesen and T. Høholdt, *A Course in Error-Correcting Codes*. Zürich: European Mathematical Society, 2004.

[42] S. B. Korada, "Polar codes for channel and source coding," Ph.D. dissertation, École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland, 2009.

[43] M. Kuijper and D. Napp, "Erasure codes with simplex locality," in *Proc. Int. Symp. Mathematical Theory of Netw. System (MTNS)*, July 2014, pp. 1606–1609.

[44] B. Li, D. Tse, K. Chen, and H. Shen, "Capacity-achieving rateless polar codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, July 2016, pp. 46–50.

[45] J. Li and K. R. Narayanan, "Rate-compatible low density parity check codes for capacity-approaching ARQ schemes in packet data communications," in *Proc. International Conference on Communications, Internet and Information Technology (CIIT)*, Nov. 2002, pp. 201–206.

[46] Q. Li, A. Jiang, and E. F. Haratsch, "Noise modeling and capacity analysis for NAND flash memories," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, June 2014, pp. 2262–2266.

[47] S. Lin and D. J. Costello, *Error Control Coding*. Upper Saddle River, NJ: Prentice Hall, 2004.

[48] R. Liu, P. Spasojevic, and E. Soijanin, "Punctured turbo code ensembles," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Mar.–Apr. 2003, pp. 249–252.

[49] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*. New York: Elsevier, 1977.

[50] J. L. Massey, *Threshold Decoding*. Cambridge, MA: MIT Press, 1963.

[51] J. Maucher, V. V. Zyablov, and M. Bossert, "On the equivalence of generalized concatenated codes and generalized error location codes," *IEEE Trans. Inf. Theory*, vol. 46, no. 2, pp. 642–649, Mar. 2000.

[52] J. Moon, J. No, S. Lee, S. Kim, S. Choi, and Y. Song, "Statistical characterization of noise and interference in NAND flash memory," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 60, no. 8, pp. 2153–2164, Aug. 2013.

[53] F. Oggier and A. Datta, "Self-repairing homomorphic codes for distributed storage systems," in *Proc. IEEE International Conference on Computer Communications (INFOCOM)*, Apr. 2011, pp. 1215–1223.

[54] L. Pamies-Juarez, H. D. Hollmann, and F. Oggier, "Locally repairable codes with multiple repair alternatives," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, July 2013, pp. 892–896.

[55] D. S. Papailiopoulos and A. G. Dimakis, "Locally repairable codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, July 2012, pp. 2771–2775.

[56] T. Parnell, N. Papandreou, T. Mittelholzer, and H. Pozidis, "Modelling of the threshold voltage distributions of sub-20nm NAND flash memory," in *Proc. IEEE Global Communications Conference (GLOBECOM)*, Dec. 2014, pp. 2351–2356.

[57] A. M. Patel, "Two-level coding for error control in magnetic disk storage products," *IBM Journal of Research and Development*, vol. 33, no. 4, pp. 470–484, July 1989.

[58] N. Prakash, G. Kamath, V. Lalitha, and P. Kumar, "Optimal linear codes with a local-error-correction property," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, July 2012, pp. 2776–2780.

[59] N. Prakash, V. Lalitha, and P. Kumar, "Codes with locality for two erasures," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, June 2014, pp. 1962–1966.

[60] A. Rawat, D. Papailiopoulos, A. Dimakis, and S. Vishwanath, "Locality and availability in distributed storage," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, June 2014, pp. 681–685.

[61] A. Rawat and S. Vishwanath, "On locality in distributed storage systems," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Sept. 2012, pp. 497–501.

[62] W. J. Reed, "The normal-Laplace distribution and its relatives," in *Advances in Distribution Theory, Order Statistics, and Inference*. Boston, MA: Birkhäuser, 2006, pp. 61–74.

[63] T. Richardson and R. Urbanke, *Modern Coding Theory*. New York: Cambridge University Press, 2008.

[64] R. Roth, *Introduction to Coding Theory*. New York: Cambridge University Press, 2006.

[65] D. N. Rowitch and L. B. Milstein, "On the performance of hybrid FEC/ARQ systems using rate compatible punctured turbo (RCPT) codes," *IEEE Trans. Commun.*, vol. 48, no. 6, pp. 948–959, June 2000.

[66] W. Ryan and S. Lin, *Channel Codes: Classical and Modern*. New York: Cambridge University Press, 2009.

[67] M. Sathiamoorthy, M. Asteris, D. Papailiopoulos, A. G. Dimakis, R. Vadali, S. Chen, and D. Borthakur, "XORing elephants: Novel erasure codes for big data," in *Proc. 39th Int. Conf. on Very Large Data Bases*, Aug. 2013, pp. 325–336.

[68] W. C. Schmid and R. Schürer. (2014) MinT: Table for Linear Codes. [Online]. Available: http://mint.sbg.ac.at/index.php

[69] C. Schoeny, F. Sala, and L. Dolecek, "Analysis and coding schemes for the flash normal-Laplace mixture channel," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, June 2015, pp. 2101–2105.

[70] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, pp. 379–423 and 623–656, 1948.

[71] N. Silberstein, A. S. Rawat, O. O. Koyluoglu, and S. Vishwanath, "Optimal locally repairable codes via rank-metric codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, July 2013, pp. 1819–1823.

[72] N. Silberstein and A. Zeh, "Optimal binary locally repairable codes via anticodes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, June 2015, pp. 1247–1251.

[73] W. Song, S. H. Dau, C. Yuen, and T. Li, "Optimal locally repairable linear codes," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 5, pp. 1019–1036, May 2014.

[74] I. Tamo and A. Barg, "Bounds on locally recoverable codes with multiple recovering sets," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, June 2014, pp. 691–695.

[75] ——, "A family of optimal locally recoverable codes," *IEEE Trans. Inf. Theory*, vol. 60, no. 8, pp. 4661–4676, Aug. 2014.

[76] I. Tamo, A. Barg, S. Goparaju, and R. Calderbank, "Cyclic LRC codes and their subfield subcodes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, June 2015, pp. 1262–1266.

[77] I. Tamo, D. S. Papailiopoulos, and A. G. Dimakis, "Optimal locally repairable codes and connections to matroid theory," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, July 2013, pp. 1814–1818.

[78] X. Tang and R. Koetter, "A novel method for combining algebraic decoding and iterative processing," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, July 2006, pp. 474–478.

[79] V. Taranalli, H. Uchikawa, and P. H. Siegel, "Channel models for multi-level cell flash memories based on empirical error analysis," *IEEE Trans. Commun.*, vol. 64, no. 8, pp. 3169–3181, Aug. 2016.

[80] T. Van Nguyen, A. Nosratinia, and D. Divsalar, "The design of rate-compatible protograph LDPC codes," *IEEE Trans. Commun.*, vol. 60, no. 10, pp. 2841–2850, Oct. 2012.

[81] A. Wang and Z. Zhang, "An integer programming-based bound for locally repairable codes," *IEEE Trans. Inf. Theory*, vol. 61, no. 10, pp. 5280–5294, Oct. 2015.

[82] ——, "Repair locality with multiple erasure tolerance," *IEEE Trans. Inf. Theory*, vol. 60, no. 11, pp. 6979–6987, Nov. 2014.

[83] ——, "Achieving arbitrary locality and availability in binary codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, June 2015, pp. 1866–1870.

[84] J. Wang, T. Courtade, H. Shankar, and R. D. Wesel, "Soft information for LDPC decoding in flash: Mutual-information optimized quantization," in *Proc. IEEE Global Telecommunications Conference (GLOBECOM)*, Dec. 2011, pp. 1–6.

[85] J. Wang, K. Vakilinia, T.-Y. Chen, T. Courtade, G. Dong, T. Zhang, H. Shankar, and R. Wesel, "Enhanced precision through multiple reads for LDPC decoding in flash memories," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 5, pp. 880–891, May 2014.

[86] J. Wolf, "On codes derivable from the tensor product of check matrices," *IEEE Trans. Inf. Theory*, vol. 11, no. 2, pp. 281–284, Apr. 1965.

[87] ——, "An introduction to tensor product codes and applications to digital storage systems," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Oct. 2006, pp. 6–10.

[88] Y. Wu, "Generalized integrated interleaved codes," *IEEE Trans. Inf. Theory*, vol. 63, no. 2, pp. 1102–1119, Feb. 2017.

[89] E. Yaakobi, L. Grupp, P. H. Siegel, S. Swanson, and J. K. Wolf, "Characterization and error-correcting codes for TLC flash memories," in *Proc. IEEE International Conference on Computing, Networking and Communications (ICNC)*, Jan.–Feb. 2012, pp. 486–491.

[90] E. Yaakobi, J. Ma, L. Grupp, P. H. Siegel, S. Swanson, and J. K. Wolf, "Error characterization and coding schemes for flash memories," in *Proc. IEEE Global Communications Conference (GLOBECOM) Workshops*, Dec. 2010, pp. 1856–1860.

[91] K. Yang and P. Kumar, "On the true minimum distance of Hermitian codes," *Springer Coding Theory and Algebraic Geometry*, pp. 99–107, 1992.

[92] A. Zeh and E. Yaakobi, "Optimal linear and cyclic locally repairable codes over small fields," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Apr.–May 2015, pp. 1–5.