

UCLA

UCLA Electronic Theses and Dissertations

Title

Optimal Signaling and Rate Adaptation for Low-Density Parity-Check Codes in Satellite Communications and Flash Memory

Permalink

<https://escholarship.org/uc/item/8hr2908m>

ISBN

9798190926859

Author

Galijasevic, Semira

Publication Date

2026-09-10

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA

Los Angeles

Optimal Signaling and Rate Adaptation for Low-Density Parity-Check Codes
in Satellite Communications and Flash Memory

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Electrical and Computer Engineering

by

Semira Galijasevic

ABSTRACT OF THE DISSERTATION

Optimal Signaling and Rate Adaptation for Low-Density Parity-Check Codes
in Satellite Communications and Flash Memory

by

Semira Galijasevic

Doctor of Philosophy in Electrical and Computer Engineering

University of California, Los Angeles, 2026

Professor Richard D. Wesel, Chair

Low-density parity-check (LDPC) codes have become a cornerstone of modern communication and storage systems for their near-capacity performance and low-complexity decoding. They are used in flash memory and adopted in standards such as CCSDS, 5G NR, Wi-Fi, and DVB-S2/T2. However, LDPC-coded systems often operate under time-varying channel conditions, which can substantially degrade performance. To address this, LDPC codes supporting multiple rates are needed to adapt to varying channel conditions, along with feedback from the receiver to inform the transmitter about the channel. This dissertation introduces a family of low-rate LDPC codes that overcome the high-error floor limitations of existing low-rate CCSDS turbo codes in such conditions. To mitigate feedback delay in free-space optical channels, we propose a channel prediction scheme that forecasts channel variations so that the appropriate LDPC code rate can be selected, thereby improving throughput and reducing packet errors. In addition to adapting LDPC code rates to time-varying channels, practical system performance is strongly influenced by interaction between coding and modulation.

When LDPC codes are combined with higher-order modulation, performance degrades due to unequal bit reliabilities within each symbol and mismatch between the input distribution and the channel conditions. For example, a standard equidistant and equiprobable M-PAM signaling is suboptimal for AWGN channel, causing up to a 1.53 dB loss compared to optimal

Gaussian signaling. This can be mitigated by constellation shaping. Probabilistic amplitude shaping (PAS) is a practical scheme that combines constellation shaping with forward error correction (FEC) encoder.

This dissertation analyzes and mitigates the error floor phenomenon for LDPC codes with PAS when low-weight codewords are the dominant errors in the error-floor region. An approximate truncated union bound (TUB) characterizes the error-floor region of bit-metric decoding (BMD) in such cases. A bit interleaver optimized to lower the approximate TUB significantly reduces the error floor. Further investigation reveals that the symbol-metric decoding (SMD) TUB can be significantly lower than the corresponding approximate BMD TUB. This dissertation develops a low-complexity post-processing technique that approaches the performance of SMD and further reduces the error floor. Applying the proposed bit-interleaving and post-processing techniques to the DVB-S2 rate-2/3 LDPC code demonstrates that bit interleaving reduces the error floor by more than an order of magnitude, while the post-processing technique provides an additional order-of-magnitude reduction.

Previous examples of PAS for LDPC codes require LDPC protographs that lack the structure that facilitates low-complexity encoding. We show that equalizing the conditional entropies of the bits that label the constellation points decouples the density evolution threshold from the assignment of bits to LDPC variable nodes. This enables linear-complexity LDPC encoding while preserving the capacity-approaching benefits of PAS.

For flash memory systems this dissertation shows that joint optimization of write voltages (M-PAM signaling) and read thresholds, based on mutual information maximization, improves LDPC decoding performance and extends device lifetime.

Overall, the dissertation demonstrates that optimizing signaling, bit interleaving, coding, and rate-adaptation provides performance gains and practical design insights for communication and storage systems employing LDPC codes.

The dissertation of Semira Galijasevic is approved.

Danijela Cabric

Dariusz Divsalar

Lara Dolecek

Ian Roberts

Richard D. Wesel, Committee Chair

University of California, Los Angeles

2026

To My Husband Edin

&

My Children Ali & Imani

Contents

List of Acronyms	xii
1 Introduction	1
1.1 Summary of Contributions	2
2 Design of LDPC Codes to Support Multiple Rates	8
2.1 Introduction	8
2.2 PBRL LDPC Code Construction	12
2.3 Comparison of LDPC Codes to Turbo Codes	19
2.4 Conclusion	27
3 Adapting LDPC Code Rate to Changing Channel Conditions	29
3.1 Introduction	29
3.2 System Architecture and Fading Channel Model	35
3.3 Normal Approximation of Frame Error Rate	42
3.4 Low-Rate Protograph-based LDPC Codes Design	45
3.5 LDPC Rate Selection To Maximize Throughput	48
3.6 Conclusions	59
4 Interleaving and Post-Processing to Lower LDPC Error Floors	61
4.1 Introduction	61
4.2 Preliminaries	64
4.3 Truncated Union Bounds	71
4.4 Truncated Union Bound for the DVB-S2 Code	79
4.5 Permuting Bits to Increase the Worst-Case Distance and Lower the Error Floor	84
4.6 SMD TUB and SMD Post Processing	90
4.7 Conclusion	94
5 Constellation Design to Enable Low-Complexity Encoding	95
5.1 Introduction	95
5.2 Probabilistic Amplitude Shaping	100
5.3 Equalizing Conditional Entropies	101
5.4 Protograph-based LDPC Code Design	106
5.5 Simulation Results	107
5.6 Conclusion	108

6	Optimization of Write Voltage Levels for TLC Flash Memory	110
6.1	Introduction	110
6.2	Writing and Reading Flash Information	113
6.3	The Mutual Information Perspective	115
6.4	Maximizing the Minimum Rate	118
6.5	Conclusion	127
7	Joint Optimization of Write and Read Thresholds for MLC Flash Memory	129
7.1	Introduction	129
7.2	Flash Channel Model	132
7.3	Read Threshold Optimization	134
7.4	Write Voltage Levels that Optimize Page MI	135
7.5	Page MI vs. P/E Cycles for Three Objectives	137
7.6	Performance with LDPC codes	141
7.7	Conclusion	143
8	Conclusion	144

List of Figures

2.1	Encoder/Decoder for low-rate PBRL LDPC codes without changing the current CCSDS code.	11
2.2	The 44×52 matrix H_1 after lifting by 4 of 11×13 LDPC protograph. Dots indicate ones.	18
2.3	Turbo code encoder block diagram from CCSDS Blue Book.	20
2.4	Interpretation of Turbo Code Permutation.	21
2.5	FER as a function of E_b/N_0 of LPDC Code compared to normal approximation and turbo codes for rates $R = 1/6, 1/4, 1/3$ and message length $K = 16384$. .	23
2.6	FER as a function of E_b/N_0 of LPDC Code with message length $K = 4096$ compared to normal approximation with $K = 4096$ and turbo codes with message lengths $K = 4096$ and $K = 4072$ for rates $R = 1/6, 1/4$ and $1/3$. .	24
2.7	FER as a function of E_b/N_0 of LPDC Code compared to turbo codes and normal approximation for rates $R = 1/6, 1/4, 1/3$ and message length $K = 1024$. . .	25
2.8	FER as a function of E_b/N_0 of LPDC Code with message length $K = 1024$ compared to turbo codes with message length $K = 1016$ for rates $R = 1/6, 1/4, 1/3$	26
2.9	FER as a function of E_b/N_0 of the CCSDS LPDC Code with message length $K = 1024$ compared to the CCSDS turbo code and normal approximation with message lengths $K = 1016$ and $K = 1024$ for rate $R = 1/2$	27
3.1	Simplified block diagram illustrating round-trip delay in using feedback for rate control.	32
3.2	High-level system block diagram for predictive LDPC rate adaptation. . . .	35
3.3	Frame Error Rate (FER) Vs. Channel Gain for LDPC code rates $8/9$ to $8/77$ in descending order from right to left.	36
3.4	Fading channel gain realizations ρ_{T_i} in dB as a function of time for turbulence coherence time of a) $5\ ms$ and b) $10\ ms$	39
3.5	Autocorrelation function $R_\rho(\tau)$ for turbulence coherence time of $\tau_0 = 10\ ms$. .	40
3.6	Frame Error Rate (FER) vs Channel Gain using Normal Approximation (NA). .	44
3.7	The crossing points in Table 3.2 are compared with crossing points using normal approximation (NA).	46
3.8	Throughput as a function of time for optical channel with coherence time of $10\ ms$	49
3.9	Comparison of actual and estimated fading channel gain data for coherence time of $10\ ms$	51

3.10	Comparison of channel gain values for quadratic and linear prediction with respect to the actual fading channel gain values.	54
3.11	a) Actual and predicted channel gain values vs. time. b) Throughput vs. time obtained using zero-delay, zero-order, linear and quadratic prediction models	55
3.12	Percentage of zero-delay throughput achieved using prediction models as a function of feedback delay when coherence time is 5 <i>ms</i> and feedback delay ranges from 1 <i>ms</i> to 5 <i>ms</i>	57
3.13	Percentage of zero-delay throughput achieved using prediction models as a function of feedback delay when coherence time is 10 <i>ms</i> and feedback delay ranges from 1 <i>ms</i> to 10 <i>ms</i>	58
3.14	Comparison of zero-order, linear and quadratic predictions of channel gain value at time instance of 65 <i>ms</i> for feedback delay of 10 <i>ms</i> and coherence time of 10 <i>ms</i>	59
4.1	Gray-labeled 8-PAM constellation with labels $B_1B_2B_3$	66
4.2	Encoder for an LDPC code with probabilistic amplitude shaping.	67
4.3	Decision regions for ML and SMD decoding.	72
4.4	Subgraph of the DVB-S2 code Tanner graph induced by Hamming weight-nine codewords.	80
4.5	Approximate TUB for BMD of the DVB-S2 LDPC code for all possible bit interleavers $\pi_0(\mathbf{e}) - \pi_5(\mathbf{e})$	87
4.6	FER of DVB-S2 code with the π_0 and π_5 bit interleavers.	89
4.7	SMD and BMD TUBs for the DVB-S2 LDPC code for all possible bit interleavers.	91
4.8	FER for the DVB-S2 code with the post-processing.	93
5.1	Comparison Cumulative Distribution Functions (CDFs) for 8-PAM constellation unde different shaping configurations.	96
5.2	Probabilistic amplitude shaping (PAS) encoder.	100
5.3	Objectives $f(\cdot)$ of (5.3), $g(\cdot)$ of (5.4) and P-EXIT threshold for $\mathbf{P}_{1-LCE}$ of (5.7) as a function of weight factor α . P-EXIT threshold is influenced by both $f(\cdot)$ and $g(\cdot)$, but the range of $f(\cdot)$ is an order of magnitude smaller than the range of $g(\cdot)$. The solution at $\alpha = 0.77$ minimizes maximum conditional entropy $g(\cdot)$ achieving the minimum threshold of 7.88 dB.	103
5.4	PMF comparison for EMB and ECE signaling.	104
5.5	Conditional entropies for ECE and MB Constellation.	105
5.6	Lifted matrix $\mathbf{H}$ from protomatrix $\mathbf{P}_{1-LCE}$	107
5.7	Encoder with linear complexity in blocklength for the optimized code in Fig. 5.6	108
5.8	FER performance for $\mathbf{P}_1$, $\mathbf{P}_{1-LCE}$, and $\mathbf{P}_{2-LCE}$	109
6.1	The eight voltage levels and Gray labeling for a TLC flash memory.	113
6.2	Flash Memory Layout as Multiple Access Channel (MAC).	114
6.3	Independent MI rates for Equally Spaced Equally Likely (ESEL) Constellations as a function of SNR.	117

6.4	Upper subplot: Comparison of Information rate $I(X; Y) = I(B_1, B_2, B_3; Y)$ for joint decoding depicted by green curve and $3 \times \min\{I_1, I_2, I_3\}$ where $I_1 = I(B_1; Y)$, $I_2 = I(B_2; Y)$, $I_3 = I(B_3; Y)$ for Equally Spaced Equally Likely (ESEL) points. Lower subplot: Percentage of Mutual Information loss for ESEL points with equal rate constraint.	117
6.5	Three 8-PAM independent MI rates corresponding to bits B_1, B_2 and B_3 respectively for optimized points locations.	122
6.6	8-PAM Optimized constellations locations as a function of SNR.	123
6.7	8-PAM Independent rates for optimized constellation positions and input PMFs.	125
6.8	8-PAM Optimized constellations locations using Alg. 4 as a function of SNR.	126
6.9	Comparison of joint MI $I(B_1, B_2, B_3; Y)$ and the sum of independent rates.	127
6.10	Equal Rate Constraint Loss.	128
7.1	Flash system model with initial and enhanced precision reads.	131
7.2	Flash model at 1600 P/E Cycles	133
7.3	Plot of $I^{Initial}$ vs P/E Cycle for the MSB and LSB for each optimization technique and original mean voltages and initial read thresholds from Wong et al.	137
7.4	Plot of $I^{Enhanced}$ vs P/E Cycle for the MSB and LSB for each optimization technique and original mean voltages and initial read thresholds from Wong et al.	138
7.5	Plot of I^{Soft} vs P/E Cycles for the MSB and LSB for each optimization technique and original mean voltages and initial read thresholds from Wong et al.	139
7.6	Comparison of three MI measures for all sets of means and thresholds at 1600 P/E Cycles	140
7.7	Block diagram of independent encoding of MSB and LSB pages.	141
7.8	Block diagram of independent decoding of MSB and LSB pages.	141
7.9	FER vs. P/E Cycles for initial read decoding of MSB and LSB pages.	142
7.10	FER vs. P/E Cycles for enhanced precision decoding.	143

List of Tables

2.1	Code Block Lengths for Supported CCSDS Code Rates and Newly Designed Code Rates	14
2.2	d_{ACE} and η for different message lengths	19
2.3	Parameters k_1 and k_2 for Specified Information Block Lengths from CCSDS Blue Book.	21
3.1	Orbital distances and corresponding round-trip feedback delays of common LEO systems.	33
3.2	LDPC codes with channel gain crossing points. Each code rate achieves FER of 10^{-6} for corresponding crossing point.	37
3.3	Constants and variables used to calculate signal current μ_1 and total noise σ_1 when signal is ON, and signal current μ_0 and total noise σ_0 when signal is OFF.	41
4.1	Amplitude probabilities of 8-PAM constellation with Maxwell-Boltzmann distribution and $\mathbb{E}[X^2] = 21$	65
4.2	Definitions of PAS labeling functions $\phi(\cdot)$ for 8-PAM	66
4.3	Worst-case distances for the Gray-Labeled 8-PAM of Fig. 4.1 where $p_i = P_A(a_i)$ and $\mathcal{X}_{w.c.} = \{x : d_E(b(x), b(x) \oplus e) = d_{w.c.}(e)\}$	71
4.4	Decision boundaries λ_i for error pattern $\pi_0(\mathbf{e}) = (4, 4, 4, 4, 4, 4, 1, 2)_8$	84
4.5	Worst-case distances for all possible error patterns $\mathbf{e}$, where $p_i = P_A(a_i)$ and $\mathcal{S}_{w.c.}(\mathbf{e}) = \{\mathbf{x} : d_E(\mathbf{c}, \mathbf{c} \oplus \mathbf{e}) = d_{w.c.}(\mathbf{e})\}$	85
4.6	Decision boundaries λ_i for error pattern $\pi_5(\mathbf{e}) = (4, 4, 4, 4, 4, 5, 6)_8$	88
6.1	8-pam ESEL constellations and gray labeling for TLC flash	115

List of Acronyms

LDPC low-density parity-check

SMD symbol-metric decoding

BMD bit-metric decoding

QC quasi-cyclic

ML maximum likelihood

AWGN additive white Gaussian noise

BER bit error rate

BICM bit-interleaved coded modulation

LLR log-likelihood ratio

MB Maxwell-Boltzmann

SE spectral efficiency

FER frame error rate

SNR signal-to-noise ratio

BI-AWGN binary-input additive white Gaussian noise

TUB truncated union bound

MAP maximum a-posteriori

PAS Probabilistic amplitude shaping

PBRL Protograph-Based Raptor-Like

CCSDS Consultative Committee for Space Data Systems

FSO free-space optical

PBRL Protograph-based Raptor-like

RCA reciprocal channel approximation

PMF probability mass function

FEC forward error correction

DE density evolution

P-EXIT protograph extrinsic information transfer

ECE equal conditional entropies

CDF Cumulative Distribution Function

GS Geometric shaping

PS Probabilistic shaping

ESEL equally spaced equally likely

EMB equidistant Maxwell-Boltzmann

DAB dynamic-assignment Blahut-Arimmoto

MLC Multi-Level Cell

MSB most-significant bit

LSB least-significant bit

MI mutual information

ACE extrinsic-message-degre

Acknowledgments

My advisor Professor Richard Wesel has been an exceptional mentor, shaping not only my growth as a researcher, but also my view of what it means to be a truly dedicated professor. I got interested in his research when I took his Probability and Statistics course as an undergraduate student. Interestingly, many years later, I was fortunate to be a teaching assistant for the same course during my last quarter at UCLA as a doctoral student. A significant part of my success, and the completion of this dissertation, is due to his support, encouragement, and guidance. I am deeply grateful for everything he has done for me throughout this journey.

I was also fortunate to have Professors Dariush Divsalar, Danijela Cabric, Lara Dolecek, and Ian Roberts as my committee members. I am sincerely grateful to each one of them for their time, support and valuable feedback. I was privileged to learn from Professors Cabric, Dolecek and Roberts as a student in their classes. Their classes built the foundation of my research.

Professor Divsalar served as co-advisor on many of the projects in this dissertation, for which I am deeply grateful. Learning from an expert in LDPC coding like him made my research both enriching and rewarding.

Professor Cabric was the first professor I met at UCLA and my undergraduate advisor. I am deeply grateful for the time she has invested in me, as well as for her encouragement, guidance, and support throughout all these years. I am also thankful to Professors Cabric and Roberts for their support and for writing letters of recommendation for my next career opportunity. Their encouragement and confidence in me mean a lot as I begin the next chapter of my journey.

I would like to thank to my collaborator Dr. Gianluigi Liva whose passion for channel coding made my research more exciting and rewarding. Thank you for always bringing such positive energy to our meetings and making every discussion enjoyable and inspiring.

Many thanks to my colleagues at the Communications Systems Laboratory: Jingchao Luo, Linfang Wang, Ava Asmani, Amaael Antonini, Hengjie Yang, Bruce Qu, Wenhui Sui, Parth Pandhare, Aki Subramaniam, Jerry Huang, Kaitlyn Su, Joshua Brazao. Some of you left shortly after I joined the lab, while others joined more recently, but I am grateful for the time we shared together at UCLA. Linfang, Jingchao, and Ava, thank you especially for your collaboration. I am grateful for the meaningful work we accomplished together. Beyond the research, I am grateful for the conversations, coffee breaks, and countless small moments that made the lab feel like a community.

Prior to transferring to UCLA as an undergraduate student, I had the opportunity to

intern at the Jet Propulsion Laboratory (JPL) for an entire academic year. I would like to thank my mentor at JPL, Dr. Dragan Nikolic who recognized my potential and encouraged me to pursue a doctoral degree. Thank you for making my time at JPL a truly rewarding learning experience and for believing in me.

I was also fortunate to meet Lejla Tricic, a lecturer at CSU Fresno, who provided invaluable guidance and mentorship as I navigated the university application process. I am truly thankful for her kindness, encouragement, and support throughout my journey. What began as academic support grew into a meaningful friendship, and she and her family have become dear friends of mine.

I am incredibly grateful to my family and friends, who have been by my side throughout this journey, especially to my parents, my mother-in-law and my brother Semir. To everyone who has been part of this journey, whether mentioned here or not, thank you. I am grateful for all the support, kindness, and encouragement I have received along the way.

And finally, to my husband, Edin, and our children, Ali and Imani: there are no words that can fully express how grateful I am to have you by my side. You have been my greatest source of strength and happiness. To my children, I hope that one day you will understand how much your love, smiles, and little moments of joy meant to me during these years. And to my husband, thank you for your unconditional love, patience and understanding. Thank you for always being there for me. I could not have done this without you.

Vita

Education

B.S. in Electrical Engineering 2019 - 2021

University of California, Los Angeles

M.S. in Electrical Engineering 2021 - 2023

University of California, Los Angeles

Ph.D. in Electrical Engineering, Expected 2026 2021 - 2026

University of California, Los Angeles

Fellowship

Eugene V. Cota-Robles Fellowship 2021 - 2022

Eugene V. Cota-Robles Fellowship 2024 - 2025

Teaching Assistant Experiences

University of California, Los Angeles

ECE 102 - Signals and Systems Jan - Apr 2024

ECE 231A - Information Theory Sep - Dec 2025

ECE 132A - Introduction to Communication Systems Jan - Apr 2026

ECE 131A - Probability and Statistics Mar - Jun 2026

Publications

S. Galijasevic and R. D. Wesel, “Optimizing Write Voltages for Independent, Equal-Rate Pages in Flash Memory,” *2022 56th Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, CA, USA, 2022, pp. 168–174, doi: 10.1109/IEEECONF56349.2022.10051932.

S. Galijasevic and R. Wesel, “Optimizing Write Voltages to Achieve Equal Reliability for All Pages in Flash Memory,” *14th Annual Non-Volatile Memories Workshop*, Mar. 2023.

S. Galijasevic, J. Lou, D. Divsalar, and R. Wesel, “Effect of Feedback Delay on Adaptive

LDPC Coding in a Fading Free-Space Optical Channel,” *2024 IEEE International Conference on Communications (ICC)*, Denver, CO, USA, June 9–13, 2024.

A. Asmani, **S. Galijasevic**, and R. Wesel, “Write Voltage Optimization to Increase Flash Lifetime in a Two-Variance Gaussian Channel,” *2024 IEEE International Symposium on Information Theory (ISIT)*, Athens, Greece, July 7–12, 2024.

S. Galijasevic, L. Wang, J. Hamkins, D. Divsalar, and R. Wesel, “Construction of Low-Rate LDPC Codes from Rate-1/2 CCSDS Standard LDPC Codes,” *2025 IEEE Aerospace Conference (AERO)*, Big Sky, MT, USA, Mar. 1–8, 2025.

S. Galijasevic, J. Luo, D. Divsalar and R. Wesel, “Channel-Prediction-Driven Rate Control for LDPC Coding in a Fading FSO Channel With Delayed Feedback,” in *IEEE Open Journal of the Communications Society*, vol. 6, pp. 3320–3331, 2025, doi: 10.1109/OJCOMS.2025.3538839. **Special Issue: Selected Best Papers of IEEE ICC 2024.**

S. Galijasevic, G. Liva, D. Divsalar, and R. Wesel, “Geometric and Probabilistic Shaping to Enable LDPC Encoding with Linear Complexity,” *2025 International Symposium on Topics in Coding (ISTC)*, Los Angeles, CA, USA, Aug. 18–22, 2025. **Best Paper Award.**

Chapter 1

Introduction

Low-density parity-check (LDPC) codes [57] are essential to modern communication and data storage systems, delivering capacity-approaching performance [98] that enables high-throughput, low-latency, and reliable communication across various applications such as cellular networks, Wi-Fi, optical communication, and Flash memory [65, 99]. LDPC codes have been adopted in standards such as 5G NR [44], Digital Video Broadcasting (DVB) Satellite/Terrestrial (S2/T2) [41, 42], Optical Communications Terminal (OCT) [104], and the Consultative Committee for Space Data Systems (CCSDS) [47]. This dissertation studies LDPC encoder and decoder design as well as the interaction between the LDPC code and the underlying channel. The primary contributions include

- design of LDPC codes to support multiple rates,
- adaptation of LDPC code rate to changing channel conditions,
- interleaving and post-processing techniques to lower LDPC error floors,
- constellation design to enable low-complexity encoding,
- and threshold optimization for the Flash memory channel.

These five primary contributions are developed in detail throughout the dissertation and are presented in the corresponding chapters as follows. Chapters 2-5 describe work on LDPC codes for communication systems, while Chapters 6 and 7 focus on optimizing read and write thresholds for the Flash memory channel.

1.1 Summary of Contributions

1.1.1 Design of LDPC Codes to Support Multiple Rates

Chapter 2 presents the Protograph-based Raptor-like (PBRL) approach to develop new low-rate LDPC codes that aim to replace existing turbo codes in the CCSDS standard. The current CCSDS LDPC codes are used by the NASA Artemis program and are proposed for the Human Landing System on the Moon. The development of LDPC codes has been highly successful, particularly for high code rates where they outperform turbo codes in both performance and decoding complexity. However turbo codes remain the published solution for lower code rates in the CCSDS standard. While the CCSDS turbo codes offer good performance in the waterfall region, their overall performance is limited by a high error floor, where the frame error rate (FER) no longer improves with increasing transmission power. The error floor for the CCSDS low-rate turbo codes for rates $1/3$, $1/4$, and $1/6$ is between FERs of 10^{-3} and 10^{-5} . For future missions at much longer distances, such as Mars, lower code rates with much better performance than turbo codes are desirable. To address the issue, I designed new low-rate LDPC codes for message lengths of 1024, 4096, and 16,384 bits that do not suffer from an error floor at least above FER of 10^{-7} for which simulation was performed. The LDPC encoder for the new low-rate codes is compatible with the current CCSDS encoder, generating additional parity check bits from the output of the current CCSDS encoder using Exclusive-OR operations. Eventually, the constructed low-rate LDPC codes will be proposed to the CCSDS standard to be used for space applications. Replacing

the low-rate turbo codes with better-performing PBRL LDPC codes would allow the CCSDS codes to be used for longer distances and would also support dynamic rate adaptation for fading channels.

1.1.2 Adapting LDPC Code Rate to Changing Channel Conditions

In Chapter 3 we explore prediction models for estimating future channel state information (CSI) in fading channels and dynamically selecting an appropriate LDPC code rate. In free-space optical (FSO) communication, LDPC rate adaptation is a widely used technique that enables communication systems to adapt to varying channel conditions by selecting an appropriate code rate based on CSI feedback from the receiver. FSO communication [77] offers numerous benefits including high data rates, vast unlicensed free spectrum, high immunity to interference, highly secure links and easy installation [70,93,112]. FSO channels can be used for communications over distances of several kilometers as well as ultra-long distances including ground-to-satellite, satellite-to-satellite communications, and interplanetary communications [112]. However, FSO links are sensitive to channel fading caused by atmospheric turbulence, varying weather conditions, and changes in the distance between the transmitter and receiver. Rate-adaptive coding in FSO channels enables reliable communication while efficiently utilizing the available channel mutual information. To adapt the code rate to current channel conditions, the transmitter must sense the channel or rely on the feedback from the receiver. FSO CSI cannot be reliably estimated at the transmitter because optical scintillation caused by atmospheric turbulence may differ for signals traveling in opposite directions. In real-world FSO systems, feedback from the receiver provides CSI. Even if the receiver sends accurate CSI, the channel conditions might change by the time the feedback reaches the transmitter and the new signal travels to the receiver. While adaptive coding has been explored in previous research, the effect of feedback delay is often overlooked. To mitigate throughput performance degradation caused by feedback delay, we developed linear and quadratic predictive models

that estimate future CSI and dynamically select the appropriate low-density parity-check (LDPC) code rate. In order to facilitate convenient rate switching, we designed PBRL LDPC codes supporting a wide range of rates, from rate 1/10 to rate 8/9. The throughput performance of predictive models are evaluated as the feedback delay is increased from no delay to a delay equal to the coherence time of a fading channel. Optical channels with coherence times of 5 *ms* and 10 *ms* are explored, where the 10 *ms* channel is meant to model the optical channel of a Low Earth Orbit (LEO) satellite.

1.1.3 Interleaving and Post-Processing to Lower LDPC Error Floors

Chapter 4 presents two complementary methods, bit interleaving and a post-processing technique, that improve the error floors of low-density parity-check (LDPC) codes with probabilistic amplitude shaping (PAS).

One of the main challenges associated with LDPC codes is the error-floor phenomenon, characterized by a flattening of the FER curve in the high signal-to-noise ratio (SNR) region. This limits the performance of LDPC codes at low error rates. Even LDPC codes that exhibit strong waterfall performance can suffer from a high error floor, which limits their practical applicability. The error floor is generally attributed to a combination of low-weight codewords and the limitations of iterative decoding algorithms when applied to graphs containing short, weakly-connected cycles. Previous research has explored these two causes of error floors, studying low-weight codewords, near-codewords [85], trapping sets [25, 64, 96, 128], pseudocodewords [51], and absorbing sets [36].

This dissertation presents an approximate truncated union bound (TUB) for LDPC codes with PAS that adjusts decision thresholds to match bit-metric decoding (BMD) and applies it to the low-weight codewords. TUB tracks BMD simulation results for the DVB-S2 rate-2/3 LDPC code with an 8-PAM Maxwell-Boltzmann constellation. Analysis of the BMD TUB identifies the nearby codewords that cause the error floor and reveals the effect of constellation

labeling on Euclidean distance. This analysis also shows how interleaving can increase the Euclidean distance of these neighbors and lower the BMD TUB (and thus the error floor) by more than an order of magnitude.

While optimized bit-interleaving can significantly lower the error floor of an LDPC code, the overall performance still differs from that of SMD decoding due to the sub-optimality of the bit-metric decoder. We derived the TUB for SMD decoder showing that the error floor can be lowered by an additional order in magnitude. This chapter develops a low-complexity post-processing technique to approach SMD decoding performance. Simulation results show that the post-processing technique lowers the error floor of the DVB-S2 rate-2/3 LDPC code by an additional order of magnitude.

1.1.4 Constellation Design to Enable Low-Complexity Encoding

The increasing demand for higher data rates and spectral efficiency in modern communication systems necessitates the use of higher-order modulation schemes. For low-noise channels, spectral efficiency is improved by using PAM or QAM, and system performance can be enhanced through constellation shaping techniques. Probabilistic amplitude shaping (PAS) [18] adjusts the distribution of the systematic bits of a forward error correction (FEC) code so that the probabilities of signaling amplitudes are optimized to approach the channel capacity. Previous examples of PAS for LDPC codes use protographs that lack the structure that facilitates low-complexity encoding. In Chapter 5 we optimize the signaling positions and probability mass function (PMF), i.e. we use both geometric and probabilistic shaping, to facilitate LDPC encoders with encoding complexity that is linear in blocklength while still harvesting the capacity-approaching benefits of PAS. The new approach equalizes the conditional entropies of the bits that label the constellation points, which decouples the LDPC design from the specific signaling PMF.

Equalizing the conditional entropies of the bit-levels allows PAS to approach the Shannon

limit using any of the numerous existing LDPC code constructions that allow low-complexity encoders, which were originally developed for the binary-input AWGN channel. The effectiveness of the proposed approach is demonstrated by modifying signaling PMF and the bit mapping of a code from [105] to support efficient encoding, showing how the modified constellation yields a performance that closely competes with the one of the original constellation-specific mapping of [105]. We then show how a protograph LDPC code optimized for the bi-AWGN channel with the proposed constellation design outperforms the design of [105].

1.1.5 Read and Write Threshold Optimization in Flash Memory

Chapters 6 and 7 apply a mutual-information (MI) maximization paradigm to optimize the voltage levels written into Flash memory cells. Beyond communication channels, similar principles apply to LDPC-coded storage systems, where signaling points represent the write-voltage levels used to store data by placing specific amounts of electrical charge onto the floating gate of a memory cell, changing its threshold voltage. To enable low-latency, each page of Flash memory stores only one coded bit in each Flash memory cell. For example, three-level cell (TL) Flash has three bit channels, one for each of three pages, that together determine which of eight voltage levels are written to each cell. Each Flash page is required to store the same number of data bits, but the various bits stored in the cell typically do not have to provide the same mutual information. In Chapter 6 This dissertation uses a mutual-information maximization paradigm to optimize the voltage levels written to cells in a Flash memory. A modified version of dynamic-assignment Blahut-Arimoto (DAB) moves the constellation points and adjusts the probability mass function for each bit channel to increase the mutual information of a worst bit channel with the goal of each bit channel providing the same mutual information. The resulting constellation provides essentially the same mutual information to each page while negligibly reducing the mutual information of

the overall constellation. The optimized constellations feature points that are neither equally spaced nor equally likely requiring PAS for practical implementation.

In Chapter 7, for a two-variance model of the Flash read channel that degrades as a function of the number of program/erase cycles, we show that jointly optimizing write voltage levels and read thresholds to maximize the minimum MI between a page and its associated initial or enhanced-precision read bits can improve LDPC decoding performance.

Chapter 7 Contributions

Chapter 7 concludes this dissertation.

Chapter 2

Design of LDPC Codes to Support Multiple Rates

2.1 Introduction

2.1.1 History of LDPC Codes

Low-density parity-check (LDPC) codes are a class of linear block codes invented by Gallager in his 1963 doctoral dissertation [57].¹ Despite offering near-capacity performance across a wide range of data-transmission and data-storage channels, implementation seemed out of reach at the time of the invention. LDPC codes were largely overlooked until Tanner's work in 1981 that brought renewed attention to them. Tanner generalized LDPC codes and introduced a graphical representation of LDPC codes, now called a Tanner graph [108].

The introduction of turbo codes by Berrou et al. [11] in 1990s sparked further interest in LDPC codes, leading to significant contributions from numerous researchers including MacKay and Luby [3, 16, 83, 84] who recognized the benefits of linear block codes with sparse (low-density) parity-check matrices.

¹Note: This chapter is largely a reprint of work published in the IEEE Aerospace Conference [55].

A fundamental LDPC code structure that facilitates efficient encoding while achieving good frame error rate (FER) performance is the Repeat Accumulate (RA) code [31] introduced in 1998. The original RA codes were regular in the sense that all variable nodes have the same degree, and all check nodes have the same degree. In 2000 RA codes were extended to include Irregular RA (IRA) codes [69]. Some advantages offered by IRA codes include greater flexibility in selecting the repetition rate for each information bit which enables the design of high-rate codes. Additionally, their irregularity enables operation that approaches capacity more closely.

In 2001 Richardson, Urbanke, and Shokrollahi [97,98] demonstrated that using density evolution to optimize the degree distribution of the variable and check nodes produces capacity approaching LDPC codes. In 2003, Thorpe [110] showed that LDPC codes can be constructed by copying and permuting a small base graph, called a protograph, with only few variable and check nodes. The parity-check matrix of the protograph is called protomatrix and the resulting code is called a Protograph LDPC code. In this chapter, protograph and protomatrix are used interchangeably.

The introduction of Accumulate Repeat Accumulate (ARA) Codes in 2004 and later in 2007 [2] achieved a significant improvement in the performance of LDPC codes. A key contribution was the ability to reduce the iterative decoding threshold to obtain capacity approaching LDPC codes without increasing the degrees of certain variable nodes as was required in [97,98]. This was accomplished by deleting (or "puncturing") selected bits from the LDPC transmitted codeword.

In 2005, Divsalar et al. [33] showed that the number of degree-2 nodes in LDPC code should be less than the number of check nodes attached to them in order to have an LDPC code with minimum distance growing linearly with block size. This discovery inspired the development of Accumulate Repeat Jagged Accumulate (ARJA) codes which were proposed to CCSDS in 2007 and became the international standard for space applications.

A few years later, Protograph-based Raptor-like (PBRL) LDPC codes were introduced

[21, 23], enabling construction of low-rate LDPC codes with good performance. PBRL codes naturally provide a rate-compatible family of codes supporting a wide range of rates. The PBRL structure features a highest rate code (HRC) and an incremental redundancy code (IRC) that provides the additional symbols used to achieve the lower rates.

The current CCSDS LDPC codes are proposed for the Human Landing System on the Moon and the NASA Artemis program to explore Mars via the Moon. While the development of LDPC codes has been highly successful, offering performance and complexity benefits over turbo codes at high code rates, turbo codes remained the optimal solution for lower code rates until now and have been utilized in the CCSDS standard [45]. Replacing the low-rate turbo codes with better-performing PBRL LDPC codes would allow the CCSDS codes to be used for longer distances and would also support dynamic rate adaptation for fading channels.

2.1.2 Contributions

This chapter presents a new class of low-rate PBRL LDPC Codes for lower rates $1/6, 1/4, 1/3$ for message lengths of $K = 1024, 4096$ and 16384 which avoid the error floor behavior seen in the current CCSDS turbo codes. Deep space communication to Mars and beyond requires transmitting data at low rates. The current CCSDS standard codes for rates less than $1/2$ are turbo codes, which are hindered by an error floor. Even increasing power cannot achieve error rates below this floor which is usually between 10^{-4} and 10^{-5} range of frame error rates.

As a remedy, this chapter provides new PBRL LDPC codes with rates below $1/2$ by attaching a Low-Density Generator Matrix (LDGM) code to the existing rate- $1/2$ CCSDS LDPC code. In software simulations, the new low-rate LDPC/LDGM codes offer better performance than the current turbo codes. By overcoming the error floor, the LDPC codes provide the very low FERs necessary to support reliable transmission of compressed data. The lower-rate LDPC encoder will use the current CCSDS encoder and generate the additional parity check bits based on the output of current CCSDS code using Exclusive OR operation

only. The overall decoder can be implemented as a single decoder that supports all code rates or the current LDPC decoder could be augmented with an LDGM decoder. Fig.2.1 shows

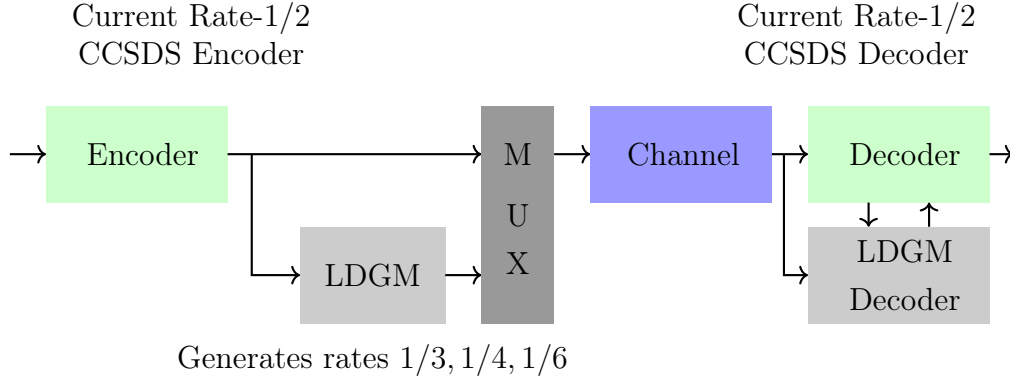


Figure 2.1: Encoder/Decoder for low-rate PBRL LDPC codes without changing the current CCSDS code.

LDPC encoding and decoding for the new low code rate codes. In this chapter, the selected HRC is exactly the existing CCSDS rate-1/2 code. The protograph for IRC parts for each message length is designed to achieve the best possible threshold for each successively lower rate according to Reciprocal Channel Approximation (RCA) [23]. The final rate-compatible LDPC codes are achieved by a two-step lifting procedure. First the protograph is lifted by four to create a protograph that has only zeros and ones and to ensure a girth of at least four. Using the concept of extrinsic message degree (EMD), the IRC protograph is lifted a second time by 128 or 512 or 2048 to optimize the approximate cycle EMD (ACE) [111] of the lowest rate code without changing the structure of HRC code, which is the rate-1/2 CCSDS LDPC code.

The addition of lower rates to the CCSDS standard would allow the codes to be used for longer distances and would also allow for dynamic rate adaptation to adapt to a fading channel. The current CCSDS LDPC codes are proposed to be used for Human Landing System on Moon and NASA Artemis program Moon/Mars. For future missions at much longer distances such as Mars, lower code rates with much better performance than turbo codes are desirable. Eventually the constructed low-rate LDPC codes will be proposed to

CCSDS standard to be used for space applications. The lower rate LDPC encoder will use the current CCSDS encoder and generates the additional parity check bits from the output of the current CCSDS encoder using Exclusive-OR operations.

In this chapter we describe the construction of the new codes and show that their performance outperforms turbo code FERs by several orders of magnitude in the error floor region. The structure of 5G-NR LDPC codes proposed in 2017 have the PBRL code structure that was jointly proposed by UCLA and JPL/Caltech in IEEE Transaction on Communications 2015 [23] and before that in a 2011 conference paper [21].

We will gain improvements in the overall performance of the deep space coding system in terms of lower required signal-to-noise ratio, increased data return, and lower FER by at least by three orders of magnitude with respect to turbo codes. Low-rate codes enable more reliable communication for the NASA Artemis program for future Moon/Mars and Human Landing Systems and exploration beyond the Moon.

2.1.3 Organization

The rest of the chapter proceeds as follows: Sec. 2.2 describes the PBRL approach to LDPC code construction and describes the newly designed LDPC codes. Sec. 2.3 compares the FER performance of the newly designed LDPC codes with the FER performance of CCSDS turbo codes having similar rates and block lengths. Sec. 2.4 concludes the chapter.

2.2 PBRL LDPC Code Construction

Let $\mathbf{P}$ given in (2.1) be an $n_c \times n_v$ protomatrix of PBRL LDPC code with n_c check nodes and n_v variable nodes:

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_{\text{HRC}} & \mathbf{0} \\ \mathbf{P}_{\text{IRC}} & \mathbf{I} \end{bmatrix}. \quad (2.1)$$

In (2.1), the highest rate code of protograph is represented by $\mathbf{P}_{\text{HRC}}$ protomatrix. The $\mathbf{0}$ and $\mathbf{I}$ matrices are respectively the all-zeros and identity matrices of appropriate sizes. Together, the $\mathbf{P}_{\text{IRC}}$ and $\mathbf{I}$ matrices specify the additional rows of the parity check matrix that lower the rate by adding additional parity symbols to codewords of the highest rate code.

Let n_p denote the number of “punctured” variable nodes, i.e. protograph variable nodes intentionally excluded from transmission in the code structure. The rate of a protograph is $R = \frac{n_v - n_c}{n_v - n_p}$. This is the lowest rate supported by the PBRL LDPC code. Our goal is to design the protomatrix $\mathbf{P}$ and lift it to produce the needed low-rate LDPC codes.

2.2.1 The Highest Rate Code (HRC)

We use the protograph of the existing rate-1/2 CCSDS LDPC code to obtain $\mathbf{P}_{\text{HRC}}$ for our new codes. The protograph of the existing rate-1/2 CCSDS LDPC code is described in (2.2). The last column of this protograph is designated for puncturing.

$$\mathbf{P}_{\text{CCSDS}} = \begin{bmatrix} 0 & 0 & 1 & 0 & 2 \\ 1 & 1 & 0 & 1 & 3 \\ 1 & 2 & 0 & 2 & 1 \end{bmatrix} \quad (2.2)$$

Since $\mathbf{P}_{\text{CCSDS}}$ does not have a favorable PBRL structure as in (2.1) to facilitate the design process, this chapter obtains $\mathbf{P}_{\text{HRC}}$ described in (2.3) by permuting rows and columns of the existing CCSDS rate-1/2 code given in (2.2) for all message lengths. Note that with the permutation the last column of CCSDS code in (2.2) becomes the first column of $\mathbf{P}_{\text{HRC}}$ in (2.3) and thus the first column in (2.3) is designated to be punctured.

$$\mathbf{P}_{\text{HRC}} = \begin{bmatrix} 3 & 1 & 1 & 1 & 0 \\ 1 & 1 & 2 & 2 & 0 \\ 2 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.3)$$

Table 2.1: Code Block Lengths for Supported CCSDS Code Rates and Newly Designed Code Rates

Message Length K	Code block length: $n - n_p$					
	CCSDS Rates			New Rates		
	4/5	2/3	1/2	1/3	1/4	1/6
1024	1280	1536	2048	3072	4096	6144
4096	5120	6144	8192	12288	16384	24576
16384	20480	24576	32768	49152	65536	98304

We use (2.3) for the PBRL design process. However, after code optimization and lifting, the columns and rows of the HRC are inversely permuted to recover the rate-1/2 parity check matrix H as described in the CCSDS standard with exactly the circulant permutations of the CCSDS standard as the HRC in the final PBRL matrix.

2.2.2 Incremental Redundancy Code (IRC) Design

The identity matrix $\mathbf{I}$ in (2.1) generates degree-1 variable nodes, corresponding to the incremental redundancy symbols of $\mathbf{P}$. The check nodes that connect to the degree-1 variable nodes also have connections to the variable nodes in the HRC part. These connections are expressed by the sub-matrix $\mathbf{P}_{\text{IRC}}$, which represents the incremental redundancy code. This section presents the design of $\mathbf{P}_{\text{IRC}}$ to support rates 1/3, 1/4 and 1/6 for message lengths of $K = 1024, 4096, 16384$. The existing rates and code block lengths of CCSDS code along with new rates and code block lengths are presented in Table 2.1.

The protomatrix $\mathbf{P}_{\text{IRC}}$ is designed row by row to achieve the best possible thresholds for each successively lower rate according to Reciprocal Channel Approximation (RCA) [23]. The RCA is a fast and accurate approximation to the density evolution algorithm with deviation in accuracy of less than 0.01 dB. The RCA for the binary-input AWGN (BI-AWGN) channel (BPSK modulation with soft output demodulation) uses a single real-valued parameter s , the effective signal-to-noise ratio (SNR), and its reciprocal r to approximate the densities of the

density evolution.

The RCA algorithm essentially models all of the message densities as Gaussian distributions and passes a single parameter to describe each Gaussian density. In the context of the RCA algorithm, s_e is the message passed along an edge e from a variable node to a check node and r_e is the message passed along an edge e from a check node to a variable node. The message s_e may be interpreted the SNR for the Gaussian message density corresponding to that edge on that iteration. The message r_e is the *reciprocal* SNR for the Gaussian message density corresponding to that edge on that iteration. The reciprocal SNR r_e is defined so that $C(s_e) + C(r_e) = 1$, where $C(x)$ is the capacity of the binary-input additive white Gaussian noise (BI-AWGN) channel with SNR x . Let s_{chl} be the channel SNR

$$s_{chl} = 2 \frac{E_c}{N_0} = 2R \frac{E_b}{N_0}, \quad (2.4)$$

where $2E_c/N_0$ is the code-symbol signal-to-noise ratio, E_b is energy per information bit, and R is the code rate. If the edge e is connected to a punctured variable node the message s_e is initialized to 0. Otherwise s_e is initialized to s_{chl} . RCA computes a sequence of messages $(s_e^{(n)}, r_e^{(n)}), n = 0, \dots, N$, where N represents the maximum number of iterations. The approximated iterative decoding threshold s_{th} is determined as the minimum s_{chl} such that $s_e^{(N)} > T$ for all edges e in the graph, where T is a stopping threshold. The full RCA algorithm can be found in [23]. Alg. 1 describes IRC design process. At every iteration i , a

Algorithm 1 Greedy Design for Protomatrix $\mathbf{P}_{\text{IRC}}$

Require: Initial $\mathbf{P}_0 = \mathbf{P}_{\text{HRC}}$, Row number $t = 8$

$i \leftarrow 1$

while $i \leq t$ **do**

$\mathbf{h}_i = \arg \min_{\mathbf{h}_i \in \mathbf{A}} \delta(\mathbf{P}_i)$,

 where $\mathbf{P}_i$ is computed from $\mathbf{P}_{i-1}$ via (2.7)-(2.8).

$i = i + 1$

end while

return $[\mathbf{h}_1^T \dots \mathbf{h}_t^T]^T$

new row $\mathbf{h}_i$ of $\mathbf{P}_{\text{IRC}}$ is designed to minimize the threshold $\delta(\mathbf{P}_i)$ of $\mathbf{P}_i$, where

$$\mathbf{P}_i = \begin{bmatrix} \mathbf{P}_{\text{HRC}} & \mathbf{0} \\ \mathbf{P}_{\text{IRC},i} & \mathbf{I} \end{bmatrix}, \quad (2.5)$$

and

$$\mathbf{P}_{\text{IRC},i} = [\mathbf{h}_1^T \dots \mathbf{h}_i^T]^T. \quad (2.6)$$

At iteration i , $\mathbf{P}_{i-1}$ has been completely specified as

$$\mathbf{P}_{i-1} = \begin{bmatrix} \mathbf{P}_{\text{HRC}} & \mathbf{0}_{3 \times i} \\ \mathbf{P}_{\text{IRC},i-1} & \mathbf{I}_{i-1 \times i-1} \end{bmatrix}, \quad (2.7)$$

so that only the row vector $\mathbf{h}_i$ is allowed to vary to improve the RCA threshold. Thus, $\mathbf{h}_i$ is selected to minimize the threshold $\delta(\mathbf{P}_i)$ (calculated using RCA [23]) of

$$\mathbf{P}_i = \begin{bmatrix} \mathbf{P}_{\text{HRC}} & \mathbf{0}_{3 \times i} & \mathbf{0}_{3 \times 1} \\ \mathbf{P}_{\text{IRC},i-1} & \mathbf{I}_{i-1 \times i-1} & \mathbf{0} \\ \mathbf{h}_i & \mathbf{0}_{1 \times i-1} & 1 \end{bmatrix}. \quad (2.8)$$

After t iterations, the final IRC protomatrix is given by:

$$\mathbf{P}_{\text{IRC}} = [\mathbf{h}_1^T \dots \mathbf{h}_t^T]^T. \quad (2.9)$$

To obtain the lowest code rate of 1/6, we set $t = 8$ and consider the following $\mathbf{h}_i$'s:

$$\mathbf{h}_i \in \mathbf{A} = \{[1 \ b_1 \ b_2 \ b_3] \mid b_j \in \{0, 1\}\}. \quad (2.10)$$

For all message lengths considered in this chapter a single protomatrix $\mathbf{P}_{\text{IRC}}$ is designed

consisting of 8 rows to support the lowest code rate of 1/6. The final designed protomatrix $\mathbf{P}$ in (2.1) is shown below, with dimensions 11×13 .

$$\mathbf{P} = \begin{bmatrix} 3 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 2 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Protographs for rates 1/4 and 1/3 are obtained by puncturing the degree-one variable nodes corresponding to the last four or six columns of $\mathbf{P}$, respectively.

2.2.3 Quasi-Cyclic LDPC Code and Lifting

CCSDS LDPC codes are members of a class of codes called Quasi-Cyclic (QC) codes. QC LDPC codes are characterized by a parity-check matrix that consists of square sub-matrices that are either the zero matrix or a cyclic permutation σ^i of the identity matrix, also called a circulant where σ is the right circular shift of the identity matrix of the same size. Such parity check matrices are constructed from a protomatrix through a two-step process known as lifting. We construct the QC-LDPC parity check matrix $\mathbf{H}$ from protomatrix $\mathbf{P}$ in (2.1) as follows.

The first step of lifting constructs the 44×52 matrix $\mathbf{H}_1$ in Fig. 2.2 by replacing each

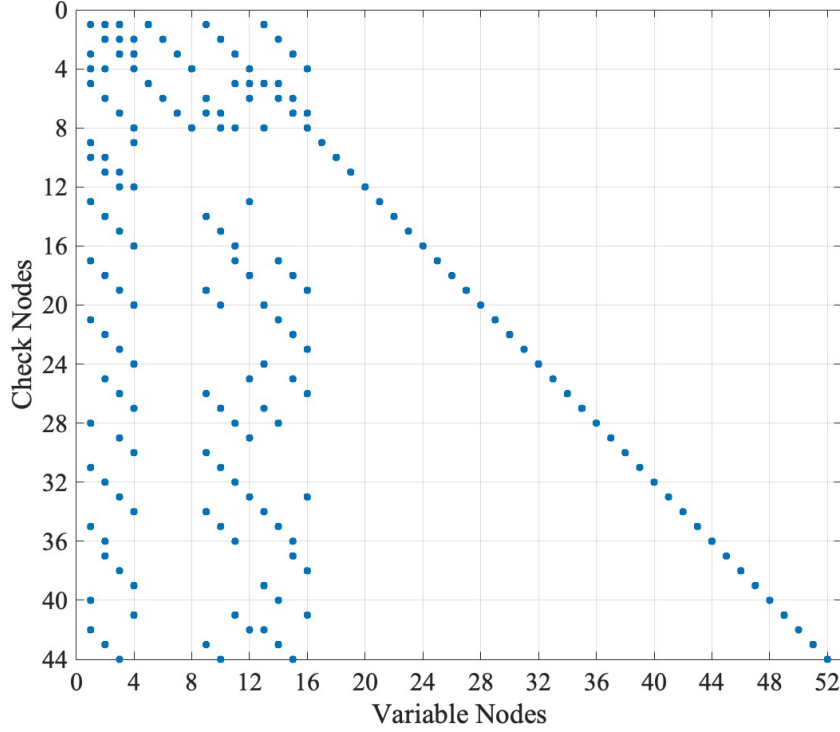


Figure 2.2: The 44×52 matrix H_1 after lifting by 4 of 11×13 LDPC protograph. Dots indicate ones.

element in $\mathbf{P}$ with square sub-matrices of size $Z_1 = 4$. If the element in $\mathbf{P}$ is 0, then the sub-matrix is a zero matrix of size $Z_1 = 4$. If the element in $\mathbf{P}$ is $k > 0$, then the sub-matrix equals to $\sigma^{s_1} + \dots + \sigma^{s_k}$, where $s_1 \dots s_k \in \{0 \dots Z_1 - 1\}$ and $s_i \neq s_j$ for $i \neq j$.

Note that the lifting by 4 also preserves original rate-1/2 CCSDS matrix with circulants, i.e. the first 12×20 part of $\mathbf{H}_1$ matrix after inverse permutation of columns is exactly the rate-1/2 CCSDS code lifted by 4 and provided in the CCSDS standard [47].

The second lifting step constructs the QC LDPC parity-check matrix $\mathbf{H}$ by replacing the elements in $\mathbf{H}_1$ with zero matrices or circulant matrices of size $Z_2 \times Z_2$ to optimize approximate-cycle extrinsic-message-degree (ACE) [111]. The value of Z_2 is 128, 512, or 2048, corresponding to message lengths $K = 1024, 4096, 16384$, respectively. The ACE algorithm with parameters of d_{ACE} and η ensures that the lifted parity check matrix has the property that all cycles whose length is $2d_{\text{ACE}}$ or less in the bipartite graph associated with the parity

Table 2.2: d_{ACE} and η for different message lengths

Message Length K	1024	4096	16384
d_{ACE}	5	5	5
η	21	26	33
Z_2	128	512	2048

check matrix have ACE values of at least η . Table 2.2 provides the values of d_{ACE} and η that were enforced for the three message lengths of interest.

2.3 Comparison of LDPC Codes to Turbo Codes

The current CCSDS standard codes with rates less than $1/2$ are turbo codes hindered by an error floor—we cannot decode data frames with arbitrarily low error rates as we increase our transmit power. In the error floor region, the slope of the turbo code FER curve becomes significantly flatter [45]. The newly constructed LDPC codes are designed to overcome the error floor drawbacks of the current CCSDS turbo codes.

Since CCSDS turbo codes do not support message lengths of 1024 and 4096 we compare the performance of LDPC codes to the turbo codes by simulation and by using normal approximation by Polyanskiy [94]. The normal approximation refines the classical Shannon capacity result by providing tight approximations of the achievable coding rate for finite block lengths n as short as 100 bits. The maximum coding rate $R^*(n, \epsilon)$ with a final block length n and error probability ϵ is approximated by:

$$R^*(n, \epsilon) \approx C - \sqrt{\frac{V}{n}} Q^{-1}(\epsilon) + \frac{\log_2 n}{2n} \quad (2.11)$$

where C is channel capacity, V is channel dispersion and $Q^{-1}(\epsilon)$ is inverse Gaussian function. Comparison of the newly constructed LDPC codes to the turbo codes is obtained by comparing frame error rates (FERs) of both codes to normal approximation in addition to simulations.

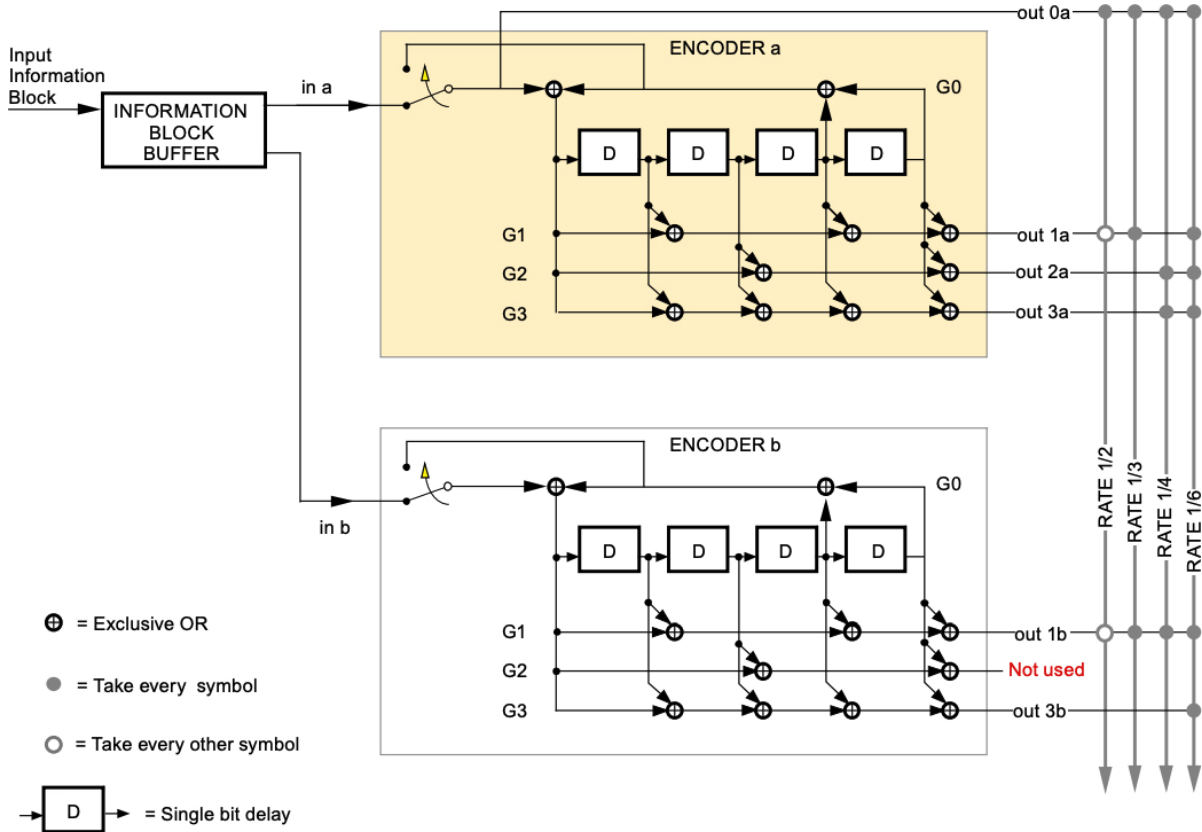


Figure 2.3: Turbo code encoder block diagram from CCSDS Blue Book.

The closer the frame error rate of the code is to the rate predicted by the normal approximation, the better the code performs.

2.3.1 Turbo Codes Simulation Setup

As described in CCSDS Blue Book [46] and presented in Fig. 2.3, the CCSDS turbo encoder consists of two encoders that generate parity symbols for two recursive convolutional codes, each with a small number of states. A main characteristic of turbo codes is the use of an interleaver, which permutes bit-wise the original K information bits before input to the second encoder. The interleaver for turbo codes is a fixed bit-by-bit permutation of the entire block of data. Since the turbo code encoder is systematic, the information bits are sent uncoded in

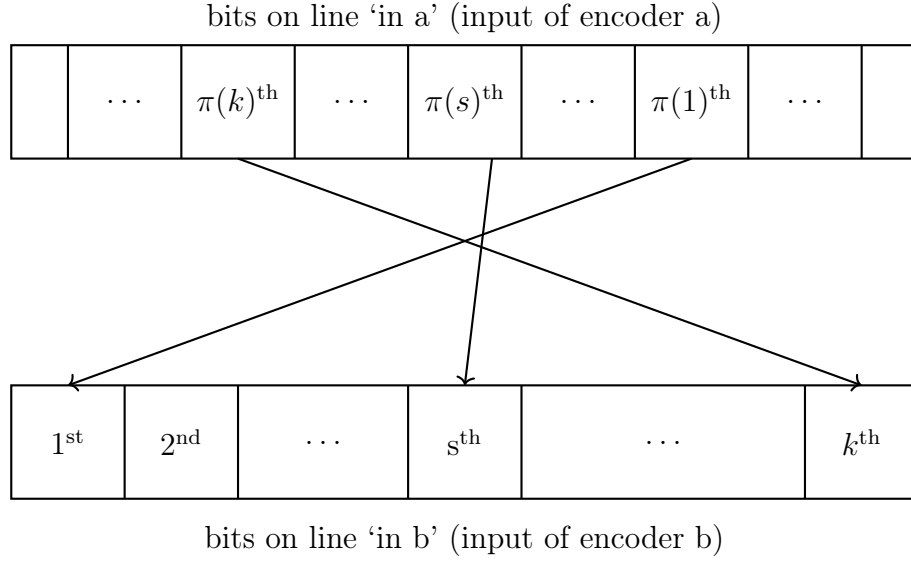


Figure 2.4: Interpretation of Turbo Code Permutation.

Table 2.3: Parameters k_1 and k_2 for Specified Information Block Lengths from CCSDS Blue Book.

Information block length	k_1	k_2
1016	8	127
1024	8	128
4072	8	509
4096	8	128×4
16384	8	128×16

output 0a of Fig. 2.3. The turbo code permutation for any specified message length K must follow a specific reordering of the integers $1, 2, \dots, K$ determined by the following algorithm which is in the CCSDS Blue Book.

1. K shall be expressed as $K = k_1 k_2$, where k_1 and k_2 for the specified block sizes are given in Table 2.3.
2. The following operations shall be performed for $s = 1$ to $s = k$ to obtain permutation

numbers $\pi(s)$:

$$\begin{aligned}
m &= (s - 1) \mod 2 \\
i &= \lfloor \frac{s - 1}{2k_2} \rfloor \\
j &= \lfloor \frac{s - 1}{2} \rfloor - ik_2 \\
t &= (19i + 1) \mod \frac{k_1}{2} \\
q &= t \mod 8 + 1 \\
c &= (p_q j + 21m) \mod k_2 \\
\pi(s) &= 2 \left(t + c \frac{k_1}{2} + 1 \right) - m
\end{aligned}$$

Where $\lfloor x \rfloor$ denotes the largest integer less than or equal to x , and p_q denotes one of the following eight prime integers

$$\begin{aligned}
p_1 &= 31; p_2 = 37; p_3 = 43; p_4 = 47; \\
p_5 &= 53; p_6 = 59; p_7 = 61; p_8 = 67
\end{aligned}$$

The permutation numbers shall be interpreted such that the s th bit read out on line 'in b' in Fig. 2.3 is the $\pi(s)$ th bit of the input information block, as shown in Fig. 2.4.

For input block size of $K = 16384$ and rates R of $1/6$, $1/4$ and $1/3$ FER performance of the turbo codes is provided in CCSDS [45]. To compare FER performance of the designed low-rate LDPC codes with CCSDS turbo codes for input block sizes $K = 1024$ and $K = 4096$ FER simulations for the turbo codes are performed. We also performed simulations for input block sizes of $K = 1016$ and $K = 4072$ to explore how input length can affect interleaver performance and error floor. For $K = 1016$ and $K = 4072$ we selected k_2 as the nearest prime number to $k_2 = K/8$ to see if a prime k_2 would improve the error floor performance. The resulting message block sizes K are 1016 and 4072, which are very close to the LDPC code

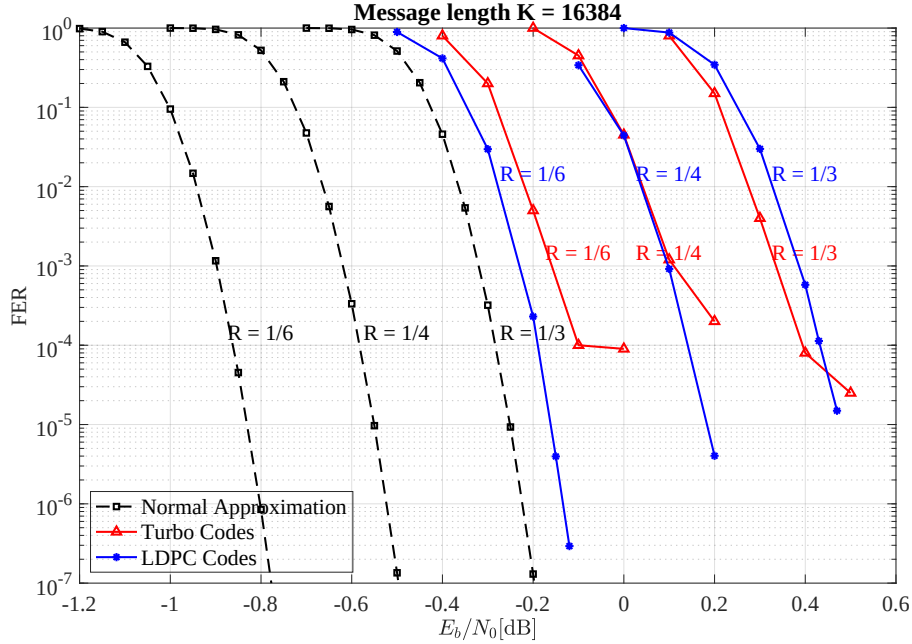


Figure 2.5: FER as a function of E_b/N_0 of LDPC Code compared to normal approximation and turbo codes for rates $R = 1/6, 1/4, 1/3$ and message length $K = 16384$.

input block sizes and sometimes provide better interleavers with lower error floors but now with a message length that is not a power of 2. In CCSDS turbo codes there are 4-bit tails to terminate the trellis, so the actual code rates for turbo codes are $RK/(K + 4)$ which are slightly lower than the ideal R s of $1/6, 1/4, 1/3$ and $1/2$ achieved by the LDPC code.

2.3.2 Message Length $K = 16384$

Fig. 2.5 shows FER of the designed LDPC code compared to FER obtained with normal approximation and FER of the CCSDS turbo code, plotted as a function of E_b/N_0 for a code rates of $R = 1/6, 1/4, 1/3$ and a message length of $K = 16384$. In the waterfall region down to FER 10^{-3} or 10^{-4} , the rate-1/6 LDPC code outperforms the rate-1/6 turbo code by less than 0.1 dB. The rate-1/4 LDPC and turbo codes have almost identical waterfall performance, and at rate-1/3 the turbo code outperforms the LDPC code in the waterfall region by less than 0.1 dB

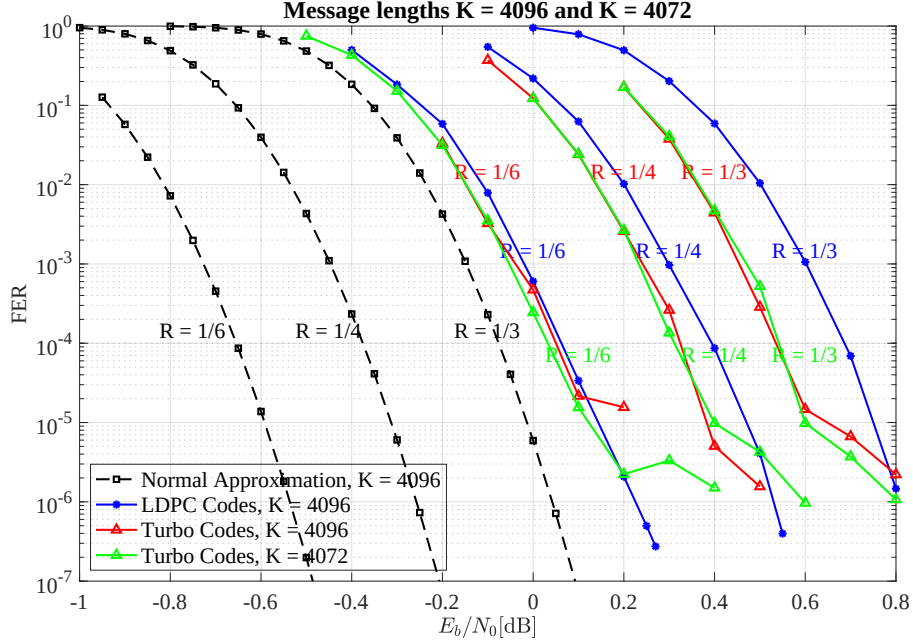


Figure 2.6: FER as a function of E_b/N_0 of LDPC Code with message length $K = 4096$ compared to normal approximation with $K = 4096$ and turbo codes with message lengths $K = 4096$ and $K = 4072$ for rates $R = 1/6$, $1/4$ and $1/3$.

After the waterfall region, the LDPC code has much better performance for all three rates. Below 10^{-4} FER, the rate-1/6 LDPC code outperforms the rate-1/6 turbo code by a few orders of magnitude since turbo code has an error floor at 10^{-4} . The rate-1/3 and rate-1/6 turbo codes have error floors at FER of 10^{-4} and rate-1/4 at FER of 10^{-3} . The LDPC codes did not show any error floor at the simulated SNRs. The data for turbo codes FER curves in Fig. 2.5 is obtained from the CCSDS standard [45].

2.3.3 Message Lengths $K = 4096$, $K = 4072$

The CCSDS standard provides turbo code for message length $K = 3568$. In order to compare FER performance of the designed LDPC code rates to FER performance of the turbo code, we simulated turbo codes for message lengths $K = 4072$ and $K = 4096$ for rates $1/6$, $1/4$ and $1/3$. Fig. 2.6 illustrates FER of the designed LDPC code for message length $K = 4096$ compared

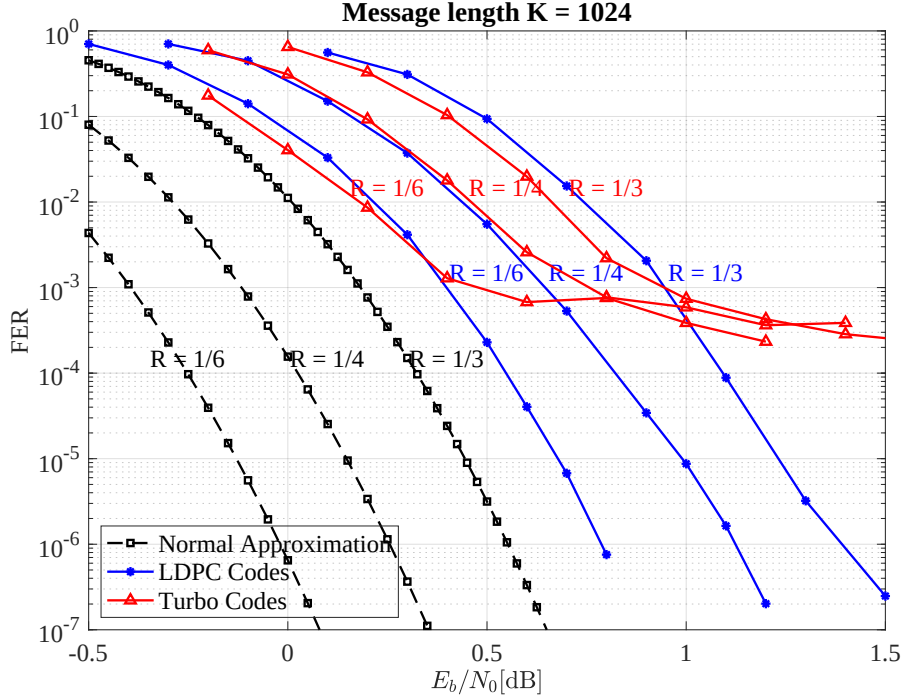


Figure 2.7: FER as a function of E_b/N_0 of LPDC Code compared to turbo codes and normal approximation for rates $R = 1/6, 1/4, 1/3$ and message length $K = 1024$.

to normal approximation and FER of the turbo code for message lengths of $K = 4096$ and $K = 4072$, plotted as a function of E_b/N_0 for code rates $R = 1/6, 1/4, 1/3$.

For rates $1/4$ and $1/3$, the designed LDPC codes show worse waterfall-region FER performance compared to the turbo codes. For rate $1/3$, the turbo code outperforms the LDPC codes by almost 0.2 dB in the waterfall region. However, turbo code is hindered with an error floor at FER of about 10^{-5} and therefore the LDPC codes perform significantly better for lower FERs. When using input block size of $K = 4072$, where the interleaver benefits from a prime k_2 , the turbo code shows slightly lower error floor for rates $1/6$ and $1/3$.

2.3.4 Message Length $K = 1024, K = 1016$

The CCSDS standard provides turbo codes for message length $K = 1784$, which is the closest message length to $K = 1024$ for which we designed LDPC code. For the purposes

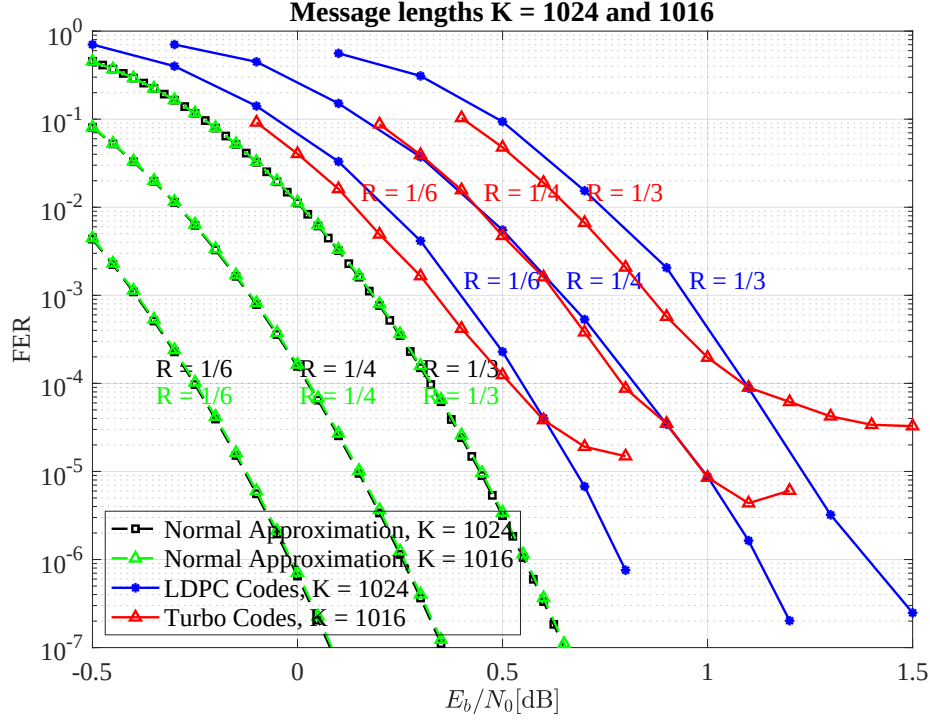


Figure 2.8: FER as a function of E_b/N_0 of LDPC Code with message length $K = 1024$ compared to turbo codes with message length $K = 1016$ for rates $R = 1/6, 1/4, 1/3$.

of performance comparison, we simulated turbo code FER performance for message lengths $K = 1024$ and $K = 1016$, where the interleaver benefits from a prime k_2 . Fig. 2.7 shows frame error rate of the designed LDPC codes, CCSDS turbo codes, and normal approximation as a function of E_b/N_0 for rates $R = 1/6, 1/4$, and $1/3$ and for message length $K = 1024$. For message length $K = 1024$ the CCSDS turbo code suffers from a high error floor at FER of 10^{-3} . Fig. 2.8 shows FER as a function of E_b/N_0 for the designed LDPC codes and normal approximation with message length $K = 1024$ and CCSDS turbo codes with message length $K = 1016$, where the interleaver benefits from a prime k_2 , for rates $R = 1/6, 1/4$, and $1/3$. For $K = 1016$ turbo code shows a lower error floor at FER of 10^{-4} because of a prime k_2 . Figs. 2.7 and 2.8 emphasize the benefit of the LDPC approach at low FERs. The LDPC code outperforms the turbo code FER by orders of magnitude. Note that LDPC code has a steep FER curve and no error floor for simulated data points down to FER of 10^{-7} .

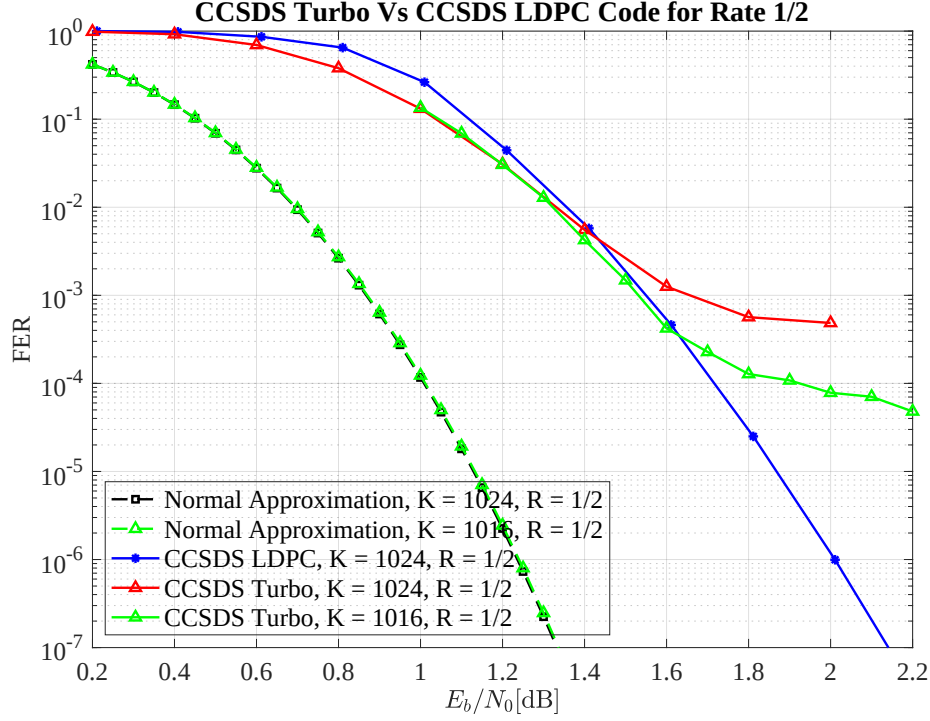


Figure 2.9: FER as a function of E_b/N_0 of the CCSDS LDPC Code with message length $K = 1024$ compared to the CCSDS turbo code and normal approximation with message lengths **$K = 1016$** and $K = 1024$ for rate $R = 1/2$.

Fig. 2.9 compares the FER vs. E_b/N_0 performance of the rate-1/2 CCSDS LDPC code with input block size of $K = 1024$ to the rate-1/2 CCSDS turbo codes with input block sizes of $K = 1024$ and $K = 1016$. The CCSDS turbo codes have error floors at FERs of 10^{-3} and 10^{-4} respectively for $K = 1024$ and $K = 1016$. The CCSDS rate-1/2 LDPC code, simulated down to FER of 10^{-7} , shows steep curve without error floor implying orders of magnitude superior performance when compared to the rate-1/2 CCSDS turbo codes.

2.4 Conclusion

Deep space communication to Mars and other distant missions requires data transmission at low rates. For rates below 1/2, the current CCSDS standard provides turbo codes which

suffer from an error floor phenomenon where the FER does not improve significantly despite increasing the transmission power beyond a certain threshold. To overcome this error floor limitation, we developed new LDPC codes with rates $R = 1/3$, $1/4$, and $1/6$ while maintaining the original rate- $1/2$ CCSDS LDPC code [47] as a part of the new code.

Simulations show that the FER of LDPC codes outperforms current CCSDS turbo codes in the error floor region of the turbo codes by orders of magnitude. The turbo codes have error floors between 10^{-3} and 10^{-5} while the newly designed LDPC codes had no error floor visible in our simulations. The newly designed LDPC codes performed similarly to the turbo codes in the waterfall region, generally within 0.1 dB better or worse than the turbo codes. The most waterfall performance loss was with the $K = 4096$ LDPC code, which was almost 0.2 dB worse than the turbo code in the waterfall region.

In summary, the turbo codes provided in the CCSDS standard have an error floor between FERs of 10^{-3} and 10^{-5} . Since the published CCSDS performance results do not provide enough data to observe the error floor for all the rates/block lengths, we simulated CCSDS turbo codes for rates $1/2$, $1/3$, $1/4$ and $1/6$ and message lengths $K = 1016, 1024, 4072$ and $K = 4096$ to provide more robust comparison with our LDPC codes. The simulated results confirm the existence of error floor of turbo codes at about FER between 10^{-3} or 10^{-5} and show that the newly designed LDPC codes do not display this error floor behavior.

Chapter 3

Adapting LDPC Code Rate to Changing Channel Conditions

3.1 Introduction

3.1.1 Background

Free-space optical (FSO) communication [77] offers numerous benefits, including high data rates, vast unlicensed free spectrum, high immunity to interference, highly secure links, and easy installation [70, 93, 112].¹ FSO channels can be used for communications over distances of several kilometers as well as ultra-long distances, including ground-to-satellite, satellite-to-satellite communications, and interplanetary communications [112].

FSO links are sensitive to channel fading caused by atmospheric turbulence, varying weather conditions, and changes in the distance between the transmitter and receiver. Because of this fading, hybrid communication systems are sometimes deployed where an RF link is used when the FSO link fails [1, 9, 71, 91]. A novel coding paradigm called "Hybrid Channel Coding" that constructs non-uniform and rate-compatible LDPC codes to achieve

¹Note: This chapter is largely a reprint of work [54] published in the IEEE Open Journal of the Communications Society.

the combined channel capacity of parallel FSO and RF channels is introduced in [43].

Simulation analysis in [43] shows that Hybrid Channel Codes can increase average throughput more than 33% compared to prior systems.

FSO fading has also been mitigated by adaptive parameter selection techniques such as those explored in [5, 28, 34, 60, 81]. The authors of [60] introduced a rate-adaptive transmission scheme employing intensity modulation and direct detection over FSO channels. The scheme uses repetition coding and variable silence periods to exploit the potential time-diversity order available in the fading channel.

Similarly, the study in [28] investigated punctured digital video broadcast satellite standard (DVB-S2) LDPC codes combined with channel interleavers to utilize time diversity. The combination of channel coding and bit interleaving in [28] improves performance under turbulence conditions.

In [34], three adaptive modulation schemes have been explored: (i) variable-rate variable-power adaptation, (ii) channel inversion, and (iii) truncated channel inversion schemes. CSI is estimated at the receiver and is fed back to the transmitter through the RF channel without considering the feedback delay. The results show that channel inversion scheme gives similar performance compared to variable-rate variable-power scheme when turbulence is weak but suffers from significant performance degradation when turbulence is strong.

To mitigate FSO fading, a scheme to estimate the channel state information (CSI) at the receiver for Raptor and punctured LDPC code rate selection is proposed in [5]. The receiver sends estimated CSI through a feedback channel to the transmitter where the code rate is selected to accommodate estimated fading channel conditions. The proposed feedback scheme for both coding schemes is evaluated over a short transmission range such that feedback delay is not significant compared to the coherence time of the fading.

A rate-adaptive scheme using LDPC codes with optimized puncturing is compared to uncoded FSO system and coded FSO system using LDPC codes with random puncturing scheme in [81]. Results show that rate-adaptive systems perform well in realistic FSO systems

over different weather conditions. For example, under rainy weather conditions uncoded FSO systems suffer from outages 87% of the time. In contrast, LDPC rate-adaptive systems can successfully utilize 75% to 80% of the signaling rate resulting in a significant increase in throughput. In [81], the rate is selected based on the CSI estimate at the receiver and sent back to the transmitter through an error-free feedback channel. However, in [81] the code rate selection does not consider how delay in the feedback channel can impact performance.

This chapter investigates the effect of feedback delay on rate-adaptive FSO system with LDPC coding. Rate-adaptive LDPC codes provide significant coding gain [89] and efficient encoding and decoding with low hardware complexity [113, 114]. LDPC codes comprise the standard coding technique in Digital Video Broadcasting - Satellite, Second Generation (DVB-S2) [41] and are also utilized by Version 3.1.0 of the Optical Communications Terminal (OCT) Standard Developed by the Space Development Agency of the United States Space Force [104].

To minimize performance degradation caused by feedback delay, this chapter uses predictive models to estimate fading channel conditions and dynamically select LDPC code rate. This achieves reliable communication while efficiently utilizing the available channel mutual information. The three predictive models explored are zero-order prediction, linear prediction, and quadratic prediction.

This work examines how these predictive models behave as the feedback delay is increased from no delay to a delay equal to the coherence time of the channel. Optical channels with coherence times of 5 *ms* and 10 *ms* are explored. Protograph-based Raptor-like (PBRL) LDPC codes with rates 8/9, 8/10, ..., 8/80 are designed using reciprocal channel approximation (RCA) [23] allowing convenient rate switching.

The feedback delay is the round-trip delay, i.e., the time between when the receiver makes the rate prediction and when the LDPC codeword at the selected rate arrives at the receiver experiencing a fade. A simple block diagram in Fig. 3.1 illustrates the round-trip feedback delay. The first half of the round-trip delay is the time it takes for the rate selection signal to

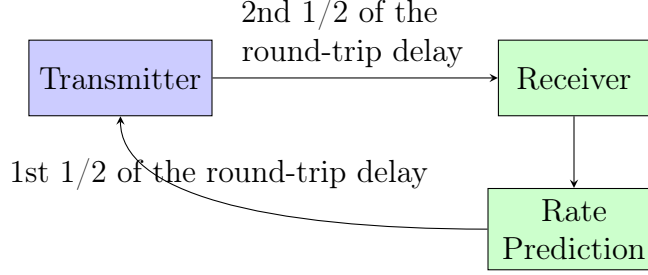


Figure 3.1: Simplified block diagram illustrating round-trip delay in using feedback for rate control. The first half of the round-trip delay is the duration from when the receiver sends back the predicted rate until the transmitter receives it. The second half of the round-trip delay is the duration for the signal encoded at the selected rate to travel from the transmitter to the receiver.

travel back to the transmitter after the receiver predicts the rate. The second half of the round-trip delay is the signal propagation time from the transmitter to the receiver. Thus, the receiver experiences the fade one round-trip time after sending the prediction.

Note that there is an additional small time at the transmitter to compute the codeword after receiving the prediction. However, this could be very small or even negligible since the transmission can start even slightly before the rate guidance arrives since the rate determines the total number of symbols and the transmitter will always send at least the number of symbols for the highest rate. An additional time also occurs for the receiver to compute the prediction, which is negligible compared to the total feedback delay.

The optical channel with a coherence time of 10 ms is meant to model the optical channel of a Low Earth Orbit (LEO) satellite. A typical LEO system usually operates at altitudes ranging from around 160 km (99 miles) to $2,000\text{ km}$ ($1,240\text{ miles}$). The distance between an Earth station and a LEO satellite corresponding to the specific round-trip feedback delay time is calculated as $c \times t_d/2$, where $c = 3 \times 10^8\text{ m/s}$ is speed of light and t_d is round-trip feedback delay time. The round-trip delay times of $2 - 10\text{ ms}$ correspond to distances of 300 to 1500 km .

Table 3.1 provides the orbital distances of various common LEO systems. The International Space Station (ISS), for example, has an orbital distance of $408 - 420\text{ km}$ while SpaceX's

Table 3.1: Orbital distances and corresponding round-trip feedback delays of common LEO systems.

LEO system	Orbital Distance (km)	Round-Trip Feedback Delay (ms)
Starlink (SpaceX)	340 – 570	2.2 – 3.8
International Space Station (ISS)	408 – 420	2.7 – 2.8
Hubble Space Telescope	535	3.6
TROPICS (NASA)	550	3.6
Planet Labs’ Dove Satellites	400 – 500	2.7 – 3.4
Kuiper (Amazon)	590 – 630	4.0 – 4.2
Landsat (NASA/USGS)	705	4.7
Sentinel (ESA)	700 – 786	4.6 – 5.2
Iridium Next	780	5.2
NOAA JPSS	824	5.5
OneWeb	1200	8.0
Globalstar	1400	9.3

Starlink satellites orbit Earth at a distance of about 340 – 570 km above sea level. The project Kuiper satellites will have an orbit between 590 and 630 kilometers. The Iridium satellites orbit at about 780 km . Earth observation satellites such as Landsat satellites operate around 705 km .

3.1.2 Contributions

The main contributions include:

- Design of 72 PBRL LDPC codes supporting a wide range of rates from 8/9 to 8/80. The rates are designed using RCA [23] to minimize the decoding threshold.
- The analysis shows when CSI is known with no delay, dynamically selecting LDPC code rate based on the CSI maximizes throughput. Such throughput is referred as the zero-delay throughput in this chapter.

- Zero-order, linear, and quadratic prediction models are explored to estimate fading channel CSI and dynamically select the LDPC code rate. The optical channel coherence time used for analysis is 10 *ms* and round-trip feedback delay times explored range from no delay to delay equal to 4 *ms*.
- Providing a detailed analysis of the channel model and the simulation method used to generate realizations of temporal fading.
- Incorporating the CSI estimate at the receiver to make prediction models more applicable for real-world implementation.
- Studying optical channels with shorter coherence times such as 5 *ms* in addition to the explored coherence time of 10 *ms*.
- Improving the linear and quadratic prediction models by optimizing the number of fading channel gain samples used to predict future channel conditions and appropriately select LDPC code rate.
- Extending round-trip feedback delay times from no delay to delay equal to the coherence time of optical channel.
- Analyzing computational complexity of the proposed prediction models.
- Exploring how round-trip delay affects common LEO systems in terms of their orbital distance.

3.1.3 Organization

The rest of the chapter proceeds as follows. Sec. 3.2 introduces the system architecture and FSO channel model. Sec. 3.3 presents a estimates FER for LDPC codes in the modeled channel using the normal approximation. Sec. 3.4 describes LDPC codes designed for a wide range of rates. Sec. 3.5 describes three prediction models and presents throughput results

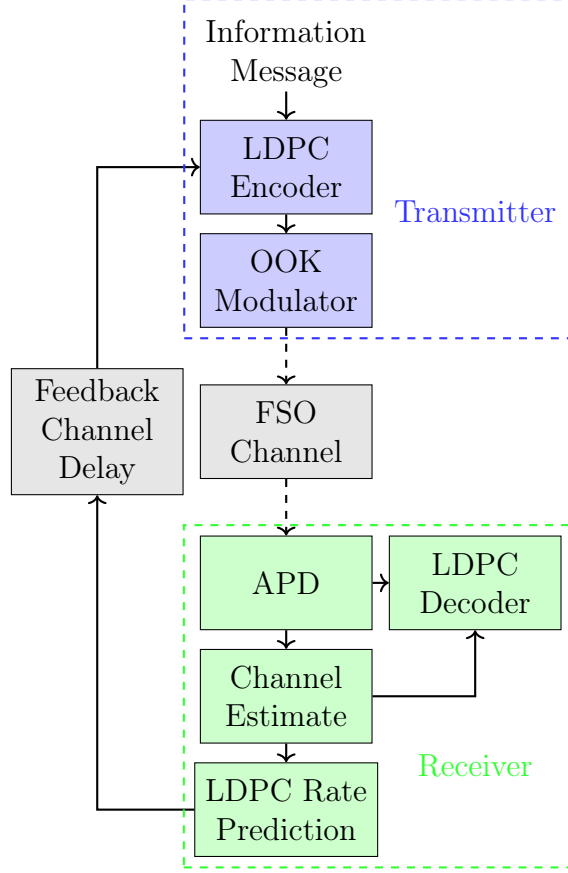


Figure 3.2: High-level system block diagram. The fading channel gain is estimated at the receiver and used to decode the current codeword as well as to predict the future channel gain and corresponding LDPC code rate based on the round-trip delay. The receiver selects the future LDPC code rate such that selected code rate achieves frame error rate (FER) lower than 10^{-6} for predicted channel gain value. The time required for the transmitter to receive a predicted codeword via the feedback channel is equal to half of the round-trip delay.

achieved by using these predictive models to select the LDPC code rate. Sec. 3.6 concludes this chapter.

3.2 System Architecture and Fading Channel Model

3.2.1 System Architecture

Fig. 3.2 describes the high-level system architecture. The receiver estimates the fading channel and uses it to decode the current codeword. The receiver also uses the estimated channel

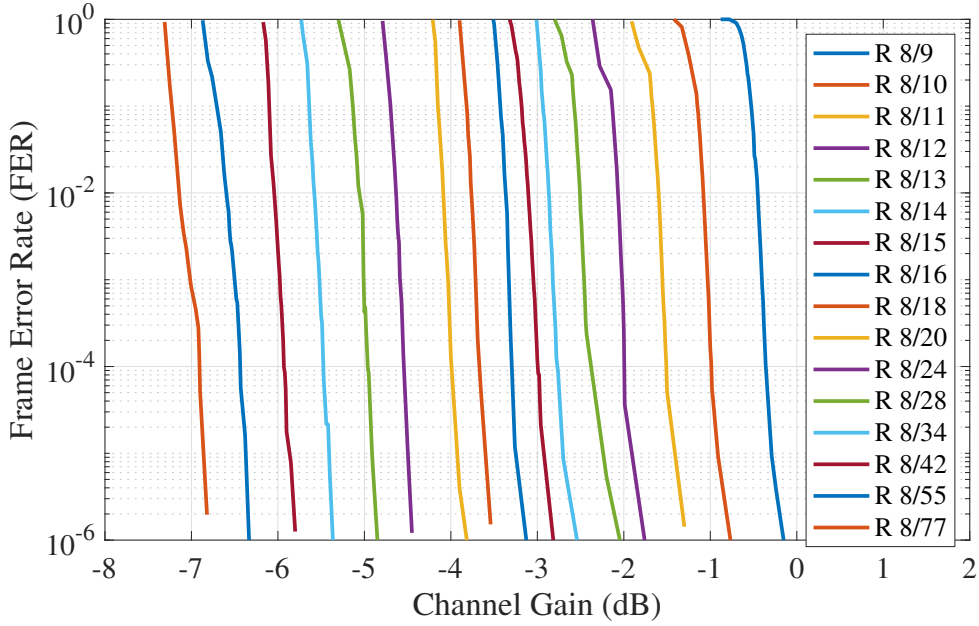


Figure 3.3: Frame Error Rate (FER) Vs. Channel Gain for LDPC code rates 8/9 to 8/77 in descending order from right to left.

gains to predict the future channel gain and corresponding LDPC code rate at a specified future time, which is based on the delay required to transmit the code rate to the transmitter. The receiver selects the future LDPC code rate such that selected code rate achieves frame error rate (FER) lower than 10^{-6} for predicted channel gain value.

Fig. 3.3 shows FER curves for a subset of the designed LDPC codes as a function of fading channel gain. None of our LDPC code rates suffer from an error floor above 10^{-6} . The highest code rate (8/9) is the rightmost curve.

The channel gain values corresponding to FERs 10^{-6} in Fig. 3.3 for each code rate are referred as “crossing points” and shown in Table 3.2 in the LDPC Simulation Crossing Point column. The crossing points in Table 3.2 are calculated by subtracting baseline average power on detector (P_{ave}) of -48.1 dBm from P_{ave} for which LDPC code rate achieves FER of 10^{-6} .

The selected LDPC code rate is sent back to the transmitter through an error-free feedback channel with round-trip feedback delay time t_d . The information message is generated at the transmitter side and encoded with LDPC encoder with rate equal to the code rate received

Table 3.2: LDPC codes with channel gain crossing points. Each code rate achieves FER of 10^{-6} for corresponding crossing point.

Code Rate	LDPC Simulation Crossing Point [dB]	Normal Approx. Crossing Point [dB]
8/9	-0.1522	-0.5550
8/10	-0.7672	-1.4700
8/11	-1.2802	-2.0800
8/12	-1.7596	-2.5400
8/13	-2.0459	-2.9200
8/14	-2.5409	-3.2500
8/15	-2.8154	-3.5300
8/16	-3.1276	-3.7800
8/18	-3.5267	-4.2100
8/20	-3.8154	-4.5000
8/24	-4.4457	-5.1600
8/28	-4.8492	-5.6300
8/34	-5.3644	-6.1900
8/42	-5.7939	-6.7700
8/55	-6.3336	-7.4800
8/77	-6.8036	-8.3200

via the delayed feedback channel.

3.2.2 Fading Channel Model

The channel model (given the fade power ρ) is an asymmetric Gaussian model based on experimentally measured gains in communications performance of a laboratory-based, free-space optical communications system using an avalanche photodiode detector (APD) at the receiver for signal detection [123]. Following [91] and [75], to compute realizations of temporal fading a sequence of independent and identically distributed (i.i.d.) Gaussian random numbers z_i is filtered by a low-pass finite impulse response (FIR) filter with the

frequency response described in (3.1).

$$H(f) = \sqrt{\sigma_L^2 \tau_0 \sqrt{\pi}} \exp \left(-\frac{1}{2} (\pi \tau_0 f)^2 \right). \quad (3.1)$$

The filtered z_i values produce a band-limited discrete sequence x_i of correlated Gaussian random variables with variance $\sigma_L^2 = \ln(\sigma_T^2 + 1)$, where σ_T^2 is the Power Scintillation Index (PSI) characterizing the depth of scintillations.

Generated samples x_i are transformed to obtain the desired log-normal distribution of correlated fading realizations ρ_{T_i} as follows:

$$\rho_{T_i} = \exp(x_i - \sigma_L^2/2). \quad (3.2)$$

Fig. 3.4 depicts fading channel realizations ρ_{T_i} in *dB* for turbulence coherence times of 5 *ms* and 10 *ms*.

The autocorrelation function of the log amplitude of scintillation is approximated as

$$R_\rho(\tau) = (\ln(\sigma_T^2 + 1))^2 \exp[-(\tau/\tau_o)^2], \quad (3.3)$$

where τ_0 denotes turbulence coherence time [123]. Turbulence coherence time represents a time interval during which the change in fading characteristics of the channel is very small.

Fig. 3.5 shows autocorrelation function for turbulence coherence time of 10 *ms* as a function of delay τ . As the delay increases, the most recent observed fade and the predicted fade are less and less correlated, and becoming completely uncorrelated when delay becomes longer than two times the coherence time. The less correlated the two fades are, the harder it is to predict the future fade.

The modulation scheme used is On-Off keying (OOK) such that each OOK slot contains either the signal (bit 1) or background noise (bit 0). Bit 1 is modulated to μ_1 and bit 0

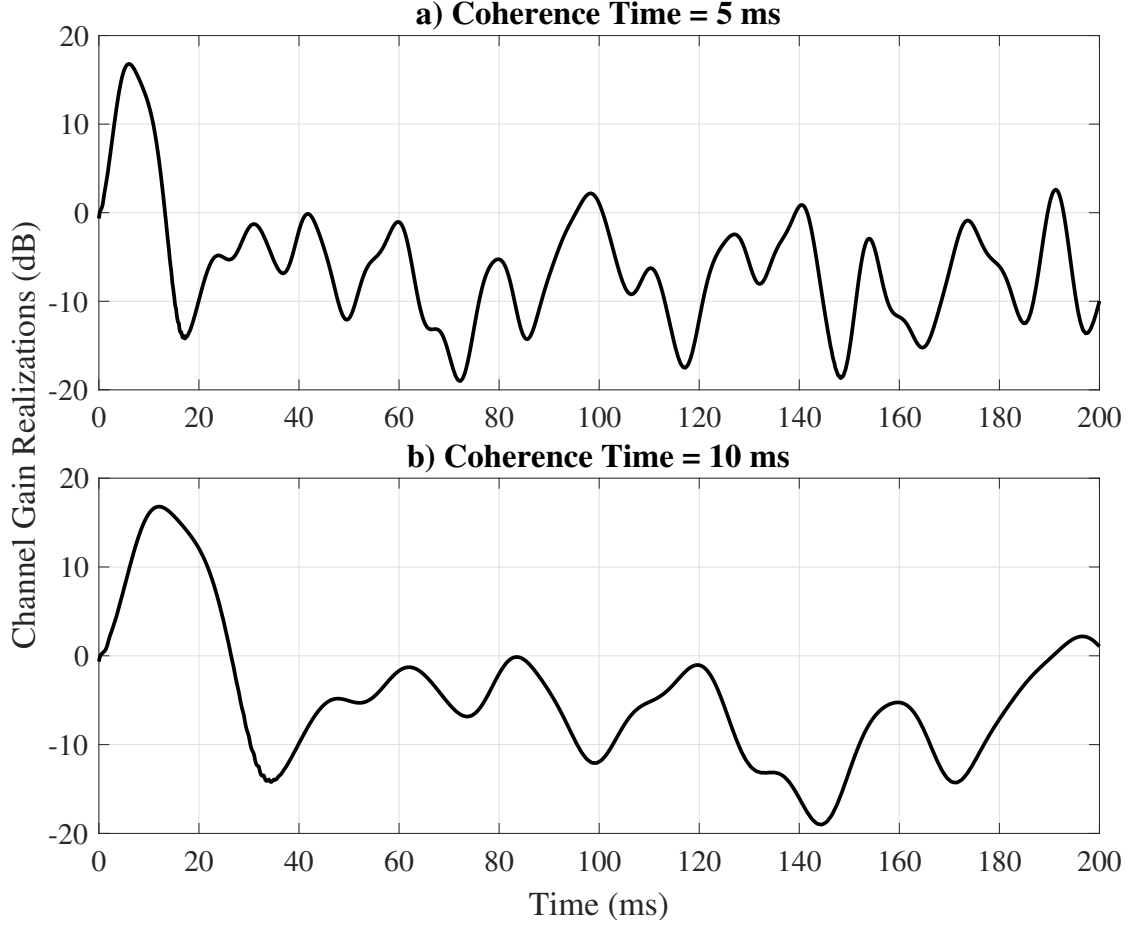


Figure 3.4: Fading channel gain realizations ρ_{T_i} in dB as a function of time for turbulence coherence time of a) 5 ms and b) 10 ms .

is modulated to μ_0 , where μ_1 is the signal current when the signal is ON and μ_0 is signal current when the signal is off. Additive white Gaussian noise (AWGN) is added to the signal so that observations for both signal (ON) for bit 1 and signal (OFF) for bit 0 are modeled using Gaussian distributions $\mathcal{N} \sim (\mu_1, \sigma_1^2)$ and $\mathcal{N} \sim (\mu_0, \sigma_0^2)$. Here, σ_1 and σ_0 denote total noise when the signal is on and off, respectively. Thus, the log-likelihood ratio (LLR) used by LDPC decoder is given by:

$$LLR = \frac{1}{2} \ln \frac{\sigma_0^2}{\sigma_1^2} + \frac{(y - \mu_0)^2}{2\sigma_0^2} - \frac{(y - \mu_1)^2}{2\sigma_1^2}. \quad (3.4)$$

Equations (3.5)–(3.12) describe the calculations of μ_1, σ_1, μ_0 and σ_0 . The meanings and

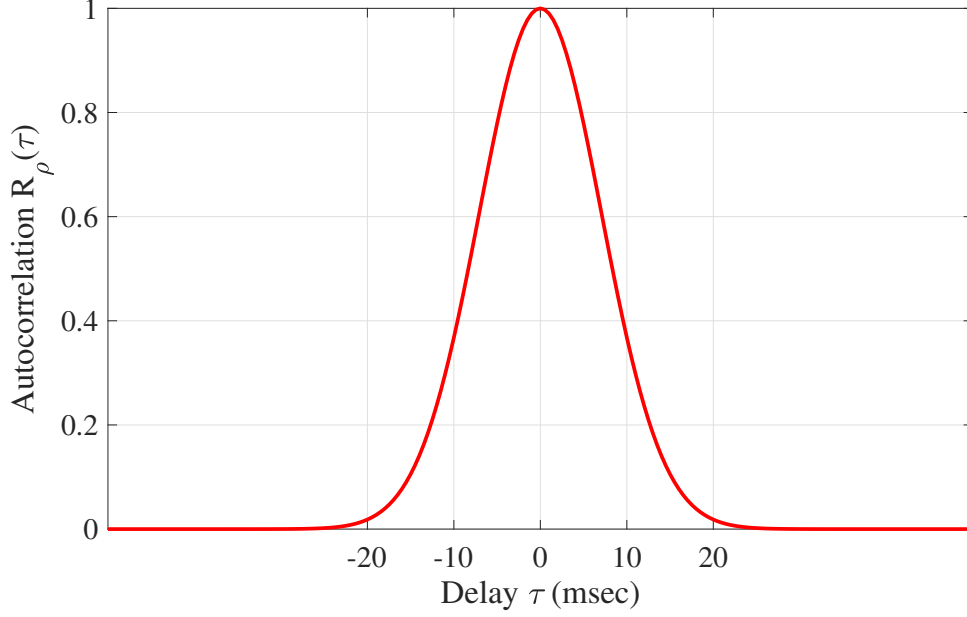


Figure 3.5: Autocorrelation function $R_\rho(\tau)$ for turbulence coherence time of $\tau_0 = 10 \text{ ms}$.

values of the variables and constants used in the equations are given in Table 3.3. Note that F is the APD excess noise factor, and $\rho(t)$ is the fading realization at time t .

$$\mu_1(t) = \frac{1}{2} \times \mathcal{R} \times G \times \rho(t) \times p_1 = \rho(t)\mu_1^* \quad (3.5)$$

$$\mu_0(t) = \frac{1}{2} \times \mathcal{R} \times G \times \rho(t) \times p_0 = \rho(t)\mu_0^* \quad (3.6)$$

$$F = k \times G + (1 - k) \times \left(2 - \frac{1}{G}\right) \quad (3.7)$$

$$S_1(t) = (2 \times q \times \mu_1(t) \times G \times F)^{\frac{1}{2}} \quad (3.8)$$

$$S_0(t) = (2 \times q \times \mu_0(t) \times G \times F)^{\frac{1}{2}} \quad (3.9)$$

$$M = (2 \times q \times F \times G^2 \times I_{dark})^{\frac{1}{2}} \quad (3.10)$$

$$\sigma_1(t) = [B \times ((S_1(t))^2 + M^2 + D_{TIA}^2)]^{\frac{1}{2}} \quad (3.11)$$

$$\sigma_0(t) = [B \times ((S_0(t))^2 + M^2 + D_{TIA}^2)]^{\frac{1}{2}} \quad (3.12)$$

Table 3.3: Constants and variables used to calculate signal current μ_1 and total noise σ_1 when signal is ON, and signal current μ_0 and total noise σ_0 when signal is OFF.

Symbol	Quantity	Value	Unit
q	electron charge	1.6×10^{-19}	C
G	Avalanche Photodiode gain APD	15	
$\mathcal{R}$	responsivity	0.9	A/W
B	electronic filter bandwidth	1250	MHz
k	impact coefficient	0.2	
I_{dark}	dark current	2.1×10^{-8}	A
D_{TIA}	Transimpedance Amplifier (TIA) input noise current density	2×10^{-12}	$\frac{A}{\sqrt{Hz}}$
P_{ave}	average power on detector	-48.1	dBm
r_{ext}	extinction ratio	11	dB
P_1	power on detector when bit 1 is sent	$P_{ave} + \frac{r_{ext}}{2}$	dBm
P_0	power on detector when bit 0 is sent	$P_{ave} - \frac{r_{ext}}{2}$	dBm
p_1	power on detector when bit 1 is sent	$10^{P_1/10}$	mW
p_0	power on detector when bit 0 is sent	$10^{P_0/10}$	mW

The electronic filter bandwidth B value of 1250 MHz is used for simulation purposes, but it can be higher in practice. Note that for the baud rate of 2.5 Giga symbols per second used in this chapter, the time occupancy of each codeword ranges from 3.6864 microseconds (μs) for the highest code rate (8/9) to 31.539 μs for the lowest code rate (8/77), which is relatively small compared to the turbulence coherence times of 5 ms and 10 ms . The fading model generates one fade value for every 1024 bits, which means that different sections of a codeword will experience a different fade. However, since the turbulence coherence time is

much longer than the time occupancy of a codeword, these differences are negligible.

3.3 Normal Approximation of Frame Error Rate

For FSO On-Off Keying (OOK) with equally likely transmission of bits 1 and 0, consider the following channel model when bit 1 (On) or bit 0 (Off) is transmitted:

$$y = \mu_i + \sigma_i n, \quad i = 0 \text{ or } 1, \quad (3.13)$$

where n is a zero mean, unit variance normal (Gaussian) random variable. The mean μ_i already includes the fading $\rho(t)$ as described in Sec. 3.2.2. The received value y is a Gaussian random variable with probability density function $f_i(y)$ which is normal distributed $\mathcal{N}(\mu_i, \sigma_i^2)$.

When bit 1 is transmitted the channel information density is

$$i_1(y) = 1 - \log_2 \left(1 + \frac{\sigma_1}{\sigma_0} e^{-\frac{1}{2\sigma_0^2}(y-\mu_0)^2 + \frac{1}{2\sigma_1^2}(y-\mu_1)^2} \right) \quad (3.14)$$

and its n th moment after change of variable is

$$m_n(i_1) = \int_{-\infty}^{\infty} f_1(y) i_1^n(y) dy = \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}} i_1^n(z) dz,$$

where

$$i_1(z) = 1 - \log_2 \left(1 + \frac{\sigma_1}{\sigma_0} e^{-\frac{1}{2\sigma_0^2}(\sigma_1 z + \mu_1 - \mu_0)^2 + \frac{1}{2} z^2} \right).$$

When bit 0 is transmitted the channel information density is

$$i_0(y) = 1 - \log_2 \left(1 + \frac{\sigma_0}{\sigma_1} e^{-\frac{1}{2\sigma_1^2}(y-\mu_1)^2 + \frac{1}{2\sigma_0^2}(y-\mu_0)^2} \right) \quad (3.15)$$

and its n th moment after change of variable is

$$m_n(i_0) = \int_{-\infty}^{\infty} f_0(y) i_0^n(y) dy = \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}} i_0^n(z) dz ,$$

where

$$i_0(z) = 1 - \log_2 \left(1 + \frac{\sigma_0}{\sigma_1} e^{-\frac{1}{2\sigma_1^2}(\sigma_0 z + \mu_0 - \mu_1)^2 + \frac{1}{2}z^2} \right) .$$

The average of channel information density with equally likely channel inputs of zero or one is

$$I = \frac{1}{2} [m_1(i_1) + m_1(i_0)] \quad (3.16)$$

and the channel dispersion with equally likely channel inputs of zero or one is

$$V = \frac{1}{2} [m_2(i_1) + m_2(i_0)] - I^2. \quad (3.17)$$

Both $I(P_{ave})$ and $V(P_{ave})$ are functions of the average received power P_{ave} at APD.

Using the Normal Approximation (NA) by Polyanskiy et al. [94], the maximal achievable rate can be approximated by

$$R^*(n, FER) = I - \sqrt{\frac{V}{n}} Q^{-1}(FER) + O\left(\frac{\log_2 n}{n}\right), \quad (3.18)$$

where $Q^{-1}(\cdot)$ denotes inverse of the Gaussian Q-function which is

$$Q(x) = \int_x^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}y^2} dy. \quad (3.19)$$

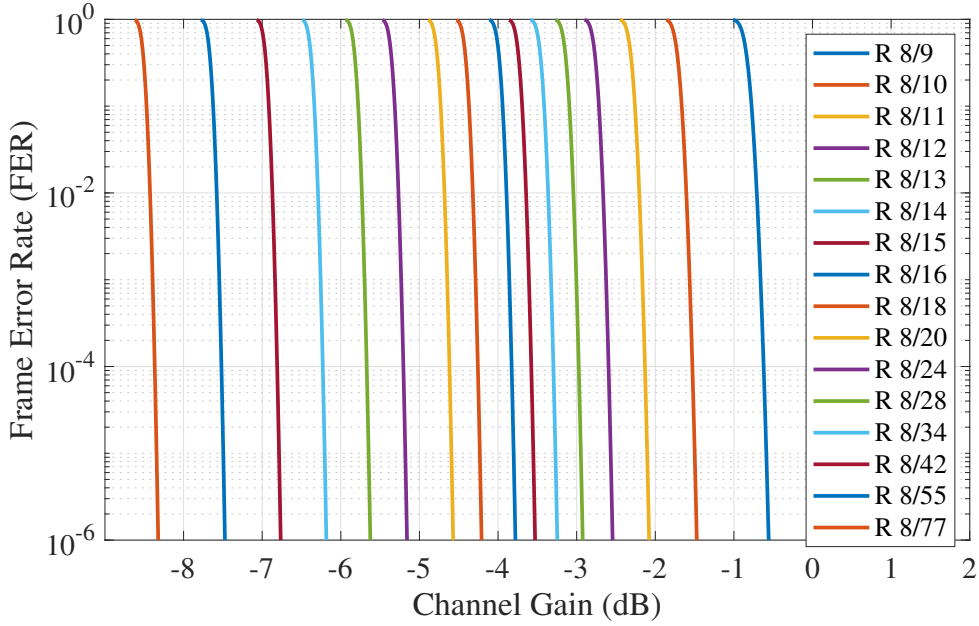


Figure 3.6: Frame Error Rate (FER) vs Channel Gain using Normal Approximation (NA).

Then the normal approximation on frame error rate FER_{NA} can be calculated as

$$\text{FER}_{\text{NA}}(P_{\text{ave}}) = Q \left(\frac{I(P_{\text{ave}}) - R + \log_2(n)/2n}{\sqrt{V(P_{\text{ave}})/n}} \right), \quad (3.20)$$

where R represent the code rate, $n = k/R$ is codeword block length, k is the message block length, and $O(\frac{\log_2 n}{n}) \approx \log_2(n)/2n$. However, for $k = 8192$ the same FERs have been obtained for all code rates by ignoring the $O(\cdot)$ term.

As an example, consider a laser with extinction ratio of 11 dB, and an APD detector, such that observations can be expressed as, $\mu_1 = \alpha_1 P_{\text{ave}}$, $\mu_0 = \alpha_0 P_{\text{ave}}$, $\sigma_1^2 = \beta_1 P_{\text{ave}} + \gamma_1$, and $\sigma_0^2 = \beta_0 P_{\text{ave}} + \gamma_0$, where $\alpha_1 = 47.9$, $\alpha_0 = 3.8$, $\beta_1 = 1.3 \times 10^{-6}$, $\beta_0 = 1 \times 10^{-7}$ and $\gamma_1 = \gamma_0 = 1.36 \times 10^{-14}$. In this chapter block fading is considered where the fade power ρ is constant over duration of codeword. This assumption is valid when the coherence time of fading is larger than duration of codeword. The fading power is normalized such that $E\{\rho\} = 1$.

In the fading channel model, P_{ave} is then replaced with ρP_{ave} . For atmospheric fading

the Power Scintillation Index $PSI = \sigma_T^2 = 10$ is assumed.

The FERs using normal approximation for rates 8/9 to 8/77 are plotted in Fig. 3.6. The crossing points representing fading channel gain values at which FERs for normal approximation are 10^{-6} are listed in Table 3.2. For LDPC simulations, crossing points are computed based on the reference point average P_{ave} of -48.1 dBm.

Fig. 3.7 presents a comparison between the crossing points derived from LDPC simulations and those obtained through the normal approximation. The gap between the simulated LDPC code FER crossing points and those resulting from the normal approximation of random code performance ranges from 0.4 dB to 1.5 dB. The close proximity of the normal approximation to the actual LDPC code performance adds confidence that the LDPC codes are well-designed across the entire range of rates.

One way to see how adaptively adjusting the rate can improve performance for slow fading is to compute the FER_{fixed} that a fixed-rate system would provide. To compute the theoretical performance of a fixed-rate scheme in slow fading without feedback, we integrate the product of the density $f(\rho)$ of ρ from [75] and the FER from (3.20) (denote that by $F(P_{ave})$) with P_{ave} replaced with ρP_{ave} for a fixed-rate random code, as shown below:

$$FER_{\text{fixed}} = \int_{\rho=0}^{\infty} FER_{\text{NA}}(\rho P_{ave}) f(\rho) d\rho, \quad (3.21)$$

assuming that the pdf $f(\rho)$ of fade power ρ is normalized such that $E\{\rho\} = 1$. Such a computation reveals that the FER performance for the fixed-rate scheme incurs a huge performance loss.

3.4 Low-Rate Protograph-based LDPC Codes Design

This chapter uses the Protograph-Based Raptor-Like (PBRL) [23] approach to design LDPC codes with information blocklength $k = 8192$ and parity check matrix $\mathbf{H}$ described by (3.22)

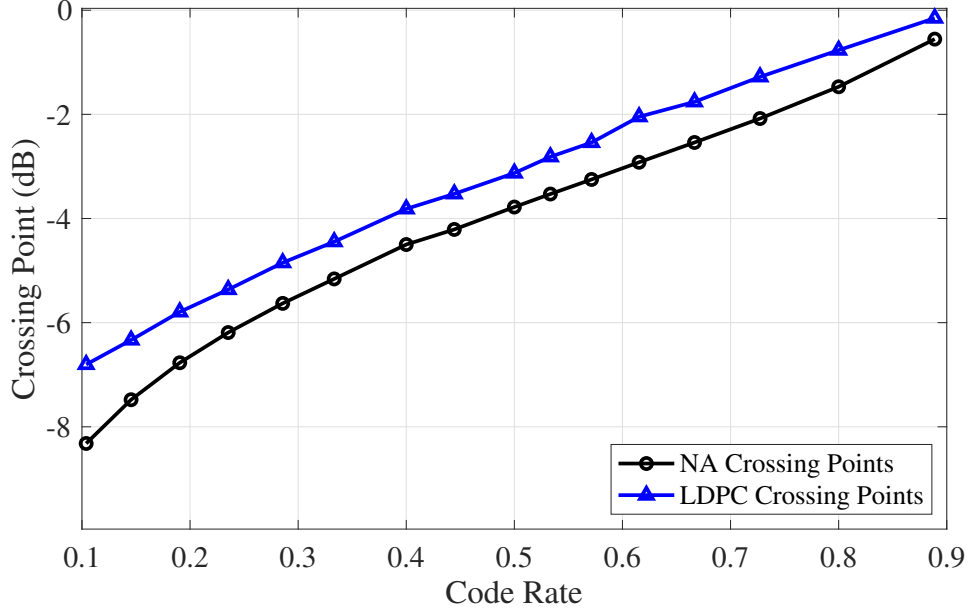


Figure 3.7: The crossing points in Table 3.2 are compared with crossing points using normal approximation (NA).

for wide range of rates:

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_{\text{HRC}} & \mathbf{0} \\ \mathbf{H}_{\text{IRC}} & \mathbf{I} \end{bmatrix}. \quad (3.22)$$

Let n_1 represent the number of variable nodes in $\mathbf{H}_{\text{HRC}}$, of which n_p are punctured to improve the iterative decoding threshold. Let m_1 represent the number of rows in $\mathbf{H}_{\text{IRC}}$ matrix. In equation (3.22) submatrix $\mathbf{H}_{\text{HRC}} \in \mathbb{F}_2^{(n_1-k) \times n_1}$ represents a highest-rate code (HRC) and submatrix $\mathbf{H}_{\text{IRC}} \in \mathbb{F}_2^{m_1 \times n_1}$ represents an incremental redundancy code (IRC). PBRL LDPC code supports rates from $\frac{k}{n_1-n_p}$ to $\frac{n_1-k}{n_1-n_p}$. The rate is selected by deciding how many of the degree-1 variable nodes associated with identity matrix in (3.22) should be punctured. This work designs the $\mathbf{H}_{\text{IRC}}$ to support the lowest code rate of 1/10. Thus, $m_1 = 72704$ and $n_1 = 9216$.

The parity check matrix $\mathbf{H}_{\text{HRC}}$ for the highest rate code is obtained by lifting a proto-

matrix. The proto-matrix $\mathbf{H}_{\text{HRC-proto}}$ in (3.23) is adopted from [91].

$$\mathbf{H}_{\text{HRC-proto}} = \begin{bmatrix} 4 & 4 & 4 & 4 & 4 & 3 & 3 & 3 & 3 \end{bmatrix} \quad (3.23)$$

Unlike LDPC codes that start with a designed lowest rate code and increase the rate by randomly puncturing variable nodes hoping to not degrade performance, PBRL design starts with a well-designed highest-rate code and obtains lower-rate codes by carefully selecting the rows of $\mathbf{H}_{\text{IRC}}$.

The design of $\mathbf{H}_{\text{IRC}}$ is completed in two steps. In the first step, the proto-matrix $\mathbf{H}_{\text{IRC}}$ [52] is designed line by line in a greedy fashion by minimizing decoding threshold of newly constructed protograph matrix computed using the reciprocal channel approximation (RCA) algorithm [23]. The RCA is a fast and accurate approximation to the density evolution algorithm with deviation in accuracy of less than 0.01 dB. The decoding threshold of a protograph matrix refers to the minimum channel noise that supports reliable iterative decoding of LDPC codes with infinite code length built from the protograph. The fully designed protograph matrix of $\mathbf{H}$ for rate 1/10 consists of 72 rows (check nodes) and 80 columns (variable nodes).

In the second step, the designed protograph matrix for lowest code rate of 1/10 is lifted using approximate-cycle extrinsic-message-degree (ACE) progressive-edge-growth (PEG) algorithm [111] to replace each element in protograph matrix with circulant matrices and obtain parity check matrix $\mathbf{H}$ with longer block-length. The ACE-PEG algorithm with parameters of $d_{\text{ACE}} = 6$ and $\eta = 7$ are selected to ensure that all the cycles in the lifted parity check matrix whose length is 12 or less have ACE values of at least 7.

The lifting process also consists of two steps. In the first step, lifting number is 4 to remove parallel edges in protograph matrix. In the second step, lifting number is 256 which gives a parity check matrix with information blocklength of 8192 bits. Fig. 3.3 shows FER as a function of fading channel gain for a subset of the designed LDPC code rates. Fig. 3.7

compares performance between designed LDPC codes and the normal approximation and random coding for FER of 10^{-6} .

3.5 LDPC Rate Selection To Maximize Throughput

This section presents different predictive models for selecting LDPC code rate based on the knowledge of channel gain and feedback time delays. The fading channel gains are estimated at the receiver and used to predict a future channel gain considering the round-trip delay required to transmit the feedback signal from the receiver back to the transmitter and new signal from the transmitter to the receiver. The receiver uses predicted channel gain values to select LDPC code rate that achieves FER lower than 10^{-6} for predicted channel gain. For the purpose of analysis, out of 72 designed code rates a subset of 16 code rates with approximate differences of 0.5 *dB* at crossing points at FER of 10^{-6} is selected. The channel gain crossing points for which each LDPC code rate decodes a codeword with FER of 10^{-6} are given in Table 3.2.

3.5.1 Instantaneous Channel State Information

As a baseline for comparison, we consider the case where the feedback delay is zero and current channel state is known. The LDPC code rate is selected to maximize throughput, i.e. the code rate selected is the highest code rate that achieves FER below 10^{-6} for the current known channel state. Actual channel gain data is represented with black curve in Fig. 3.4 for coherence times of 5 and 10 *ms*. Throughput achieved when the transmitter knows the CSI with no delay is referred as zero-delay throughput and it is used as a reference to evaluate the performance of prediction models when feedback delay is not zero.

Fig. 3.8 shows throughput as a function of time for optical channel with coherence time of 10 *ms* for a) rate-adaptive system when CSI is known at the transmitter with no delay, referred as zero-delay throughput, b) when single LDPC code rate 8/9 is used, and c) when

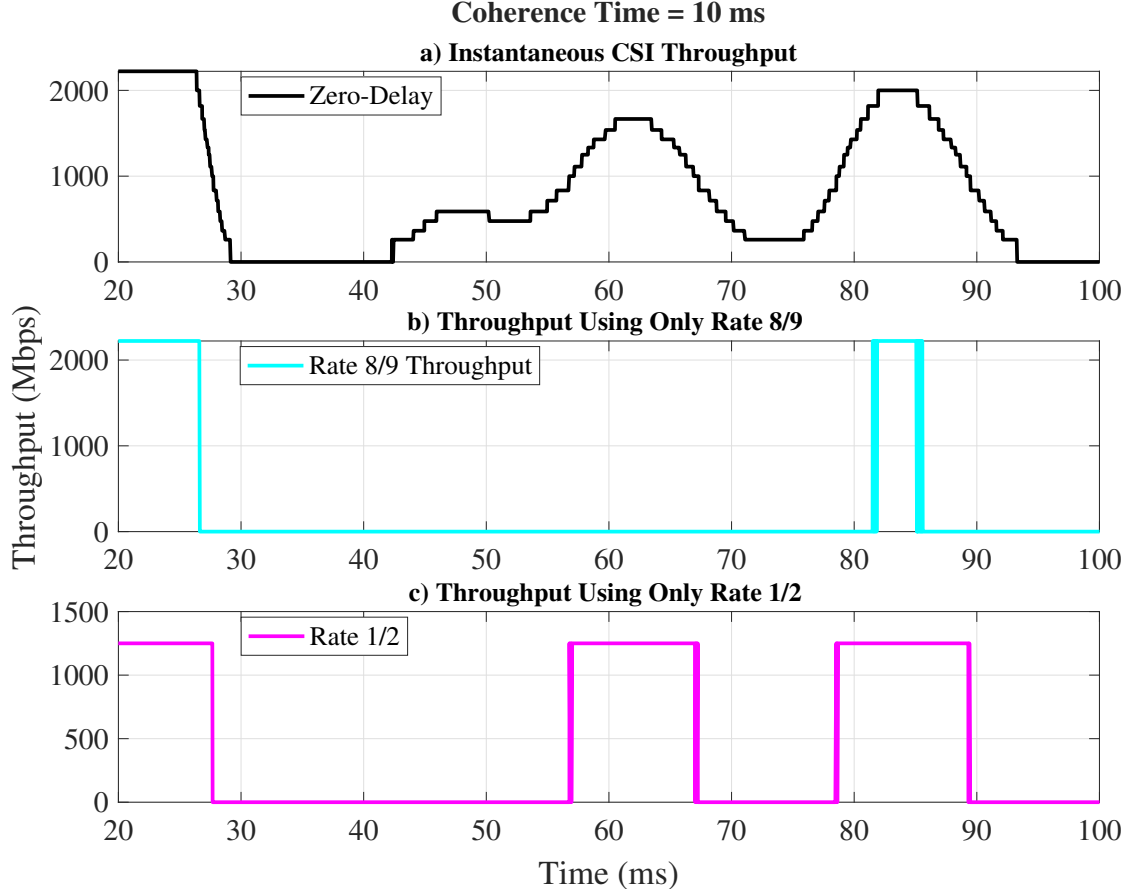


Figure 3.8: Throughput as a function of time for optical channel with coherence time of 10 *ms*. a) Throughput achieved using rate-adaptive LDPC coding when channel state information (CSI) is known. b) Throughput achieved when using LDPC code rate 8/9 only. c) Throughput achieved when using LDPC code rate 1/2 only.

single LDPC code rate 1/2 is used. The simulations, performed over 100 *ms* of data, indicate that throughput achieved when using single LDPC code rate is approximately 55.9% and 60.3% of zero-delay throughput for rates 8/9 and 1/2 respectively. Note that the optical channel realizations used for simulations consist of 30% of very good turbulence conditions boosting the performance of both 8/9 and 1/2 rates. As turbulence conditions become worse, the performance of using a fixed single rate drastically decreases, which is evident between 30 *ms* and 55 *ms* in Fig. 3.8 where both rates suffer from constant failure, making the throughput zero for a significant fraction of the simulated time segment.

3.5.2 Delayed Channel State Information

Now we consider the practical scenario in which the feedback delay is not zero.

CSI Estimation

Based on practical experience of the optical fading channel, our fading model assumes that fading is constant over 1024 OOK symbols. Even between groups of 1024 OOK symbols the fading changes very slowly. Because of this slow change in the fading, estimation of the fading at the receiver is not the key issue, the ability to predict future fading based on the knowledge of the current fading is the critical issue.

A simple fading channel gain estimation $\hat{\rho}(t)$ at the receiver is performed as follows. From equations (3.5) and (3.6) $\mu_1^* = \frac{1}{2} \times R \times G \times p_1$ and $\mu_0^* = \frac{1}{2} \times R \times G \times p_0$. Then $\mu_1(t) = \mu_1^* \times \hat{\rho}(t)$ and $\mu_0(t) = \mu_0^* \times \hat{\rho}(t)$. For a single value of $\hat{\rho}(t)$ we see 1024 observations of $\mu_1(t)$ and $\mu_0(t)$. In our estimation model we assume that out of 1024 bits there is equal possibility of getting bit 1 or bit 0, i.e. we generate 1024 Bernoulli distributed zeros and ones². The receiver sees 1024 different observations of noisy signal $y(t) = \mu_i(t) + Z\sigma_i$, where $i \in \{0, 1\}$ and Z is Gaussian random variable with zero mean and unit variance. Then estimated $\hat{\rho}(t)$ is given by

$$\hat{\rho}(t) = \frac{\frac{1}{512} \sum_{j=1}^{1024} y_j(t)}{\mu_1^* + \mu_0^*}. \quad (3.24)$$

To improve the estimation of the noisier channels, once $\hat{\rho}(t)$ is obtained by (3.24) we average 9 adjacent measurements and use that value for $\hat{\rho}(t)$. Note that 9 channel gain samples, each corresponding to 1024 bits, represent the duration of the codeword with the highest code rate of 8/9.

²The generation of received noisy sequences of 1024 using Bernoulli distributed zeros and ones can be justified since in optical transmitter and specifically in the CCSDS optical standard usually a Pseudo-Randomizer is used at transmitter (see [26]). In addition, in the optical standard, Gold code sequences are used for frame synchronization (frame synch marker) with almost balanced zeros and ones with a difference of 1 between the two counts, which can also be used for CSI estimation provided that the coherence time is longer than the duration of synch marker and a codeword.

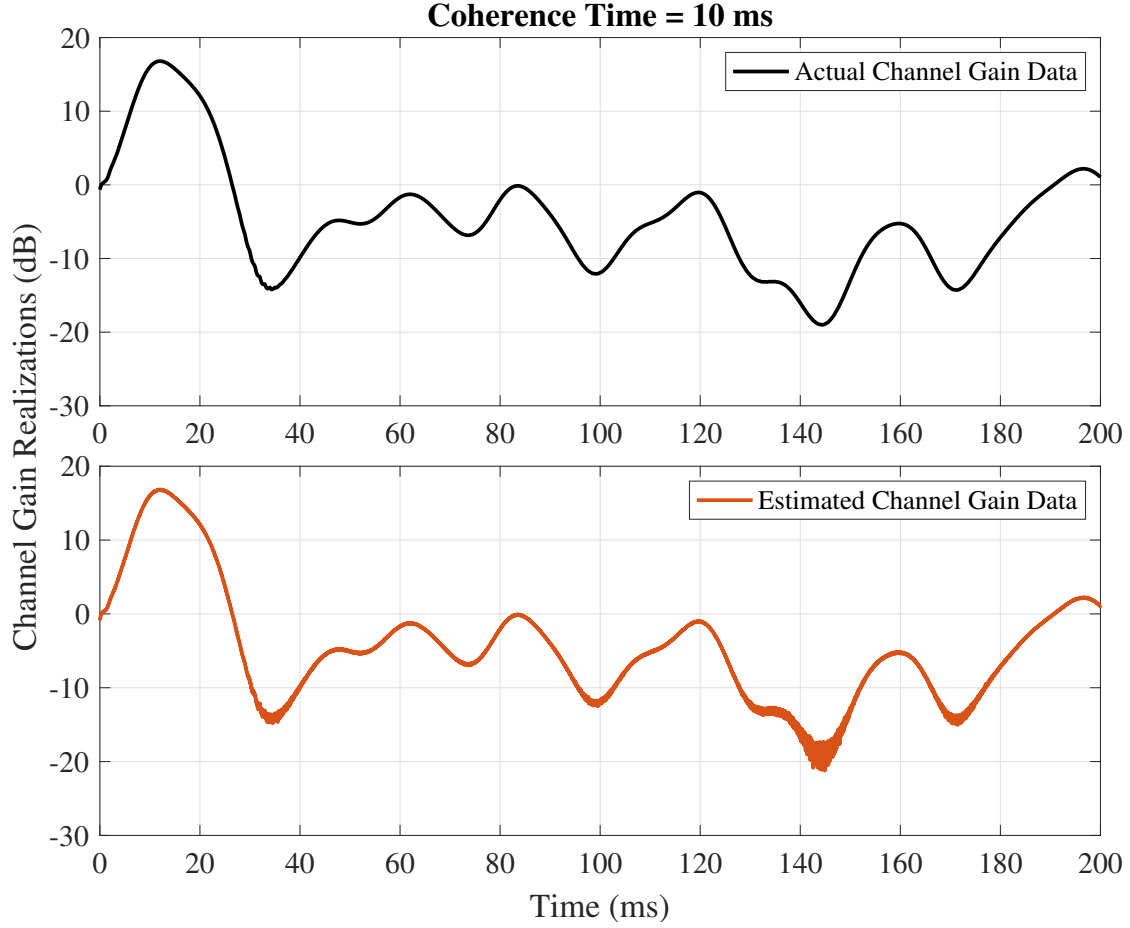


Figure 3.9: Comparison of actual and estimated fading channel gain data for coherence time of 10 *ms*.

Fig. 3.9 shows actual fading channel gain data generated in MATLAB using fading channel model in Sec. 3.2.2 and estimated fading channel gain data at the receiver given by (3.24). When the channel is relatively good (e.g. above -10 *dB*), the received signal is less affected by noise, making the channel estimation more reliable and closer to the actual channel gain values resulting in low estimation error. When the channel is poor (e.g. below -10 *dB*), the received signal is heavily corrupted by noise. This leads to higher uncertainty in the estimation process, causing the estimated fading channel gains to deviate significantly from the actual values and appear noisy which can be observed near 35 *ms*, 140 *ms* and 170 *ms* in Fig. 3.9.

Zero-Order Prediction

The zero-order prediction model predicts fading channel gain value in the future to be the same as the current channel estimate at the receiver. Let fading channel gain value estimated at the receiver at time t_k be $\hat{\rho}_k$ in dB and let t_d denote the feedback channel delay time. The predicted channel gain value $\tilde{\rho}_{k+d}$ at time $(t_k + t_d)$ is the same as estimated value $\hat{\rho}_k$.

Linear Prediction

Let $\hat{\boldsymbol{\rho}} = [\hat{\rho}_1, \hat{\rho}_2, \dots, \hat{\rho}_n]$, $\hat{\rho}_i \in \mathbb{R}$ represent fading channel gain values estimated by the receiver. Let $\mathbf{t} = [t_1, t_2, \dots, t_n]$, $t_i \in \mathbb{R}$ represent time instances that correspond to the estimated fading channel gain values in $\hat{\boldsymbol{\rho}}$. Note that $\hat{\rho}_n$ corresponds to the current time instance t_n , and all other samples in $\hat{\boldsymbol{\rho}}$ are past samples estimated at the receiver. The receiver only has knowledge of the past fading channel gain data measured at the receiver. In order to make a prediction of channel gain in the future, the estimated n samples is used to fit a polynomial of a form:

$$p(t) = x_1 + x_2 t + \dots + x_{m+1} t^m. \quad (3.25)$$

For each polynomial coefficient x a vector of errors $\mathbf{e} = [p(t_1) - \hat{\rho}_1, p(t_2) - \hat{\rho}_2, \dots, p(t_n) - \hat{\rho}_n]$ is formed. As described in [14], to find a polynomial that minimizes the norm of the error vector $\mathbf{e}$ following norm approximation problem is solved:

$$\min_{\mathbf{x}} \|\mathbf{e}\| = \|\mathbf{A}\mathbf{x} - \hat{\boldsymbol{\rho}}\|, \quad (3.26)$$

where $A_{ij} = t_i^{j-1}$, $i = 1, 2, \dots, n$, $j = 1, 2, \dots, (m+1)$. MATLAB function *polyfit* is used to solve problem in (3.26) to obtain polynomial coefficients $\mathbf{x}$. Once the polynomial model is fitted, its coefficients are used to predict future fading channel gain $\tilde{\rho}_{t_k}$ at some time instance t_k by plugging t_k into (3.25). The coding rate is predicted based on this predicted fading

channel gain value. In the simulations when the codeword sent at this rate arrives at the receiver it will experience fade coming from the actual fading channel data which receiver never saw before. Note that the polynomial coefficients change for every prediction as the receiver keeps retraining the model in an “online” manner based on the new data it receives. Since the receiver does not have knowledge of the future true data, there is no testing or validation data. The only way to know if the prediction was accurate is by observing if the codeword that was sent at the predicted rate was successfully decoded.

For linear prediction model, polynomial degree is $m = 1$ and number of samples used in fitting/training is $n = 1220$.

In this chapter, the receiver starts prediction calculations upon the receipt of the first codeword which code rate is chosen to be 8/16 for simulation purposes. Since each fade represents 1024 bits, the receiver will estimate 16 channel gain values. This is the smallest number of samples used to fit a polynomial to predict a future channel gain. As more samples arrive at the receiver, each new prediction is modeled using more samples. For the purpose of evaluating prediction models the codewords received until first codeword encoded with predicted rate arrives at the receiver are considered as pilots and not used for analysis.

Training Samples Selection

The number of channel gain samples used in the prediction models is determined using a greedy optimization approach by solving the following minimization problem:

$$\min_j MSE = \frac{1}{N} \sum_{i=1}^N (\hat{\rho}_i - \tilde{\rho}_i^{(j)})^2, \quad j = 1, 2, \dots, l \in \mathbb{N}, \quad (3.27)$$

where $\hat{\rho}_i$ are estimated fading channel gain values at the receiver, $\tilde{\rho}_i^{(j)}$ are predicted channel gain values solving (3.26), N is the number of data points and j represents the index of vector with length l containing number of data points in ms starting from 0.5 ms up to coherence time of optical channel in increments of 0.5 ms . The number of data points N used in (3.27)

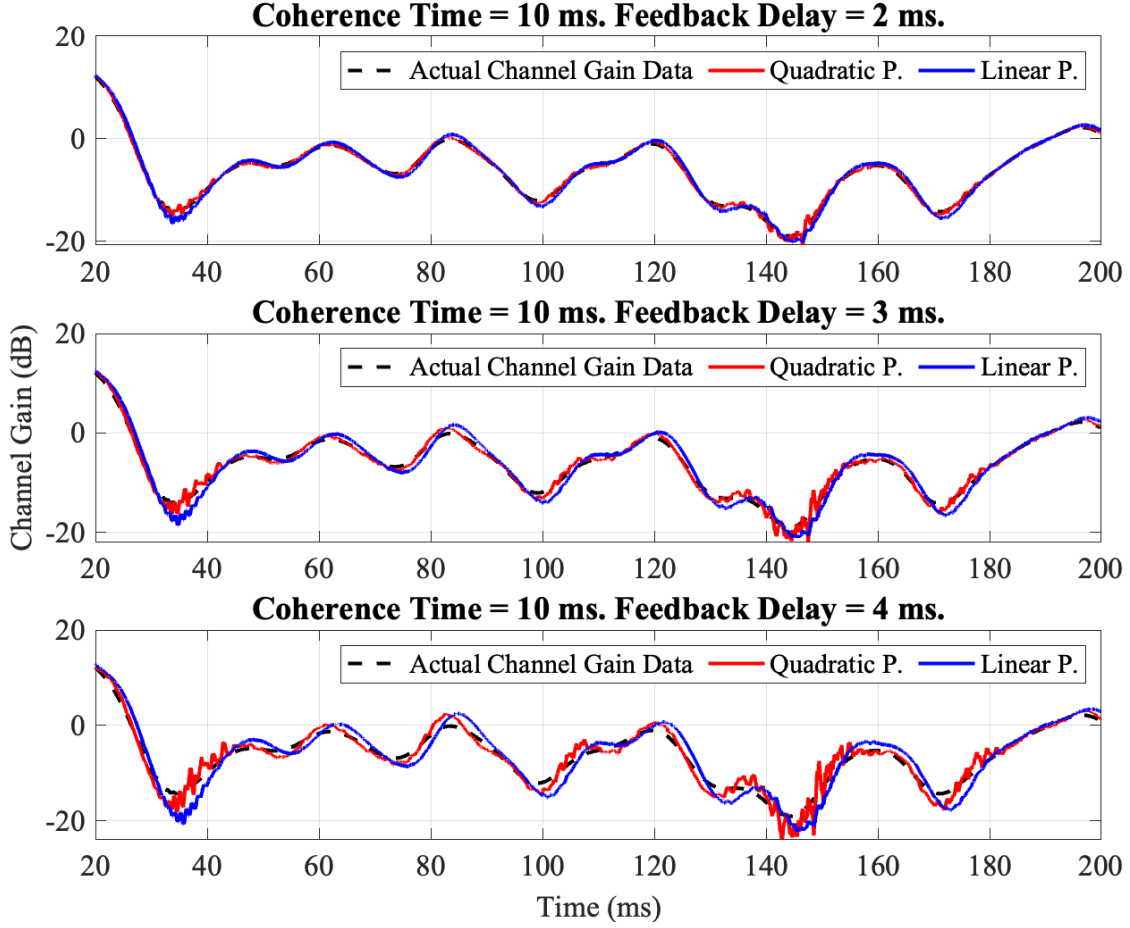


Figure 3.10: Comparison of channel gain values for quadratic and linear prediction with respect to the actual fading channel gain values when coherence time is 10 *ms* and feedback delay ranges from 2 *ms* to 4 *ms*.

corresponds to number of predicted $\hat{\rho}_i^{(j)}$ in (3.26) over duration of 200 *ms* for every j . For example, for a channel coherence time of 5 *ms*, j would range from 1 to 10, where index $j = 1$ represents 0.5 *ms* worth of data points and index $j = l = 10$ represents 5 *ms* worth of data. The solution of (3.27) for the linear prediction model suggests that 0.5 *ms* worth of data gives the best fitting performance for both coherence times considered.

Quadratic Prediction

The quadratic prediction model uses $m = 2$ in (3.25), using past and current estimated CSI in a quadratic model to predict future CSI for LDPC code rate selection. The number of

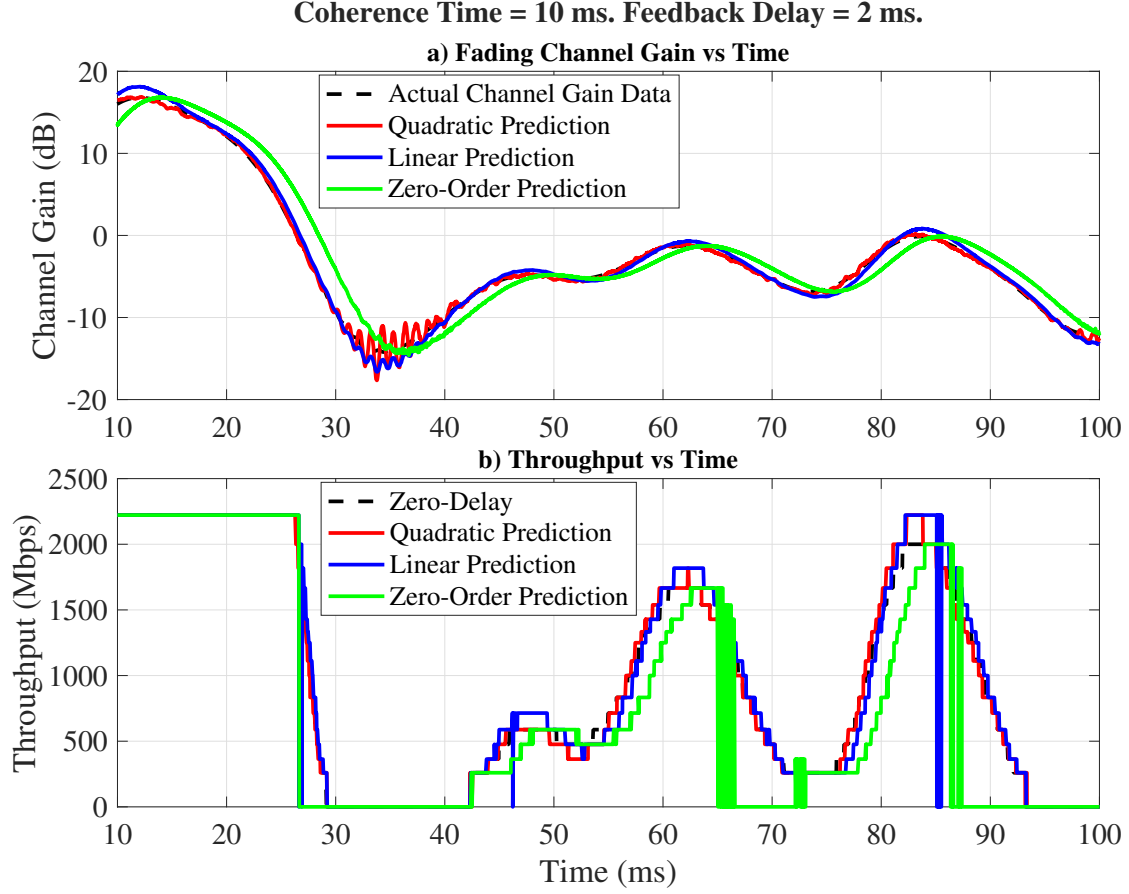


Figure 3.11: a) Actual and predicted channel gain values vs. time. b) Throughput vs. time obtained using zero-delay, zero-order, linear and quadratic prediction models for fading channel conditions in a). Turbulence coherence time is 10 *ms*, feedback delay is 2 *ms* and baud rate is 2.5 *Gbps*.

channel gain samples used in quadratic prediction is equal to 1 *ms* and 2 *ms* for optical channels with coherence times of 5 *ms* and 10 *ms* respectively. These values are obtained by solving (3.27). For all prediction models, we observed that adding a small margin of 0.25 *dB* to the original crossing points determined in Table 3.2 when selecting LDPC code rate at the receiver improved our FER performance.

Fig. 3.10 compares channel gain data obtained using linear and quadratic prediction models when feedback delay increases from 2 *ms* to 4 *ms* with actual channel gain data over the time interval of 10 *ms* to 100 *ms*. As feedback delays increases, the prediction becomes worse. Quadratic prediction becomes noisy when the channel is very bad, mainly due to

noisy fading channel gain estimation in this area.

Fig. 3.11 a) shows actual and predicted channel gain values for zero-order, linear and quadratic prediction when turbulence coherence time is 10 *ms* and feedback channel delay is 2 *ms*. Fig. 3.11 b) shows throughput as a function of time obtained using prediction models in Fig. 3.11 a). Linear and quadratic prediction models give similar performance as baseline zero-delay model, while zero-order prediction show loss in throughput compared to other models.

3.5.3 Computational Complexity

Polynomial regression fits a polynomial function of degree m to a data set. For linear ($m = 1$) and quadratic ($m = 2$) prediction models, the computational complexity depends on the number of data points n and polynomial degree m . The computational complexity of the prediction models involves the following steps:

- Complexity $\mathcal{O}(nm)$: Input data are transformed into polynomial features. For linear prediction ($m = 1$) this includes t_i term, and for quadratic prediction ($m = 2$) this includes t_i and t_i^2 terms, where $i = 1, 2, \dots, n$. We are not counting column of all ones in matrix $\mathbf{A}$.
- Complexity $\mathcal{O}(nm^2 + nm + m^3)$: The least squares solution involves solving:

$$\mathbf{x} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \hat{\boldsymbol{\rho}}.$$

- Compute $\mathbf{A}^T \mathbf{A}$ (matrix multiplication): $\mathcal{O}(nm^2)$
- Compute the inverse of $\mathbf{A}^T \mathbf{A}$: $\mathcal{O}(m^3)$
- Compute $\mathbf{A}^T \hat{\boldsymbol{\rho}}$: $\mathcal{O}(nm)$

Total complexity for both prediction models is $\mathcal{O}(nm^2 + 2nm + m^3)$. For the linear prediction model, $n = 1220$ for both coherence times of 5 *ms* and 10 *ms*. For the quadratic prediction

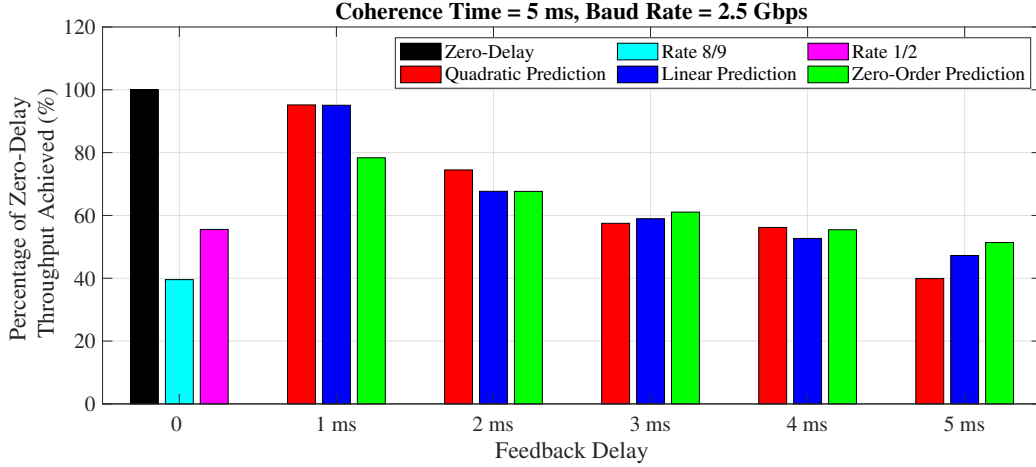


Figure 3.12: Percentage of zero-delay throughput achieved using prediction models as a function of feedback delay when coherence time is 5 *ms* and feedback delay ranges from 1 *ms* to 5 *ms*. Quadratic prediction model gives the best performance for shorter feedback delay times (1 *ms* and 2 *ms*). Zero-Order prediction model gives the best performance for longer feedback delay times. Simulation is performed for 80 *ms*.

model, the number of samples n is 2440 and 4880 for fading channels with coherence times 5 and 10 *ms*, respectively.

3.5.4 Prediction Models Performance

Fig. 3.12 shows the percentage of zero-delay throughput achieved using each prediction model and single rates 8/9 and 1/2 as a function of feedback delay when coherence time is 5 *ms*. The feedback delay ranges from 1 *ms* to 5 *ms* and data is collected for 160 *ms*. The quadratic prediction model gives the best performance for shorter delay times achieving 95.2% and 74.5% of zero-delay throughput for feedback delays of 1 *ms* and 2 *ms*, respectively. For feedback delays 3 – 5 *ms*, zero-order prediction outperforms linear and quadratic models. Throughput achieved using single rates 8/9 and 1/2 are 39.6% and 55.5%, respectively.

Fig. 3.13 depicts the percentage of zero-delay throughput achieved using each prediction models and single rates (8/9 and 1/2) as a function of feedback delay when coherence time is 10 *ms*. The linear prediction model gives the best performance for 1 *ms* and 2 *ms* feedback delays achieving 100.8% and 101.2% of zero-delay throughput, respectively. Note that 101.3%

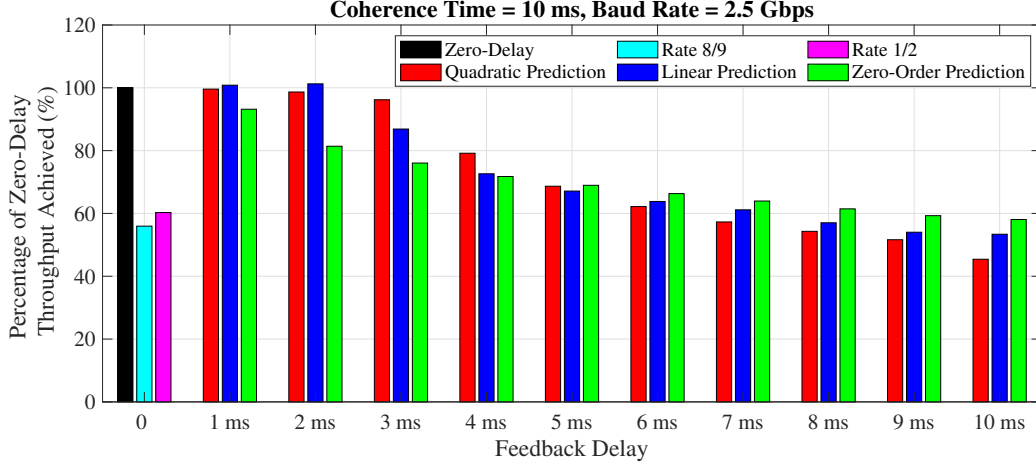


Figure 3.13: Percentage of zero-delay throughput achieved using prediction models as a function of feedback delay when coherence time is 10 *ms* and feedback delay ranges from 1 *ms* to 10 *ms*. Quadratic prediction model gives the best performance for shorter feedback delay times (1 *ms* to 5 *ms*). Zero-Order prediction model gives the best performance for longer feedback delay times. Simulation is performed for 160 *ms*.

is due to the linear prediction model occasionally overestimating channel gain values at the peaks where the values change direction from increasing to decreasing. At these peaks, the model sometimes successfully selects a higher code rate compared to zero-delay model. Since we are considering crossing points where the LDPC code achieves FER below 10^{-6} , the selected code rate might still have a high chance of success, which happened in the simulation for 1 and 2 *ms* feedback delay.

Quadratic model outperforms other prediction models for 3 – 5 *ms* feedback delays achieving from 68.7% to 95.2% of zero-delay throughput. Zero-order prediction achieves the highest throughput for longer feedback delays 6 – 10 *ms*.

Quadratic prediction is worse than linear and zero-order predictions for longer feedback delays because it overestimates channel gain values near peaks in data. For example, Fig. 3.14 compares prediction models when predicting channel gain value at time instance of 65 *ms* for feedback delay of 10 *ms* and coherence time of 10 *ms*. Quadratic prediction overestimates channel gain values by 10 *dB*, which results in selecting rate 8/9 that the true channel condition cannot support as evident in Fig. 3.8 b). Zero-order prediction is the

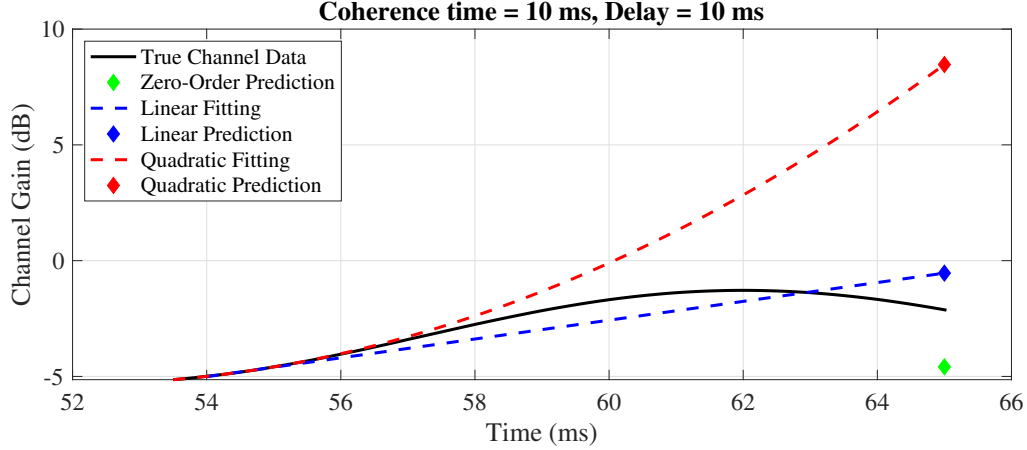


Figure 3.14: The amount of data used for linear and quadratic predictions is equal to 0.5 ms and 2 ms , respectively.

only model that underestimates channel gain values at peaks and selects a lower rate than what channel can support, which results in successful decoding but smaller throughput than baseline zero-delay throughput.

The single rate choice of using only rate $1/2$ seems to be a preferable choice for feedback delays longer than 8 ms . However, this is true only if the atmospheric turbulence is not too strong to severely degrade the channel. In such cases, long outages are possible as seen approximately between 30 ms and 80 ms in Fig. 3.8 b).

3.6 Conclusions

For an FSO fading channel when CSI is known with no delay, the throughput is maximized by selecting the rate accordingly. This chapter presents three prediction models to mitigate FSO fading when feedback delay is not zero.

The simulation results indicate that the quadratic prediction model is the overall best choice for round-trip feedback delays shorter than half of the fading channel coherence time τ_0 . For $\tau_0 = 5\text{ ms}$, quadratic model achieves from 95.2% to 74.6% of the zero-delay throughput as round-trip feedback delay increases from 1 ms to $\lfloor \tau_0/2 \rfloor\text{ ms}$. For a fading optical channel

with a coherence time of 10 *ms* simulation results show that using only a single LDPC code rate 8/9 or 1/2 results in up to 55.9% or 60.3% of zero-delay throughput, respectively. In comparison, for the same channel for shorter feedback delays (1 *ms* to 5 *ms*) quadratic model achieves (99.6 – 68.7)% of zero-delay throughput. For feedback delays 6 *ms* and 7 *ms* zero-order prediction model achieves (66.3 – 64.0)% of the zero-delay throughput.

While quadratic model works very well for shorter feedback delays, prediction for longer feedback delays can be improved. Quadratic polynomial fitting fails to predict the change in fading channel gain at the peaks where the channel gain values change direction from increasing to decreasing, and this leads to poor prediction for longer feedback delays. Our future work will aim to mitigate this problem through various approaches, including machine learning techniques such as neural network regression.

Chapter 4

Interleaving and Post-Processing to Lower LDPC Error Floors

4.1 Introduction

With capacity-approaching performance and low-complexity decoding [98], low-density parity-check (LDPC) codes [57] are widely used in contemporary communication systems, including digital video broadcasting (DVB) [41, 42]. For low-noise channels, spectral efficiency is improved by using PAM or QAM, and system performance can be further enhanced through constellation shaping techniques. Probabilistic amplitude shaping (PAS) [18] adjusts the distribution of the systematic bits of a forward error correction (FEC) code so that the marginal probabilities of signaling amplitudes are optimized to approach the channel capacity.

PAS has been adopted for optical communications [104], and is being considered for 6G cellular mobile networks [4]. Although PAS improves frame error rate (FER) performance of LDPC codes in the waterfall region, standard PAS decoding does not inherently mitigate the error floor phenomenon, characterized by a change in the slope of the FER performance curve that make take place at moderate-high signal-to-noise (SNR) ratios. The error floor phenomenon can hinder the use of LDPC codes, including PAS LDPC codes, in applications

requiring low error rates.

Because many communication standards [41, 42, 47] require very low error rates, the development of techniques to predict and mitigate error floors is essential. The error floor is generally attributed to a combination of low-weight codewords and the limitations of iterative decoding algorithms when applied to graphs containing short, weakly-connected cycles. Previous research has explored these two causes of error floors, studying low-weight codewords, near-codewords [85], trapping sets [25, 64, 96, 128], pseudocodewords [51], and absorbing sets [36].

An example LDPC code from a published communication standard that suffers from a high error floor on the BI-AWGN channel [73] is the rate-2/3 and blocklength-64800 DVB-S2 code¹ [41]. When PAS is applied to this code, a high error floor is also present [18, 105].

Kim et al. [73] found that the error floor region of the DVB-S2 code on the BI-AWGN channel is dominated by weight-nine codewords. Based on this finding, Kim et al. modified the DVB-S2 code to eliminate the weight-nine codewords, thereby lowering the error floor on the BI-AWGN channel. This modification exhibited a dominant $(10, 1)$ trapping set, which required further modification.

The remainder of this chapter refers to the rate-2/3 and length-64800 DVB-S2 LDPC code with 8-PAM PAS using a Maxwell-Boltzmann (MB) distribution as “the DVB-S2 code.” This chapter lowers the error floor of the DVB-S2 code without changing the underlying LDPC code specified by the standard.

4.1.1 Contributions

This chapter develops an approximate truncated union bound (TUB) for PAM (and QAM) constellations with PAS on the AWGN channel. The new TUB restricts attention to low-weight codewords and computes decision thresholds to match bit-metric decoding (BMD) behavior. When the dominant errors in the error-floor region are caused by low-weight

¹DVB-S2 LDPC codes are combined with an outer algebraic code that lowers the error floor.

codewords, selecting the bit interleaving to optimize the BMD TUB can significantly lower the error floor. The accuracy of the BMD TUB and proposed bit-interleaving technique is demonstrated using a rate-2/3 DVB-S2 LDPC code combined with an 8-PAM constellation that follows a MB distribution. The approximate TUB tracks simulation results, confirming that the weight-nine codewords are the cause of the error floor, and guides the design of a new bit-interleaver for 8-PAM that lowers the error floor by more than an order of magnitude by improving the worst-case Euclidean distance. While the results are shown for 8-PAM (i.e. 64-QAM), the methodology is general.

As an additional contribution, this chapter compares the BMD TUB to the TUB for symbol-metric decoding (SMD). For the example PAS scheme considered in this chapter, the SMD TUB indicates that SMD can lower the error floor an additional order of magnitude below the error floor of BMD. To harvest the SMD advantage with practical LDPC decoding, post-processing compares the SMD metric for each decoded codeword to the SMD metric of the neighboring codewords closest in Euclidean distance to the initially decoded codeword. Among these, the codeword with the best SMD metric is selected. Monte Carlo simulation results show that this post-processing lowers the error floor for the rate-2/3 DVB-S2 code with 8-PAM PAS by another order of magnitude.

4.1.2 Organization

This chapter is organized as follows: Sec. 4.2 reviews LDPC codes with PAS and presents the concept of worst-case Euclidean distance. Sec. 4.3 derives a TUB for SMD decoding and an approximate TUB for BMD. Sec. 4.4 applies the BMD TUB of Sec. 4.3 specifically to the DVB-S2 LDPC code with bit interleaving as described in [105]. For each of six distinct bit interleavers available for the DVB-S2 LDPC code, Sec. 4.5 computes the worst-case distances and plots the approximate BMD TUBs. Then, Sec. 4.5 compares FER simulations of decoding for the bit interleaver of [105] and the bit interleaver with the best BMD TUB.

Sec. 4.6 reveals that SMD decoding can significantly outperform BMD in the error floor region for the best bit interleaver and develops a post-processing technique to approach the SMD decoding performance.

4.2 Preliminaries

4.2.1 Low-Density Parity-Check Codes

LDPC codes [58], are (n, k) linear block codes that feature a sparse $(n - k) \times n$ parity-check matrix $\mathbf{H}$. The matrix $\mathbf{H}$ can be represented by a Tanner graph [108] $\mathcal{G}$ consisting of n variable nodes (VNs) and $n - k$ check nodes (CNs), such that CN i is connected to VN j whenever element $\mathbf{H}_{i,j}$ in $\mathbf{H}$ is 1. A protograph is a small Tanner graph of n_p VNs and $(n_p - k_p)$ CNs. It is expanded to a larger graph with n VNs and $(n - k)$ CNs through a "copy-and-permute" process, known as lifting, in which each edge in the protograph is replaced by a Q edges.

Equivalently, a protograph can be described by a $(n_p - k_p) \times n_p$ base matrix. When the base matrix is binary, the lifting operation yields a larger matrix, where each "0" element of the base matrix is replaced by $Q \times Q$ zero matrix and each "1" element is replaced by a $Q \times Q$ permutation matrix. If the permutation matrices are circulants, i.e. cyclic permutation matrices, then the lifted $(n - k, k)$ code (with $n = n_p Q$ and $k = k_p Q$) is a quasi-cyclic (QC) [100] LDPC code. A QC LDPC code constructed in this way can be fully represented by a matrix whose entries specify $Q \times Q$ circulant submatrices.

For a QC code, a block index refers to the position of an entry in the base matrix. Specifically, $r \in [0, n_p - k_p - 1]$ denotes the row-block index corresponding to a CN in the protograph, and $c \in [0, n_p - 1]$ denotes the column-block index corresponding to a VN in the protograph.

Table 4.1: Amplitude probabilities of 8-PAM constellation with Maxwell-Boltzmann distribution and $\mathbb{E}[X^2] = 21$.

i	1	2	3	4
a_i	1.8162	5.4485	9.0808	12.7132
$p_i = P_A(a_i)$	0.5819	0.3131	0.0908	0.0142

4.2.2 LDPC codes with Probabilistic Amplitude Shaping

Consider using LDPC codes with PAS on the discrete-time AWGN channel $Y = X + Z$, where X , Y , and Z are random variables² with Gaussian noise $Z \sim \mathcal{N}(0, \sigma^2)$ and transmit symbol X distributed on a 2^m -PAM constellation $\mathcal{X} \in \{\pm a_1, \pm a_2, \dots, \pm a_{2^m-1}\}$. The signal-to-noise ratio is $\text{SNR} = \mathbb{E}[X^2]/\sigma^2$, where σ^2 is the Gaussian noise variance.

PAS seeks to create a stream of modulated constellation symbols that have a marginal distribution that approaches the optimal signaling distribution according to information theory. With PAS, the transmit symbol is factored as $X = A \times S$, where A and S are random variables with $A = |X|$ and $S = \text{sign}(X)$. Specifically for the AWGN channel, the optimal marginal distribution is a Gaussian. To approach this distribution, PAS uses parity bits for the sign S so that $P_S(1) = P_S(-1) = 1/2$, and the amplitude A for PAS typically follows a MB distribution.

As an example, this chapter uses PAS with an 8-PAM constellation, which was also studied in [18]. Following [18], transmitted symbols are drawn according to the MB distribution yielding the amplitude probabilities $P_A(\cdot)$ shown in Table 4.1 with average energy $\mathbb{E}[X^2] = 21$, which matches the energy of equally likely 8-PAM signaling with $\mathcal{X} = \{\pm 1, \pm 3, \pm 5, \pm 7\}$.

Furthermore, in the PAS construction A and S are stochastically independent so that P_X is symmetric, i.e., $P_X(x) = P_X(-x)$, and can be factored as

²While the transmitted sequence x^N is a deterministic function of the message v^K , it is also helpful in understanding PAS to think about a single symbol X as a random variable with a well-defined marginal distribution.

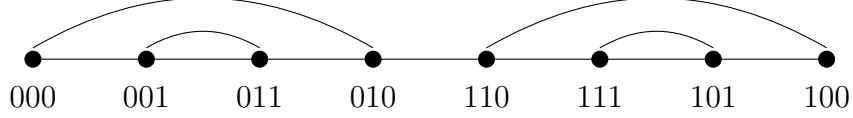


Figure 4.1: Gray-labeled 8-PAM constellation with labels $B_1B_2B_3$. The symbols are equidistant with distance δ_0 between adjacent symbols. The Euclidean distances δ_0 and $3\delta_0$ for error pattern $\mathbf{e} = 010$ are shown. From Table 4.1, $\delta_0 = 3.6324$.

Table 4.2: Definitions of PAS labeling functions $\phi(\cdot)$ for 8-PAM

$x \in \mathcal{X}$	$-a_4$	$-a_3$	$-a_2$	$-a_1$	a_1	a_2	a_3	a_4
$\phi_X(x) = \mathbf{b}$	000	001	011	010	110	111	101	100
$\phi_A(x) = b_2b_3$	00	01	11	10	10	11	01	00
$\phi_S(\text{sign}(x)) = b_1$	0	0	0	0	1	1	1	1
$\phi_2(x) = b_2$	0	0	1	1	1	1	0	0

$$P_X(x) = P_A(|x|)P_S(\text{sign}(x)), \quad \forall x \in \mathcal{X}, \quad (4.1)$$

where $P_A(|x|)$ denotes probability of amplitude A .

Labeling for PAS typically uses a binary reflected Gray code [61], so that adjacent constellation points have binary labels that differ in only one bit position as shown in Fig. 4.1. As in [17], labeling functions $\phi(\cdot)$ assign binary labels to the various elements of a signaling alphabet. As shown in Table 4.2, $\phi_X : \mathcal{X} \rightarrow \{0, 1\}^m$ assigns an m -bit binary label $b_1b_2 \dots b_m$ to each 2^m -PAM symbol $x \in \mathcal{X}$. Similarly, $\phi_A : \mathcal{A} \rightarrow \{0, 1\}^{m-1}$ assigns an $(m-1)$ -bit label to each amplitude $a \in \mathcal{A}$, while $\phi_S : \mathcal{A} \rightarrow \{0, 1\}$ provides the sign bit. Finally, $\phi_i(x) = b_i$.

Encoding

Fig. 4.2 shows an encoder for encoding an LDPC code with PAS that takes an input K -bit message V^K and produces a sequence X^N of constellation points for transmission. The distribution matcher processes the input message sequence V^K to produce the sequence A^N of amplitudes whose empirical marginal distribution approximates the desired $P_A(\cdot)$. For long

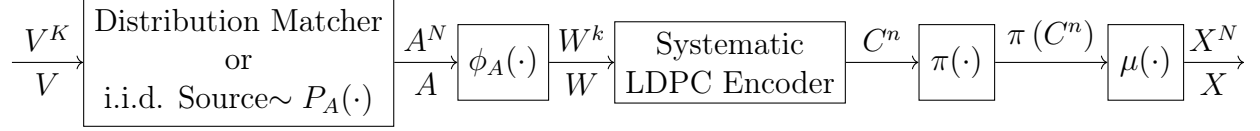


Figure 4.2: Encoder for an LDPC code with probabilistic amplitude shaping as described in Sec. 4.2.2. The distribution matcher can be modeled as an i.i.d. source.

block lengths and with a well-designed distribution matcher often $K \approx NH(P_A)$, so that the number of input messages 2^K matches the approximate number $2^{NH(P_A)}$ of typical sequences for amplitudes drawn i.i.d. from the distribution $P_A(\cdot)$. For the MB distribution described in Table 4.1, $H(P_A) = 1.3805$. For the DVB-S2 code, $N = 21,600$ so that $NH(P_A) = 29,819$.

The amplitudes A are converted into binary labels by the function $\phi_A(\cdot)$, which produces the binary sequence W^k . For the DVB-S2 code, $\phi_A(\cdot)$ is shown in Table 4.2, and $k = 2N = 43,200$ since W^k includes N b_2 bits and N b_3 bits. A systematic LDPC encoder then produces the binary sequence C^n consisting of the systematic bits followed by N parity (b_1) bits computed from the systematic bits. These bits are then interleaved producing $\pi(C^n)$ which has the bits ordered so that constellation point is labeled by appropriate b_1 , b_2 , and b_3 bits. Finally, the mapper takes each three-bit label and produces an 8-PAM symbol yielding the sequence X^N of constellation points. For the DVB-S2 code the mapper $\mu(\cdot)$ uses the labeling of Fig 4.1 and Table 4.2.

Decoding

LDPC codes used with PAS are decoded with message passing decoders using BMD [18, 103, 105, 106]. The message-passing decoder can select any of the 2^k LDPC codewords resulting from a k -bit input. Thus the message passing decoder can select an LDPC codeword even if the corresponding k -bit input could not have been produced by that distribution matcher. This is referred as *codebook mismatch*³ in [87].

³One PAS system that avoids codebook mismatch entirely is probabilistically shaped trellis-coded modulation [120], which performs serial list Viterbi decoding until a valid distribution matcher input is identified.

In fact, the set of $2^K \approx 2^{NH(P_A)}$ valid distribution-matcher outputs is significantly smaller than the set of 2^k valid LDPC encoder inputs. For example, for the DVB-S2 code $K = 29819$ while $k = 43200$. Essentially, the distribution matcher adds redundancy behaving like an error control code with rate $\frac{H(P_A)}{m-1}$, which is 0.69025 for the DVB-S2 code with the MB constellation of Table 4.1.

To improve the performance of message passing in the PAS framework, BMD is biased [18, 103, 105, 106] to match the expected PMF for each of the label bits so that it will tend to favor valid distribution matcher outputs.

Define a random vector $\mathbf{B} = \phi_X(X) = B_1 B_2 \dots B_m$, where $B_j = \phi_j(X)$. The j th random bit level is $B_j = \phi_j(X)$. B_1 designates the random sign S , while $B_2 \dots B_m$ specify the random amplitude A .

The bias is included in BMD as the a-priori information in the computation of the log likelihood L_i for each bit-level i , as shown in (4.2)

$$L_i = \underbrace{\log \frac{P_{B_i}(0)}{P_{B_i}(1)}}_{\text{a-priori information}} + \underbrace{\log \frac{f_{Y|B_i}(y|0)}{f_{Y|B_i}(y|1)}}_{\text{channel LLR}}, \quad (4.2)$$

where P_{B_i} and $f_{Y|B_i}$ are calculated as

$$P_{B_i}(b_i) = \sum_{b_j \in \{0,1\} \text{ for } j \neq i} P_{\mathbf{B}}(b_1 b_2 b_3), \quad (4.3)$$

$$f_{Y|B_i}(y|b_i) = \sum_{b_j \in \{0,1\} \text{ for } j \neq i} \frac{P_{\mathbf{B}}(b_1 b_2 b_3)}{P_{B_i}(b_i)} f_{Y|\mathbf{B}}(y|b_1 b_2 b_3).$$

For example, with the 8-PAM constellation with the MB distribution described in Fig. 4.1 and Table 4.1 the marginal probabilities using (4.3) are $P_{B_1}(1) = 0.5$, $P_{B_2}(1) = 0.8950$, and $P_{B_3}(1) = 0.4038$ [18], so that the a-priori information terms in (4.2) are -2.142863 for L_2 and 0.389656 for L_3 .

For a typical situation in which the input message bits V in Fig. 4.2 are i.i.d. and equally

likely to be one or zero, the distribution matcher produces equally likely amplitude sequences A^N , each with probability 2^{-K} .

Thus, the bias term is not meant to help the decoder distinguish one equally likely distribution matcher output from another. Rather, the bias terms help the decoder favor actual distribution matcher outputs from other inputs that could not have come from the distribution matcher. Thus, the decoder uses the bias term to overcome the mismatch of operating on a larger codebook than the one used by the encoder [87].

With the bias in (4.2), BMD usually favors the distribution-matcher inputs over the other possible encoder inputs, but there are exceptions.

Consider a constant-composition distribution matcher (CCDM) [102] where the output sequences form a set of valid shaped codewords, each with the same fixed amplitude composition determined by the target amplitude distribution $P_A(\cdot)$. When the LDPC decoder with BMD and the bias in (4.2) considers the full LDPC codebook, it also considers codewords whose amplitude composition differs from the target distribution $P_A(\cdot)$. The prior distribution of such codewords will differ from that of the valid shaped codewords, and thus the 2^k possible codewords are not treated as equally likely.

A decoding metric that incorporates prior information can help compensate for codebook mismatch by assigning different metric values to these codewords according to their consistency with the shaped input distribution [87]. The BMD LDPC decoder incorporates prior information at the bit level. From the perspective of the biased BMD LDPC decoder, the effective probabilities are the products of the marginal probabilities of each bit level, i.e.

$$P_{\text{BMD}}(X^N) = \prod_{i=1}^N \prod_{j=1}^3 P_{B_j}(\phi_j(X_i)). \quad (4.4)$$

As an alternative to the bit-wise approach of BMD and BMD, the SMD metric [87] calculates the metric for the sequence X^N of symbols as the product of the marginal

probabilities of each symbol:

$$P_{\text{SMD}}(X^N) = \prod_{i=1}^N P_X(x_i). \quad (4.5)$$

SMD usually favors valid distribution-matcher inputs over the other possible encoder inputs, but there are exceptions. For example, SMD applied to an LDPC code with a distribution matcher with target distribution $P_A(\cdot)$ prefers most of all the LDPC codewords that use only (or almost only) the most-likely amplitude according to $P_A(\cdot)$. Such codewords are not even permitted by the CCDDM distribution matcher. Fortunately, such sequences are relatively sparse.

Both BMD and SMD decoders use prior distributions to compensate for the fact that the decoder is looking at the larger codebook, without explicitly restricting the decoder to the CCDDM output set. However, $P_{\text{BMD}}(X^N) \neq P_{\text{SMD}}(X^N)$ because the bit levels of a shaped symbol are generally not independent. Because it considers the dependence of the bit levels, SMD provides a more accurate representation of the shaping prior than BMD.

4.2.3 Worst-Case Euclidean distance

Consider the binary error pattern $\mathbf{e} = \mathbf{c} \oplus \mathbf{c}'$, where $\oplus$ indicates bit-wise exclusive-or and $\mathbf{c}$ is the binary transmitted codeword that is erroneously decoded as the binary codeword $\mathbf{c}' = \mathbf{c} \oplus \mathbf{e}$. We refer to the Euclidean distance between $\mathbf{c}$ and $\mathbf{c}'$ as

$$d_E(\mathbf{c}, \mathbf{c}') = \|\mu(\mathbf{c}) - \mu(\mathbf{c}')\|_2, \quad (4.6)$$

i.e. to the Euclidean distance between the PAM modulations of $\mathbf{c}$ and $\mathbf{c}'$. For a fixed binary error pattern $\mathbf{e}$, the Euclidean distance $d_E(\mathbf{c}, \mathbf{c} \oplus \mathbf{e})$ depends on the transmitted codeword $\mathbf{c}$. Following [121], define the worst-case Euclidean distance $d_{\text{w.c.}}(\mathbf{e})$ for error pattern $\mathbf{e}$ as

$$d_{\text{w.c.}}(\mathbf{e}) = \min_{\mathbf{c} \in \mathcal{C}} d_E(\mathbf{c}, \mathbf{c} \oplus \mathbf{e}). \quad (4.7)$$

Table 4.3: Worst-case distances for the Gray-Labeled 8-PAM of Fig. 4.1 where $p_i = P_A(a_i)$ and $\mathcal{X}_{w.c.} = \{x : d_E(b(x), b(x) \oplus e) = d_{w.c.}(e)\}$

e (octal)	1	2	3	4	5	6	7
e (binary)	001	010	011	100	101	110	111
$d_{w.c.}(e)$	δ_0	δ_0	$2\delta_0$	δ_0	$2\delta_0$	$4\delta_0$	$3\delta_0$
$P(\mathcal{X}_{w.c.})$	1	$p_2 + p_3$	1	p_1	$p_1 + p_2$	1	$p_1 + p_3$

For example, consider the Gray-labeled 8-PAM constellation in Fig. 4.1. For the binary error $e = 010$ associated with a single PAM symbol x , Fig. 4.1 shows how the distance $d_E(\phi_X(x), \phi_X(x) \oplus 010)$ can be either δ_0 or $3\delta_0$, depending on x . Thus $d_{w.c.}(010) = \delta_0$. Table 4.3 shows the worst case distances for all seven possible error patterns for the Gray-labeled constellation of Fig. 4.1. Table 4.3 also provides, for each specified error pattern, the probability that a transmitted symbol will produce the worst case distance for that error pattern.

4.3 Truncated Union Bounds

This section reviews maximum likelihood (ML) decoding, SMD, and BMD and presents the TUBs for ML and SMD as well as an approximate TUB for BMD.

4.3.1 Thresholds for ML, SMD, and BMD on the AWGN channel

Let $\mathbf{x} = \mu(\mathbf{c})$ denote the modulated transmit signal for the binary codeword $\mathbf{c}$. The Maximum-likelihood (ML) decoding rule selects the codeword $\hat{\mathbf{c}}$ that makes the received sequence most likely, i.e the codeword that maximizes the ML metric J_{ML} :

$$\hat{\mathbf{c}} = \arg \max_{\mathbf{c} \in \mathcal{C}} J_{ML}(\mathbf{y}|\mathbf{x}) , \quad (4.8)$$

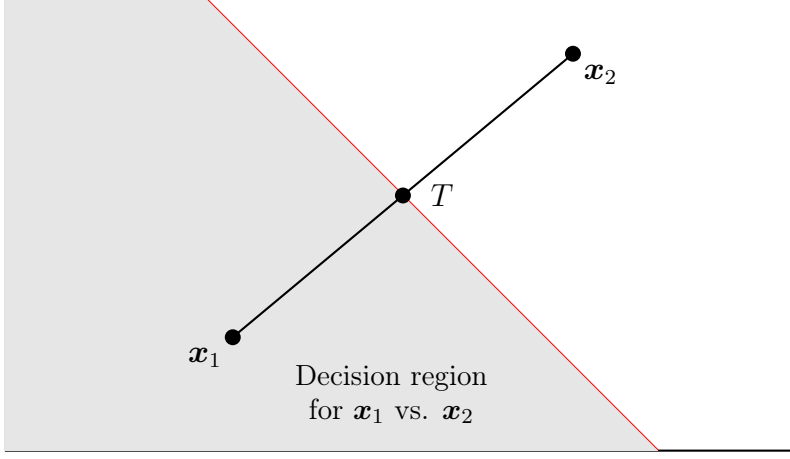


Figure 4.3: Decision regions for ML and SMD decoding. Decision boundary is a half-plane perpendicular to the line segment between $\mathbf{x}_1$ and $\mathbf{x}_2$ and intersect the line segment at the threshold point $T = \lambda \mathbf{x}_1 + (1 - \lambda) \mathbf{x}_2$. For ML decoding $\lambda = 0.5$. For SMD decoding, λ is defined in (4.18). For BMD, the decision regions are not separated by a half plane as shown in this figure.

where the ML metric J_{ML} is

$$J_{\text{ML}}(\mathbf{y}|\mathbf{x}) = \prod_{i=1}^N f_{Y|X}(y_i|x_i), \quad (4.9)$$

where N is the number of 2^m -PAM symbols in each sequence and $f_{Y|X}$ is the conditional probability density induced by the AWGN channel:

$$f_{Y|X}(y_i|x_i) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(\frac{-(y_i - x_i)^2}{2\sigma^2}\right). \quad (4.10)$$

When all codewords are equally likely, ML decoding minimizes the probability of decoding error. ML decoding can also be accomplished by maximizing the log of the ML metric:

$$\hat{\mathbf{c}} = \arg \max_{\mathbf{c} \in \mathcal{C}} \ln J_{\text{ML}}(\mathbf{y}|\mathbf{x}) \quad (4.11)$$

$$= \arg \min_{\mathbf{c} \in \mathcal{C}} \sum_{i=1}^N (y_i - x_i)^2. \quad (4.12)$$

When deciding only between two modulated codewords $\mathbf{x}_1$ and $\mathbf{x}_2$ on the AWGN channel,

the ML decision boundary is a half-plane that is perpendicular to the line segment between $\mathbf{x}_1$ and $\mathbf{x}_2$ and intersects that line segment at the ML threshold point T_{ML} , where

$$T_{\text{ML}} = \frac{\mathbf{x}_1 + \mathbf{x}_2}{2}. \quad (4.13)$$

Fig. 4.3 shows how a half-plane provides the boundary between the decision regions of the two codewords $\mathbf{x}_1$ and $\mathbf{x}_2$.

SMD selects the codeword that maximizes $J_{\text{SMD}}(\mathbf{x}, \mathbf{y})$, the SMD decoding metric:

$$J_{\text{SMD}}(\mathbf{x}, \mathbf{y}) = \prod_{i=1}^N P_X(x_i) f_{Y|X}(y_i|x_i). \quad (4.14)$$

As with ML, SMD can also be accomplished by maximizing the log of the metric, which is equivalent to the following minimization for the AWGN channel:

$$\hat{\mathbf{c}} = \arg \min_{\mathbf{c} \in \mathcal{C}} \sum_{i=1}^N \frac{(y_i - x_i)^2}{2\sigma^2} - \ln \prod_{i=1}^N P_X(x_i) \quad (4.15)$$

$$= \arg \min_{\mathbf{c} \in \mathcal{C}} \sum_{i=1}^N \frac{(y_i - x_i)^2}{2\sigma^2} - \ln P(\mathbf{x}) \quad (4.16)$$

As with ML decoding, when deciding only between two modulated codewords $\mathbf{x}_1$ and $\mathbf{x}_2$ on the AWGN channel, the SMD decision boundary is a half-plane that is perpendicular to the line segment between $\mathbf{x}_1$ and $\mathbf{x}_2$ as shown in Fig. 4.3. With SMD decoding, the half-plane intersects that line segment at the SMD threshold point T_{SMD} , where

$$T_{\text{SMD}} = \lambda_{\text{SMD}} \mathbf{x}_1 + (1 - \lambda_{\text{SMD}}) \mathbf{x}_2, \quad (4.17)$$

and

$$\lambda_{\text{SMD}}(\mathbf{x}_1, \mathbf{x}_2) = \frac{1}{2} \left(1 + \frac{2\sigma^2}{\|\mathbf{x}_2 - \mathbf{x}_1\|^2} \ln \left(\frac{P(\mathbf{x}_2)}{P(\mathbf{x}_1)} \right) \right). \quad (4.18)$$

When $P(\mathbf{x}_1) = P(\mathbf{x}_2)$, $\lambda_{\text{SMD}} = \frac{1}{2}$, and $T_{\text{SMD}} = T_{\text{ML}}$; when $P(\mathbf{x}_1) > P(\mathbf{x}_2)$, T_{SMD} moves closer to $\mathbf{c}_2$ to increase the decision region of the more probable codeword.

When LDPC codes are used with PAS as in [18], BMD is employed. With BMD, the decoder separately considers the marginal probability (and corresponding log likelihood) of each bit labeling the constellation. As described in [63], the symbol-wise decoding metric $q(x_i, y_i)$ for BMD is product of m bit decoding metrics:

$$q(x_i, y_i) = \prod_{j=1}^m q_j(\phi_j(x_i), y_i) , \quad (4.19)$$

where $\phi_j(x)$ denotes j -th bit of symbol x and the j -th bit decoding metric $q_j(b, y)$ is given by

$$q_j(b, y_i) = \sum_{x \in \mathcal{X}_b^j} P_X(x) f_{Y|X}(y|x) , \quad (4.20)$$

where $\mathcal{X}_b^j$ denotes the set of 2^m -PAM signaling points x for which the j th bit of the binary label $\phi_j(x)$ has value $b \in \{0, 1\}$. In [63], (4.19) is referred as the bit-interleaved coded modulation (BICM) maximum a-posteriori (MAP) metric.

BMD essentially replaces the joint probability of the bit label with the product of its marginals, producing the following joint distribution for the transmitted signaling sequence $\mathbf{x}$ and received symbol sequence $\mathbf{y}$ as

$$q(\mathbf{x}, \mathbf{y}) = \prod_{i=1}^N \prod_{j=1}^m q_j(\phi_j(x_i), y_i) , \quad (4.21)$$

which we saw earlier as (4.4). This is the distribution assumed by BMD to compute its metric so that

$$J_{\text{BMD}}(\mathbf{x}, \mathbf{y}) = q(\mathbf{x}, \mathbf{y}) . \quad (4.22)$$

Using this metric the decoder selects a binary codeword $\hat{\mathbf{c}}$ according to

$$\hat{\mathbf{c}} = \arg \max_{\mathbf{c} \in \mathcal{C}} J_{\text{BMD}}(\mathbf{x}, \mathbf{y}) \quad (4.23)$$

and we can find the same $\hat{\mathbf{c}}$ by maximizing the log of J_{BMD} :

$$\log J_{\text{BMD}}(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^N \sum_{j=1}^m \log(q_j(\phi_j(x_i), y_i)). \quad (4.24)$$

When deciding only between two modulated codewords $\mathbf{x}_1$ and $\mathbf{x}_2$ on the AWGN channel, the BMD decision boundary is *not* simply a half-plane that is perpendicular to the line segment between $\mathbf{x}_1$ and $\mathbf{x}_2$ as shown in Fig. 4.3.

However, we can still compute where the decision boundary intersects the line segment between $\mathbf{x}_1$ and $\mathbf{x}_2$. We still refer to this as the BMD threshold point T_{BMD} , where

$$T_{\text{BMD}} = \lambda_{\text{BMD}} \mathbf{x}_1 + (1 - \lambda_{\text{BMD}}) \mathbf{x}_2, \quad (4.25)$$

where λ_{BMD} can be computed by numerically finding the value of $\lambda \in (0, 1)$ that satisfies

$$J_{\text{BMD}}(\mathbf{x}_1, \mathbf{y}_\lambda) = J_{\text{BMD}}(\mathbf{x}_2, \mathbf{y}_\lambda), \quad (4.26)$$

where $\mathbf{y}_\lambda = \lambda \mathbf{x}_1 + (1 - \lambda) \mathbf{x}_2$. While we cannot provide a closed form expression for λ_{BMD} , we will see below that it is generally not equal to λ_{SMD} .

4.3.2 Truncated Union Bound for SMD

Let $\mathbf{c}$ denote binary transmitted codeword, let $\bar{\mathbf{c}} = \mathbf{c} \oplus \mathbf{e}$ denote a different, competing codeword that is under consideration by the decoder, and let $\mathbf{y}$ denote the received signal. The codewords $\mathbf{c}$ and $\bar{\mathbf{c}}$ are modulated into $\mathbf{x} = \mu(\mathbf{c})$ and $\bar{\mathbf{x}} = \mu(\bar{\mathbf{c}})$ respectively.

The union bound [63] provides an upper bound on the probability of message error for a

linear block code under ML or SMD decoding as follows:

$$P_e \leq \sum_{\mathbf{e}} \sum_{\mathbf{x}} P(\mathbf{x}) \text{ PEP}(\mathbf{c}, \mathbf{c} \oplus \mathbf{e} \mid \mathbf{x}), \quad (4.27)$$

where PEP is the pairwise error probability

$$\text{PEP}(\mathbf{c}, \mathbf{c} \oplus \mathbf{e} \mid \mathbf{x}) \triangleq P\{J(\tilde{\mathbf{x}}, \mathbf{y}) > J(\mathbf{x}, \mathbf{y})\}. \quad (4.28)$$

$\text{PEP}(\mathbf{c}, \mathbf{c} \oplus \mathbf{e} \mid \mathbf{x})$ is the probability that the ML or SMD decoder prefers codeword $\bar{\mathbf{c}} = \mathbf{c} \oplus \mathbf{e}$ over $\mathbf{c}$ when $\mathbf{c}$ is transmitted.

For the binary-input AWGN channel with noise variance σ^2 , the PEP under ML decoding is

$$\text{PEP}_{\text{ML}}(\mathbf{c}, \mathbf{c} \oplus \mathbf{e} \mid \mathbf{x}) = Q\left(\frac{d_E(\mathbf{c}, \bar{\mathbf{c}})}{2\sigma}\right), \quad (4.29)$$

where $d_E(\mathbf{c}, \bar{\mathbf{c}})$ denotes the Euclidean distance between the codewords $\mathbf{c}$ and $\bar{\mathbf{c}} = \mathbf{c} \oplus \mathbf{e}$, and $Q(\cdot)$ is Gaussian tail function:

$$Q(x) = \int_x^\infty \frac{1}{\sqrt{2\pi}} e^{-t^2/2} dt. \quad (4.30)$$

Since for SMD decoding the decision boundary between two codewords is not necessarily half-way and codewords are not necessarily equally likely, the pairwise error probability in (4.29) is given by:

$$\text{PEP}_{\text{SMD}}(\mathbf{c}, \mathbf{c} \oplus \mathbf{e} \mid \mathbf{x}) = Q\left(\frac{\lambda_{\text{SMD}}(\mathbf{c}, \bar{\mathbf{c}}) d_E(\mathbf{c}, \bar{\mathbf{c}})}{\sigma}\right), \quad (4.31)$$

where $\lambda_{\text{SMD}} \in [0, 1]$ denotes the scaling factor along the line segment connecting the transmit-

ted and competing codewords, as described in (4.18). The union bound under SMD decoding is:

$$P_e^{\text{SMD}} \leq \sum_{\mathbf{e}} \sum_{\mathbf{x}} P(\mathbf{x}) Q \left(\frac{\lambda_{\text{SMD}}(\mathbf{c}, \bar{\mathbf{c}}) d_E(\mathbf{c}, \bar{\mathbf{c}})}{\sigma} \right) \quad (4.32)$$

If performance is dominated by low-weight codewords, (4.32) can be approximated by truncating to only include the worst-case distances of low-weight codewords:

$$P_e^{\text{SMD}} \approx \sum_{\mathbf{e} \in \mathcal{S}_t} \sum_{\mathbf{x} \in S_{\text{w.c.}}(\mathbf{e})} P(\mathbf{x}) Q \left(\frac{\lambda_{\text{SMD}}(\mathbf{c}, \bar{\mathbf{c}}) d_{\text{w.c.}}(\mathbf{e})}{\sigma} \right)$$

where $\mathcal{S}_t = \{\mathbf{e} : w_{\text{H}}(\mathbf{e}) \leq t\}$ is the set of all errors with Hamming weight less or equal to t , and

$$S_{\text{w.c.}}(\mathbf{e}) = \{\mathbf{c} : d_E(\mathbf{c}, \mathbf{c} \oplus \mathbf{e}) = d_{\text{w.c.}}(\mathbf{e})\} \quad (4.33)$$

is the set of all transmitted codewords $\mathbf{c}$ whose Euclidean distance from the competing codeword $\bar{\mathbf{c}} = \mathbf{c} \oplus \mathbf{e}$ is equal to the worst-case distance $d_{\text{w.c.}}(\mathbf{e})$.

The value of λ_{SMD} is different for different transmitted codewords $\mathbf{c}$. Decompose $S_{\text{w.c.}}(\mathbf{e})$ into subsets that share a common λ_{SMD} as follows:

$$S_{\text{w.c.}}(\mathbf{e}) = \cup_{i=1}^{N_{\lambda}} S_{\text{w.c.}}(\mathbf{e}, \lambda_i), \quad (4.34)$$

where

$$S_{\text{w.c.}}(\mathbf{e}, \lambda_i) = \{\mathbf{c} : \mathbf{c} \in S_{\text{w.c.}}(\mathbf{e}), \lambda_{\text{SMD}}(\mathbf{c}, \mathbf{c} \oplus \mathbf{e}) = \lambda_i\}, \quad (4.35)$$

and where N_{λ} is the number of subsets of transmitted codewords $\mathbf{c}$ such that the codewords in the same class i have the same decision boundary $\lambda_{\text{SMD}} = \lambda_i$ using (4.18). Using this

decomposition, the SMD TUB can be written as:

$$P_e^{\text{SMD}} \approx \sum_{\mathbf{e} \in \mathcal{S}_t} \sum_{i=1}^{N_\lambda} P(\mathcal{S}_{\text{w.c.}}(\mathbf{e}, \lambda_i)) Q\left(\frac{\lambda_i d_{\text{w.c.}}(\mathbf{e})}{\sigma}\right), \quad (4.36)$$

where

$$P(\mathcal{S}_{\text{w.c.}}(\mathbf{e}, \lambda_i)) = \sum_{\mathbf{c} \in \mathcal{S}_{\text{w.c.}}(\mathbf{e}, \lambda_i)} P(\mu(\mathbf{c})). \quad (4.37)$$

4.3.3 Approximate Truncated Union Bound for BMD

Since the PAS decoder employs bit-metric decoding (BMD), pairwise decoding errors probabilities cannot be computed using the $Q(\cdot)$ function so that a truncated union bound of the form of (4.36) cannot be applied precisely. However, in the error floor region when the dominant errors are caused by the low-weight codewords, we choose to *approximately* apply the TUB given in (4.36) to BMD with the λ values computed using (4.26) as follows:

$$P_e^{\text{BMD}} \approx \sum_{\mathbf{e} \in \mathcal{S}_t} \sum_{i=1}^{N_{\lambda_{\text{BMD}}}} P(\mathcal{S}_{\text{w.c.}}(\mathbf{e}, \lambda_i)) Q\left(\frac{\lambda_i d_{\text{w.c.}}(\mathbf{e})}{\sigma}\right) \quad (4.38)$$

where as with the SMD case, the set of error events $\mathbf{e}$ is truncated to include only error events with Hamming distance t or less,

$$\mathcal{S}_t = \{\mathbf{e} : w_{\text{H}}(\mathbf{e}) \leq t\}. \quad (4.39)$$

For the low-weight error events in $\mathcal{S}_t$, we further restrict attention to the transmitted codewords that lead to the worst-case distance:

$$\mathcal{S}_{\text{w.c.}}(\mathbf{e}) = \{\mathbf{x} : d_E(\mathbf{c}, \mathbf{c} \oplus \mathbf{e}) = d_{\text{w.c.}}(\mathbf{e})\}, \quad (4.40)$$

and we decompose $\mathcal{S}_{\text{w.c.}}(\mathbf{e})$ into subsets that share a common λ_{BMD} as follows:

$$\mathcal{S}_{\text{w.c.}}(\mathbf{e}) = \cup_{i=1}^{N_\lambda} \mathcal{S}_{\text{w.c.}}(\mathbf{e}, \lambda_i), \quad (4.41)$$

where

$$\mathcal{S}_{\text{w.c.}}(\mathbf{e}, \lambda_i) = \{\mathbf{c} : \mathbf{c} \in \mathcal{S}_{\text{w.c.}}(\mathbf{e}), \lambda_{\text{BMD}}(\mathbf{c}, \mathbf{c} \oplus \mathbf{e}) = \lambda_i\}, \quad (4.42)$$

and where N_λ is the number of subsets of transmitted codewords $\mathbf{c}$ such that the codewords in the same class i have the same decision boundary $\lambda_{\text{BMD}} = \lambda_i$ using (4.26).

4.4 Truncated Union Bound for the DVB-S2 Code

This section uses (4.38) to approximate the error floor region of the DVB-S2 code under BMD decoding. The error floor region of the DVB-S2 code both for BPSK and the 8-PAM Maxwell Boltzmann constellation is dominated by errors with Hamming weight nine, as noted in [18, 73, 105], and observed in our Monte-Carlo simulations. Thus we choose $t = 9$ and truncate to only consider the set $\mathcal{S}_9$ of errors with Hamming weight nine.

By symmetry that we will discuss below, we can replace the outer sum in (4.38) with the number A_9 of weight-nine errors. This yields the approximate BMD TUB for Rate-2/3 DVB-S2 code as follows:

$$P_e^{\text{BMD}} \approx A_9 \sum_{i=1}^{N_{\lambda_{\text{BMD}}}} P(\mathcal{S}_{\text{w.c.}}(\mathbf{e}, \lambda_i)) Q\left(\frac{\lambda_i d_{\text{w.c.}}(\mathbf{e})}{\sigma}\right). \quad (4.43)$$

Numerical computation of the approximate BMD TUB of (4.43) requires the number of Hamming weight-nine neighbors A_9 , the value of the worst-case distance $d_{\text{w.c.}}$ for the closest weight-nine errors, and the values of λ_i and $P(\mathcal{S}_{\text{w.c.}}(\mathbf{e}, \lambda_i))$ for each distinct decision boundary. These required values are established in the subsections below.

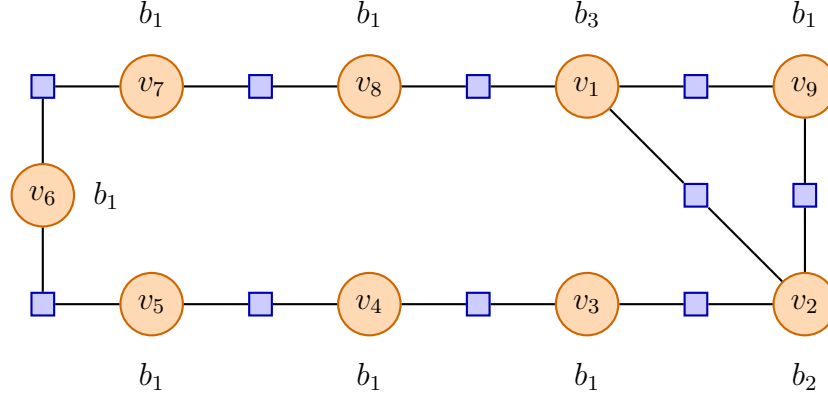


Figure 4.4: Subgraph of the DVB-S2 code Tanner graph induced by Hamming weight-nine codewords. Degree-3 variable nodes v_1 and v_2 represent information bits mapped to bit levels b_3 and b_2 , respectively, of an 8-PAM constellation. Degree-2 variable nodes v_3 to v_9 represent parity bits, each mapped to the bit level b_1 of a distinct modulation symbol.

4.4.1 The Number A_9 of Weight-Nine Errors

To analytically establish the number of Hamming weight-nine neighbors A_9 , we permute the rows and parity columns of DVB-S2 parity check matrix to transform it into QC structure with $Q = 360$ and base matrix of size 60×180 , with one exception (see Sec. 6.5 in [100]): the code is not QC after the permutation because the non-zero $Q \times Q$ submatrix is not a circulant. This is due to a missing edge that would connect the first CN with the last VN in parity check matrix. This missing edge is inherent to codes whose parity-check matrix parity part has dual-diagonal structure to enable low-complexity encoding [125]. Despite this exception, the block-circulant structures of the parity-check matrix with $Q = 360$ results in 360 weight-nine codewords.

Fig. 4.4 shows the subgraph of the DVB-S2 code Tanner graph, induced by each of the weight-nine codewords. As discussed in Sec. 4.2, each 8-PAM symbol has three bit levels described by the binary labels (b_1, b_2, b_3) . As shown in Fig. 4.4, for weight-nine error events the two degree-3 variable nodes are v_1 and v_2 , which correspond to bit levels described by binary labels b_3 and b_2 , respectively. The remaining seven degree-2 variable nodes in Fig. 4.4 correspond to parity bits and are mapped to bit levels described by binary label b_1

corresponding to distinct 8-PAM constellation symbols.

Consistent with the block-circulant structure, there are exactly 360 weight-nine errors each involving column block indices 23, 68, 128, 129, 130, 131, 132, 133, 143. The errors associated with the weight-nine codewords involve two degree-3 variable nodes associated with message bits located in the column block indices 23 and 68. The block-circulant structure allows us to replace the outer sum in (4.38) with the number A_9 of weight-nine errors because all of the Hamming weight-nine errors are related by cyclic shifts that have the same values of λ_i and $P(\mathcal{S}_{w.c.}(\mathbf{e}, \lambda_i))$.

4.4.2 Worst-Case Distance $d_{w.c.}$ for the Standard Bit Interleaving

This chapter explores the DVB-S2 code used with PAS where the amplitude bits appear as the systematic bits in the LDPC codeword beginning with all of the b_3 bits followed by the all of b_2 bits, as in [18, 105]. This section, Sec. 4.4, focuses on the parity-bit mapping used in [105], where the b_1 bits are assigned to the PAM symbols in the order they are encoded. Sec. 4.5 explores alternative interleaving of the parity bits.

For the parity bit interleaving of [105] each bit of a Hamming weight-nine error corresponds to a distinct 8-PAM symbol, i.e., no two bits of the error belong to the same modulation symbol. As a result, the error pattern $\mathbf{e}$ for the nine affected PAM symbols can be represented as $\mathbf{e} = (4, 4, 4, 4, 4, 4, 4, 1, 2)_8$ in octal, e.g. the 4's correspond to b_1 bits.

Sec. 4.5 will consider how permutations $\pi_i(\mathbf{e})$ of the LDPC codeword bits affect the worst-case distance, TUB, and simulated performance. For clarity, we will refer to $\mathbf{e} = (4, 4, 4, 4, 4, 4, 4, 1, 2)_8$ as $\pi_0(\mathbf{e})$ since it is the identity permutation that produces the default ordering of the LDPC codeword bits.

Let δ_0 denote the minimum Euclidean distance between two adjacent symbols in the equidistant 8-PAM, and let p_i denote probability of each 8-PAM amplitude a_i , as in Table 4.1. The worst-case distances $d_{w.c.}(\mathbf{e})$ for each possible three-bit error pattern for the Gray-labeled

constellation in Fig. 4.1 are given in Table 4.3. We can compute the worst-case Euclidean distance for the weight-nine error $\pi_0(\mathbf{e}) = (4, 4, 4, 4, 4, 4, 4, 1, 2)_8$ using the $e = 001$, $e = 010$ and $e = 100$ columns of Table 4.3, which correspond, respectively, to random variables B_3 , B_2 and B_1 associated with bit levels b_3 , b_2 , and b_1 . The bottom two rows of Table 4.3 can be deduced using Fig. 4.1.

The worst-case Euclidean distance $d_{\text{w.c.}}(\mathbf{e})$ for the weight-nine error $\pi_0(\mathbf{e})$ is calculated as:

$$d_{\text{w.c.}}(\mathbf{e}) = \sqrt{7\delta_0^2 + \delta_0^2 + \delta_0^2} = 3\delta_0 = 10.90. \quad (4.44)$$

Specifically, this worst-case distance occurs when a transmitted codeword uses constellation points for which the binary symbol errors take each of the nine relevant transmitted symbols to a neighboring constellation point that is δ_0 away.

For the seven parity-bit errors, the error $e = 100$ leads to a neighbor δ_0 away only for the two constellation points with the smallest amplitude $a_1 = 1.8162$. The probability of a constellation point with this magnitude being transmitted is $p_1 = 0.5820$. For the message bit v_1 corresponding to $e = 001$, this error takes every constellation point, i.e. all four amplitudes, to a neighbor that is δ_0 away, so the probability of such a constellation point being transmitted is one. For the message bit v_2 corresponding to $e = 010$, this error takes constellation points with amplitudes $a_2 = 5.4485$ and $a_3 = 9.0808$ to neighbors that are δ_0 away so the probability of such a constellation point being transmitted is $p_2 + p_3 = 0.4040$.

Multiplying the nine probabilities together, the probability of the set of modulated codewords $\mathbf{x} = \mu(\mathbf{c})$ for which $\pi_0(\mathbf{e}) = (4, 4, 4, 4, 4, 4, 4, 1, 2)_8$ produces a neighboring codeword $\mathbf{x}' = \mu(\mathbf{c} \oplus \mathbf{e})$ with the worst-case Euclidean distance $3\delta_0$ is:

$$P(\mathcal{S}_{\text{w.c.}}(\mathbf{e})) = p_1^7(p_2 + p_3) = 9.138 \times 10^{-3}. \quad (4.45)$$

4.4.3 Values of λ_i and $P(\mathcal{S}_{\text{w.c.}}(\mathbf{e}, \lambda_i))$

As explained in Sec. 4.3, the decision boundaries for SMD and BMD correspond to λ_{SMD} and λ_{BMD} . These λ values depend on the probabilities of the codewords following (4.18) and (4.26). The codeword probabilities depend on the sequence of amplitudes of the transmitted codeword, and the codeword probability affects λ specifically through the amplitudes of the symbols where the errors occur.

As discussed in the derivation of (4.45), there is only one possible amplitude for the seven B_1 errors with $e = 100$ errors, there are two amplitudes for the B_2 error with $e = 010$, and four amplitudes for the B_3 error with $e = 001$. Overall, there are eight distinct sequences of amplitudes that can produce the worst-case distance for $\pi_0(\mathbf{e}) = (4, 4, 4, 4, 4, 4, 4, 1, 2)_8$.

Table 4.4 shows the eight values of λ_i corresponding to the eight possible amplitude sequences as well as the probability $P(\mathcal{S}_{\text{w.c.}}(\mathbf{e}, \lambda_i))$ of a transmitted codeword having each sequence. The rows of Table 4.4 are organized into pairs of codewords that are separated by $d_{\text{w.c.}}$, i.e. that play the roles of $\mathbf{x}_1$ and $\mathbf{x}_2$ in Fig. 4.3. The sum of λ_i for two rows in such a pair is approximately one. Following (4.18), within such a pair of rows the SMD value of λ_i is smaller for the lower value of the $P(\mathcal{S}_{\text{w.c.}}(\mathbf{e}, \lambda_i))$, thus giving more decision region area to the more likely codeword.

The Monte Carlo threshold in Table 4.4 was found by simulating the LDPC decoder for closely spaced points along the line segment between the transmitted codeword $\mathbf{x} = \mu(\mathbf{c})$ and the neighboring codeword $\mathbf{x}' = \mu(\mathbf{c} \oplus \mathbf{e})$ to identify the point where the LDPC decoder stops selecting $\mathbf{x} = \mu(\mathbf{c})$ and begins selecting $\mathbf{x}' = \mu(\mathbf{c} \oplus \mathbf{e})$. Since the LDPC decoder uses BMD, we expect to see agreement with the BMD λ_i value, which is confirmed in Table 4.4.

Table 4.4 also reveals that for the error $\pi_0(\mathbf{e})$ the SMD and BMD thresholds have only a slight difference.

Table 4.4: Decision boundaries λ_i for error pattern $\pi_0(\mathbf{e}) = (4, 4, 4, 4, 4, 4, 4, 1, 2)_8$.

Transmit Amplitudes		λ_i				
B_2 $e = 010$	B_3 $e = 001$	Monte Carlo	BMD	SMD	$P(\mathcal{S}_{w.c.}(\mathbf{e}, \lambda_i))$	
a_2	a_1	0.54	0.54	0.55	$p_1^8 p_2$	4.12×10^{-3}
a_3	a_2	0.44	0.45	0.45	$p_1^7 p_2 p_3$	6.43×10^{-4}
a_2	a_2	0.51	0.51	0.52	$p_1^7 p_2^2$	2.22×10^{-3}
a_3	a_1	0.48	0.48	0.48	$p_1^8 p_3$	1.20×10^{-3}
a_2	a_3	0.58	0.57	0.58	$p_1^7 p_2 p_3$	6.43×10^{-4}
a_3	a_4	0.42	0.42	0.42	$p_1^7 p_3 p_4$	2.92×10^{-5}
a_2	a_4	0.48	0.48	0.48	$p_1^7 p_2 p_4$	1.01×10^{-4}
a_3	a_3	0.51	0.51	0.52	$p_1^7 p_3^2$	1.86×10^{-4}

4.5 Permuting Bits to Increase the Worst-Case Distance and Lower the Error Floor

This section shows how different mappings of the LDPC code bits to the 8-PAM symbols can produce different worst-case distances and thus different values for the approximate BMD TUB. The section enumerates the possible permutations and identifies the permutation with the largest worst-case Euclidean distance and the lowest error probability according to the BMD TUB as derived in (4.38).

4.5.1 Bit Interleaving to Increase the Worst-Case Distance

Sec. 4.4 considered the worst-case Euclidean distance and the BMD TUB for the interleaving of LDPC code bits to 8-PAM symbols used in [105]. Different interleavings can produce

Table 4.5: Worst-case distances for all possible error patterns $\mathbf{e}$, where $p_i = P_A(a_i)$ and $\mathcal{S}_{\text{w.c.}}(\mathbf{e}) = \{\mathbf{x} : d_E(\mathbf{c}, \mathbf{c} \oplus \mathbf{e}) = d_{\text{w.c.}}(\mathbf{e})\}$.

i	$\pi_i(\mathbf{e})$ (in Octal)	$d_{\text{w.c.}}(\pi_i(\mathbf{e}))$		$P(\mathcal{S}_{\text{w.c.}}(\mathbf{e}))$	
0	$(4, 4, 4, 4, 4, 4, 4, 1, 2)_8$	$3\delta_0$	10.90	$p_1^7(p_2 + p_3)$	0.0091
1	$(4, 4, 4, 4, 4, 4, 4, 3)_8$	$\sqrt{11}\delta_0$	12.05	p_1^7	0.0226
2	$(4, 4, 4, 4, 4, 4, 2, 5)_8$	$\sqrt{11}\delta_0$	12.05	$p_1^6(p_2 + p_3)(p_1 + p_2)$	0.0141
3	$(4, 4, 4, 4, 4, 4, 7)_8$	$\sqrt{15}\delta_0$	14.07	$p_1^6(p_1 + p_3)$	0.0261
4	$(4, 4, 4, 4, 4, 4, 1, 6)_8$	$\sqrt{23}\delta_0$	17.42	p_1^6	0.0389
5	$(4, 4, 4, 4, 4, 5, 6)_8$	$5\delta_0$	18.16	$p_1^5(p_1 + p_2)$	0.0598

different values of $d_{\text{w.c.}}$, λ_i , and $P(\mathcal{S}_{\text{w.c.}}(\mathbf{e}, \lambda_i))$ and thus different error probability BMD TUBs (4.38) when interleaving changes the way that bits of error patterns affect 8-PAM symbols.

For the specific error pattern of interest with Hamming weight nine consisting of seven B_1 bits, one B_2 bit, and one B_3 bit, there are six distinct ways for the bits of the error pattern to affect 8-PAM symbols. Table 4.5 shows the worst-case Euclidean distances $d_{\text{w.c.}}$ and probabilities $P(\mathcal{S}_{\text{w.c.}}(\mathbf{e}))$ for the six possible permutations $\pi(\mathbf{e})$ ordered by increasing worst-case distance.

The seven B_1 bits must each correspond to a different 8-PAM symbol. The six distinct sets of values of $d_{\text{w.c.}}$ and $P(\mathcal{S}_{\text{w.c.}}(\mathbf{e}))$ correspond to how the B_2 and B_3 bits of the error pattern are mapped to 8-PAM symbols. For the original permutation $\pi_0(\mathbf{e})$ the B_2 and B_3 bits are each mapped to separate 8-PAM symbols and neither of these symbols contains one of the seven B_1 bits that are part of the error pattern. Thus the error pattern involves nine 8-PAM symbols.

For permutations $\pi_2(\mathbf{e})$ and $\pi_4(\mathbf{e})$, the B_2 and B_3 bits are each mapped to separate 8-PAM symbols but one of these two bits is mapped to the same symbol as one of the seven B_1 bits

that are part of the error pattern. Thus the error pattern involves eight 8-PAM symbols. For $\pi_4(\mathbf{e})$, B_2 and B_1 are mapped to the same symbol. Permutations $\pi_1(\mathbf{e})$ and $\pi_3(\mathbf{e})$ map the B_2 and B_3 bits to the same 8-PAM symbol. For $\pi_3(\mathbf{e})$ this 8-PAM symbol contains one of the seven B_1 bits so that the error involves only seven 8-PAM symbols, but for $\pi_2(\mathbf{e})$ it does not so that the error involves eight 8-PAM symbols.

The final possible permutation, $\pi_5(\mathbf{e})$, has the B_2 and B_3 bits each mapped to different 8-PAM symbols, but both of these symbols contain one of the seven B_1 bits that are part of the error pattern. This permutation involves seven 8-PAM symbols and produces the largest worst-case distance of $5\delta_0$.

4.5.2 Approximate BMD TUBs for the Six Possible Mappings

Fig. 4.5 plots the approximate BMD TUB (4.38) for all six possible mappings. These curves are not fully determined by $d_{\text{w.c.}}$ and $P(\mathcal{S}_{\text{w.c.}}(\mathbf{e}))$, since the TUB also depends on the values of λ_i and $P(\mathcal{S}_{\text{w.c.}}(\mathbf{e}, \lambda_i))$ as well as $d_{\text{w.c.}}$. For $\pi_5(\mathbf{e})$, which achieves the lowest approximate BMD TUB curve, Table 4.6 shows the eight values of λ_i for SMD and BMD corresponding to the eight possible amplitude sequences as well as the probability $P(\mathcal{S}_{\text{w.c.}}(\mathbf{e}, \lambda_i))$ of a transmitted codeword having each sequence.

While the approximate BMD TUB is not completely determined by $d_{\text{w.c.}}$ and $P(\mathcal{S}_{\text{w.c.}}(\mathbf{e}))$, the ordering of the curves in Fig. 4.5 is consistent with these values as presented in Table 4.5: Higher values of $d_{\text{w.c.}}$ in Table 4.5 correspond to lower approximate BMD TUB curves in Fig. 4.5. In the case of $\pi_1(\mathbf{e})$ and $\pi_2(\mathbf{e})$, where the values of $d_{\text{w.c.}}$ are both $\sqrt{11}\delta_0 = 12.05$, the lower value of $P(\mathcal{S}_{\text{w.c.}}(\mathbf{e}))$ produces the lower approximate BMD TUB curve.

As with Table 4.4, the rows of Table 4.6 are organized into pairs of codewords that are separated by $d_{\text{w.c.}}$, i.e. that play the roles of $\mathbf{x}_1$ and $\mathbf{x}_2$ in Fig. 4.3. Also shown is the threshold observed in Monte Carlo simulations of the LDPC decoder along the line segment between the transmitted codeword $\mathbf{x} = \mu(\mathbf{c})$ and the neighboring codeword $\mathbf{x}' = \mu(\mathbf{c} \oplus \mathbf{e})$.

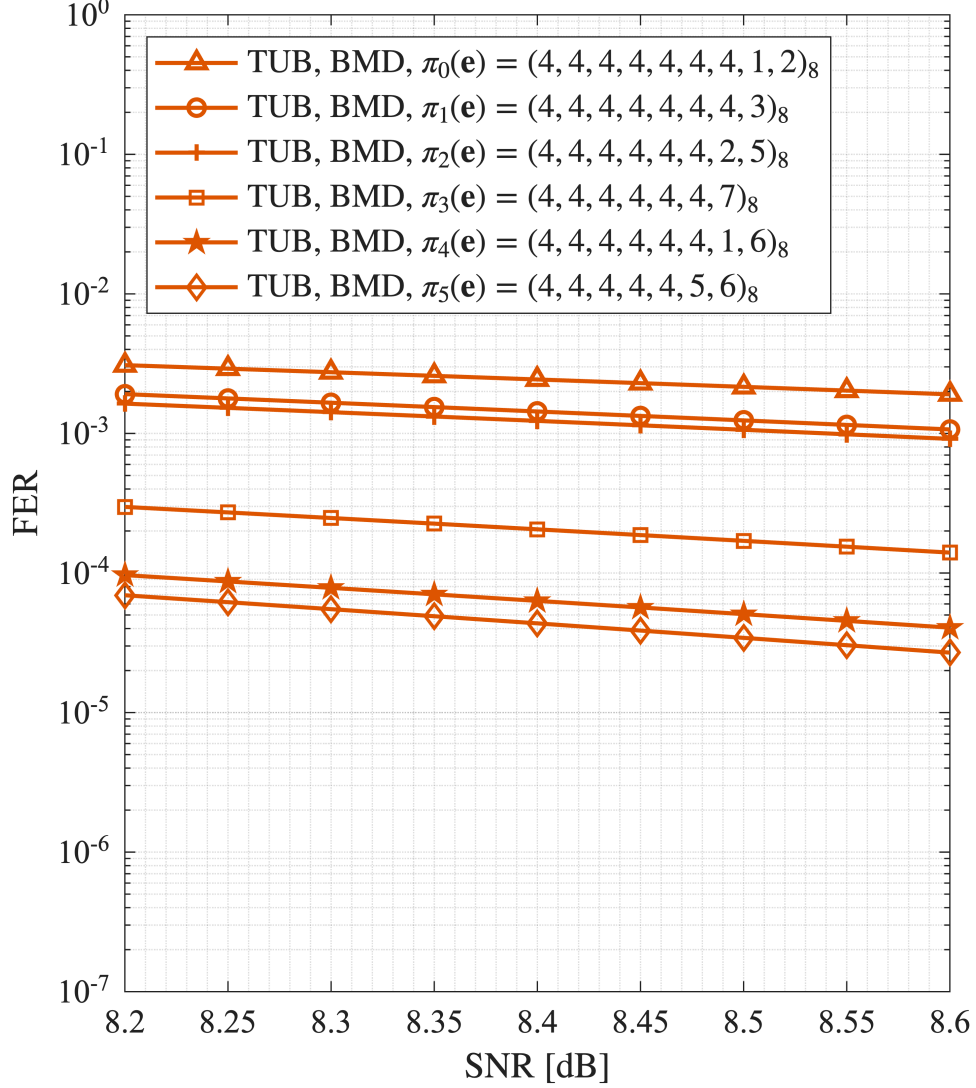


Figure 4.5: Approximate truncated union bound (TUB) for bit-metric decoding (BMD) of the DVB-S2 LDPC code for all possible bit interleavers $\pi_0(\mathbf{e}) - \pi_5(\mathbf{e})$ corresponding to neighboring codewords that differ from the transmitted codeword by Hamming distance nine and $d_{w.c.}$.

Comparing with Table 4.4 in Sec. 4.4.3, which provided these same values for $\pi_0(\mathbf{e})$, there is an important difference. In Table 4.6, the BMD λ_i values do not always agree with the SMD λ_i values. In the first pair of rows and the last pair of rows, we see that the BMD λ_i values favor the *less likely* codeword while the SMD λ_i values favor the more likely codeword as required by (4.18). We will explore this difference in BMD and SMD behavior further in Sec. 4.6.

Table 4.6: Decision boundaries λ_i for error pattern $\pi_5(\mathbf{e}) = (4, 4, 4, 4, 4, 5, 6)_8$.

Transmit Amplitudes		λ_i				
$B_1 B_2$ $e = 110$	$B_1 B_3$ $e = 101$	Monte Carlo	BMD	SMD	$P(\mathcal{S}_{w.c.}(\mathbf{e}, \lambda_i))$	
a_1	a_1	0.46	0.45	0.54	p_1^7	2.26×10^{-2}
a_4	a_2	0.52	0.54	0.46	$p_1^5 p_2 p_4$	2.97×10^{-4}
a_3	a_1	0.41	0.41	0.49	$p_1^6 p_3$	3.53×10^{-3}
a_2	a_2	0.57	0.58	0.51	$p_1^5 p_2^2$	6.55×10^{-3}
a_2	a_1	0.59	0.59	0.52	$p_1^6 p_2$	1.22×10^{-2}
a_3	a_2	0.40	0.40	0.48	$p_1^5 p_2 p_3$	1.90×10^{-3}
a_4	a_1	0.54	0.55	0.47	$p_1^6 p_4$	5.52×10^{-4}
a_1	a_2	0.45	0.44	0.53	$p_1^6 p_2$	1.22×10^{-2}

4.5.3 Simulation of $\pi_5(\mathbf{e})$ with Bit Metric Decoding

Sec. 4.5.2 revealed that $\pi_5(\mathbf{e})$ achieved the lowest approximate BMD TUB curve from among all possible bit mappings for the DVB-S2 code. This section simulates BMD on the DVB-S2 code with interleaving of the parity bits to achieve $\pi_5(\mathbf{e})$. The interleaver permutes the parity bits such that the weight-nine errors conform to the bit mapping specified by $\pi_5(\mathbf{e}) = (4, 4, 4, 4, 4, 5, 6)_8$. Specifically, the original parity bit sequence of [105] is written column-by column into 60 columns of 360 bits each and then read out row by row from the 360 rows.

Fig. 4.6 compares the FER performance of the DVB-S2 code with the interleaving that achieves $\pi_5(\mathbf{e})$ (shown with diamond markers) with the standard DVB-S2 code, characterized by the error pattern $\pi_0(\mathbf{e})$ (shown as triangular markers). Monte Carlo simulations were performed using a CUDA-based GPU implementation with floating-point precision and 100

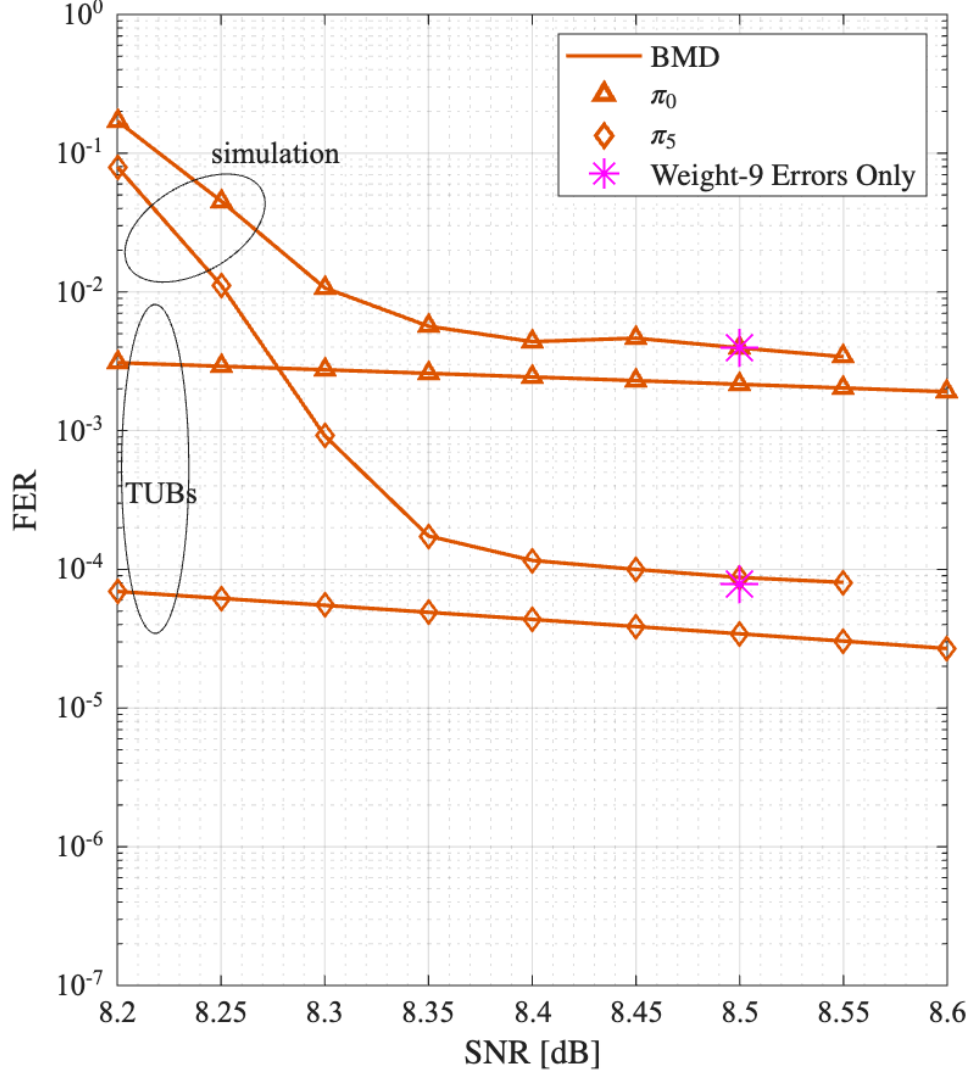


Figure 4.6: Frame error rate (FER) performance of DVB-S2 code with the π_0 (triangular markers) and π_5 (diamond markers) bit interleavers. Also shown are the approximate BMD TUBs for these two bit interleavers.

decoding iterations. The decoder used is the layered min-sum-with-correction decoder as described in [100]. In the error-floor region ($\text{SNR} > 8.35$ dB), the interleaved parity code achieves more than one-order-of-magnitude improvement in FER over the standard DVB-S2 code.

Both simulations approach their respective approximate BMD TUB curves, but there is a gap between the simulation and the TUB. This gap is due in part to the truncation, i.e. the simulation has frame errors that are caused by other decoding failures than neighboring

codewords that differ from the transmitted codeword by Hamming distance nine and $d_{w.c.}$. However, the difference between simulation and the approximate BMD TUB is also due in part to the approximate nature of the TUB, since the BMD decoding regions are more complicated than the regions defined by half planes that are used to construct the approximate TUB.

To explore how much of the gap is due to truncation and how much is due to the approximation of the TUB decision boundaries by half-planes, we include points shown as magenta asterisks that plot the FER including only the frame errors caused by neighboring codewords that differ from the transmitted codeword by Hamming distance nine. If the approximation of the TUB decision boundaries by half-planes was precisely accurate, then we would expect the asterisk to lie exactly on the approximate BMD TUB curve. Fig. 4.6 shows that the asterisks actually lay on the simulation curve indicating that the inaccuracy introduced by the approximation of the TUB decision boundaries by half-planes accounts for most of the gap between the BMD TUB and the simulated error floor.

4.6 SMD TUB and SMD Post Processing

This section compares the approximate BMD TUBs and SMD TUBs for the DVB-S2 code for all possible weight-nine error patterns, and develops the post-processing technique to approach SMD decoding.

4.6.1 Comparing BMD and SMD TUBs for DVB-S2 Code

In Sec. 4.5, Table 4.6 revealed that the BMD λ_i values do not always agree with the SMD λ_i values. Sometimes, the BMD λ_i values favor the less likely codeword while the SMD λ_i values always favor the more likely codeword as required by (15). When the the BMD λ_i favors the less likely codeword, this will yield a higher FER than the optimal SMD decoding.

To find out how much loss BMD decoding incurs for these misplaced thresholds, Fig. 4.7

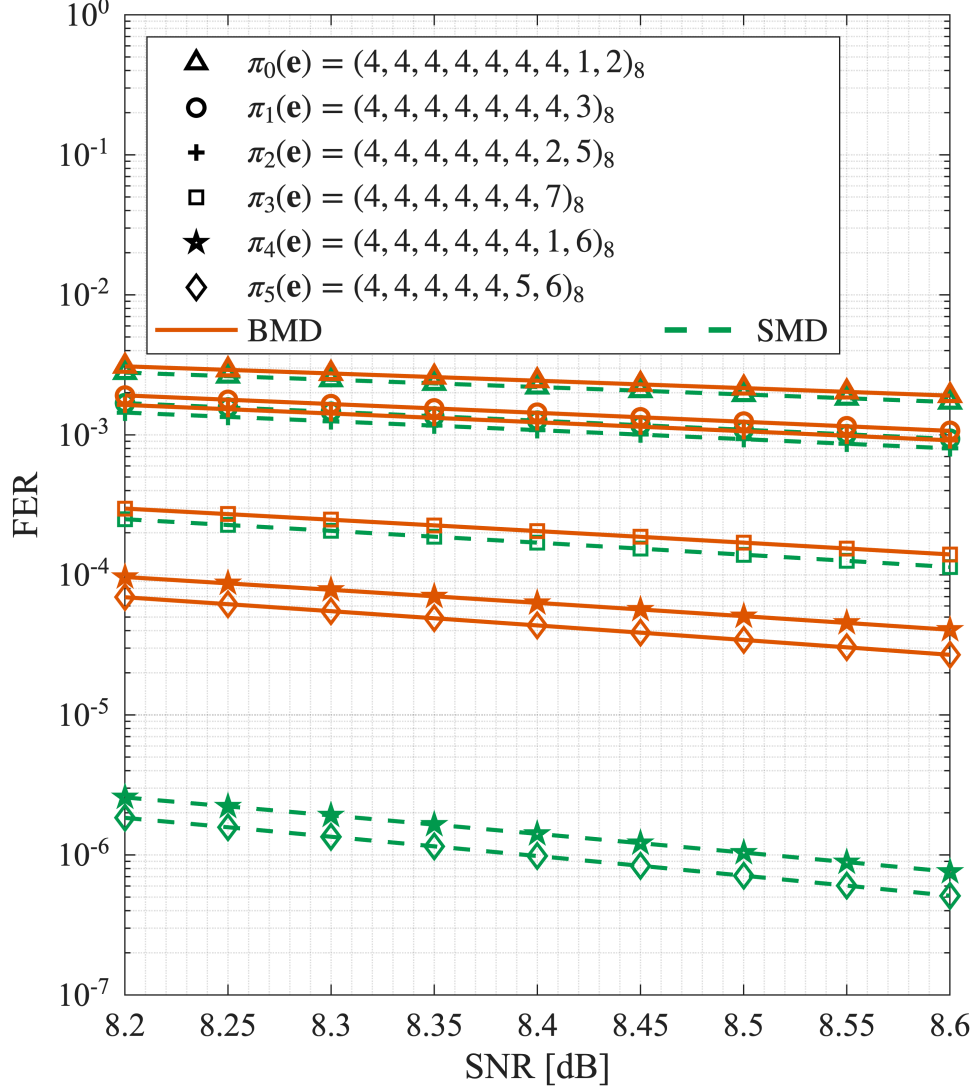


Figure 4.7: SMD and BMD TUBs for the DVB-S2 LDPC code for all possible bit interleavers $\pi_0(\mathbf{e}) - \pi_5(\mathbf{e})$ corresponding to neighboring codewords that differ from the transmitted codeword by Hamming distance nine and $d_{w.c.}$.

shows both SMD and BMD TUBs for the truncation to errors caused by neighboring codewords that differ from the transmitted codeword by Hamming distance nine and $d_{w.c.}$. As with Fig. 4.5, TUBs are shown for all of the possible bit mappings $\pi_0(\mathbf{e}) - \pi_5(\mathbf{e})$. For $\pi_0(\mathbf{e}) - \pi_3(\mathbf{e})$, the SMD TUB curves are slightly below, but very similar to the approximate BMD TUBs for those bit mappings. However, for $\pi_4(\mathbf{e})$ and $\pi_5(\mathbf{e})$, the SMD TUB curves are significantly below their BMD counterparts. In fact, for these two cases the SMD TUBs are more than an

order of magnitude better than the approximate TUB curves.

Both $\pi_4(\mathbf{e})$ and $\pi_5(\mathbf{e})$ contain the octal 6 error pattern, i.e. $e = 110$, and in both cases we observe the behavior that BMD favors the less likely codeword when amplitude a_1 or a_4 is transmitted in the symbol position where the $e = 110$ error takes place.

4.6.2 Post-processing to Approach SMD Decoding Performance

As shown in Fig 4.7, when the dominant errors in the error-floor region are caused by low-weight codewords and BMD decoding favors the less likely codeword in its decision region for some of the low-weight codewords that dominate the error floor, SMD decoding should provide significantly improved LDPC decoding. However, the graph-based message-passing decoding used for LDPC does not generally support SMD decoding, and BMD decoding is adopted out of necessity.

However, with the knowledge of the specific dominant error patterns where the sub-optimal BMD decision regions are causing a high error floor, the SMD post-processing technique described in Algorithm 2 can mitigate the loss due to those specific sub-optimal decision regions and thus lower the error floor.

Algorithm 2 SMD Post-Processing of dominant errors

1. Perform standard message-passing LDPC decoding for each codeword.
 2. Generate the A_w weight- w neighboring codewords.
For the DVB-S2 code $A_w = 360$, $w = 9$.
 3. Compute the SMD metric for each neighbor.
For the parity-interleaved DVB-S2 code, each comparison involves only seven symbols.
 4. Select the candidate with the best SMD metric.
-

Fig. 4.8 shows that applying Algorithm 2 to the DVB-S2 code with the π_5 bit mapping lowers the error floor by additional order of magnitude, resulting in total of two orders in magnitude error floor reduction compared to the DVB-S2 code with standard π_0 bit mapping and standard message-passing decoding. The FER performance of the parity-interleaved

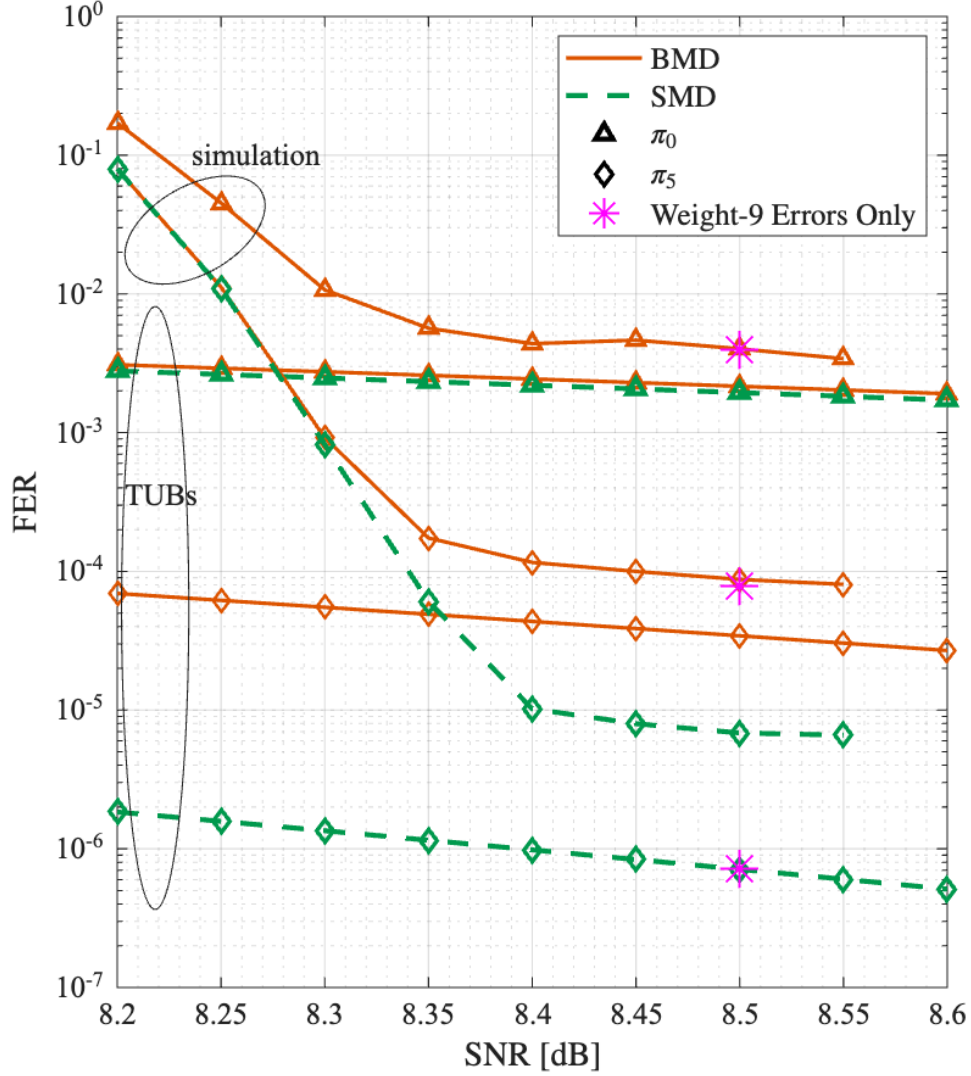


Figure 4.8: This figure builds on Fig. 4.6, adding as green dashed lines the SMD TUB curves for π_0 (triangle markers) and π_5 (diamond markers) and the FER performance of LDPC decoding with the post-processing of Algorithm 2 for the DVB-S2 code with the π_5 bit mapping (diamond markers). The magenta asterisk at SNR of 8.5 dB is computed using neighboring codewords that differ from the transmitted codeword by Hamming distance nine.

DVB-S2 code with post-processing as described in Algorithm 2 is represented by green-dashed simulation curve with diamond markers in Fig. 4.8. The magenta asterisk marker represents the FER at SNR = 8.5 dB obtained using the post-processing technique and counting only weight-nine errors. It lies exactly on the TUB for SMD decoding, shown by the dashed green line with diamond markers. Thus, the behavior of the weight-nine errors is captured precisely

by the SMD TUB and the overall FER lies above the SMD TUB because there are additional errors other than the weight-nine errors that are now affecting the error floor.

4.7 Conclusion

This chapter derives an analytical TUB for SMD decoding and an approximate TUB for BMD decoding for LDPC codes with PAS. The BMD TUB facilitates the design of improved bit interleavers that reduce the error floor in scenarios where the dominant error events of the LDPC code are associated with low-weight codewords. The rate-2/3 DVB-S2 LDPC code coupled with PAS and 8-PAM constellation is used as an example application to demonstrate the effectiveness of the proposed analysis. The proposed bit-interleaving method achieves more than an order of magnitude reduction in the error floor region. The BMD TUB gives good estimate of the error floor.

A comparison of the BMD and SMD TUBs shows the sub-optimality of BMD for DVB-S2 code with PAS for certain bit mappings where BMD favors the less likely codeword, implying that further error floor reduction is possible by using a SMD decoder. This chapter develops a low-complexity post-processing technique that reduces the error floor of the DVB-S2 code in these cases by an additional order of magnitude, resulting in an overall error floor reduction of more than two orders of magnitude compared to the DVB-S2 code with standard bit interleaving and standard message-passing decoding.

Chapter 5

Constellation Design to Enable Low-Complexity Encoding

5.1 Introduction

This chapter considers shaping for low density parity check (LDPC) codes [58] from the perspective of encoder complexity.

5.1.1 Background

For the additive white Gaussian noise (AWGN) channel with an average power constraint, a gain up to 1.53 dB can be attained with optimized signaling as compared to equally spaced equally likely (ESEL) constellation points [49].¹ Geometric shaping (GS) optimizes the placement and location of signaling points to match the optimal distribution while keeping the probability of each point uniform [50]. Probabilistic shaping (PS) optimizes the probability mass function (PMF) of equidistant signaling points. Fig. 5.1 shows how shaping affects the widths and heights of the stairs in the Cumulative Distribution Function (CDF) of the 8-PAM

¹This chapter is largely a reprint of work previously published in the IEEE International Symposium on Topics in Coding [53] except Fig. 5.1 and Fig 5.2 and their discussions. Section 5.2 was modified since Chapter 4 provides discussion on PAS system

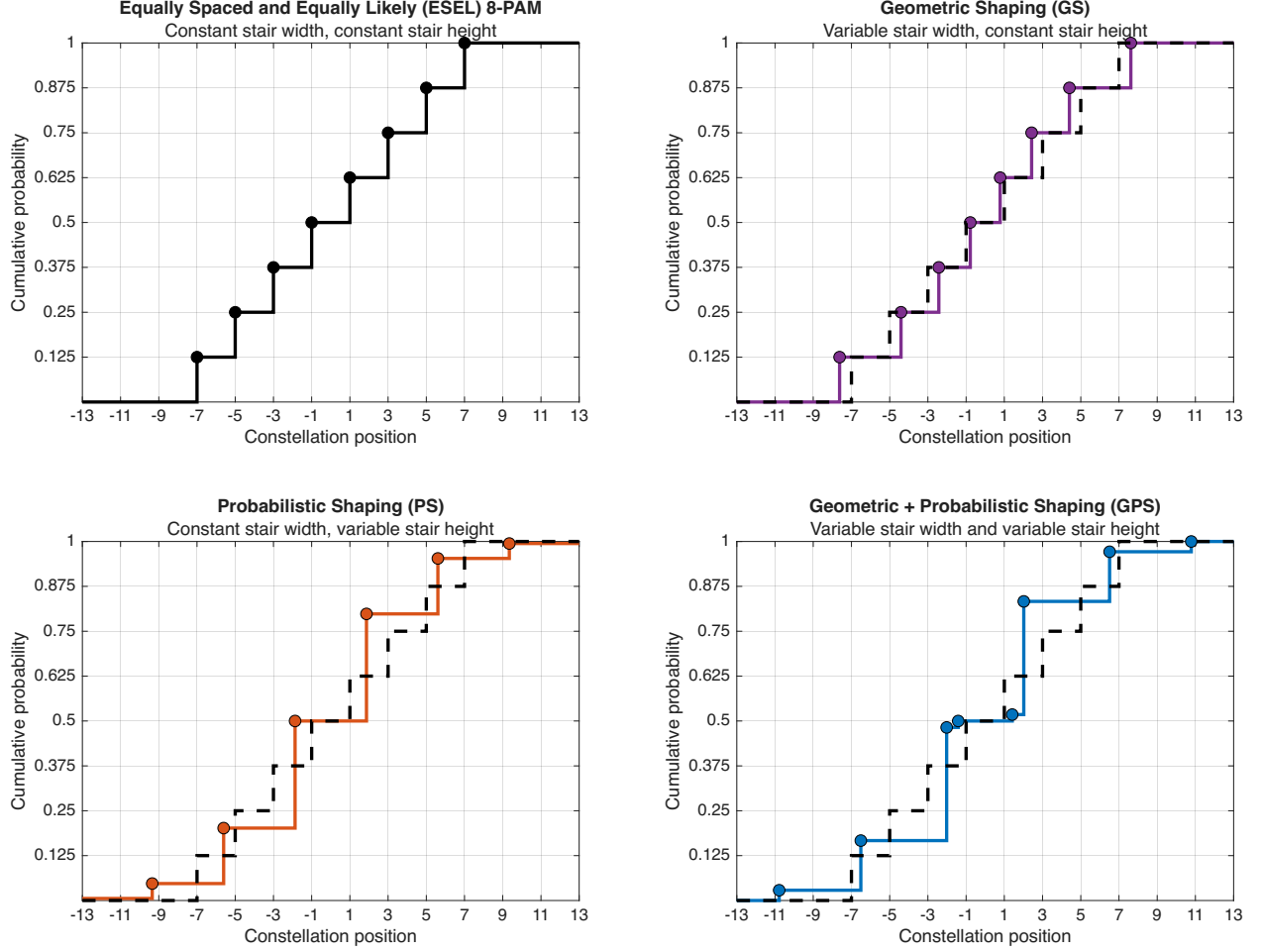


Figure 5.1: Comparison Cumulative Distribution Functions (CDFs) for 8-PAM constellation under different shaping configurations.

constellation under the different shaping techniques. Each shaped constellation is scaled so that the average power is equal to the average power of the ESEL 8-PAM constellation with amplitudes 1, 3, 5, and 7.

The widths and heights of the stairs in the CDF of the ESEL 8-PAM constellation are constant. For reference, the CDF of the ESEL 8-PAM constellation is also added as a dashed black line to the three other subplots in Fig. 5.1, which represent CDFs of the three possible shaping strategies: geometric shaping, probabilistic shaping, and the combination of both geometric and probabilistic shaping.

Geometric shaping has constant stair height, but variable stair width. The geometrically shaped constellation in Fig. 5.1 is shaped using the method described in [50]. Probabilistic shaping has constant stair width but variable stair height. The probabilistically shaped constellation in Fig. 5.1 is shaped according to the MB distribution, which was described in Chapter 4. Geometric and probabilistic shaping has the most flexibility with variable stair height and stair width. The geometrically and probabilistically shaped constellation in Fig. 5.1 is shaped to equalize conditional entropies, as will be described later in this chapter.

Some practical applications of probabilistic shaping are discussed in [59, Sec. IV.B], [15, 37, 48, 72, 76, 90]. When combined with binary forward error correction (FEC) codes, such shaping schemes may require iterative demapping-decoding at the receiver, which increases receiver complexity [18].

Probabilistic amplitude shaping (PAS) as proposed by Bocherer et al. [18] is a practical coded modulation scheme that combines a non-uniform PMF on equidistant signaling points with binary FEC without requiring iterative demapping [88] at the receiver. Instead, bit-metric decoding (BMD) [63, 86] is used. At the transmitter, a distribution matcher (DM) [102] produces a sequence of signaling amplitudes according to the desired PMF. The DM is followed by systematic encoder that adds parity bits to the binary labels of the DM amplitudes while preserving the PMF on the amplitudes. The parity bits specify the sign of each signaling point, and that sign is applied to the DM amplitude to specify the final signaling level. Results in [18] show that at frame error rate (FER) of 10^{-3} , the PAS scheme operates within less than 1.1 dB of the AWGN capacity over a wide range of spectral efficiencies.

Bit-interleaved coded modulation (BICM) with BMD [20, 127] is widely used in wireless and satellite communication systems, in conjunction with Gray mapping and capacity-approaching binary iterative coding schemes, to achieve high spectral efficiencies with low-complexity decoding. When BICM is coupled with LDPC codes, the code design should consider the different reliabilities of the bit levels in the constellation label. For example, in [80] a bit-reliability-mapping strategy is proposed, which carefully allocated codeword bits to the

constellation bit levels by matching the corresponding variable node degrees to the bit level reliability. This approach was followed also in [68, 78]. In [66], the LDPC code design explicitly considers the target modulation, and, hence, the different reliability of the bit levels. The approach of [66] relies on a density evolution (DE) analysis, which is tailored to BMD by introducing the idea of channel adapters to address the issue of the missing symmetry in the BICM bit-channels.

The problem of adapting an LDPC code to different bit-level reliabilities is present in PAS as well. In 2016, Steiner et al. [105] proposed a protograph-based LDPC code design that jointly optimizes the code structure and the mapping of the coded bits to the BICM bit-levels. The proposed optimization scheme is applied to both uniform and non-uniform inputs. Following observations that the LDPC code optimized on surrogate channel performs similarly when evaluated on the target channel [92], authors of [105] represent each bit-channel by a uniform binary-input symmetric-output AWGN surrogate channel, which describes both the BICM bit-channels and the non-uniform input distribution. To determine the belief propagation convergence threshold of the designed protograph ensembles, the authors used the protograph extrinsic information transfer (P-EXIT) analysis [82].

When modeling the observation of a bit at the output of the demapper as the observation of the bit at the output of a surrogate BI-AWGN channel, the analysis of [105] requires matching the conditional entropy of the bit given the demapper output with the conditional entropy of the surrogate BI-AWGN channel.

5.1.2 The Problem

Although the LDPC codes designed by Steiner et al. [105] show excellent FER performance, their simulation process relied on channel adapters instead of encoding actual information bits, leaving the practical encoding complexity unclear. In fact, the PAS scheme requires a systematic encoder, which ensures that the bits representing amplitudes are systematic

and the bit representing the sign is a parity bit. When the LDPC code is optimized with respect to the different constellation bit-level reliabilities, this may result in protographs that have high variable node degrees assigned to parity bits, making it difficult to construct a low-complexity encoder using standard techniques, such as the repeat-accumulator (RA) structure [32], or the Raptor-like structure [21, 22]. The codes designed in [105] have these problematic protographs.

5.1.3 Contributions

This chapter proposes combined geometric and probabilistic shaping approach that builds on the PAS architecture of [18] to simplify the encoder. Optimizing the positions and probabilities of PAM constellation points to equalize the conditional entropies of the different bit-levels decouples the LDPC code design (and bit mapping) problem from the underlying constellation properties. This renders the performance of an LDPC code independent of the specific choice of the bit mapping. It turns out that the equalization of the conditional entropies can be accomplished with a negligible impact on the achievable rates of PAS. This allows the employment of a bit mapping that facilitates low-complexity encoding, for example, by associating the sign bits to the output of the inner accumulator in an RA-like LDPC code construction.

Equalizing the conditional entropies of the bit-levels allows PAS to approach the Shannon limit using any of the numerous existing LDPC code constructions that allow low-complexity encoders, which were originally developed for the BI-AWGN channel. We demonstrate the effectiveness of the proposed approach by modifying signaling PMF and the bit mapping of a code from [105] to support efficient encoding, showing how the modified constellation yields a performance that closely competes with the one of the original constellation-specific mapping of [105]. We then show how a protograph LDPC code optimized for the BI-AWGN channel with the proposed constellation design outperforms the design of [105].

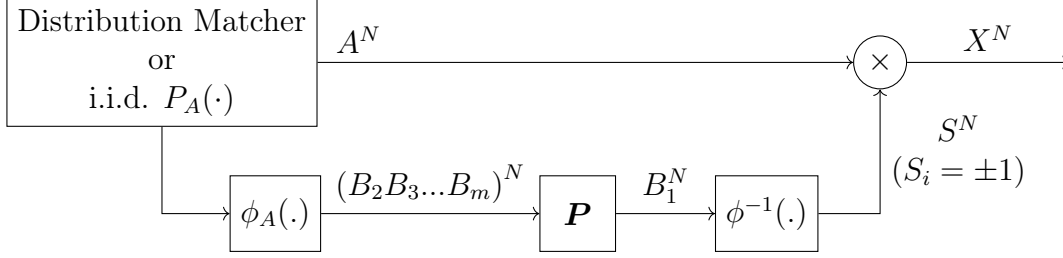


Figure 5.2: Probabilistic amplitude shaping (PAS) encoder.

5.1.4 Organization

The chapter is organized as follows: Section 5.2 reviews PAS. Section 5.3 introduces the optimization of the constellation points to equalize conditional entropy across bit levels. Section 5.4 introduces a protograph LDPC code design optimized for the BI-AWGN channel, which admits linear-time encoding. Numerical results are provided in Section 5.5, and conclusions follow in Section 5.6.

5.2 Probabilistic Amplitude Shaping

Consider the discrete-time AWGN channel $Y = X + N$ where the transmit symbol X is drawn from a 2^m -PAM signaling with alphabet $\mathcal{X} = \{\pm a_1, \pm a_2, \dots, \pm a_{2^{m-1}}\} \in \mathbb{R}$ and the noise N is Gaussian random variable with zero mean and variance σ^2 . The signal-to-noise ratio (SNR) is defined as $\text{SNR} = P/\sigma^2$, where σ^2 is the noise variance.

A full PAS encoding block diagram is presented in Chapter 4 Sec. 4.2.2. The constellation labeling in PAS is typically a binary reflected Gray code [61], such that adjacent constellation points have binary labels that differ in only one bit position. Recall that, with PAS, the transmit symbol is factored as $X = A \times S$, where $A = |X|$ is the amplitude and $S = \text{sign}(X)$ is the sign of X . Fig. 5.2 shows an encoder for encoding an LDPC code with PAS, starting from the distribution matcher and forming a sequence X^N of constellation points for transmission. The distribution matcher or a discrete memoryless source $P_A(\cdot)$ outputs a sequence of

amplitudes A^N that are i.i.d. according to the distribution $P_A(\cdot)$. A labeling function $\phi_A : \mathcal{A} \rightarrow \{0, 1\}^{m-1}$ assigns an $(m-1)$ -bit binary label to each amplitude A_i , producing a binary sequence $(B_2 B_3 \dots B_m)^N$ consisting of $(m-1) \times N$ bits. This sequence is then multiplied by the parity matrix $\mathbf{P}$ to generate N sign labels, which are subsequently transformed into the corresponding signs ($S_i = \pm 1$), producing a sequence S^N of signs. The parity matrix $\mathbf{P}$ is obtained from the systematic LDPC encoder $\mathbf{G} = [\mathbf{I}_k | \mathbf{P}]$, where $k = (m-1) \times N$. Each amplitude A_i is then multiplied by the corresponding sign S_i in the order in which they were produced to form a sequence X^N of N PAM symbols for transmission. The sign S is uniformly distributed, i.e. $P_S(1) = P_S(-1) = 1/2$ and A and S are stochastically independent. P_X is symmetric around zero, i.e., $P_X(x) = P_X(-x)$, and can be factored as

$$P_X(x) = P_A(|x|)P_S(\text{sign}(x)), \forall x \in \mathcal{X}. \quad (5.1)$$

Our designs constrain the average power $P = \mathbb{E}[X^2]$ of the constellation to be that of equally likely equidistant 2^m -PAM signaling with $\mathcal{X} = \{\pm 1, \pm 3, \dots, \pm(2^m - 1)\}$.

In this chapter, LDPC codes with PAS are decoded with message passing decoding using BMD as described in Chapter 4, Sec. 4.2.2. The equation used to calculate the soft information L_i for each bit-level is given in (4.2).

5.3 Equalizing Conditional Entropies

An achievable rate for BMD [17, 18] is as follows:

$$R_{\text{BMD}} = \left[H(\mathbf{B}) - \sum_{i=1}^m H(B_i | Y) \right]^+, \quad (5.2)$$

where $H(\cdot)$ denotes entropy and $[x]^+ = \max\{0, x\}$. As shown in [105], the iterative decoding threshold, computed via P-EXIT analysis, depends on conditional entropies of the bit levels

associated with the protograph variable nodes. Hence, optimizing a protograph for the iterative decoding threshold results in a specific mapping between bit-levels and variable nodes according to conditional entropies $H(B_i|Y)$. A signaling scheme that yields, at the desired working SNR, equal conditional entropies $H(B_i|Y)$ allows the iterative decoding threshold computed via P-EXIT analysis to be independent of the assignment of bit levels to variable nodes. Consider the two objective functions

$$f(\mathcal{X}, P_X) = \sum_{j=1}^m H(B_j|Y) \quad (5.3)$$

$$g(\mathcal{X}, P_X) = \max_j \{H(B_j|Y)\}_{j=1}^m. \quad (5.4)$$

Assuming constant $H(\mathbf{B})$, minimizing $f(\cdot)$ maximizes R_{BMD} in (5.2). Also, minimizing $g(\cdot)$ for a given R_{BMD} equalizes conditional entropies $H(B_i|Y)$. A weighted sum of $f(\cdot)$ and $g(\cdot)$ allows a trade-off between equalizing conditional entropies $H(B_i|Y)$ and maximizing R_{BMD} as follows:

$$\begin{aligned} \min_{\mathcal{A}, \{P_A(|x_i|)\}_{i=1}^{2^m-1}} & [\alpha f(\mathcal{X}, P_X) + (1 - \alpha)g(\mathcal{X}, P_X)] \\ \text{s.t. } & x_k < x_{k+1}, \quad \mathbb{E}[X^2] = P, \quad H(\mathbf{B}) = h, \end{aligned} \quad (5.5)$$

where $\mathcal{A} = \{|x_1|, |x_2|, \dots, |x_{2^m-1}|\}$ and $\{P_A(|x_i|)\}_{i=1}^{2^m-1}$ are the amplitude probabilities. The entropy $H(\mathbf{B})$ is fixed to a constant value h according to the desired spectral efficiency. P is the average power, and α is the weighting factor between $f(\cdot)$ and $g(\cdot)$. Setting α low focuses on equalizing conditional entropies by placing more weight on $g(\cdot)$. In contrast, choosing α closer to 1 focuses on maximizing R_{BMD} .

5.3.1 Example Application to 8-PAM

Consider achieving a target spectral efficiency of 1.34 bits per channel use at an SNR of 8.3 dB using 8-PAM and an $R_{\text{FEC}} = 2/3$ code. The spectral efficiency (SE) for PAS is

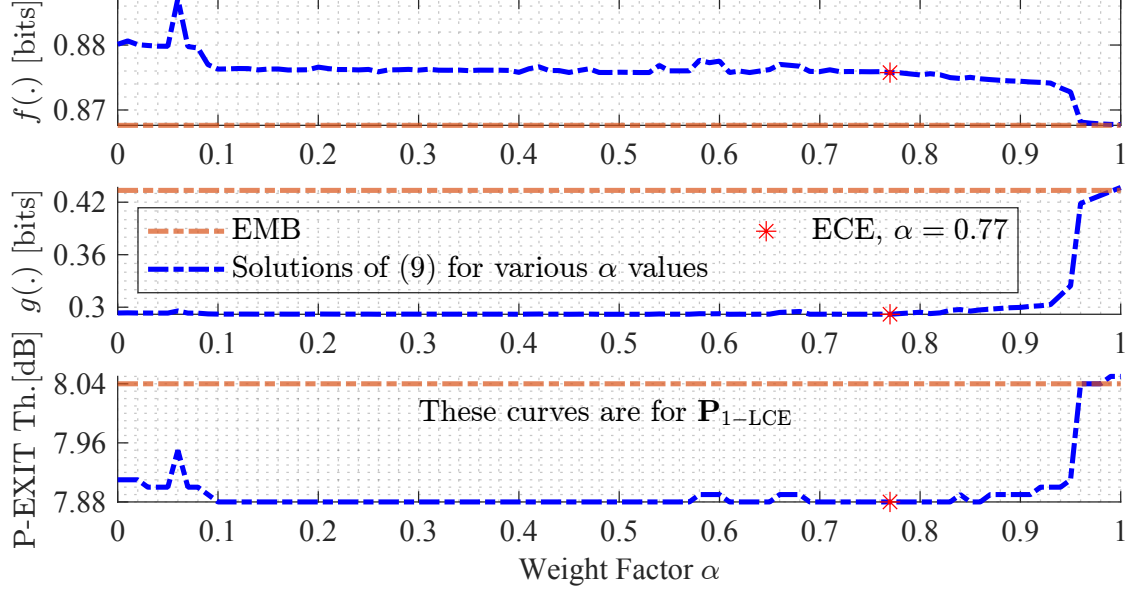


Figure 5.3: Objectives $f(\cdot)$ of (5.3), $g(\cdot)$ of (5.4) and P-EXIT threshold for $\mathbf{P}_{1-LCE}$ of (5.7) as a function of weight factor α . P-EXIT threshold is influenced by both $f(\cdot)$ and $g(\cdot)$, but the range of $f(\cdot)$ is an order of magnitude smaller than the range of $g(\cdot)$. The solution at $\alpha = 0.77$ minimizes maximum conditional entropy $g(\cdot)$ achieving the minimum threshold of 7.88 dB.

calculated as $SE = H(\mathbf{B}) - m(1 - R_{FEC}) = H(\mathbf{B}) - 1$. Thus $H(\mathbf{B})$ in (5.5) must be equal to $h = 2.34$. Fig. 5.3 shows $f(\cdot)$, $g(\cdot)$, and the P-EXIT threshold, (which will be discussed below) as a function of α in steps of 0.01 from zero to 1. For each α , Fig. 5.3 shows these three properties for solutions to the optimization problem (5.5) found using the MATLAB solver *MultiStart*, which returns the best of the multiple local solutions found by starting from various initial conditions. Fig. 5.4 shows the signaling PMF for equidistant equidistant Maxwell-Boltzmann (EMB) signaling for $SE = 1.34$ bits per channel use as used in [105]. This EMB signaling PMF was also the initial condition provided to *MultiStart* for each α . Fig. 5.4 also shows the signaling PMF found by *MultiStart* for $\alpha = 0.77$.

Fig. 5.5 compares the conditional entropies for the signaling PMFs given in Fig. 5.4. Selecting $\alpha = 0.77$ achieves equal conditional entropies (ECE) with only a slight increase in the sum of the conditional entropies from 0.8676 to 0.8758. This increase leads to a reduction of 0.0082 in R_{BMD} of (5.2).

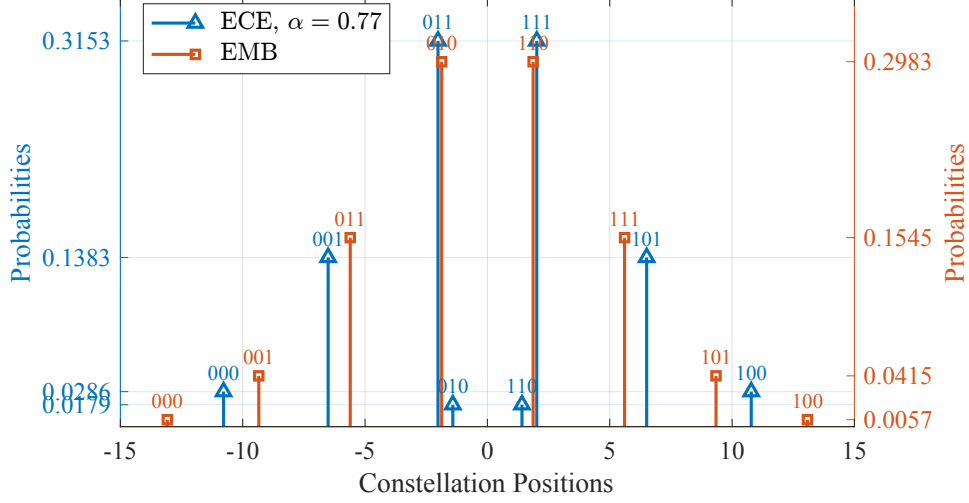


Figure 5.4: Equidistant Maxwell-Boltzmann (EMB) signaling PMF [105] that was the initial condition for (5.5) and the solution found by *MultiStart* for $\alpha = 0.77$ which achieves equal conditional entropies (ECE). $SE = 1.34$ for both.

Now let's explore how achieving ECE affects the P-EXIT threshold for specific protographs. For $R_{\text{FEC}} = 2/3$, a protograph-based code (5.6) designed by Steiner et al. for 8-PAM with EMB signaling has the following protomatrix $\mathbf{P}_1$:

$$\mathbf{P}_1 = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 6 \\ 2 & 2 & 1 & 1 & 2 & 6 \end{pmatrix}. \quad (5.6)$$

$\underbrace{\hspace{1.5cm}}_{B_2} \quad \underbrace{\hspace{1.5cm}}_{B_3} \quad \underbrace{\hspace{1.5cm}}_{B_1}$

The protomatrix (5.6) has P-EXIT threshold of 7.77 dB for EMB signaling, but the high-degree column associated to sign bit B_1 makes construction of an efficient encoder difficult. Since this protograph has two degree-2 columns, it is possible to construct a linear-time encoder by performing an appropriate lifting of the protograph based on the irregular-repeat-accumulate (IRA) structure [69], provided that these degree-2 columns are mapped to B_1 .

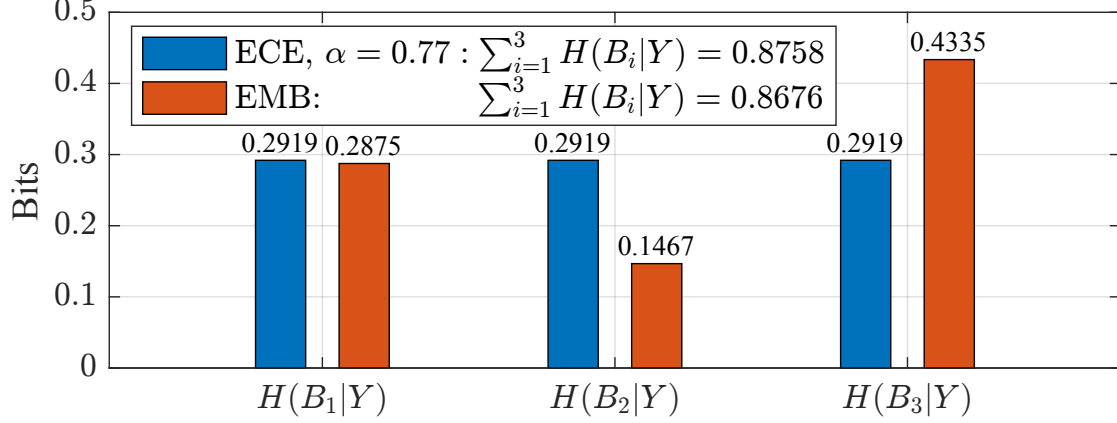


Figure 5.5: Conditional entropies for the two signaling PMF's shown in Fig. 5.4. The $\alpha = 0.77$ solution achieves equal conditional entropies.

Permuting the columns of the Steiner et al. protomatrix (5.6) results in the following:

$$\mathbf{P}_{1-\text{LCE}} = \begin{pmatrix} 6 & 1 & 1 & 1 & 1 & 1 \\ 6 & 2 & 2 & 2 & 1 & 1 \end{pmatrix}, \quad (5.7)$$

$\underbrace{\hspace{1.5cm}}_{B_2} \quad \underbrace{\hspace{1.5cm}}_{B_3} \quad \underbrace{\hspace{1.5cm}}_{B_1}$

where subscript LCE indicates that the protograph facilitates low-complexity encoding. This change in bit mapping to enable low-complexity encoding increases the P-EXIT threshold for EMB signaling from 7.77 dB to 8.04 dB. Fig. 5.3 shows the P-EXIT threshold calculated for the protomatrix (5.7) for the signaling PMFs that optimize (5.5) as a function of the weight factor α . In particular, the $\alpha = 0.77$ solution minimizes $g(\cdot)$ to achieve equal conditional entropies (ECE) and achieves the minimum P-EXIT threshold of 7.88 dB. Note that there are a wide range of signaling PMFs for $0.1 < \alpha < 0.85$ that achieve (or almost achieve) ECE and have excellent thresholds near 7.88 dB.

As α approaches 1 and $g(\cdot)$ increases, the ECE condition is not met, and the P-EXIT threshold increases to 8.04 dB, which is the threshold of the EMB solution. The ECE condition decouples bit mapping from the threshold so that both $\mathbf{P}_{1-\text{LCE}}$ and $\mathbf{P}_1$ have the same threshold of 7.88 dB, but with the new ECE signaling PMF, we can approach the

original threshold while also implementing an efficient encoder.

5.4 Protograph-based LDPC Code Design

$\mathbf{P}_1$ was optimized for EMB signaling, but $\mathbf{P}_{1\text{-LCE}}$ was not optimized for the ECE signaling. This section introduces a low-complexity encoding (LCE) protograph-based LDPC code specifically designed for ECE signaling, which is the same as optimizing for the BI-AWGN channel. Optimizing the reciprocal channel approximation (RCA) [30] threshold yields the following rate-2/3 protomatrix:

$$\mathbf{P}_{2\text{-LCE}} = \begin{pmatrix} 1 & 0 & 0 & 2 & 0 & 1 & 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 & 2 & 1 & 0 & 0 \\ 2 & 1 & 2 & 1 & 2 & 0 & 1 & 0 & 1 & 1 \\ 1 & 2 & 1 & 2 & 1 & 0 & 2 & 0 & 1 & 1 \end{pmatrix}. \quad (5.8)$$

$$\underbrace{\quad}_{B_2} \quad \underbrace{\quad}_{B_3} \quad \underbrace{\quad}_{punct.} \quad \underbrace{\quad}_{B_1} \quad (5.9)$$

The first parity column of $\mathbf{P}_{2\text{-LCE}}$ in (5.8) is punctured and the rest of the parity columns are mapped to the sign bit B_1 . The message bits are mapped to bits B_2 and B_3 .

The lifting of $\mathbf{P}_{2\text{-LCE}}$ constrains the final matrix $\mathbf{H}$ to have a form of Fig. 5.6, where $\mathbf{H}_{p_1}$ and $\mathbf{H}_{p_3}$ both have the quasi-cyclic repeat-accumulate (QCRA) [109] structure. The upper right corners of the sub-matrices $\mathbf{H}_{p_1}$ and $\mathbf{H}_{p_3}$ initially both contained a left circular 1800×1800 identity matrix $\mathbf{I}_1$ after lifting, with all zeros in the first row except for a “1” in the last column. This “1” is removed from $\mathbf{H}_{p_1}$ and $\mathbf{H}_{p_3}$ in Fig. 5.6 to enable a low-complexity encoder without affecting the P-EXIT threshold or FER performance. Fig. 5.7 shows a systematic LDPC encoder for $\mathbf{H}$. The columns and rows of $\mathbf{H}$ corresponding to block matrices $\mathbf{H}_{p_1}$ and $\mathbf{H}_{p_3}$ can be permuted so that both $\mathbf{H}_{p_1}$ and $\mathbf{H}_{p_3}$ become “dual-diagonal” matrices [69]. This allows encoding with the complexity that grows linearly in blocklength by concatenating

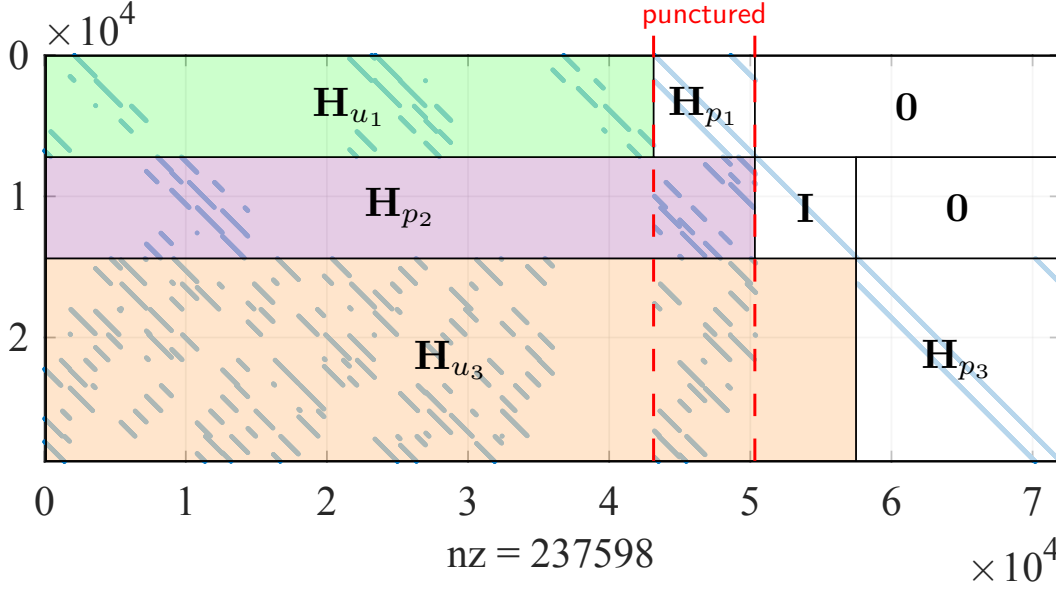


Figure 5.6: Lifted matrix $\mathbf{H}$ from protomatrix $\mathbf{P}_{1\text{-LCE}}$ (5.8). The message length $k = 43,200$, number of variable nodes $N = 72,000$, number of punctured nodes $n_p = 7,200$, blocklength $n = 64,800$.

an accumulator with sparse matrix multiplication with $\mathbf{H}_{u_1}^T$, low-density generator matrix $[\mathbf{I} \ \mathbf{H}_{p_2}^T]$ and an accumulator with sparse matrix multiplication with $\mathbf{H}_{u_3}^T$.

Lifting also optimized approximate-cycle extrinsic-message-degree (ACE) [111]. All the cycles whose length is $2d_{\text{ACE}} = 12$ or less in the bipartite graph associated with the parity check matrix have ACE values of at least $\eta_{\text{ACE}} = 5$.

5.5 Simulation Results

Fig. 5.8 shows FER performance and P-EXIT thresholds of codes $\mathbf{P}_1$ (5.6), $\mathbf{P}_{1\text{-LCE}}$ (5.7), and $\mathbf{P}_{2\text{-LCE}}$ (5.8) using the ECE and EMB signaling described in Fig. 5.8. $\mathbf{P}_{2\text{-LCE}}$ (5.8) with LCE and ECE signaling ($\alpha = 0.77$) outperforms $\mathbf{P}_1$, which does admit LCE) using EMB signaling by 0.08 dB for a target FER of 10^{-3} . With ECE signaling, the FER performance of the permuted code $\mathbf{P}_{1\text{-LCE}}$ is improved by 0.16 dB for a target FER of 10^{-3} compared to the same code using EMB signaling, and 0.12 dB worse than original code (5.6) without LCE

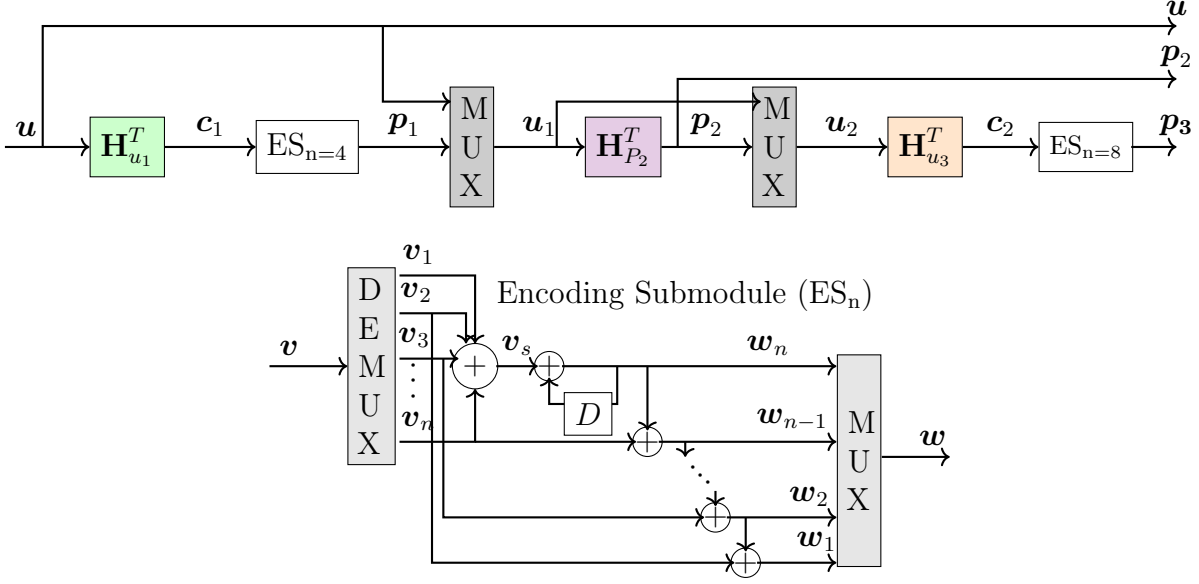


Figure 5.7: Encoder with linear complexity in blocklength for the optimized code in Fig. 5.6. An input to Encoding Submodule ES_n is a binary vector $\mathbf{v}$ consisting of n serially concatenated block vectors $\mathbf{v}_i$, $i = 1, 2, \dots, n$, each of size (1×1800) , and an output is the same size binary vector $\mathbf{w}$ representing encoded parity bits. For example, $ES_{n=4}$ means that $\mathbf{c}_1$ consists of 4 serially concatenated block vectors each of size (1×1800) . MUX concatenates vectors as follows: $\mathbf{u}_1 = [\mathbf{u} \ \mathbf{p}_1]$, $\mathbf{u}_2 = [\mathbf{u}_1 \ \mathbf{p}_2]$, $\mathbf{w} = [\mathbf{w}_1, \dots, \mathbf{w}_n]$.

and using EMB signaling.

The gap $\Delta\text{SNR}|_{\text{dB}} = 0.57$ dB of the code $\mathcal{C}$ described by parity check matrix in Fig. 5.6 to AWGN capacity is calculated as $\Delta\text{SNR}|_{\text{dB}} = \text{SNR}(\text{FER}, \mathcal{C})|_{\text{dB}} - 10 \log_{10}(2^{2R} - 1)$,

where $\text{SNR}(\text{FER}, \mathcal{C})|_{\text{dB}}$ denotes the SNR at which the code achieves $\text{FER} = 10^{-3}$ and $R = 1.34$ is the transmission rate.

5.6 Conclusion

Optimizing positions and probabilities of PAM signaling to equalize the conditional entropies of the different bit-levels enables the construction of LDPC encoders with linear encoding complexity while harvesting capacity-approaching benefits of PAS. The proposed scheme decouples LDPC code design (and bit mapping) from the specific PAM constellation. An example demonstrates capacity-approaching performance with a PAS LDPC code that features

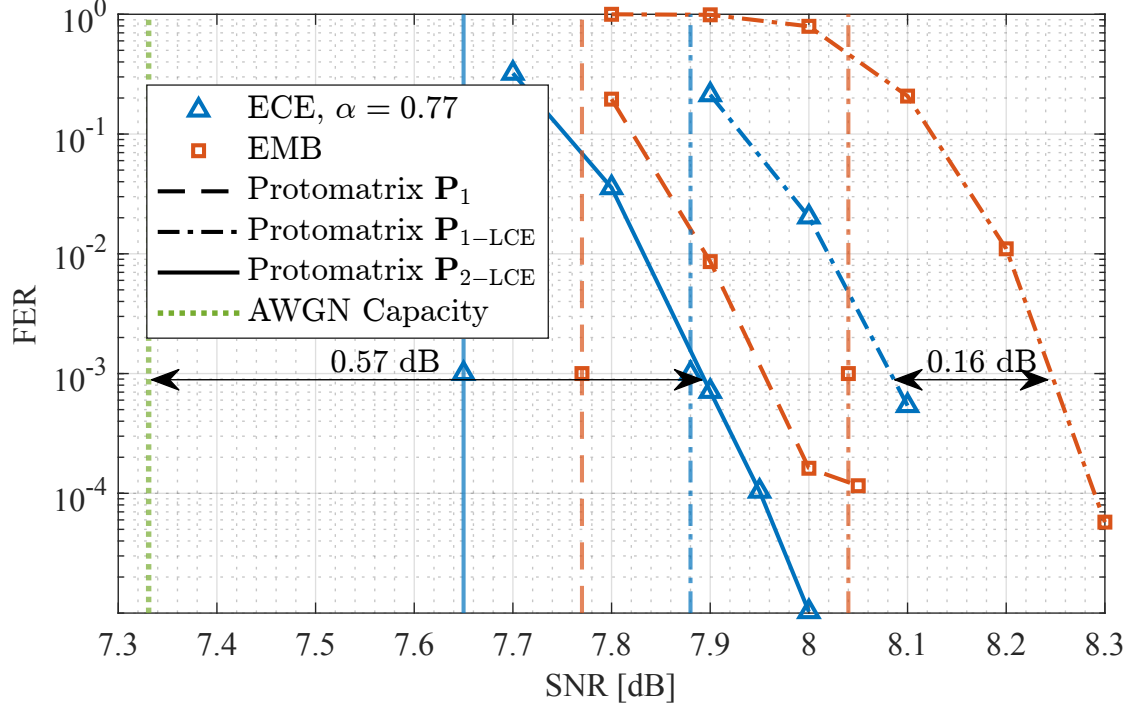


Figure 5.8: FER as a function of SNR for: 1) Protomatrix $\mathbf{P}_1$ (5.6) without low-complexity encoder (LCE) using Equidistant Maxwell-Boltzmann (EMB) signaling (dashed orange curve, square marker). 2) Protomatrix $\mathbf{P}_{1-LCE}$ (5.7) with LCE using equal conditional entropies (ECE) signaling (blue dash-dot curve, triangular marker) and using EMB signaling (dash-dot orange curve, square marker). 3) Designed protomatrix $\mathbf{P}_{2-LCE}$ (5.8) with LCE using ECE signaling (full blue curve, triangular marker). Dotted green vertical line shows the SNR where AWGN capacity is 1.34 bits per channel-use. The rest of vertical lines represent P-EXIT thresholds with values 7.65 dB, 7.77 dB, 7.88 dB, 8.04 dB from left to right.

a low-complexity encoder. We note that equidistant points can facilitate a low-complexity demapper, especially in systems with limited analog-to-digital converter resolution. Future works will investigate constellations that simultaneously can achieve both equal conditional entropies and low-complexity demapping.

Chapter 6

Optimization of Write Voltage Levels for TLC Flash Memory

6.1 Introduction

6.1.1 Background

Flash memory is a non-volatile storage medium invented in the 1980s that quickly became a prominent segment within the semiconductor industry [35].¹ With fast reads and no moving parts, flash memory is widely used for storage and data transfer in consumer devices such as phones, digital cameras, SD cards, tablets and laptops, enterprise systems, data centers and industrial applications. Flash memory can retain data for a long period of time regardless of whether a flash-equipped device is powered on or off. It is small, reliable, and inexpensive which makes it attractive for mobile and miniature products, two major market demands for electronic devices.

To meet demands for information density, flash memory has introduced multilevel cell techniques and technology scaling which degrades the Flash read channel [62]. Sometimes,

¹This chapter is largely a reprint of work published in [56].

the values manufacturers provide in publicly available datasheets differ from the actual Flash device performance, which is significantly worse and highly variable as demonstrated in [62]. These trends exacerbate inter-cell interference and program/erase (P/E) cycling effects. A P/E cycle includes writing (programming) the data to the cells in a page, reading the data from a page (possibly multiple times) and then erasing the data. The Flash memory read channel degrades over time as a function of number of P/E cycles or the cumulative amount of charge that is written into and subsequently erased from the memory cell [24, 79]. Multilevel cell techniques store multiple bits in each cell. For example, triple-level-cell (TLC) Flash stores three bits per cell by using eight levels of write voltage. However, to minimize the read latency, each bit in the same cell is mapped to a different page.

In this manner the bits corresponding to the same cell are encoded independently.

While nearly all practical devices encode each bit in the cell independently as part of a different page, the academic literature sometimes explores Flash memory capacity for the joint-encoding case where all the bits belonging to the same cell are jointly encoded as part of one codeword. Example papers considering joint encoding for Flash memory include [35], [101], [118], [117], [6], [39], [40]. Various methods for maximizing mutual information for encoding and decoding Flash read channels were explored in [118], [117], [6], [39], [40] and [99]. Recent work considering independent encoding in comparison with joint encoding include [122], [67], and [10].

In [122] the benefit of joint encoding for multi-level cell (MLC) is examined for traditional hard decoding and enhanced precision decoding in terms of mutual information (MI). It is shown that for MLC the difference in sum-rate MI between joint and independent encoding is small for hard decoding and negligible for enhanced precision decoding as long as the pages are permitted to have different rates.

In [67], the focus is the differing performance of the possible binary labelings of the cell levels for independent encoding for MLC and TLC flash for various decoding schemes. The results obtained in [67] show that for P/E cycling model the largest sum rate is achieved with

Gray labeling for Treating Interference as Noise (TIN) decoding. Very recently, [10] explored both joint and independent encoding for TLC. They found that joint encoding increases the mutual information and improves hard-input error correction performance with LDPC codes [10].

6.1.2 Contributions

While joint encoding provides an information-theoretic benefit, this chapter focuses exclusively on the practical scenario of independent encoding of Flash under the constraint that all pages must convey the same number of bits. We use a mutual information maximizing paradigm similar to [124] to adapt the locations and probabilities of write levels to increase the mutual information of the weakest bit channel and hence improve the reliability of its corresponding page. In this way, we seek a constellation of write levels that delivers the same amount of mutual information to the bit channel for each page. For simplicity, we restrict attention to TLC Flash with a simplistic additive white Gaussian noise (AWGN) channel model.

6.1.3 Organization

The remaining parts of this chapter are organized into four main sections. Sec. 6.2 describes Flash paradigm for writing and reading information. Sec. 6.3 introduces our problem statement from the mutual information perspective. Sec. 6.4 describes two algorithms for maximizing the minimum information rate in under power and symmetry constraints. Alg. 3 adapts only the positions of write levels in the constellation, seeking to maximize the minimum rate. Alg. 4 adapts both the position and the probability of each levels, again seeking to maximize the minimum rate. Sec. 6.5 concludes the chapter.

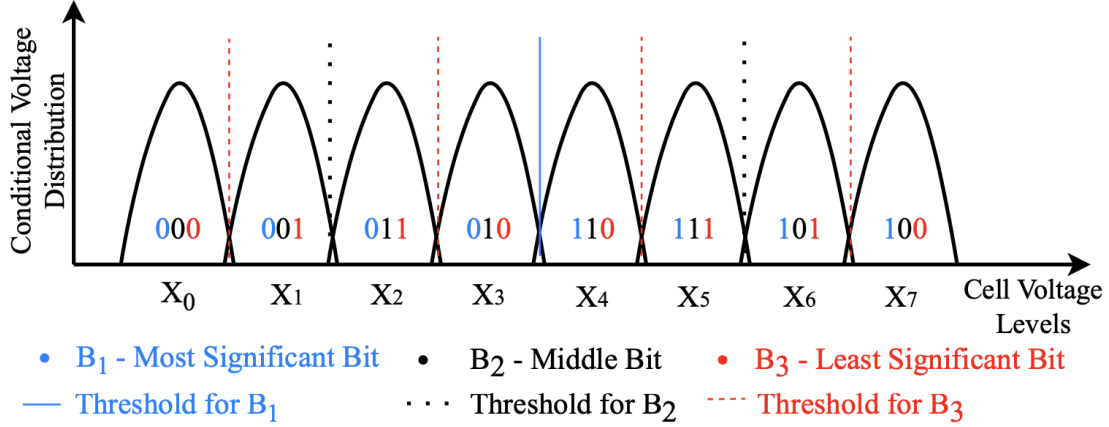


Figure 6.1: The eight voltage levels and Gray labeling for a TLC flash memory. B_1, B_2, B_3 represent three bits corresponding to three different pages. $X_0, \dots, X_7$ are cell voltage levels. Vertical lines represent thresholds for each bit. The read process provides a single bit at a time checking a threshold and reports whether the voltage is above or below that threshold.

6.2 Writing and Reading Flash Information

A basic cell of a flash memory is composed of a floating gate transistor. The amount of charge in the floating gate controls the voltage at which the transistor turns on. We often refer to the write-level as a voltage because the read process will apply a voltage and learn whether the transistor is on or not via a sense-amp comparator.

The data stored in the cell is represented by cell voltage levels. The number of voltage levels corresponds to the number of bits stored in the cell. There are 2^m voltage levels when m coded bits are stored in each cell. Flash technologies are named based on the number of bits they can store. For example, Single Level Cell (SLC) stands for one bit per cell, Multi Level Cell (MLC) indicates two bits per cell, Triple Level Cell (TLC) indicates three bits per cell, etc. Cells are grouped into pages which are grouped into blocks. Pages are the smallest unit for write and read operations.

Since bits belonging to the same cell are mapped independently to different pages, the read process does not provide the actual voltage level, but a single bit at a time checking a threshold and reporting whether the voltage is above or below that threshold. Fig. 6.1

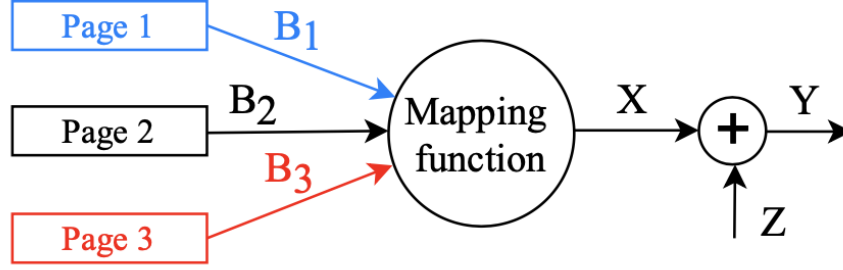


Figure 6.2: Flash Memory Layout as Multiple Access Channel (MAC). The noise Z is drawn i.i.d. from a Gaussian distribution with variance N .

(following [67]) illustrates the eight voltage levels and Gray labeling for a cell in TLC flash memories. During each read process the information obtained is whether the voltage is on the left or the right side of the threshold depicted by vertical lines. We notice that for decoding three pages a total of 7 reads are needed. However, bit B_1 requires only one threshold check, B_2 requires two checks, and B_4 requires four thresholds to be checked. Additional threshold reads can provide soft information [122].

Consider a TLC flash memory with eight voltage levels as illustrated in Fig. 6.1. The three bits B_1, B_2 and B_3 written to a cell for the three independent pages together cause the threshold voltage X to be written to the Flash cell. As noted in [67], this is analogous to a multiple access channel (MAC) with three users as illustrated in Fig. 6.2. When the cell position is read, distortion causes the actual threshold voltage at the time of reading to be Y . Therefore, the system model is given by following equation:

$$Y = X + Z, \quad Z \sim \mathcal{N}(0, N) \quad (6.1)$$

The noise Z is assumed to be independent of the signal X .

We can model the Flash write levels as M-ary pulse amplitude modulation (M-PAM). For TLC, we use 8-PAM constellations to store 3 bits per page. We will consider equally spaced equally likely (ESEL) 8-PAM as illustrated in the Table 6.1 as a baseline for comparison,

Table 6.1: 8-pam ESEL constellations and gray labeling for TLC flash

Voltage Levels	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7
ESEL	-7	-5	-3	-1	1	3	5	7
B_1	0	0	0	0	1	1	1	1
B_2	0	0	1	1	1	1	0	0
B_3	0	1	1	0	0	1	1	0

although practical Flash write levels are not equally spaced.

6.3 The Mutual Information Perspective

6.3.1 Mutual Information for Three Independent Pages

Let p_1 denote the probability of bit B_1 being equal to 0, i.e. $p_1 = P(B_1 = 0)$. Similarly, let p_2 and p_3 denote probability of B_2 and B_3 being equal to 0, respectively. Let the 8-PAM alphabet $\mathcal{X} = \{x_0, x_1, \dots, x_7\}$ and let $\mathcal{X}_B^{(0)}$ be a subset of alphabet $\mathcal{X}$ for which $B_j = 0$ and $\mathcal{X}_B^{(1)}$ be a subset of alphabet $\mathcal{X}$ for which $B_j = 1$, where $j = 1, 2, 3$. We use AWGN for channel model, i.e., $Z \sim \mathcal{N}(0, N)$ and i.i.d.

Mutual information rates $I(B_1; Y)$, $I(B_2; Y)$ and $I(B_3; Y)$ for independent encoding of pages are calculated as follows:

$$I(B_j; Y) = \int_y f(B_j = 0, y) \log_2 \left(\frac{f(B_j = 0, y)}{P(B_j = 0)f(y)} \right) dy \\ + \int_y f(B_j = 1, y) \log_2 \left(\frac{f(B_j = 1, y)}{P(B_j = 1)f(y)} \right) dy$$

$$\text{where } P(B_j = 0) = p_j, \quad P(B_j = 1) = 1 - p_j,$$

$$\text{and } j = 1, 2, 3.$$

We can expand the terms inside the integrals as follows:

$$\begin{aligned} f(B_j = 0, y) &= P(B_j = 0)f(y|B_j = 0) \\ &= p_j f(y|B_j = 0) \end{aligned} \quad (6.2)$$

$$\begin{aligned} f(B_j = 1, y) &= P(B_j = 1)f(y|B_j = 1) \\ &= (1 - p_j)f(y|B_j = 1) \end{aligned} \quad (6.3)$$

$$f(y|B_j = 0) = \sum_{x_i \in \mathcal{X}_{\mathcal{B}}^{(0)}} p(x_i) f(y|x_i) \quad (6.4)$$

$$f(y|B_j = 1) = \sum_{x_i \in \mathcal{X}_{\mathcal{B}}^{(1)}} p(x_i) f(y|x_i) \quad (6.5)$$

$$\begin{aligned} f(y) &= \sum_{x_i \in \mathcal{X}} p(x_i) f(y|x_i) \\ &= \sum_{x_i \in \mathcal{X}} p(x_i) \frac{1}{\sqrt{2\pi N}} \exp \frac{-(y - x_i)^2}{2N} \end{aligned} \quad (6.6)$$

Where $p(x_i) = P(X = x_i)$ or equivalently,

$$p(x_i) = P(B_1 = b_1(i))P(B_2 = b_2(i))P(B_3 = b_3(i)) \quad (6.7)$$

where i ranges from 0 to 7 and $b_1(i)$ is the most significant bit of the binary representation of i , i.e. the blue bit in Fig. 6.1. Similarly, $b_2(i)$ is the middle bit and $b_3(i)$ is the least significant bit.

The penalty for decoding the three pages independently (rather than jointly) is calculated as follows:

$$\begin{aligned} &I(B_1, B_2, B_3; Y) - I(B_1; Y) - I(B_2; Y) - I(B_3; Y) \\ &= I(B_1; B_2|Y) + I(B_1, B_2; B_3|Y) \geq 0 \end{aligned} \quad (6.8)$$

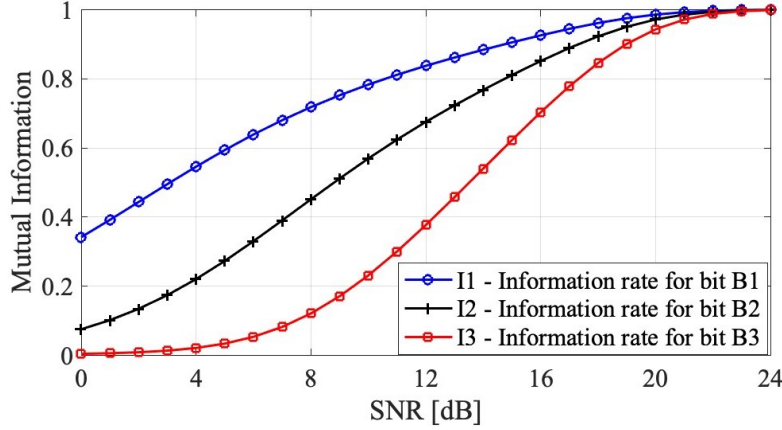


Figure 6.3: Independent MI rates for Equally Spaced Equally Likely (ESEL) Constellations as a function of SNR.

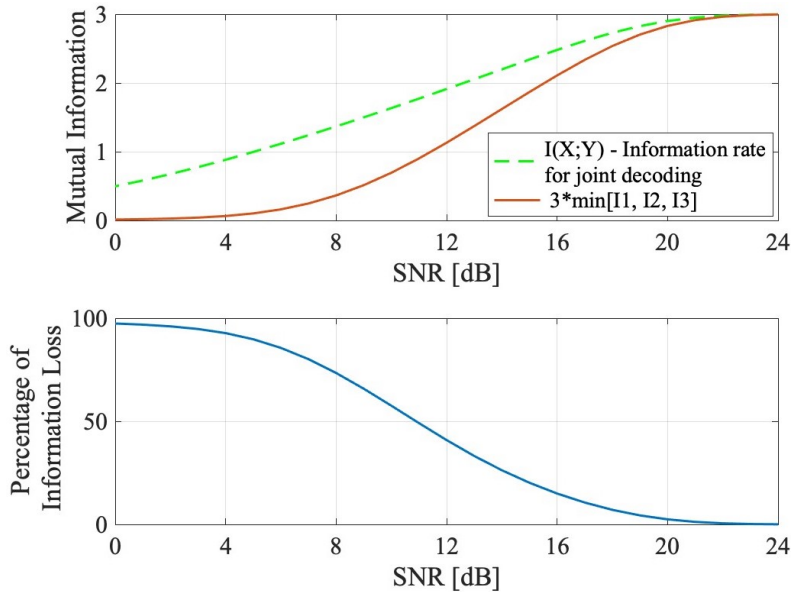


Figure 6.4: Upper subplot: Comparison of Information rate $I(X;Y) = I(B_1, B_2, B_3; Y)$ for joint decoding depicted by green curve and $3 \times \min\{I_1, I_2, I_3\}$ where $I_1 = I(B_1; Y)$, $I_2 = I(B_2; Y)$, $I_3 = I(B_3; Y)$ for Equally Spaced Equally Likely (ESEL) points. Lower subplot: Percentage of Mutual Information loss for ESEL points with equal rate constraint.

6.3.2 Joint Vs Independent Encoding for ESEL Constellations

For ease of notation let $I_1 = I(B_1; Y)$, $I_2 = I(B_2; Y)$ and $I_3 = I(B_3; Y)$. Plots of three independent rates I_1 , I_2 and I_3 as a function of signal-to-noise ratio (SNR) for ESEL constellations are given in Fig. 6.3. Fig. 6.4 illustrates the mutual information $I(X;Y) = I(B_1, B_2, B_3; Y)$

for the overall ESEL constellations and $3 \times \min_j (I_j)$ for $j = 1, 2, 3$ as a function of SNR. Observing the Fig. 6.4 we note that there would be an improvement if we could find a way to equalize the rates so that $3 \times \min_j (I_j)$ is closer to $I(X; Y)$.

6.3.3 Maximizing the minimum page mutual information

We formulate following optimization problem to maximize the minimum page mutual information to seek a solution for which all three page rates in TLC Flash memory are equal:

$$\begin{aligned}
& \max_{x \in \mathcal{X}, p_2, p_3} \min_j I(B_j; Y) \quad j = 1, 2, 3. \\
& \text{s.t.} \quad \sum_i p(x_i) x_i^2 = E_{\text{ESEL}}, \quad i = 0, \dots, M-1 \\
& \quad x_k = -x_{M-1-k}, \quad k = 0, 1, 2, \frac{M}{2} - 1 \\
& \quad p(x_i) = P(B_1 = b_1(i))P(B_2 = b_2(i))P(B_3 = b_3(i)) \\
& \quad \sum_i p(x_i) x_i = 0
\end{aligned}$$

With $M = 8$ for TLC Flash and $\mathcal{X} = \{x_0, x_1, \dots, x_7\}$ having the ESEL values shown in Table 6.1, E_{ESEL} , the average power for equally spaced equally likely constellation points given by:

$$E_{\text{ESEL}} = \frac{\sum_{i=0, \dots, M-1} x_i^2}{M} = 21 \quad (6.9)$$

We seek to design an alphabet $\mathcal{X}$ subject to symmetry and the power constraint $EX^2 \leq E_{\text{ESEL}}$ that maximizes the minimum MI between binary source B_j and a real valued output Y .

6.4 Maximizing the Minimum Rate

In this section we present two algorithms for solving optimization problem in Sec. 6.3.3. The first algorithm retains equally likely points but moves points to increase the minimum value

Algorithm 3 Dynamic Write-Level Position Assignment

Initialization:

$M \leftarrow 8$

$\epsilon = 10^{-4}$

$\mathcal{X}^{(1)} = \{x_1, x_2, \dots, x_{|\mathcal{X}^{(1)}|}\}$

▷ /* Initial ESEL points */

$E = \frac{1}{M} \sum_{x \in \mathcal{X}^{(1)}} x^2$

$p_1 = p_2 = p_3 = 0.5$

$i \leftarrow 1$

Iterations:

$I_j = \min(I_1, I_2, \dots, I_{\log_2 M})$

1) Optimize constellation locations $\mathcal{X}$:

a) Determine direction vector $\tilde{\mathcal{D}}^{(k)}$ to adjust $\mathcal{X}$

b) Compute

$$\mathcal{X}^{(k+1)} = \tilde{d} \left(\mathcal{X}^{(k)} + \lambda^* \tilde{\mathcal{D}}^{(k)} \right), \quad (6.10)$$

where $\tilde{d} = \text{diag}(d_1, \dots, d_M)$ ensures the power constraint, and the optimal movement factor λ^* is determined as

$\lambda^* = \arg \max_{\lambda} I_j \left(\tilde{d} \left(\mathcal{X}^{(k)} + \lambda \tilde{\mathcal{D}}^{(k)} \right), p_1^{(k)}, p_2^{(k)}, p_3^{(k)} \right)$ where $I_j(\mathcal{X}, \mathcal{P})$ is the mutual information $I(B_j; Y)$ calculated from input PMF characterized by $\mathcal{X}$ and p_1, p_2, p_3 . Note that in the maximization that yields λ^* , the resulting new values must satisfy $I_i(\mathcal{X}^{(k+1)}, \mathcal{P}) \geq I_j(\mathcal{X}^{(k)}, \mathcal{P})$ for all $i \neq j, 0 \leq i \leq 2$.

Stop If $(I_1 - I_j) \leq \epsilon, (I_2 - I_j) \leq \epsilon, \dots, (I_{\log_2 |\mathcal{X}^{(1)}|} - I_j) \leq \epsilon$

Or if maximum number of iterations is reached

$i \leftarrow i + 1$

of $I(B_j; Y)$. The second algorithm additionally adjusts the probability of the constellation points x_i under the constraint that the three bits that determine the constellation label are independent as described by (6.7), as required for independent encoding.

Alg. 3 enumerates the steps for maximizing the minimum rate through dynamic assignment of the write level positions. Alg. 4 enumerates the steps for maximizing the minimum rate by optimizing both positions and probabilities of the write levels.

6.4.1 Dynamic Assignment of Write Level Positions

Alg. 3 maximizes the minimum rate under the constraint that the constellation points are equally likely, i.e.,

$$P(B_j = 0) = P(B_j = 1) = \frac{1}{2} \quad (6.11)$$

for $j = 1, 2, 3$. The details of the steps in Alg. 3 are as follows:

Initialization: Select the modulation number $M = 8$ for 8-PAM TLC Flash memory. Tolerance ϵ is the maximum acceptable distance from any two independent rates $I(B_j; Y)$ for $j = 1, 2, 3$. Initialize constellation (point) locations to $\mathcal{X} = \{-7, -5, -3, -1, 1, 3, 5, 7\}$ and average power $E = 21$ for ESEL constellations. In Alg. 3 we will optimize only point locations and therefore we fix the input PMFs to be $p_1 = p_2 = p_3 = 0.5$.

Iterations: Adjust Write-level Positions $\mathcal{X}$: The process at each iteration is similar to Step 4 in the power-constrained AWGN dynamic-assignment Blahut-Arimmoto (DAB) algorithm introduced by Xiao in [124]. Each iteration begins by identifying the minimum MI $I(B_j; Y)$ and optimizing one symmetric pair of constellation points to increase the selected minimum MI $I(B_j; Y) = I_j(\mathcal{X}, p_1, p_2, p_3)$ calculated using MATLAB *integral*($\cdot$) function.

For each symmetric pair of constellation points we select the direction $\tilde{\mathcal{D}}^{(k)} = e_l - e_r$ with e_j being a j -th standard basis vector as described in [124]. Define l and r such that $r = |\mathcal{X}^{(k)}| - l + 1$. Thus, for the two symmetric points, l corresponds to the left location being moved and r corresponds to the right location being moved.

Once the direction is found we perform a line search routine (e.g. using `fminbnd` in MATLAB) to find the λ^* that maximizes minimum rate $I_j(\mathcal{X}, p_1, p_2, p_3)$ where $\mathcal{X}$ is a scaled version of $\mathcal{X}^{(k)} + \lambda \tilde{\mathcal{D}}^{(k)}$. The interval for line search constrained so that other two non-minimum rates stay above $I_j(\mathcal{X}, p_1, p_2, p_3)$.

To enforce the power constraint, the constellation needs to be scaled to increase power if

the two points move closer to the origin or to decrease power if the two points move further from the origin. To do this, we define a diagonal matrix $\tilde{d} = \text{diag}(d_1, \dots, d_M)$ to scale the remaining points to meet the power constraint as follows:

Let x_i be the i -th element of $\mathcal{X}^{(k)} + \lambda \tilde{\mathcal{D}}^{(k)}$. Since MATLAB is used as an optimization tool, we will consider the indices of constellation points to start at 1, i.e. $i = 1, \dots, 8$. Similar to above, define the symmetric pair of positions being moved to be l and r such that $r = |\mathcal{X}^{(k)}| - l + 1$.

If $l = 1$ or $l = \frac{M}{2}$ the following $\tilde{d}$ satisfies the power constraint:

$$d_i \in \tilde{d} \text{ and } d_i = 1 \text{ if } i \in \{l, r\}, \text{ o.w. } d_i = \alpha \quad (6.12)$$

such that $P_{\tilde{d}} = E_{\text{ESEL}}$, where

$$P_{\tilde{d}} = p(x_l)x_l^2 + p(x_r)x_r^2 + \sum_{i \notin [l, r]} p(x_i)\alpha^2 x_i^2. \quad (6.13)$$

Otherwise, only the outermost points are scaled as follows:

$$d_i \in \tilde{d} \text{ and } d_i = \alpha \text{ if } i \in \{1, M\}, \text{ o.w. } d_i = 1 \quad (6.14)$$

such that $P_{\tilde{d}} = E_{\text{ESEL}}$, where

$$P_{\tilde{d}} = p(x_1)\alpha^2 x_1^2 + p(x_M)\alpha^2 x_M^2 + \sum_{1 < i < M} p(x_i)x_i^2. \quad (6.15)$$

Note that in this section $p(x_i) = 1/8$ since we are only optimizing point locations and keep PMFs the same.

All three rates are calculated with updated alphabet consisting of optimized symmetric pair of points and new minimum rate is determined to continue iteration process. The algorithm stops if all three rates are within epsilon distance from each other or if we reach

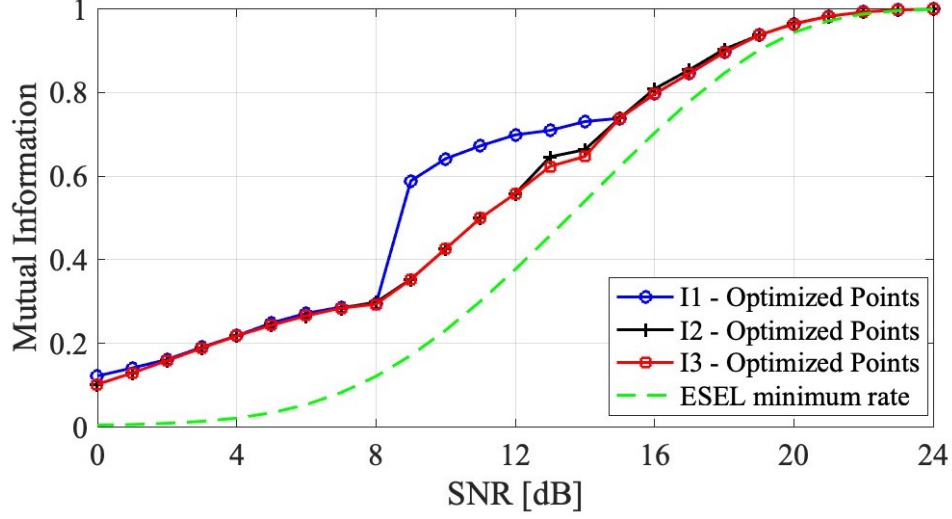


Figure 6.5: Three 8-PAM independent MI rates corresponding to bits B_1, B_2 and B_3 respectively for optimized points locations using Alg. 3. Green dotted curve is minimum Information rate for Equally Spaced Equally Likely Points

maximum number of iterations. For most SNR values maximum minimum rate is achieved for less than 50 iterations.

Fig. 6.5 shows optimized independent rates with respect to point locations as a function of SNR. For reference we also show the minimum rate for the ESEL constellation. The difference between the lowest of the optimized rates and the minimum rate for the ESEL constellation represents the an improvement in page rate for the actual system. Fig. 6.6 shows optimized point locations as a function of SNR.

Alg. 3 was able to achive essentially equal values for I_1, I_2 and I_3 for SNR values above 15 dB and below 8 dB. However, it was unable to achieve equal values in the region between 8 dB and 15 db. Seeking to improve on the performance Alg. 3, the next section introduces Alg. 4 which adds the capability to adapt the probability of each write level so that the constellation points are no longer equally likely.

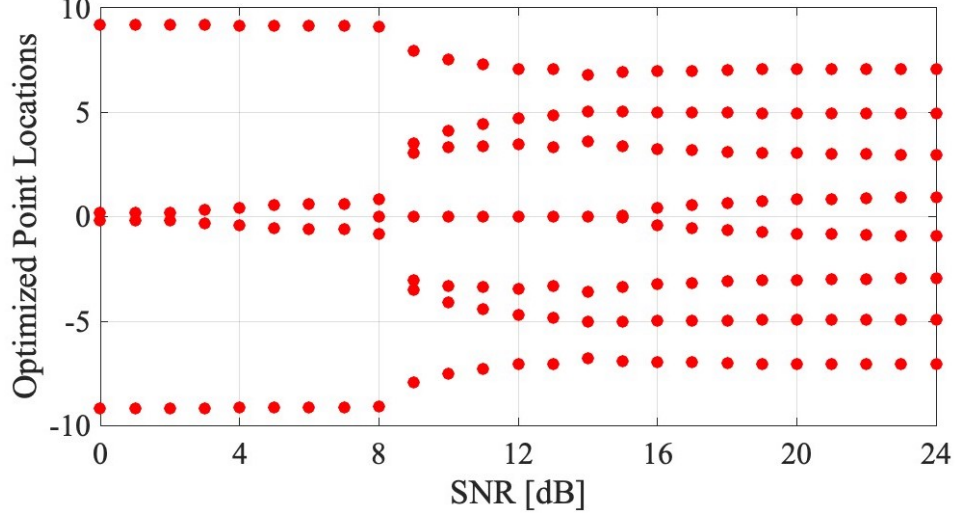


Figure 6.6: 8-PAM Optimized constellations locations as a function of SNR.

6.4.2 Optimizing Write-Level Positions and Probabilities

This section presents Alg. 4, which optimizes both the positions and the probabilities of the write levels. The constellation point probabilities must still be symmetric, which means that p_1 remains fixed at $\frac{1}{2}$. Functionally, the main difference between Alg. 4 and Alg. 3 is that Alg. 4 contains one additional step that optimizes the PMFs corresponding to bit channels B_2 and B_3 , which are defined by the parameters p_2 and p_3 . Gradient descent adjusts p_2 and p_3 while scaling the constellation points to maintain the power constraint.

Let I_j be the minimum rate. Since objective function is maximizing I_j , let $f = -I_j(\mathcal{X}, p_1, p_2, p_3)$ so that we change the problem into minimizing f with respect to p_2 and p_3 . With this notation, gradient descent method is as follows:

$$\Delta p := -\nabla f \quad (6.17)$$

$$p = p + \alpha \Delta p. \quad (6.18)$$

We used $\alpha = 0.01$ as our gradient step to optimize PMFs p_2 and p_3 . Referring back to Fig.6.1 and Table 6.1, note that the PMF of the write levels $P_X = p(x)$ can be directly computed

Algorithm 4 Optimizing Positions and Probabilities

Initialization:

$M \leftarrow 8$

$\alpha \leftarrow 0.01$

▷ /*gradient step*/

$\epsilon = 10^{-4}$

$\mathcal{X}^{(1)} = \{x_1, x_2, \dots, x_{|\mathcal{X}^{(1)}|}\}$

▷ /* Initial ESEL points */

$E = \frac{1}{M} \sum_{x \in \mathcal{X}^{(1)}} x^2$

$p_1 = p_2 = p_3 = 0.5$

$i \leftarrow 1$

Iterations:

1) Optimize constellation locations $\mathcal{X}$:

for index = $1:\frac{M}{2}$ **do**

$I_j = \min(I_1, I_2, \dots, I_{\log_2 M})$

a) Determine direction vector $\tilde{\mathcal{D}}^{(k)}$ to adjust $\mathcal{X}$

b) Compute

$$\mathcal{X}^{(k+1)} = \tilde{d} \left(\mathcal{X}^{(k)} + \lambda^* \tilde{\mathcal{D}}^{(k)} \right), \quad (6.16)$$

where $\tilde{d} = \text{diag}(d_1, \dots, d_M)$ ensures the power constraint, and the optimal movement factor λ^* is determined as

$\lambda^* = \arg \max_{\lambda} I_j \left(\tilde{d} \left(\mathcal{X}^{(k)} + \lambda^* \tilde{\mathcal{D}}^{(k)} \right), p_1^{(k)}, p_2^{(k)}, p_3^{(k)} \right)$ where $I_j(\mathcal{X}, \mathcal{P})$ is the mutual information $I(B_j; Y)$ calculated from input PMF characterized by $\mathcal{X}$ and p_1, p_2, p_3 . Note that in the maximization that yields λ^* , the resulting new values must satisfy $I_i(\mathcal{X}^{(k+1)}, \mathcal{P}) \geq I_j(\mathcal{X}^{(k)}, \mathcal{P})$ for all $i \neq j$, $0 \leq i \leq 2$.

end for

$I = \min(I_1, I_2, \dots, I_{\log_2 M})$

2) Optimize PMFs p_2 and p_3 :

Use gradient descent to optimize p_2 and p_3 while

satisfying power constraint by scaling constellations.

Stop If $(I_1 - I_2) \leq \epsilon$ && ... && $(I_1 - I_{\log_2 |\mathcal{X}^{(1)}|}) \leq \epsilon$

Or if maximum number of iterations is reached

$i \leftarrow i + 1$

from p_1, p_2 and p_3 as follows:

$$P_X = [p_1 p_2 p_3, p_1 p_2 (1 - p_3), p_1 (1 - p_2) (1 - p_3), p_1 (1 - p_2) p_3, \\ (1 - p_1) (1 - p_2) p_3, (1 - p_1) (1 - p_2) (1 - p_3), (1 - p_1) p_2 (1 - p_3), (1 - p_1) p_2 p_3] \quad (6.19)$$

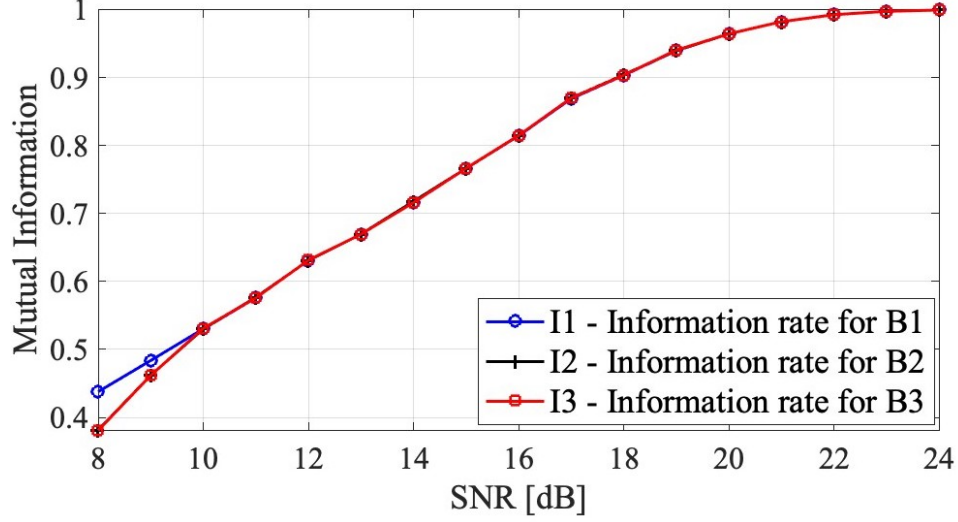


Figure 6.7: 8-PAM Independent rates for optimized constellation positions and input PMFs.

The gradient descent is complicated by the need to maintain the power constraint. Let Q_X be the previous constellation PMF used in step 1) of Alg. 4. We perform two separate steps in the power constrained gradient descent to separately adjust p_2 and p_3 as follows:

Step 1: Adjust p_2 by using (6.17) and (6.18) with $p = p_2$. Use the new p_2 to compute the new P_X according to (6.19). Scale the constellation points to satisfy power constraint as follows:

$$a_i = \frac{Q_X(i)}{P_X(i)}, i = 1, \dots, M. \quad (6.20)$$

$$x_i^{(k+1)} = \sqrt{a_i} x_i^{(k)} \quad (6.21)$$

After a gradient step in direction of p_2 is performed and points are scaled recalculate all three rates and check if they are all greater or equal to minimum rate. If this condition is not satisfied adopt initial p_2 and constellation points and move to Step 3. Otherwise, keep the solution and move to Step 3.

Step 2: Adjust p_3 . This step is identical to Step 1 except applied to p_3 instead of p_2 . We fix p_2 obtained from Step 2) and perform the gradient with respect to p_3 while scaling

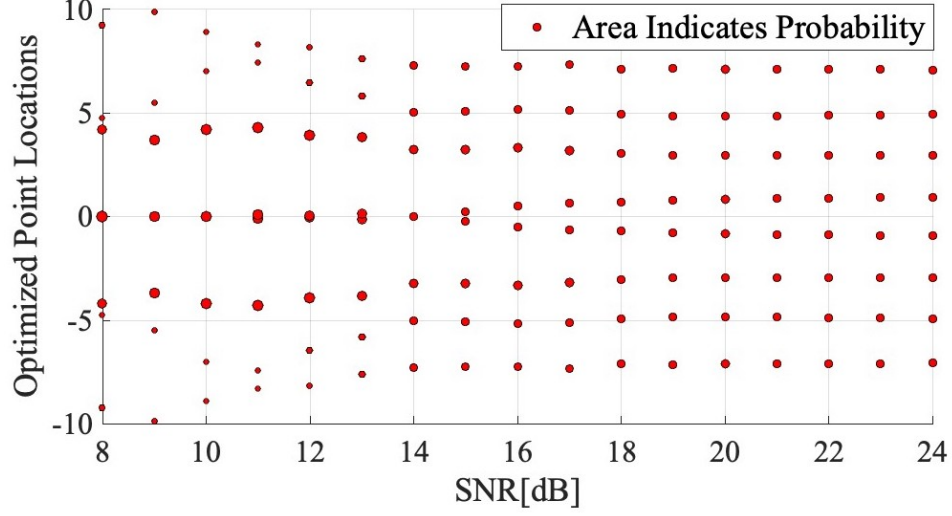


Figure 6.8: 8-PAM Optimized constellations locations using Alg. 4 as a function of SNR.

points to maintain the power constraint. Step 1 and Step 2 are repeated until there is no more improvement in maximizing the minimum rate.

Fig. 6.7 shows the independent rates achieved by the optimized point locations and input PMFs obtained using Alg. 4. Fig. 6.8 shows optimized point positions and input PMFs resulting from using Alg. 4 to achieve independent rates in Fig. 6.7. In Fig. 6.9, the joint mutual information resulting from optimized points and PMFs in Alg. 4 is compared to MI of DAB optimized input PMFs given in [124] as well as to 3 times minimum information rate and sum of independent rates for optimized points and PMFs. Fig. 6.10 shows three losses for flash system:

1. Shaping loss: $I(X; Y)_{\text{DAB}} - I(B_1, B_2, B_3; Y)$

2. Independent encoding loss:

$$I(B_1, B_2, B_3; Y) - I(B_1 : Y) - I(B_2; Y) - I(B_3; Y)$$

3. Equal rate constraint loss:

$$I(B_1; Y) + I(B_2; Y) + I(B_3; Y) - 3 * \min_j \{I(B_j : Y)\}$$

The results in Fig. 6.10 show that equal rate constraint loss is negligible for nearly all

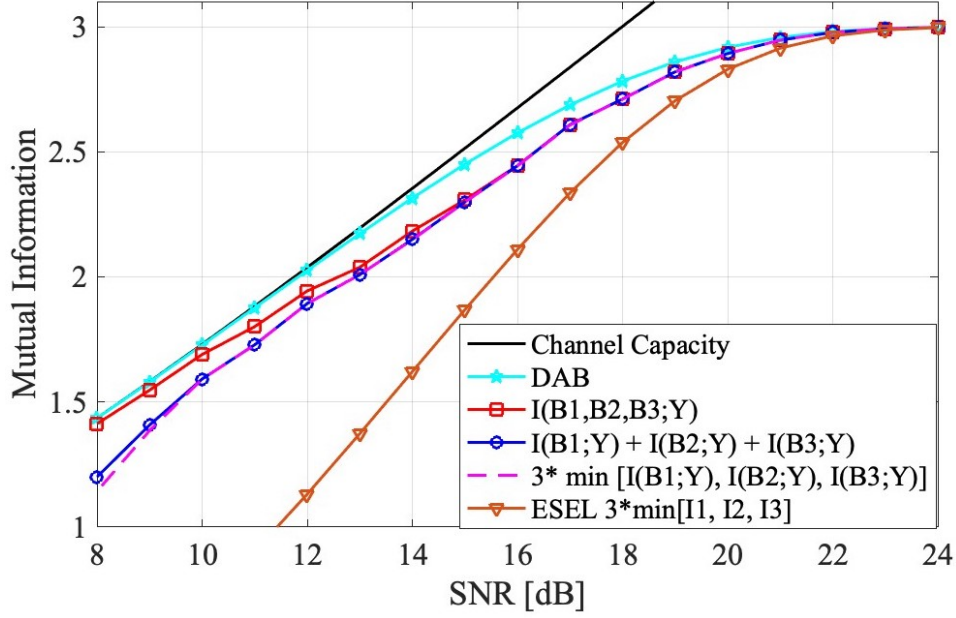


Figure 6.9: Joint mutual information rate $I(B_1, B_2, B_3; Y)$ for optimized points and PMFs is compared to the sum of independent rates, 3 times minimum independent rate and joint MI for DAB optimized input PMFs. Equal rate constraint loss is nearly zero for all SNRs. Independent encoding loss is negligible for $SNR = 15$ dB and higher. The most evident loss comes from shaping.

SNRs. Independent encoding loss is higher for very low SNRs but zero after 15 dB. Shaping loss is evident mostly from 10 to 22 dB with highest value slightly less than 0.15 bits. Shaping loss could be improved by explicitly adding a step to improve the joint mutual information.

6.5 Conclusion

This chapter focuses on the practical scenario of independent encoding of Flash under the constraint that all pages have to store the same amount of information. By optimizing the positions of the write levels in the constellation and input probabilities to maximize a mutual information rate of the bit channel with worst information rate, we were able to significantly improve the minimum page rate. The independent encoding loss and equal rate constraint loss are negligible (zero) for operational SNRs. Shaping loss is the most significant loss in an interesting range between 10 and 22 dB, with the largest loss approaching 0.15 bits. This loss

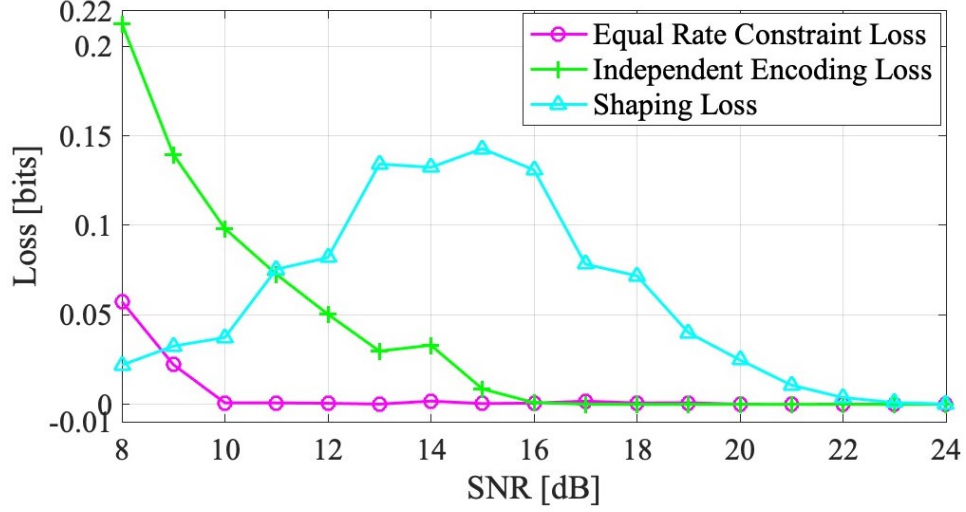


Figure 6.10: Equal Rate Constraint Loss:

$I(B1 : Y) + I(B2; Y) + I(B3; Y) - 3 * \min[I(B1 : Y), I(B2; Y), I(B3; Y)]$, where $I(B1 : Y), I(B2; Y), I(B3; Y)$ are information rates for optimized constellations and input PMFs. Independent encoding loss: $I(B1, B2, B3; Y) - (I(B1 : Y) + I(B2; Y) + I(B3; Y))$ given in Eqn. (9). Shaping Loss: Difference between DAB Mutual Information Rate $I(X; Y)$ given in [124] and $I(B1, B2, B3; Y)$ for optimized points and PMFs.

may be reduced by an additional optimization step, which is the subject of future research.

We note that 8-PAM constellations with points that are not equally likely can be supported by coded modulation techniques such as probabilistic amplitude shaping [13, 18, 27, 119].

Acknowledgment

We thank Derek Xiao, an author of paper referenced in [124], for providing DAB algorithm MATLAB code and DAB optimized PMFs and points needed for DAB rate in Fig. 6.9.

Chapter 7

Joint Optimization of Write and Read Thresholds for MLC Flash Memory

7.1 Introduction

7.1.1 Background

As discussed in Chapter 6, Flash memory is widely utilized data storage technologies in modern electronic devices.¹ Techniques that prolong reliable Flash memory performance are therefore worthy of study in response to rapidly increasing data storage demands.

Programming requires applying charge to the Flash cell to set the voltage level, or write voltage, at which current will flow. Each Multi-Level Cell (MLC) has four possible write voltages, one for each two-bit input. Each bit maps to a different page in the Flash cell [19]. Data is read through the application of various voltages or read thresholds on the gate to sense the presence or absence of current [12].

One iteration of programming, reading and erasing data is referred to as a Program and Erase Cycle or P/E Cycle. As the number of P/E Cycles of a device increases, the device

¹This chapter is largely a reprint of work published in the IEEE International Symposium on Information Theory [8]

becomes less reliable until the frame error rate (FER) becomes too high, indicating the end of its useful lifetime [79]. Sources of distortion in the Flash read channel include programming noise, cell-to-cell interference, and charge leakage induced by device wear-out [116]. Therefore, this research strives to find proper write voltage levels and read thresholds that minimize degradation of the device, subject to noise of the cell, while still allowing for successful memory storage.

For a specified LDPC code rate such as 8/9, when the mutual information (MI) between the write voltage and the bits read from the Flash read channel falls below the LDPC code rate, page failure is unavoidable. Wong et al. [122] optimizes initial and enhanced precision read thresholds by maximizing the MI between the input and output of the Flash read channel. However, in [122], the most-significant bit (MSB) page fails about 250 P/E cycles before the least-significant bit (LSB) page for the same precision level. Wong et al. also does not explore optimizing write voltage levels.

Researchers have previously explored the benefits of optimizing initial read [126] or write [39] voltage levels using the techniques of maximizing MI as introduced in [117]. [38] proposes a search algorithm of write level optimization to minimize error probability in multilevel coding and bit-interleaved coded modulation (BICM). In [74], voltage level optimization is performed to accomplish minimizing the overall bit error rate (BER) and equalizing page BERs. Both [38] and [74] consider hard decoding reads without analyzing the benefits of optimizing enhanced precision thresholds, write voltages and hard read thresholds simultaneously. Optimization of both read and write voltage levels is explored in [6] and [7]. In [6] write voltages are optimized such that the overall probability of error is minimized, while read thresholds are optimized for maximizing total MI in the same manner as [117], but with a different Flash channel model. In both papers optimization techniques are not compatible with practical Flash memory coding structure where each page is encoded independently.

This chapter uses MI maximization to jointly optimize write voltages and read thresholds in the context of practical Flash memories, which use equally-sized pages, encode each bit

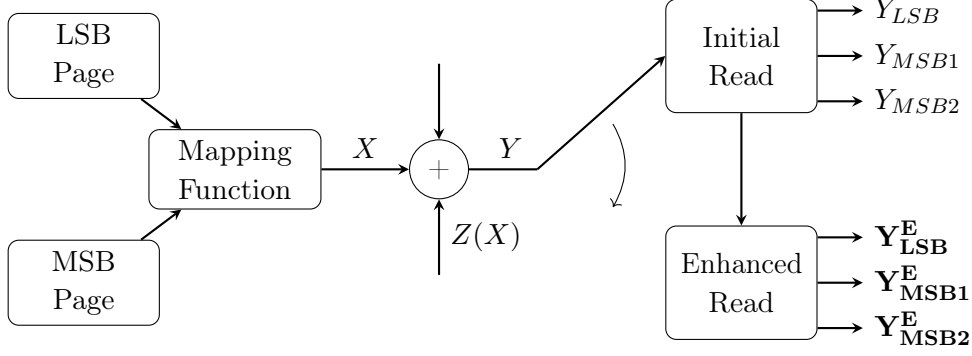


Figure 7.1: Flash system model with initial and enhanced precision reads.

of a cell into a distinct page, and use progressive reads with enhanced precision to decode pages when the initial read fails. To maximize the Flash lifetime, we seek to equalize the performance of the pages associated with different bits as in, e.g., [38, 74]. Since data is written before the controller knows whether enhanced precision is needed, the same write voltages must be used for both initial reads and the progressive reads. Therefore, a choice must be made between selecting the write voltages that optimize the performance of the initial read or the enhanced reads.

Thus, we consider jointly optimizing write voltages and read thresholds to maximize three distinct MIs. One approach is to select the write voltages and read thresholds to maximize minimum page MI between the page bit and the initial reads. This is the scenario of [38, 74], although those papers minimize bit error rate rather than maximize MI. Our second approach jointly optimizes the write voltage levels, initial read thresholds, and enhanced precision read thresholds to maximize minimum page MI between the page bit and all receiver information available after the enhanced precision read. Our third approach optimizes write voltages for the limiting case of enhanced precision where full “soft” information is available, i.e. controller learns the exact voltage where current begins to flow, as in [56] and Chapter 6.

7.1.2 Contributions

The main contributions of this chapter are as follows:

- This chapter compares three approaches to optimizing write voltage levels, maximizing the minimum of a specific MI 1) the MI for the initial read or 2) the MI after read with enhanced precision or 3) the soft MI. The minimum MI for the initial read provided the best overall performance for the channel studied.
- Among these three choices, information theoretic analysis and LDPC simulations show that maximizing the minimum MI for the initial read seems to be preferable for increasing the lifetime in P/E cycles until the LDPC frame error rate (FER) exceeds 10^{-6} .

7.1.3 Organization

The remainder of this chapter is organized as follows. Sec. II discusses the model of the Flash read channel. Sec. III summarizes the optimization of read thresholds for the initial read and for enhanced-precision to maximize the MI between the channel input and the specific outputs of that read operation. Sec. IV describes the joint optimization of write voltages and read thresholds to maximize one of three distinct MIs. Sec. V compares the three proposed approaches of Sec. IV to the write voltages and read thresholds used by [122]. Sec. VI presents the FER performance of LDPC simulations using the original write voltages of [122] and write voltages of each optimization technique. Sec. VII concludes the chapter.

7.2 Flash Channel Model

Following [122], Fig. 7.1 shows the system model and read channel where the bits of the LSB and MSB page induce the written voltage X , which is distorted by signal-dependent Gaussian noise $Z(X)$ to produce Y . As in [122], $Z(X)$ is approximated by a Gaussian distribution with two possible variances. Following the general behavior of Flash cells, the lowest value of X corresponding to an un-programmed cell has the larger variance and the

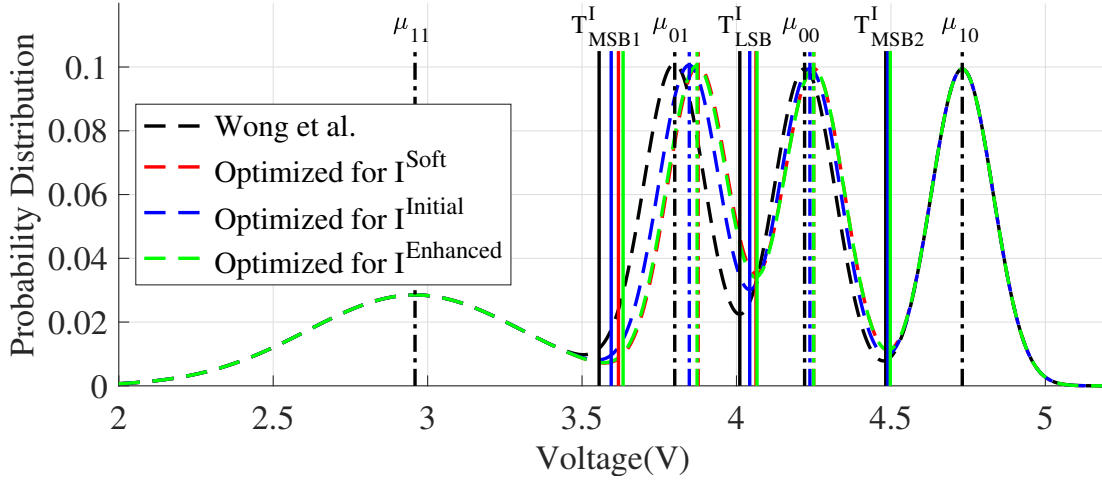


Figure 7.2: Flash model at 1600 P/E Cycles. Enhanced precision read thresholds would be placed to the left and right of each initial read threshold T_{MSB1}^I , T_{LSB}^I and T_{MSB2}^I . Solid lines represent initial read thresholds and dotted-dashed lines represent write voltages μ_{11} , μ_{01} , μ_{00} and μ_{10} .

noise for all of the other cells has the smaller variance because the feedback loop associated with programming reduces certain distortions. Note that the mean of $Z(X)$ is not X because there are distortions such as retention loss and read disturb that shift the mean in addition to distortions that add variance [19].

As shown in Fig. 7.1, the initial read for the LSB page produces the bit Y_{LSB} . For the MSB page, the initial read produces two bits Y_{MSB1} and Y_{MSB2} . Y_{LSB} , Y_{MSB1} and Y_{MSB2} represents whether the output voltage Y is above or below the initial read thresholds T_{LSB} , T_{MSB1} and T_{MSB2} , respectively. If the LDPC decoder fails to find a valid codeword with the initial read, an enhanced precision read provides additional bits. The enhanced read produces the variables $\mathbf{Y}_{LSB}^E$, $\mathbf{Y}_{MSB1}^E$ and $\mathbf{Y}_{MSB2}^E$ which are each vectors of two variables representing whether the real number output voltage Y is above or below the enhanced precision read thresholds placed to the left and right of each initial read threshold.

Fig. 7.2 shows an example of the two-variance Gaussian model for an MLC at 1600 P/E cycles. The means of the middle two means or write voltages and initial read thresholds are

placed four ways. One way is according to [122], which is considered as a baseline. The three new placements are according to the three optimization objectives described later in this chapter. The means and variances used in [122] for each multiple of 100 P/E cycles can be found in [115]. As the number of P/E cycles increase, retention losses causes the spacing between the input means to decrease and the variance of the programmed levels to increase.

7.3 Read Threshold Optimization

Wong et al. [122] optimized initial read thresholds T_{LSB}^I and $\mathbf{T}_{MSB}^I$ (a vector containing T_{MSB1}^I and T_{MSB2}^I) as a function of the means and variances of the two-variance Gaussian channel, according to the following two optimizations:

$$\max_{\{T_{LSB}^I\}} I_{LSB}^I \quad \text{and} \quad \max_{\{\mathbf{T}_{MSB}^I\}} I_{MSB}^I, \quad \text{with} \quad (7.1)$$

$I_{LSB}^I = I(X_{LSB}; Y_{LSB}^I)$ and $I_{MSB}^I = I(X_{MSB}; \mathbf{Y}_{MSB}^I)$, where X_{LSB} and X_{MSB} are the two input bits, one for the LSB page and one for the MSB page using this MLC Flash cell. $\mathbf{Y}_{MSB}^I$ is a vector containing Y_{MSB1}^I and Y_{MSB2}^I . As observed in [122], since T_{LSB}^I affects only Y_{LSB}^I and not $\mathbf{Y}_{MSB}^I$ and conversely $\mathbf{T}_{MSB}^I$ affects only $\mathbf{Y}_{MSB}^I$ and not Y_{LSB}^I , maximizing I_{LSB}^I and I_{MSB}^I are independent optimizations when the write voltages are assumed to be fixed and not included in the optimization.

Enhanced precision read thresholds $\mathbf{T}_{LSB}^E$ (a vector containing T_{LSB}^{left} and T_{LSB}^{right}) and similarly defined vectors $\mathbf{T}_{MSB1}^E$ and $\mathbf{T}_{MSB2}^E$ are found by solving the following two optimization problems

$$\max_{\{\mathbf{T}_{LSB}^E\}} I_{LSB}^E \quad \text{and} \quad \max_{\{\mathbf{T}_{MSB1}^E, \mathbf{T}_{MSB2}^E\}} I_{MSB}^E, \quad \text{with} \quad (7.2)$$

$I_{LSB}^E = I(X_{LSB}; Y_{LSB}^I, \mathbf{Y}_{LSB}^E)$ and $I_{MSB}^E = I(X_{MSB}; \mathbf{Y}_{MSB}^I, \mathbf{Y}_{MSB1}^E, \mathbf{Y}_{MSB2}^E)$. $\mathbf{Y}_{LSB}^E$ is a vector containing Y_{LSB}^{left} and Y_{LSB}^{right} . $\mathbf{Y}_{MSB1}^E$ and $\mathbf{Y}_{MSB2}^E$ are similarly defined vectors. As with

the initial reads, the I_{LSB}^E and I_{MSB}^E calculations for the enhanced reads are independent optimizations when the write voltages are assumed to be fixed and not included in the optimization. In [122], enhanced precision read thresholds are chosen after initial read thresholds are found according to Equation 7.1.

7.4 Write Voltage Levels that Optimize Page MI

For MLC Flash, there are four voltage levels, which we will refer to according to the two-bit input label that is associated with each level. Looking again at Fig. 7.2, the write voltages from left to right are μ_{11} , μ_{01} , μ_{00} , and μ_{10} .

In this chapter, the write voltages are assumed to completely determine the means of the Gaussian distributions observed during the read operation. In practice charge retention issues and read-disturb in real Flash memory systems cause the means of the read distributions to vary from the voltages written to the device [19]. Our optimization methods can easily incorporate a known relationship between the write voltage and the mean of the read distribution.

This section describes three techniques, with each technique maximizing the minimum of a specific page MI. We assume that the erased state mean μ_{11} cannot be changed by the controller. Similarly, we assume that the rightmost mean μ_{10} is already maximized and would not be further adjusted by the controller. Thus, our optimization only modifies the middle two write voltage levels, μ_{01} and μ_{00} .

All three optimization objectives explored in this chapter are solved in MATLAB using `fminsearchcon` function [29] which utilizes the Nelder-Mead Simplex Method to perform a direct search over multidimensional space. When using the `fminsearchcon` function to maximize the minimum of I^{Soft} , the write voltages are initialized to those found in [115].

7.4.1 Maximizing Minimum of $I^{Initial}$ for LSB and MSB

The first technique jointly optimizes the write voltage levels μ_{01} , μ_{00} and all initial read thresholds to maximize the minimum page MI for the initial read. Since the write voltage levels are deterministic of both I_{LSB}^I and I_{MSB}^I a joint optimization is required as follows:

$$\max_{\{T_{LSB}^I, \mathbf{T}_{MSB}^I, \mu_{01}, \mu_{00}\}} \min(I_{LSB}^I, I_{MSB}^I), \quad (7.3)$$

I_{LSB}^I and I_{MSB}^I MIs are $I^{Initial}$. The optimization is initialized with the write voltage levels and initial read thresholds found to optimize I^{Soft} described below. Enhanced precision read thresholds for this optimization objective are found by applying the optimized write voltages and initial read thresholds to Eqn. 7.2.

7.4.2 Maximizing Minimum of $I^{Enhanced}$ for LSB and MSB

The second technique jointly optimizes the write voltage levels μ_{01} , μ_{00} , the initial read thresholds and the enhanced precision read thresholds to maximize the minimum page MI for the enhanced read. Since the write voltage levels are deterministic of both I_{LSB}^E and I_{MSB}^E , a joint optimization is required as follows:

$$\max_{\{T_{LSB}^I, \mathbf{T}_{MSB}^I, \mathbf{T}_{LSB}^E, \mathbf{T}_{MSB1}^E, \mathbf{T}_{MSB2}^E, \mu_{01}, \mu_{00}\}} \min(I_{LSB}^E, I_{MSB}^E), \quad (7.4)$$

I_{LSB}^E and I_{MSB}^E are considered $I^{Enhanced}$. The optimization is initialized with the write voltage levels, initial and enhanced precision read thresholds found to optimize I^{Soft} described below.

7.4.3 Maximizing Minimum of I^{Soft} for LSB and MSB pages

The third technique finds write voltages that maximize the page MIs $I(X_{MSB}; Y)$ and $I(X_{LSB}; Y)$ where Y is the real number output voltage of the Flash read channel. These MIs are considered I^{Soft} because they utilize the soft output Y . Write voltages are deterministic

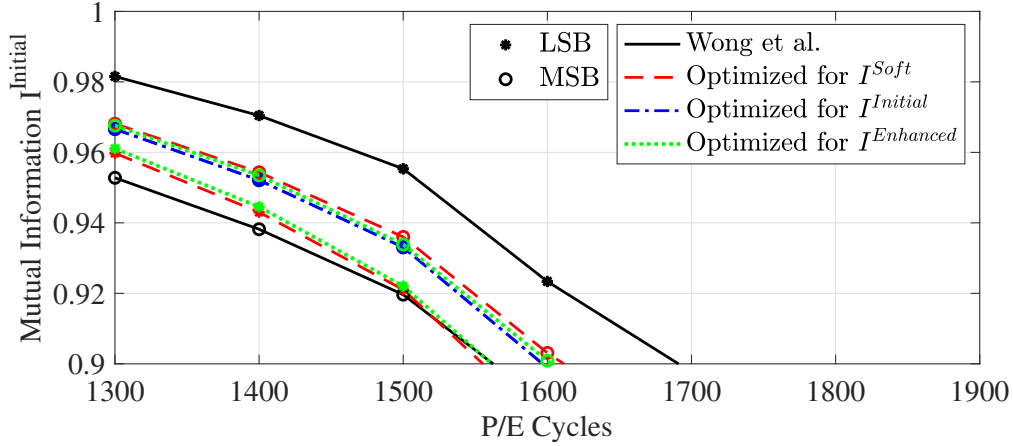


Figure 7.3: Plot of $I^{Initial}$ vs P/E Cycle for the MSB and LSB for each optimization technique and original mean voltages and initial read thresholds from Wong et al.

of both $I(X_{MSB}; Y)$ and $I(X_{LSB}; Y)$, so they must be jointly optimized. When performing the optimization to find the minimum page MI of I^{Soft} , the write voltages are initialized to those found in [115]. Initial read thresholds and enhanced precision read thresholds for this optimization objective are found by applying these write voltages to Eqn. 7.1 and Eqn. 7.2, respectively. Methods outlined in methods [116] and [107] allow the realization of the variances and means of the erase state and highest voltage input of the Flash device every 100 P/E cycles. Optimized thresholds and write voltages found using methods outlined in this chapter can then be pulled from a lookup table formulated offline. Learning the parameters of the Flash channel has been researched extensively to allow novel optimization techniques, like the one presented in this research, to extend the lifetime of Flash memory devices.

7.5 Page MI vs. P/E Cycles for Three Objectives

7.5.1 Comparing Performance for $I^{Initial}$ vs. P/E Cycles

$I^{Initial}$ is the MI between the output found from the initial reads and the LSB or MSB page input. Fig. 7.3 shows plots of $I^{Initial}$ for each page calculated with the voltage levels and

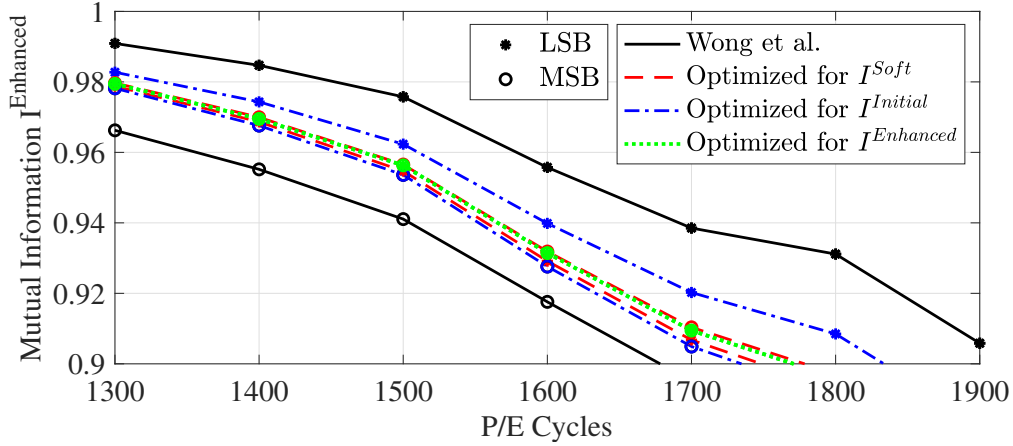


Figure 7.4: Plot of $I^{Enhanced}$ vs P/E Cycle for the MSB and LSB for each optimization technique and original mean voltages and initial read thresholds from Wong et al.

initial read thresholds from [122] and each of the three optimization objectives.

For the mean voltages optimized to maximize $I^{Initial}$, the MSB and LSB page fail (have an MI below 0.9) simultaneously at 1590 P/E cycles. Thus, mean voltages optimized for $I^{Initial}$ increase the initial-read lifetime of the device by around 40 P/E Cycles compared to values from [122]. Operating the Flash device with mean voltages and initial read thresholds optimized for $I^{Enhanced}$ leads to a similar initial-read lifetime as using values from [122]. Mean voltages optimized for I^{Soft} decreases the initial-read lifetime of the device by around 7 P/E cycles.

7.5.2 Comparing Performance for $I^{Enhanced}$ vs. P/E Cycles

$I^{Enhanced}$ measures the MI between the output bits produced by the initial and enhanced reads and the LSB or MSB of the input. Fig. 7.4 shows the values of $I^{Enhanced}$ for each page calculated using the write voltages, initial read thresholds and enhanced precision thresholds from [122] and each of the three optimization objectives.

The mean voltages optimized for $I^{Enhanced}$ the MSB and LSB page fail simultaneously at about 1767 P/E cycles, increasing the Flash memory's enhanced-read lifetime by around 90 P/E cycles compared to the original mean voltages and initial read thresholds from Wong et

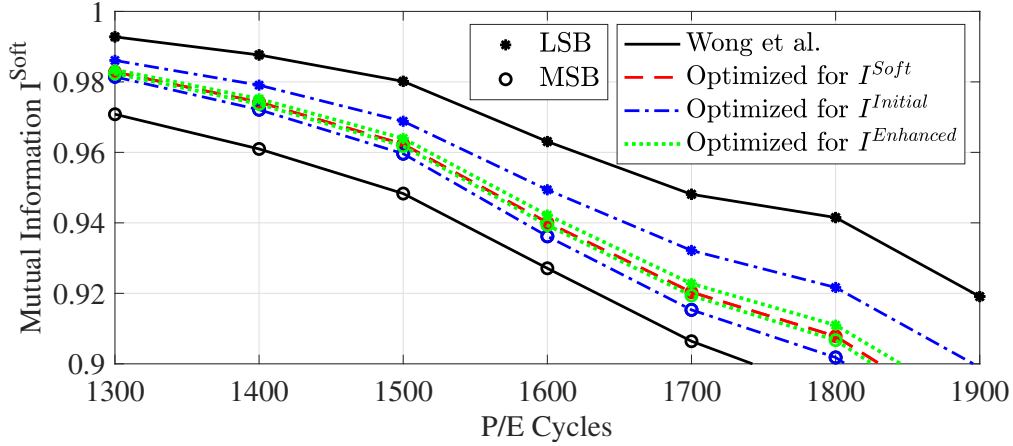


Figure 7.5: Plot of I^{Soft} vs P/E Cycles for the MSB and LSB for each optimization technique and original mean voltages and initial read thresholds from Wong et al.

al. The mean voltages optimized for $I^{Initial}$ and I^{Soft} have slightly worse initial-read lifetimes than the mean voltages optimized for $I^{Enhanced}$, but both increase the Flash memory's enhanced-read lifetime significantly as compared to the original mean voltages from Wong et al.

7.5.3 Comparing Performance for I^{Soft} vs. P/E Cycles

I^{Soft} calculates the amount of information the real number output voltage reveals about the binary input LSB or MSB. Fig. 7.5 shows I^{Soft} of each page calculated using the write voltages from [122] and each of the three optimization objectives.

Write voltage levels optimized for a real number output lengthen Flash memory lifetime by approximately 90 P/E Cycles compared to the lifetime with the original means of [122] and provides the greatest lifetime change of all three optimization methods. However, practical Flash devices do not have access to the real valued voltage Y . Instead, they must decode the page data using only a few bits of information about Y obtained by comparing Y with various read thresholds.

Fig. 7.6 shows the values $I^{Initial}$, $I^{Enhanced}$, and I^{Soft} at 1600 P/E cycles for all four ways of selecting the mean voltages. The lifetime of the flash device is determined by the

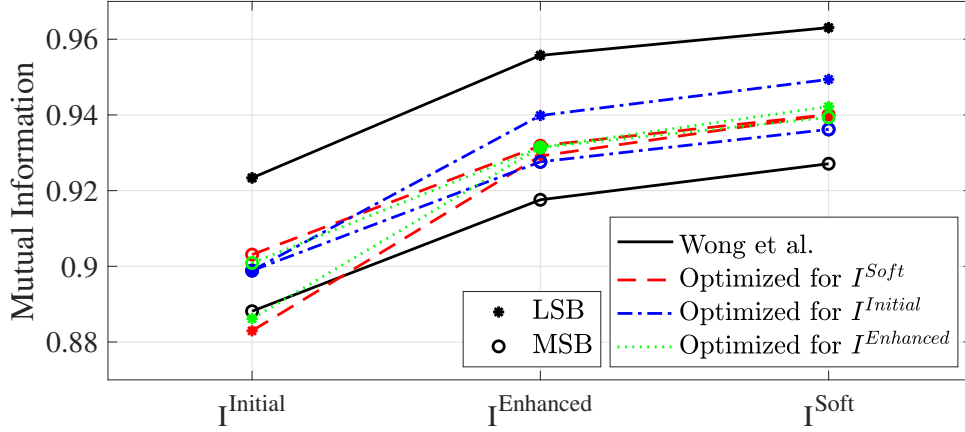


Figure 7.6: Comparison of three MI measures for all sets of means and thresholds at 1600 P/E Cycles

page with the shortest lifetime. While the MSB page lifetime decreases for all optimization techniques as compared to [122], the LSB page lifetime is increased so the lifetime of the device with optimized write voltages is increased. For each of the three possible MIs $I^{Initial}$, $I^{Enhanced}$, and I^{Soft} , the mean voltages optimized to maximize the minimum of that MI result in the MSB and LSB page MIs of that type being essentially identical. The mean voltages optimized for other MIs have a lower minimum page MI. Figs. 7.3, 7.4, 7.5, and 7.6 reveal that parameters optimized to maximize one MI do not generate optimal results for other MIs.

Write voltages are selected before knowing whether the initial read or the enhanced read will be required for LDPC decoding. One optimization approach must be picked. Since I^{Soft} is not practically relevant, let's focus attention on $I^{Initial}$ and $I^{Enhanced}$. Looking at $I^{Initial}$ in Fig. 7.6, the performance of the means optimized for $I^{Enhanced}$ is significantly worse than the performance of the means optimized for $I^{Initial}$. In contrast, looking at $I^{Enhanced}$ in Fig. 7.6, the means optimized for $I^{Initial}$ perform only slightly worse than the means optimized for $I^{Enhanced}$. Thus, for the channel studied, the MI results indicate that the best overall performance will be obtained by using $I^{Initial}$ to optimize the write voltages. This will give the absolute best performance on the initial read and the performance for the enhanced read will be almost as good as having optimized specifically for that case.

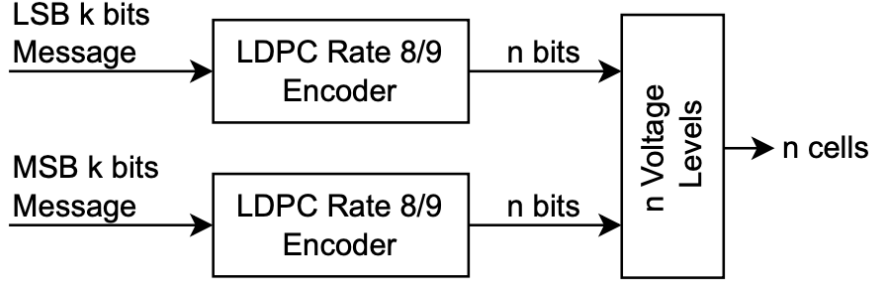


Figure 7.7: Block diagram of independent encoding of MSB and LSB pages.

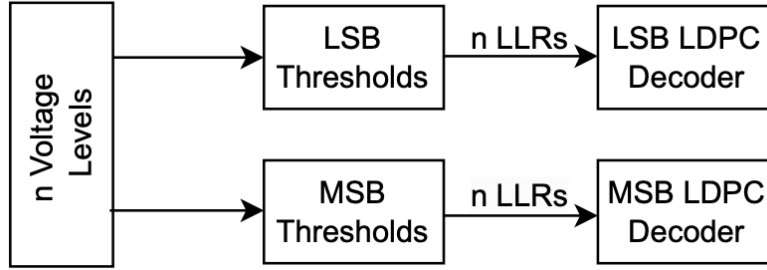


Figure 7.8: Block diagram of independent decoding of MSB and LSB pages.

7.6 Performance with LDPC codes

7.6.1 Independent Encoding and Decoding Procedure

A protograph-based low-density parity check (LDPC) code [95] of rate 8/9 encoding 14400 message bits into 16200 codeword bits is used to simulate frame error rates (FERs) for initial and enhanced read decoding. Fig. 7.7 illustrates that two independent k -bit messages, each for the LSB and MSB pages, are encoded into two separate codewords of n bits, which are then combined and mapped to n voltage levels. The decoder in Fig. 7.8 uses the n voltage levels to determine the log-likelihood ratio (LLR) for each page independently in accordance with the thresholds. For initial read decoding, Y_{LSB}^I will produce two LLR regions for the LSB page, while $\mathbf{Y}_{MSB}^I$ produce three LLR regions for the MSB page. For enhanced-read decoding, Y_{LSB}^I and $\mathbf{Y}_{LSB}^E$ will produce four LLR regions for the LSB page, while $\mathbf{Y}_{MSB}^I$, $\mathbf{Y}_{MSB2}^E$ and $\mathbf{Y}_{MSB1}^E$ will result in seven LLR regions for the MSB page.

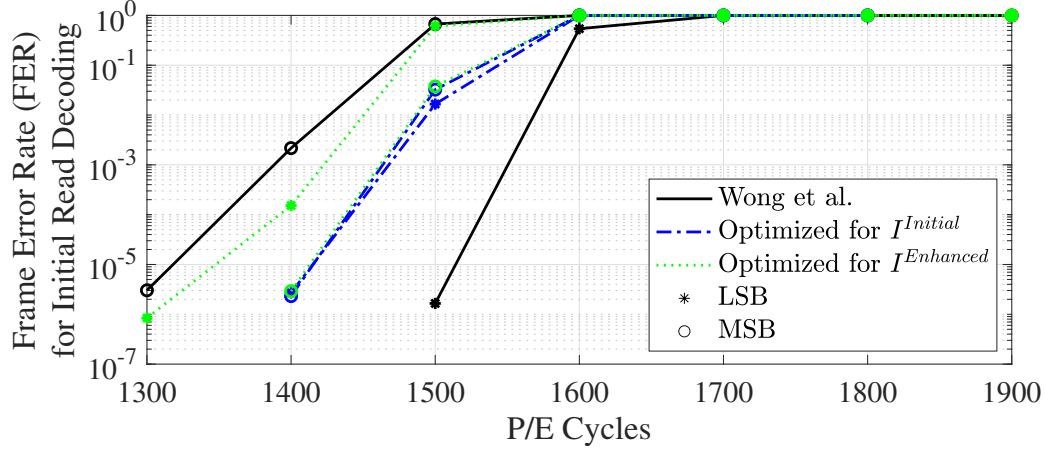


Figure 7.9: FER vs. P/E Cycles for initial read decoding of MSB and LSB pages.

7.6.2 LDPC Simulations Results

Fig. 7.9 shows initial read decoding FER vs. P/E cycles for simulations obtained using optimized mean voltages and initial-read thresholds for $I^{Initial}$ and $I^{Enhanced}$ and FER curves obtained using original mean voltages and initial read thresholds from Wong et al. Construing lifetime as when FER of either MSB or LSB page exceeds 10^{-6} , the initial-read lifetime for mean voltages optimized for $I^{Initial}$ is about 1400 P/E cycles, offering a 100 P/E cycle improvement over the means used in Wong et al. [122]. Initial-read lifetime for mean voltages optimized for $I^{Enhanced}$ is about 1300 P/E cycles, slightly better than the means used in Wong et al. [122]. Consistent with Fig. 7.3, the FER performance for both MSB and LSB pages is similar for the means optimized for $I^{Initial}$.

Fig. 7.10 shows enhanced-read-decoding FER vs. P/E cycles for simulations obtained using the various mean voltages. Construing lifetime as when FER of either page exceeds 10^{-6} , the enhanced-read lifetime for mean voltages optimized either for $I^{Initial}$ and for $I^{Enhanced}$ is about 1510 P/E cycles, improving the lifetime over means used in Wong et al. [122]. Consistent with Fig. 7.4, FER performance for both MSB and LSB pages is similar for the means optimized for $I^{Initial}$.

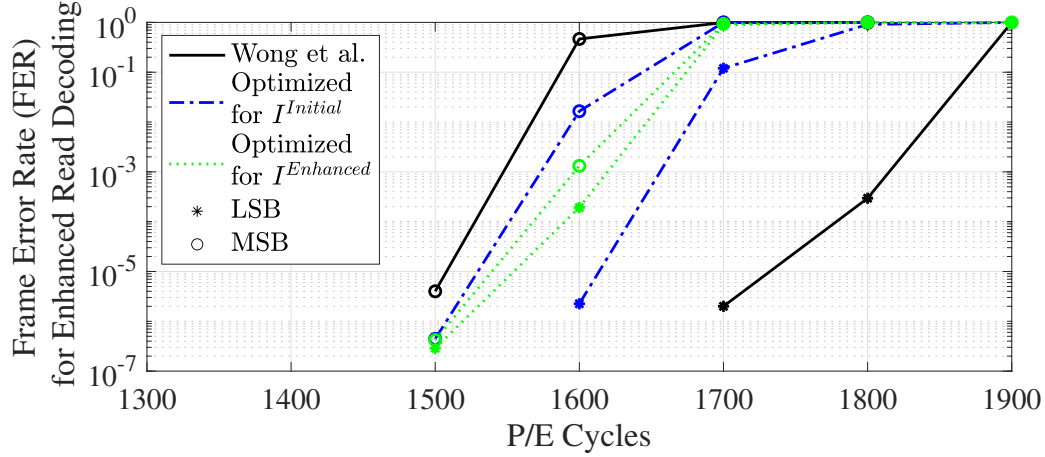


Figure 7.10: FER vs. P/E Cycles for enhanced precision decoding.

7.7 Conclusion

This research demonstrates the additional benefits to Flash memory lifetime that can be achieved by optimizing write voltage levels and thresholds for Flash memory with progressive reads. Optimizing $I^{Initial}$, i.e. for initial-read performance, was the most effective method of maximizing the minimum MI of the MSB and LSB pages in terms of increasing lifetime, whether lifetime is construed as the number of P/E cycles until the MI of a page fell below 0.9 or until the FER of a page rose above 10^{-6} . LDPC simulations showed a strong correlation between MI and LDPC FER performance, confirming that selecting mean voltages to maximize the minimum MI of a page also obtains mean voltages that achieve similar FER performance, and thus extended device lifetime.

Chapter 8

Conclusion

Forward error correction codes are essential for improving the reliability of data transmission over fading channels in satellite communications. LDPC codes deliver capacity-approaching performance with relatively low-complexity decoding, making them well suited for modern satellite communication systems. However, the current CCSDS standard relies on turbo codes for low-rates, and the high error floor of these turbo codes can limit their effectiveness over longer distances. This dissertation addressed the limitations of existing coding solutions for low-rate satellite communication by developing a new class of low-rate PBRL LDPC codes from the rate-1/2 CCSDS LDPC code. In particular, the proposed LDPC codes support rates 1/3, 1/4 and 1/6 for message lengths 1024, 4096, and 16384, offering FER performance similar to existing turbo codes in the waterfall region and no visible error floor for the simulated FER range. These results demonstrate the potential of the proposed codes as a practical alternative to low-rate turbo codes in next-generation satellite communication systems. The new codes also support dynamic adaptation for fading channels.

Dynamic adaptation is an efficient technique that adjusts the coding rate according to the current channel conditions. This allows the system to use stronger error protection (lower code rate) when the channel is poor to maintain reliable transmission. Conversely, when the channel is good, the system can use higher data rates to increase throughput. Thus, dynamic

adaptation keeps transmission both reliable and efficient. In Chapter 3, we explore the effect of round-trip feedback delay in rate-adaptive systems. Specifically, we mitigate performance degradation caused by feedback delay by proposing prediction models to estimate fading channel conditions and dynamically select appropriate LDPC code rate. The behavior of predictive models is examined as feedback delay is increased from no delay to delay equal to coherence time. The simulation results show that quadratic prediction model is the overall best choice for round-trip feedback delays shorter than half of the fading channel coherence time. For coherence time of 10 *ms* this model achieves from 95.2% to 68.7% of ideal zero-delay throughput as round-trip feedback delay is increased from 1 *ms* to 5 *ms*. The channel with coherence time of 10 *ms* is meant to model the optical channel of LEO satellite.

As communication systems continue to demand higher spectral efficiency and reliability, PAS combined with LDPC codes provides a promising technique for approaching the fundamental limits of communication. This dissertation has investigated several approaches to improve the performance of such systems, with a particular focus on code design, error-floor mitigation, and encoding complexity. The proposed techniques address performance limitations in both the waterfall and error-floor regions, providing significant improvements in error-rate performance while maintaining practical implementation complexity.

In Chapter 4 we presented bit-interleaving and post-processing techniques to improve error floors of LDPC-coded systems with PAS when the dominant errors in the error-floor region are low-weight codewords. When these techniques are applied to the rate-2/3 DVB-S2 LDPC code the error floor is reduced by more than two orders of magnitude.

In Chapter 5 we presented a geometric and probabilistic shaping optimization scheme to equalize the conditional entropies of the different bit-levels in the PAM constellation label. Such scheme enables the construction of LDPC encoders with linear encoding complexity using standard design approach for BI-AWGN channel, while harvesting capacity-approaching benefits of PAS. We demonstrated that a protograph LDPC code optimized for the BI-AWGN channel with the proposed constellation that equalizes the conditional entropies outperforms

the code design optimized for unequal bit-level protection in [105].

In the last two chapters of this dissertation, we extend our investigation of LDPC codes with optimal signaling to data storage systems, with a particular focus on Flash memory. For TLC Flash we demonstrated that optimizing the positions of the write levels in the constellation and input probabilities to maximize a mutual information rate of the bit channel with the lowest information rate can significantly improve the minimum page rate, with negligible encoding loss and equal rate constraint loss for operational SNRs. For a two-variance Gaussian channel we demonstrated that jointly optimizing write voltage levels and read thresholds of MLC Flash memory can extend its operational lifetime.

Overall, this dissertation introduced several optimization techniques for LDPC-based communication and storage systems, demonstrating significant performance gains and potential for practical implementation in real-world systems.

Bibliography

- [1] M. M. Abadi, Z. Ghassemlooy, S. Zvanovec, M. R. Bhatnagar, and Y. Wu. Hard switching in hybrid FSO/RF link: Investigating data rate and link availability. In *2017 IEEE International Conference on Communications Workshops (ICC Workshops)*, pages 463–468, 2017.
- [2] A. Abbasfar, D. Divsalar, and K. Yao. Accumulate-Repeat-Accumulate Codes. *IEEE Transactions on Communications*, 55(4):692–702, 2007.
- [3] N. Alon and M. Luby. A linear time erasure-resilient code with nearly optimal recovery. *IEEE Transactions on Information Theory*, 42(11):1732–1736, November 1996.
- [4] J. G. Andrews, T. E. Humphreys, and T. Ji. 6G Takes Shape. *IEEE BITS the Information Theory Magazine*, 4(1):1–38, Mar 2024.
- [5] J. A. Anguita, M. A. Neifeld, B. Hildner, and B. Vasic. Rateless Coding on Experimental Temporally Correlated FSO Channels. *Journal of Lightwave Technology*, 28(7):990–1002, 2010.
- [6] C. A. Aslam, Y. L. Guan, and K. Cai. Dynamic write-level and read-level signal design for MLC NAND flash memory. In *2014 9th International Symposium on Communication Systems, Networks & Digital Sign (CSNDSP)*, pages 336–341, 2014.

- [7] C. A. Aslam, Y. L. Guan, and K. Cai. Read and Write Voltage Signal Optimization for Multi-Level-Cell (MLC) NAND Flash Memory. *IEEE Transactions on Communications*, 64(4):1613–1623, 2016.
- [8] A. Asmani, S. Galijasevic, and R. D. Wesel. Write Voltage Optimization to Increase Flash Lifetime in a Two-Variance Gaussian Channel. In *2024 IEEE International Symposium on Information Theory (ISIT)*, pages 1143–1148, 2024.
- [9] B. Bag, A. Das, I. S. Ansari, A. Prokeš, C. Bose, and A. Chandra. Performance Analysis of Hybrid FSO Systems Using FSO/RF-FSO Link Adaptation. *IEEE Photonics Journal*, 10(3):1–17, 2018.
- [10] D. N. Bailon, S. Shavgulidze, and J. Freudenberger. Cell-wise encoding and decoding for TLC flash memories. In *2022 IEEE 12th International Conference on Consumer Electronics (ICCE-Berlin)*, pages 1–6, 2022.
- [11] C. Berrou and A. Glavieux. Near optimum error correcting coding and decoding: turbo-codes. *IEEE Transactions on Communications*, 44(10):1261–1271, 1996.
- [12] R. Bez, E. Camerlenghi, A. Modelli, and A. Visconti. Introduction to flash memory. *Proceedings of the IEEE*, 91(4):489–502, 2003.
- [13] G. Böcherer, P. Schulte, and F. Steiner. Probabilistic shaping and forward error correction for fiber-optic communication systems. *Journal of Lightwave Technology*, 37(2):230–244, 2019.
- [14] S. Boyd and L. Vandenberghe. *Convex Optimization*, chapter 6.5.3. Cambridge University Press, 2004.
- [15] F. Buchali, F. Steiner, G. Böcherer, L. Schmalen, P. Schulte, and W. Idler. Rate Adaptation and Reach Increase by Probabilistically Shaped 64-QAM: An Experimental Demonstration. *J. Lightw. Technol.*, 34(7):1599–1609, Apr. 2016.

- [16] J. Byers, M. Luby, M. Mitzenmacher, and A. Rege. A digital fountain approach to reliable distribution of bulk data. *ACM SIGCOMM Computer Communication Review*, 28(4), October 1998.
- [17] G. Böcherer. Probabilistic Amplitude Shaping. *Foundations and Trends in Communications and Information Theory*, 20(4):390–511, 2023.
- [18] G. Böcherer, F. Steiner, and P. Schulte. Bandwidth Efficient and Rate-Matched Low-Density Parity-Check Coded Modulation. *IEEE Trans. Comm.*, 63(12):4651–4665, Oct. 2015.
- [19] Y. Cai, Y. Luo, E. F. Haratsch, K. Mai, and O. Mutlu. Data retention in MLC NAND flash memory: Characterization, optimization, and recovery. In *2015 IEEE 21st International Symposium on High Performance Computer Architecture (HPCA)*, pages 551–563, 2015.
- [20] G. Caire, G. Taricco, and E. Biglieri. Bit-interleaved coded modulation. *IEEE Transactions on Information Theory*, 44(3):927–946, May 1998.
- [21] T.-Y. Chen, D. Divsalar, J. Wang, and R. D. Wesel. Protograph-Based Raptor-Like LDPC Codes for Rate Compatibility with Short Blocklengths. In *2011 IEEE Global Telecommunications Conference - GLOBECOM 2011*, Houston, TX, USA, Dec. 2011.
- [22] T.-Y. Chen, D. Divsalar, and R. D. Wesel. Protograph-based Raptor-like LDPC codes with low thresholds. In *Proceedings of the IEEE International Conference on Communications*, Ottawa, ON, Canada, 2012.
- [23] T.-Y. Chen, K. Vakilinia, D. Divsalar, and R. D. Wesel. Protograph-Based Raptor-Like LDPC Codes. *IEEE Transactions on Communications*, 63(5):1522–1532, 2015.

- [24] T.-Y. Chen, A. R. Williamson, and R. D. Wesel. Increasing flash memory lifetime by dynamic voltage allocation for constant mutual information. In *2014 Information Theory and Applications Workshop (ITA)*, pages 1–5, 2014.
- [25] S. K. Chilappagari, S. Sankaranarayanan, and B. Vasic. Error Floors of LDPC Codes on the Binary Symmetric Channel. In *2006 IEEE International Conference on Communications*, volume 3, pages 1089–1094, 2006.
- [26] Consultative Committee for Space Data Systems (CCSDS). Non-Coherent Optical Communications Coding and Synchronization. Technical Report CCSDS 142.0-P-1.1, CCSDS, November 2023. Draft Recommended Standard (Pink Sheets).
- [27] M. C. Coşkun, G. Durisi, T. Jerkovits, G. Liva, W. Ryan, B. Stein, and F. Steiner. Efficient error-correcting codes in the short blocklength regime. *Physical Communication*, 34:66–79, 2019.
- [28] M. Czaputa, T. Javornik, E. Leitgeb, G. Kandus, and Z. Ghassemlooy. Investigation of punctured LDPC codes and time-diversity on free-space optical links. In *Proceedings of the 11th International Conference on Telecommunications*, pages 359–362, 2011.
- [29] J. D’Errico. fminsearchbnd, fminsearchcon. <https://www.mathworks.com/matlabcentral/fileexchange/8277-fminsearchbnd-fminsearchcon>. MATLAB Central File Exchange.
- [30] D. Divsalar, S. Dolinar, C. R. Jones, and K. Andrews. Capacity-approaching protograph codes. *IEEE Journal on Selected Areas in Communications*, 27(6):876–888, Aug. 2009.
- [31] D. Divsalar, H. Jin, and R. McEliece. Coding theorems for ‘turbo-like’ codes. In *Proceedings of the 1998 Allerton Conference*, 1998.

- [32] D. Divsalar, H. Jin, and R. J. McEliece. Coding Theorems for Turbo-like Codes. In *Proceedings by Allerton Conference on Communication, Control, and Computing*, Monticello, IL, USA, Sep 1998.
- [33] D. Divsalar, C. Jones, S. Dolinar, and J. Thorpe. Protograph based LDPC codes with minimum distance linearly growing with block size. In *GLOBECOM '05. IEEE Global Telecommunications Conference, 2005.*, volume 3, pages 5 pp.–, 2005.
- [34] I. B. Djordjevic. Adaptive Modulation and Coding for Free-Space Optical Channels. *Journal of Optical Communications and Networking*, 2(5):221–229, 2010.
- [35] L. Dolecek and Y. Cassuto. Channel Coding for Nonvolatile Memory Technologies: Theoretical Advances and Practical Considerations. *Proceedings of the IEEE*, 105(9):1705–1724, 2017.
- [36] L. Dolecek, P. Lee, Z. Zhang, V. Anantharam, B. Nikolic, and M. Wainwright. Predicting error floors of structured LDPC codes: deterministic bounds and estimates. *IEEE Journal on Selected Areas in Communications*, 27(6):908–917, 2009.
- [37] L. Duan, B. Rimoldi, and R. Urbanke. Approaching the AWGN channel capacity without active shaping. In *Proceedings of IEEE International Symposium on Information Theory*, Ulm, Germany, July 1997.
- [38] C. Duangthong, W. Phakphisut, and P. Supnithi. Search algorithm of write voltage optimization in NAND flash memory. In *2017 International Electrical Engineering Congress (iEECON)*, pages 1–4, 2017.
- [39] C. Duangthong, W. Phakphisut, and P. Supnithi. Capacity Enhancement of Asymmetric Multi-Level Cell (MLC) NAND Flash Memory using Write Voltage Optimization. In *2019 34th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC)*, pages 1–4, 2019.

- [40] C. Duangthong, W. Phakphisut, and P. Supnithi. Read Voltage Optimization in MLC NAND Flash Memory via the Density Evolution. In *2019 26th International Conference on Telecommunications (ICT)*, pages 361–365, 2019.
- [41] DVBS2. Digital Video Broadcasting (DVB); 2nd generation framing structure, channel coding and modulation systems for Broadcasting, Interactive Services, News Gathering and Other Broadband Satellite Applications (DVB-S2). Technical Report EN 302 307 V1.2.1, European Telecomm. Standards Inst. (ETSI), 2009.
- [42] DVBT2. Digital Video Broadcasting (DVB), Frame structure, channel coding and modulation for a second generation digital terrestrial television broadcasting system (DVB-T2). Technical Report EN 302 755 V1.3.1, institution = European Telecomm. Standards Inst. (ETSI),, 2012.
- [43] A. Eslami, S. Vangala, and H. Pishro-Nik. Hybrid Channel Codes for Efficient FSO/RF Communication Systems. *IEEE Transactions on Communications*, 58(10):2926–2938, 2010.
- [44] ETSI. 5G NR; Multiplexing and channel coding (3GPP TS 38.212 version 15.2.0 Release 15). Technical Specification ETSI TS 138 212 V15.2.0, European Telecommunications Standards Institute, July 2018.
- [45] C. C. for Space Data Systems (CCSDS). TM synchronization and Channel coding—summary of concept and rationale. *Report Concerning Space Data System Standards (Green Book). Informational Report CCSDS, CCSDS 131.1-G-3*, 2020.
- [46] C. C. for Space Data Systems (CCSDS). TM Synchronization and Channel coding—summary of concept and rationale. *Blue Book*, 2023.

- [47] C. C. for Space Data Systems (CCSDS). Low density parity check codes for use in near-earth and deep space applications. *Experimental Specification, CCSDS 131.1-O-1. Orange Book, Issue 1.*, August 2006.
- [48] G. Forney. Trellis shaping. *IEEE Transactions on Information Theory*, 38(2):281–300, Mar. 1992.
- [49] G. Forney, R. Gallager, G. Lang, F. Longstaff, and S. Qureshi. Efficient Modulation for Band-Limited Channels. *IEEE Journal on Selected Areas in Communications*, 2(5):632–647, Sept. 1984.
- [50] C. Fragouli, R. Wesel, D. Sommer, and G. Fettweis. Turbo codes with non-uniform constellations. In *ICC 2001. IEEE International Conference on Communications. Conference Record (Cat. No.01CH37240)*, volume 1, pages 70–73 vol.1, 2001.
- [51] B. Frey, R. Kötter, and A. Vardy. Skewness and pseudocodewords in iterative decoding. In *IEEE International Symposium on Information Theory*, pages 148–, Aug. 1998.
- [52] S. Galijasevic. LDPC HRC and IRC proto-matrices for FSO Channel.
- [53] S. Galijasevic, G. Liva, D. Divsalar, and R. Wesel. Geometric and Probabilistic Shaping to Enable LDPC Encoding with Linear Complexity. In *2025 13th Int. Symposium on Topics in Coding (ISTC)*, pages 1–5, 2025.
- [54] S. Galijasevic, J. Luo, D. Divsalar, and R. Wesel. Channel-Prediction-Driven Rate Control for LDPC Coding in a Fading FSO Channel With Delayed Feedback. *IEEE Open Journal of the Communications Society*, 6:3320–3331, 2025.
- [55] S. Galijasevic, L. Wang, J. Hamkins, R. Wesel, and D. Divsalar. Construction of Low-Rate LDPC Codes from Rate-1/2 CCSDS Standard LDPC Codes. In *2025 IEEE Aerospace Conference*, pages 1–9, 2025.

- [56] S. Galijasevic and R. D. Wesel. Optimizing Write Voltages for Independent, Equal-Rate Pages in Flash Memory. In *2022 56th Asilomar Conference on Signals, Systems, and Computers*, pages 168–174, 2022.
- [57] R. G. Gallager. Low-Density Parity-Check Codes. *Cambridge, MA: MIT Press, 1963*, 1963.
- [58] R. G. Gallager. *Low-Density Parity-Check Codes*. MIT Press, Cambridge, MA, Sept. 1963.
- [59] R. G. Gallager. *Information Theory and Reliable Communication*. Wiley, Hoboken, NJ, USA, Jan. 1968.
- [60] A. García-Zambrana, C. Castillo-Vázquez, and B. Castillo-Vázquez. Rate-adaptive FSO links over atmospheric turbulence channels by jointly using repetition coding and silence periods. *Opt. Express*, 18(24):25422–25440, Nov 2010.
- [61] F. Gray. Pulse code communications, 1953.
- [62] L. M. Grupp, A. M. Caulfield, J. Coburn, S. Swanson, E. Yaakobi, P. H. Siegel, and J. K. Wolf. Characterizing flash memory: Anomalies, observations, and applications. In *2009 42nd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pages 24–33, 2009.
- [63] A. Guillén i Fàbregas, A. Martinez, and G. Caire. Bit-Interleaved Coded Modulation. *Foundations and Trends in Communications and Information Theory*, 5(1–2):1–153, 2008.
- [64] Y. Han and W. E. Ryan. Low-floor decoders for LDPC codes. *IEEE Transactions on Communications*, 57(6):1663–1673, 2009.

- [65] J. Hou, P. Siegel, L. Milstein, and H. Pfister. Capacity-approaching bandwidth-efficient coded modulation schemes based on low-density parity-check codes. *IEEE Transactions on Information Theory*, 49(9):2141–2155, 2003.
- [66] J. Hou, P. Siegel, L. Milstein, and H. Pfister. Capacity-approaching bandwidth-efficient coded modulation schemes based on low-density parity-check codes. *IEEE Transactions on Information Theory*, 49(9):2141–2155, Sept. 2003.
- [67] P. Huang, P. H. Siegel, and E. Yaakobi. Performance of Multilevel Flash Memories With Different Binary Labelings: A Multi-User Perspective. *IEEE Journal on Selected Areas in Communications*, 34(9):2336–2353, 2016.
- [68] C. Häger, A. Graell i Amat, A. Alvarado, F. Brännström, and E. Agrell. Optimized bit mappings for spatially coupled LDPC codes over parallel binary erasure channels. In *Proceedings of the IEEE International Conference on Communications*, Sydney, Australia, Jun. 2014.
- [69] H. Jin, A. Khandekar, and R. McEliece. Irregular Repeat-Accumulate Codes. In *Proceedings of the International Symposium on Turbo Codes & Related Topics*, Brest, France, Sept. 2000.
- [70] M. A. Khalighi and M. Uysal. Survey on Free Space Optical Communication: A Communication Theory Perspective. *IEEE Communications Surveys & Tutorials*, 16(4):2231–2258, 2014.
- [71] M. N. Khan, S. O. Gilani, M. Jamil, A. Rafay, Q. Awais, B. A. Khawaja, M. Uzair, and A. W. Malik. Maximizing Throughput of Hybrid FSO-RF Communication System: An Algorithm. *IEEE Access*, 6:30039–30048, 2018.

- [72] A. Khandani and P. Kabal. Shaping multidimensional signal spaces. I. Optimum shaping, shell mapping. *IEEE Transactions on Information Theory*, 39(6):1799–1808, Aug. 1993.
- [73] K.-J. Kim, S. Myung, and H. Jeong. Lowering error floors by removing dominant trapping sets of low-density parity-check codes for broadcasting systems. In *2015 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting*, pages 1–3, 2015.
- [74] Y. Kim, J. Kim, J. J. Kong, B. V. K. V. Kumar, and X. Li. Verify level control criteria for multi-level cell flash memories and their applications. *EURASIP Journal on Advances in Signal Processing*, 2012.
- [75] Z. Kolka, V. Biolková, and D. Biolek. Channel Model for Monte-Carlo Simulation of Data Transmission on Terrestrial FSO Paths. In *International Conference on Emerging Trends in Engineering and Technology*, 2014.
- [76] S. Le Goff, B. Sharif, and S. Jimaa. A new bit-interleaved coded modulation scheme using shaping coding. In *IEEE Global Telecommunications Conference, 2004. GLOBECOM '04.*, Dallas, TX, USA, Jan. 2004.
- [77] M. Leba, S. Riurean, and A. Lonica. LiFi — The path to a new way of communication. In *2017 12th Iberian Conference on Information Systems and Technologies (CISTI)*, pages 1–6, 2017.
- [78] J. Lei and W. Gao. Matching graph connectivity of LDPC codes to high-order modulation by bit interleaving. In *Proceedings by Allerton Conference on Communication, Control, and Computing*, Monticello, IL, USA, Sept. 2008.

- [79] Q. Li, A. Jiang, and E. F. Haratsch. Noise modeling and capacity analysis for NAND flash memories. In *2014 IEEE International Symposium on Information Theory*, pages 2262–2266, 2014.
- [80] Y. Li and W. Ryan. Bit-reliability mapping in LDPC-coded modulation systems. *IEEE Communications Letters*, 9(1):1–3, Jan. 2005.
- [81] L. Liu, M. Safari, and S. Hranilovic. Rate-adaptive FSO communication via rate-compatible punctured LDPC codes. In *2013 IEEE International Conference on Communications (ICC)*, pages 3948–3952, 2013.
- [82] G. Liva and M. Chiani. Protograph LDPC Codes Design Based on EXIT Analysis. In *IEEE GLOBECOM 2007 - IEEE Global Telecommunications Conference*, Washington, DC, USA, Nov. 2007.
- [83] D. MacKay. Good error correcting codes based on very sparse matrices. *IEEE Transactions on Information Theory*, 45(3):399–431, March 1999.
- [84] D. MacKay and R. Neal. Good codes based on very sparse matrices. In C. Boyd, editor, *Cryptography and Coding, 5th IMA Conference*, Berlin, October 1995. Springer-Verlag.
- [85] D. Mackay and M. Postol. Weaknesses of Margulis and Ramanujan-Margulis Low-Density Parity-Check Codes. *Electronic Notes in Theoretical Computer Science*, 74:97–104, Apr. 2003.
- [86] A. Martinez, A. Guillen i Fabregas, G. Caire, and F. Willems. Bit-Interleaved Coded Modulation Revisited: A Mismatched Decoding Perspective. *IEEE Transactions on Information Theory*, 55(6):2756–2765, June 2009.
- [87] N. Merhav and G. Böcherer. Codebook Mismatch can be Fully Compensated by Mismatched Decoding. *IEEE Transactions on Information Theory*, 69(4):2152–2164, 2023.

- [88] N. Merhav and G. Böcherer. Codebook Mismatch can be Fully Compensated by Mismatched Decoding. *IEEE Transactions on Information Theory*, 69(4):2152–2164, 2023.
- [89] T. Mizuochi. Recent progress in forward error correction and its interplay with transmission impairments. *IEEE Journal of Selected Topics in Quantum Electronics*, 12(4):544–554, 2006.
- [90] M. Mondelli, S. H. Hassani, and R. L. Urbanke. How to Achieve the Capacity of Asymmetric Channels. *IEEE Transactions on Information Theory*, 64(5):3371–3393, May 2018.
- [91] J. Nguyen, E. M. Liang, L. Wang, R. D. Wesel, T. Drullinger, and T. Chauvin. Comparison of Integrated and Independent RF/FSO Transceivers on a Fading Optical Channel. In *2020 54th Asilomar Conference on Signals, Systems, and Computers*, pages 699–701, 2020.
- [92] F. Peng, W. Ryan, and R. Wesel. Surrogate-channel design of universal LDPC codes. *IEEE Communications Letters*, 10(6):480–482, June 2006.
- [93] H. T. T. Pham, N. T. Dang, L. T. Vu, and H. T. Bui. A survey of performance improvement methods for free-space optical communication systems. In *2014 International Conference on Advanced Technologies for Communications (ATC 2014)*, pages 770–775, 2014.
- [94] Y. Polyanskiy, H. V. Poor, and S. Verdú. Channel Coding Rate in the Finite Blocklength Regime. *IEEE Transactions on Information Theory*, 56(5):2307–2359, 2010.
- [95] S. V. S. Ranganathan. Flash LDPC code: rate 8/9. http://www.seas.ucla.edu/csl/codes/Flash_Rate_8_9_LDPC.txt. UCLA Coding Theory Group.

- [96] T. Richardson. Error-floors of LDPC codes. In *41st Annual Allerton Conference on Communication, Control, and Computing*, Illinois, USA, Sept. 2003.
- [97] T. Richardson, M. Shokrollahi, and R. Urbanke. Design of capacity-approaching irregular low-density parity-check codes. *IEEE Transactions on Information Theory*, 47(2):619–637, 2001.
- [98] T. J. Richardson and R. L. Urbanke. The capacity of low-density parity-check codes under message-passing decoding. *IEEE Transactions on Information Theory*, 47(2):599–618, Feb. 2001.
- [99] F. J. C. Romero and B. M. Kurkoski. LDPC Decoding Mappings That Maximize Mutual Information. *IEEE Journal on Selected Areas in Communications*, 34(9):2391–2401, 2016.
- [100] W. E. Ryan, S. Lin, and S. G. Wilson. *Channel Codes: Classical and Modern, second edition*. Cambridge University Press, 2025.
- [101] C. Schoeny, F. Sala, and L. Dolecek. Analysis and coding schemes for the flash normal-laplace mixture channel. In *2015 IEEE International Symposium on Information Theory (ISIT)*, pages 2101–2105, 2015.
- [102] P. Schulte and G. Böcherer. Constant Composition Distribution Matching. *IEEE Transactions on Information Theory*, 62(1):430–434, Jan. 2016.
- [103] P. Schulte, W. Labidi, and G. Kramer. Joint Decoding of Distribution Matching and Error Control Codes. In *Proceedings of the International Zurich Seminar on Information and Communication (IZS)*, pages 36–37, Zurich, Switzerland, February 2020.
- [104] Space Development Agency. Optical Communications Terminal (OCT) Standard Version 3.1.0. Technical report, Space Development Agency, 2023.

- [105] F. Steiner, G. Böcherer, and G. Liva. Protograph-Based LDPC Code Design for Shaped Bit-Metric Decoding. *IEEE Journal on Selected Areas in Communications*, 34(2):397–407, Feb. 2016.
- [106] F. Steiner, E. B. Yacoub, B. Matuz, G. Liva, and A. G. i Amat. One and Two Bit Message Passing for SC-LDPC Codes With Higher-Order Modulation. *J. Lightwave Technol.*, 37(23):5914–5925, Dec 2019.
- [107] W. Sun and J. Zheng. A Low-Complexity Retention Noise Parameter Estimation for MLC NAND Flash Memory. In *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, pages 1–6, 2019.
- [108] R. Tanner. A recursive approach to low complexity codes. *IEEE Transactions on Information Theory*, 27(5):533–547, 1981.
- [109] R. M. Tanner. On Quasi-Cyclic Repeat-Accumulate Codes. In *Allerton Conference on Communication, Control, and Computing*, Monticello, IL, USA, Jan. 2000.
- [110] J. Thorpe. Low-Density Parity-Check (LDPC) Codes Constructed from Protographs,. *JPL IPN Progress Report 42-154*, August 15, 2003.
- [111] T. Tian, C. Jones, J. Villasenor, and R. Wesel. Selective avoidance of cycles in irregular LDPC code construction. *IEEE Transactions on Communications*, 52(8):1242–1247, Aug. 2004.
- [112] M. Uysal and H. Nouri. Optical wireless communications — An emerging technology. In *2014 16th International Conference on Transparent Optical Networks (ICTON)*, pages 1–7, 2014.
- [113] B. Vasic and I. B. Djordjevic. Iteratively Decodable Block Codes for Long Haul Optical Transmission Systems. *Journal of Optical Communications*, 23(5):182–186, 2002.

- [114] B. Vasic and O. Milenkovic. Combinatorial constructions of low-density parity-check codes for iterative decoding. *IEEE Transactions on Information Theory*, 50(6):1156–1176, 2004.
- [115] H. Wang. Flash model: mean and standard deviation. <http://www.seas.ucla.edu/csl/codes/enhnPrecisionMuSigma.txt>. UCLA Coding Theory Group.
- [116] H. Wang, N. Wong, T.-Y. Chen, and R. D. Wesel. Using Dynamic Allocation of Write Voltage to Extend Flash Memory Lifetime. *IEEE Transactions on Communications*, 64(11):4474–4486, 2016.
- [117] J. Wang, T. Courtade, H. Shankar, and R. D. Wesel. Soft Information for LDPC Decoding in Flash: Mutual-Information Optimized Quantization. In *2011 IEEE Global Telecommunications Conference - GLOBECOM 2011*, pages 1–6, 2011.
- [118] 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 Journal on Selected Areas in Communications*, 32(5):880–891, 2014.
- [119] L. Wang, D. Song, F. Areces, and R. D. Wesel. Achieving Short-Blocklength RCU Bound via CRC List Decoding of TCM with Probabilistic Shaping. In *ICC 2022 - IEEE International Conference on Communications*, pages 2906–2911, 2022.
- [120] L. Wang, D. Song, F. Areces, T. Wiegart, and R. D. Wesel. Probabilistic Shaping for Trellis-Coded Modulation With CRC-Aided List Decoding. *IEEE Transactions on Communications*, 71(3):1271–1283, 2023.
- [121] R. Wesel, X. Liu, J. Cioffi, and C. Komninakis. Constellation labeling for linear encoders. *IEEE Transactions on Information Theory*, 47(6):2417–2431, 2001.

- [122] N. Wong, E. Liang, H. Wang, S. V. S. Ranganathan, and R. D. Wesel. Decoding Flash Memory with Progressive Reads and Independent vs. Joint Encoding of Bits in a Cell. In *2019 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6, 2019.
- [123] M. W. Wright, M. Srinivasan, and K. Wilson. Improved Optical Communications Performance Using Adaptive Optics with an Avalanche Photodiode Detector. *Interplanetary Network Progress Report*, 42-161:1–13, May 2005.
- [124] D. Xiao, L. Wang, D. Song, and R. D. Wesel. Finite-Support Capacity-Approaching Distributions for AWGN Channels. In *2020 IEEE Information Theory Workshop (ITW)*, pages 1–5, 2021.
- [125] M. Yang, W. Ryan, and Y. Li. Design of efficiently encodable moderate-length high-rate irregular ldpc codes. *IEEE Transactions on Communications*, 52(4):564–571, 2004.
- [126] Y. Yeh, A. Fazeli, and P. H. Siegel. Optimal Placement of Read Thresholds for Coded NAND Flash Memory. In *ICC 2021 - IEEE International Conference on Communications*, pages 1–7, 2021.
- [127] E. Zehavi. 8-PSK trellis codes for a Rayleigh channel. *IEEE Trans. Comm.*, 40(5):873–884, May 1992.
- [128] Y. Zhang and W. E. Ryan. Toward low LDPC-code floors: a case study. *IEEE Transactions on Communications*, 57(6):1566–1573, 2009.