

**Design and Analysis of
Parity-Check-Code-Based
Optical Recording Systems**

Design and Analysis of Parity-Check-Code-Based Optical Recording Systems

KUI CAI

(M. Eng., National University of Singapore)

A THESIS SUBMITTED
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY IN THE
ELECTRICAL AND COMPUTER ENGINEERING DEPARTMENT
OF NATIONAL UNIVERSITY OF SINGAPORE
2007

Design and Analysis of Parity-Check-Code-Based Optical Recording Systems

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de Technische
Universiteit Eindhoven, op gezag van de Rector Magnificus,
prof.dr.ir. C.J. van Duijn, voor een commissie aangewezen door
het College voor Promoties in het openbaar te verdedigen op
maandag 12 maart 2007 om 14.00 uur

door

Kui Cai

geboren te Liao Ning, China

Dit proefschrift is goedgekeurd door de promotoren:

prof.dr.ir. J.W.M. Bergmans
en
prof.dr. T.C. Chong

Copromotor:
dr. G. Mathew

About the cover: the illustration at the back cover was kindly provided by Dr. Jean Schleipen of Philips Research.

© Copyright 2007 Kui Cai

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior written permission from the copyright owner.

CIP-DATA LIBRARY TECHNISCHE UNIVERSITEIT EINDHOVEN

Cai, Kui

Design and analysis of parity-check-code-based optical recording systems / by Kui Cai. - Eindhoven :

Technische Universiteit Eindhoven, 2007.

Proefschrift. - ISBN 978-90-386-2003-9

NUR 959

Trefw.: optische registratie / digitale signaalverwerking / foutencorrigerende codes / dataopslag.

Subject headings: optical storage / signal processing / error correction codes / magnetic disc storage.

Printed by: University printing office, Eindhoven, The Netherlands

promotiecommissie:

prof.dr.ir. J.W.M. Bergmans

prof.dr. T.C. Chong

prof.dr. H. Corporaal

prof.dr. H.K. Garg

dr. G. Mathew

prof.dr.ir. A.J.H. Vinck

dr.ir. F.M.J. Willems

to my parents and my family

Acknowledgments

I would like to express my greatest gratitude to my supervisors, Prof. Jan W. M. Bergmans and Asst. Prof. George Mathew, for their invaluable guidance, constant and tremendous help and encouragement along my way to my Ph.D. Without their help, this thesis would not exist in its current form.

I wish to express my heartfelt thanks to Prof. Jan W. M. Bergmans from Technical University of Eindhoven, The Netherlands. It has been a great honor and privilege to work under his supervision. I owe a lot to him for his high quality supervision and enthusiastic and continuous help for my Ph.D. He has spent much time in reviewing my work, providing advice and giving directions, and refining my writing. I have gained tremendous wisdom from his constructive criticism and invaluable advice, and enlightening discussions with him. I would like to express my special thanks to him for arranging my visit to Technical University of Eindhoven in 2003, and facilitating the valuable meetings and discussions with key experts in Philips Research Laboratories, The Netherlands. I am truly grateful to him.

I am deeply indebted to Asst. Prof. George Mathew from National University of Singapore. He dedicated large amounts of his time and expertise in painstakingly teaching me, reviewing my work, providing critical reviews and comments, and pointing out challenging problems and directions during every step of my Ph.D. work. It would have simply been impossible to finish this work without his tremendous help and supervision. His systematic and rigorous approach to research has been a constant source of challenge to achieve greater heights, and it will have a long-lasting effect on my career in the future. His guidance and help for both my Ph.D. and my life will never be forgotten.

I would also like to express my sincere appreciation to Prof. Kees A. Schouhamer Immink. He has freely shared his time and insights with me and provided precious advice to this work. The knowledge I gained from him was of great help in my Ph.D. work. I am deeply grateful to him for

the tremendous source of knowledge and inspiration that he has given to me.

I am grateful to Dr. Victor Y. Krachkovsky, Agere Systems, Allentown, USA, who was one of my supervisors during the initial period, for his involvement and contributions in the initial stage of my research.

I would like to thank Prof. Frans Willems, Technical University of Eindhoven, and Dr. Wim M. J. Coene, Dr. Alexander Padiy, Dr. Stan Baggen, and Dr. Bin Yin, all of Philips Research Laboratories, The Netherlands. They shared with me a lot of valuable opinions and suggestions for my research.

I would like to thank Mr. Foo Yong Ning, National University of Singapore, for his support in Sections 6.2.1 and 6.2.2 of the thesis. I am thankful to my colleagues at the Data Storage Institute, Dr. Qin Zhiliang, Mr. Ye Weichun, Dr. Chan Kheong Sann, Mr. Li Qing, Dr. Lin Yu, Mr. Zou Xiaoxin, and Mr. Peh Chester, who have helped me in one way or another.

Last but by no means least, I wish to thank my parents, my husband and my daughter, for their unfailing love, encouragement, support, patience and sacrifice. I dedicate this thesis to them.

Summary

Design and Analysis of Parity-Check-Code-Based Optical Recording Systems

The increasing demand for high-density and high-speed digital optical recording systems has made the development of advanced coding and signal processing techniques for fast and reliable data recovery increasingly important. In recent years, the parity-check (PC)-code-based reception technique has been widely studied for magnetic recording systems, and it is projected to be highly promising for high-density optical recording. The PC code is an inner error correction code (ECC), which can detect dominant short error events of the system, using only a few parity bits. This reduces the loss in error correction capability of the outer ECC due to random short errors and results in a simple and efficient solution to improve the overall performance. This thesis is dedicated to the design and analysis of PC-code-based recording systems to achieve higher recording capacity with low implementation complexity, for high-density blue laser disc systems. In particular, most of the key components of the PC-code-based optical recording system have been designed and optimized for different recording densities, and different proportions of white noise and media noise.

During the development of advanced coding and detection techniques, it becomes necessary to investigate the system's performance with different coding schemes and recording densities. In Chapter 2 of the thesis, we propose a generalized Braat-Hopkins model for optical recording channels, which provides a fair basis for the performance comparison of detection schemes over different coding schemes and recording densities, for channels with additive, white Gaussian noise (AWGN) and media noise.

Various basic issues associated with the PC-code-based systems are investigated in Chapter 3. These include bounds for bit error rates and error

event probabilities, the dominant error events at channel detector output, different PC codes, simple and efficient post-processors, the impact of error events that are split across data block boundaries and the corresponding remedy, as well as the effects of code rates and recording densities. Simulation results show that a 4-bit PC code achieves the best performance. The corresponding bit error rates (BERs) are very close to the performance bounds, at both nominal and high densities.

Constrained codes, which serve as a key component in the read-write channel of data storage systems, are desired to have a high code rate with low-complexity encoder/decoder. In Chapter 4, we investigate the design of certain capacity-approaching constrained codes for optical recording systems. In particular, we derive analytically the relationship between the number of encoder states and the maximum size of these codes. We identify the minimum number of encoder states that maximizes the code rate, for any desired codeword length.

The design of constrained PC codes is key to the development of PC-code-based systems, and the systematic design of efficient constrained PC codes remains a challenging problem. In Chapter 5, we propose a general and systematic code design technique for constructing capacity-approaching constrained PC codes, which can detect any type of error events of the system, with the minimum code rate loss. Compared to the rate 2/3 code without parity, the newly designed constrained 4-bit PC code achieves a performance gain of 2 dB at nominal density, and 1.5 dB at high density, at $\text{BER} = 10^{-5}$.

As the dominant noise for high-density optical recording systems, media noise severely degrades the performance of channel detectors and post-processors that are designed for AWGN. In Chapter 6, we propose two novel modifications to the bit detector to combat media noise. We further develop a data-dependent post-processing scheme for error correction. Compared to the system designed without considering media noise and without PC codes, the overall performance gain of the developed scheme can be more than 11 dB at high media noise levels.

In data storage systems, the ECC failure rate (EFR) serves as the ultimate measure of the data recovery performance. In Chapter 7, we develop semi-analytical approaches for estimating the EFR, and analyze EFRs for the developed PC-code-based systems, for the cases without and with interleaving. Our analysis shows that with the optimum interleaving degrees, compared to the rate 9/13 code without parity, the 2-bit PC code achieves a gain of around 0.5 dB, and the 4-bit PC code gains 0.6 dB, at high record-

ing density and $\text{EFR} = 10^{-16}$. The thesis concludes in Chapter 8 with some suggestions for further work.

Contents

Acknowledgments	vii
Summary	ix
Glossary	xvii
List of Figures	xxi
List of Tables	xxv
1 Introduction	1
1.1 Optical Recording Technology	1
1.2 Coding and Detection for Optical Recording	3
1.2.1 Optical Recording Systems	3
1.2.2 Imperfections in Optical Recording Channels	6
1.2.3 Overview of Coding and Detection Techniques	8
1.2.4 Performance Measures	12
1.3 Motivation and Scope of the Current Work	13
1.4 Contributions of the Thesis	15
1.5 Organization of the Thesis	16
2 Channel Model	17
2.1 Conventional Braat-Hopkins Model	17
2.2 Generalized Model for Channel with White Noise	21
2.3 Generalized Model for Channel with Media Noise	25
2.3.1 Media Noise in Rewritable Optical Recording Systems	25
2.3.2 Channel Modeling for Media Noise	26
2.4 Model Summary and Numerical Results	30
2.5 Conclusion	32

3	Parity-Check Coded Channels Based on Rate 2/3 Code	35
3.1	Bit Error Rate Analysis	36
3.1.1	Introduction	36
3.1.2	Performance Bounds	38
3.1.3	Comparison of Numerical and Simulation Results, without Parity-Check (PC) Codes	43
3.2	Parity-Check Codes	47
3.2.1	Cyclic Redundancy Check (CRC) Codes	47
3.2.2	A New Single-Bit Parity-Check (PC) Code	48
3.2.3	Hierarchical Scheme for Parity-Check (PC) Coding	48
3.3	Post-Processing Schemes	50
3.3.1	Analysis of Parity-Check (PC) based Post-Processing	50
3.3.2	Multiple-Error-Event Correction Post-Processor	52
3.4	Study of Boundary Error Events	55
3.4.1	Boundary Error Event Analysis	55
3.4.2	A Novel Remedy Scheme	56
3.5	BER Performance and Discussion	57
3.5.1	Choice of Codeword Length	58
3.5.2	Comparison of Theoretical BER Bounds with Simu- lations, with Parity-Check (PC) Codes	61
3.5.3	Simulation Results	62
3.6	Conclusion	65
4	Capacity-Approaching $d = 1$ Codes	67
4.1	Introduction	68
4.1.1	Constrained Codes for Blue Laser Disc Systems	68
4.1.2	Constrained Code Design Techniques	70
4.1.3	Motivation for the Current Work	72
4.2	Design of Capacity-Approaching $d = 1$ Codes	73
4.3	Optimum Number of Encoder States	74
4.3.1	Fibonacci and Generalized Fibonacci Numbers	74
4.3.2	Relationship Between Encoder States and Code Size	75
4.3.3	Minimum Number of Encoder States	80
4.4	Conclusion	88
5	Capacity-Approaching Constrained Parity-Check Codes	89
5.1	Introduction	90
5.2	General Principle of the New Code Design	92
5.3	Code Design in NRZI Format	94
5.3.1	Encoder Description	94

5.3.2	Design of the Component Codes	95
5.3.3	Decoder Description	98
5.4	Code Design in NRZ Format	98
5.5	Examples of New Codes	100
5.6	Performance Evaluation	105
5.7	Conclusions	107
6	Parity-Check Coded Channels with Media Noise	109
6.1	Introduction	110
6.2	Detection for Media Noise Channels	111
6.2.1	Monic Constrained MMSE Equalization	111
6.2.2	Viterbi Detection with Data-Dependent Noise Variance	114
6.2.3	BER Simulation Results	116
6.3	PC Codes and Post-processing for Media Noise Channels .	118
6.3.1	Parity-Check Codes	118
6.3.2	Data-Dependent Post-Processing	120
6.3.3	BER Simulation Results	121
6.4	Conclusion	123
7	Performance Analysis with Error Correction Code	125
7.1	Introduction	126
7.2	Byte Error Rate of PC Coded Systems	128
7.3	Semi-Analytical Approaches for Analyzing EFR	129
7.3.1	Multinomial Model	129
7.3.2	Block Multinomial Model	130
7.3.3	Accuracy Analysis for Non-Interleaved ECC	131
7.3.4	Generalized Multinomial/Block Multinomial Methods	134
7.3.5	Accuracy Analysis for Interleaved ECC	135
7.4	ECC Failure Rate of PC Coded Systems	137
7.4.1	Failure Rate With Non-Interleaved ECC	137
7.4.2	Failure Rate With Interleaved ECC	140
7.5	Conclusion	150
8	Epilogue	153
8.1	Conclusions	153
8.2	Suggestions for Future Work	158
A	Simplification of the MAP Post-Processor	161

B Criteria for Designing PRC Codes with $d = 2$ Constraint	163
C Interleaving	165
Samenvatting	169
Publications Originated from This Thesis	173
C.1 List of Papers	173
C.2 List of Patents	174
References	177
Curriculum Vitae	189

Glossary

List of Abbreviations

ACH	Adler-Coppersmith-Hassner
AWGN	Additive White Gaussian Noise
BD	Blu-ray Disc
BER	Bit Error Rate
BIS	Burst Indicator Subcode
ByER	Byte Error Rate
CD	Compact Disc
CIRC	Cross-Interleaved Reed-Solomn Code
DVD	Digital Versatile Disc
ECC	Error Correction Code
EFM	Eight-to-Fourteen Modulation
EFR	ECC Failure Rate
ETM	Eight-to-Twelve Modulation
FS	Fibonacci Sequence
FSM	Finite-State Machine
GB	Giga-Bytes
GFS	Generalized Fibonacci Sequence
GS	Guided Scrambling
HD-DVD	High-Definition Digital Versatile Disc
HDD	Hard Disk Drive
ISI	Inter-Symbol Interference
KB	Kilo-Bytes
LDC	Long Distance Code
MAP	Maximum <i>a Posteriori</i>
MB	Mega-Bytes
ML	Maximum Likelihood
MLSD	Maximum Likelihood Sequence Detector

MMSE	Minimum Mean Squared Error
MSE	Mean Squared Error
MTR	Maximum Transition Run
NA	Numerical Aperture
NC	Normal Constrained
NRZ	Non-Return to Zero
NRZI	Non-Return to Zero Inverse
PC	Parity-Check
PR	Partial Response
PRC	Parity-Related Constrained
PRML	Partial Response Maximum Likelihood
RMTR	Repeated Minimum Transition Runlength
ROM	Read-Only Media
RS	Reed-Solomn
RW	Rewritable Media
SBS	Symbol-by-Symbol
SNR	Signal-to-Noise Ratio
VD	Viterbi Detector

List of Symbols

a_k	constrained coded bits, with $a_k \in \{-1, +1\}$
\hat{a}_k	estimate of a_k
c_k	PR target output
C_R	capacity of constrained code
C_{pc}	capacity of constrained PC code
\mathbf{d}_k	channel input data pattern
e_k	total noise at channel detector input
\mathbf{e}	error event
$f(t)$	continuous-time channel impulse response
f_k	discrete-time channel impulse response
f_c	optical cut-off frequency
g_k	PR target
$h(t)$	continuous-time channel symbol response
h_k	discrete-time channel symbol response
k	discrete-time index
L_g	length of PR target
n_k	electronics noise

N_{pc}	number of PC codewords in each ECC codeword
p	number of parity bits
P_f	ECC failure rate
P_{If}	failure rate of interleaved ECC
P_{event}	error event probability
P_{ber}	bit error probability
$P_{ber.c}$	bit error probability with a given PC code
$Q(x)$	probability function of zero-mean unit-variance Gaussian tail (x, ∞)
q_k	detector input samples
$r(t)$	continuous-time readback (replay) signal at the channel output
r	total number of encoder states
r_1	number of encoder states of Type 1
r_2	number of encoder states of Type 2
R	code rate
SNR_u	user signal-to-noise ratio
T	channel bit period
T_u	user bit period
t	number of symbol errors that can be corrected in an ECC codeword
u_k	residual ISI channel coefficients
$U(\mathbf{e})$	probability of selecting an admissible data sequence that supports \mathbf{e}
w_k	equalizer coefficients
ε_k	random fluctuations in the reflectivity of the disc
$\tilde{\Delta}_\varepsilon$	normalized power of disc reflectivity fluctuations in the user bandwidth
γ_k	media noise
σ_u^2	variance of electronics noise in the user bandwidth
σ_n^2	variance of electronics noise in the channel bandwidth
Ω	frequency normalized by the channel bit rate
Ω_c	optical cut-off frequency normalized by the channel bit rate
$\tilde{\Omega}_u$	optical cut-off frequency normalized by the user bit rate, with ECC
Ω_u	optical cut-off frequency normalized by the user

α bit rate, without ECC
interleaving degree

List of Figures

1.1	Block diagram of digital optical recording system. . . .	4
2.1	Continuous-time model of optical recording channel with additive noise.	18
2.2	Illustration of NRZI to NRZ precoding.	19
2.3	Discrete-time model of optical recording channel with additive noise.	20
2.4	Continuous-time model of the channel with media noise, for an erased track.	26
2.5	Discrete-time counterpart of the channel model of Figure 2.4.	27
2.6	Discrete-time model of the channel with media noise and AWGN.	29
2.7	Channel SNR versus code rate for 20 dB user SNR. . .	31
2.8	Discrete-time channel symbol response h_k with (a) $\Omega_u = 0.5$; (b) $\Omega_u = 0.375$	32
3.1	Block diagram of $d = 1$ channel with parity-check (PC) code and post-processing. The channel noise is assumed to be AWGN.	37
3.2	State transition diagram for $d = 1$ codes in NRZ format.	41
3.3	Comparison of theoretical and simulated BER performances, without parity-check (PC) codes. (a) $\Omega_u = 0.5$; (b) $\Omega_u = 0.375$	44
3.4	Histogram of dominant error events obtained from theoretical analysis, without parity-check (PC) codes. (a) $\Omega_u = 0.5$; (b) $\Omega_u = 0.375$	46
3.5	A 2-level hierarchical parity-check (PC) coding scheme.	49

3.6	Block scheme of the receiver for parity-check (PC) coded channels with matched-filtering type multiple-error-event correction post-processor.	53
3.7	BER increase due to boundary error events for different block lengths (theoretical).	56
3.8	Analysis of boundary error events.	57
3.9	BER performance (theoretical) at Viterbi output, taking into account the code rate loss for different codeword lengths per parity bit. The ‘user SNR’ for each plot is shown in the brackets. (a) $\Omega_u = 0.5$; (b) $\Omega_u = 0.375$	59
3.10	BER performance at the post-processor output, as a function of codeword length per parity bit. The PC code corresponds to $g(x) = 1 + x + x^4$. The ‘user SNR’ for each plot is shown in the brackets. (a) $\Omega_u = 0.5$; (b) $\Omega_u = 0.375$	60
3.11	Comparison of BER bounds obtained from theory and simulations, with parity-check (PC) codes, $\Omega_u = 0.5$	62
3.12	BER performance of various parity-check (PC) codes in conjunction with rate $2/3$ code. (a) $\Omega_u = 0.5$; (b) $\Omega_u = 0.375$	63
4.1	Finite-state encoder.	70
4.2	Sliding-block encoder, with $v = 2$ and $u = 1$	71
5.1	Block diagram for encoding a constrained parity-check (PC) code in NRZI format.	95
5.2	Block diagram for encoding a constrained parity-check (PC) code in NRZ format.	100
5.3	BER performance with various codes, $\Omega_u = 0.5$	106
5.4	BER performance with various codes, $\Omega_u = 0.375$	107
6.1	Optical recording channel with electronics noise and media noise, and the PRML receiver.	112
6.2	BER performance comparison of various detection approaches, without parity-check (PC) codes, at $\Omega_u = 0.375$. (a) $\tilde{\Delta}_\varepsilon = 2\%$; (b) $\tilde{\Delta}_\varepsilon = 3\%$	117

6.3	Histogram of dominant error events, without parity-check (PC) codes, at $\Omega_u = 0.375$. (a) $\tilde{\Delta}_\varepsilon = 2\%$; (b) $\tilde{\Delta}_\varepsilon = 3\%$	119
6.4	BER performance of modified VD with parity-check (PC) code and different post-processors, at $\Omega_u = 0.375$. (a) $\tilde{\Delta}_\varepsilon = 2\%$; (b) $\tilde{\Delta}_\varepsilon = 3\%$	122
7.1	Block diagram of optical recording system with ECC and constrained parity-check (PC) code.	127
7.2	ByER performance with various constrained PC codes, $\tilde{\Omega}_u = 0.43$	128
7.3	Flow chart to compute $p_{j,t+1}$	131
7.4	Comparison of failure rate evaluation methods for non-interleaved ECC, with RS [248, 238] code and $\tilde{\Omega}_u = 0.39$	133
7.5	Comparison of failure rate evaluation methods for non-interleaved ECC, with RS [248, 216] code and $\tilde{\Omega}_u = 0.43$	134
7.6	Comparison of failure rate evaluation methods for interleaved ECC, with interleaved RS [248, 238] code, $\tilde{\Omega}_u = 0.39$, and user SNR = 14.5 dB.	136
7.7	Comparison of failure rate evaluation methods for interleaved ECC, with RS [248, 216] code, $\alpha = 5$, and $\tilde{\Omega}_u = 0.43$. (a) rate 9/13 code; (b) rate 277/406 code.	138
7.8	Failure rates of non-interleaved RS-ECC, with various constrained PC codes and $\tilde{\Omega}_u = 0.43$	139
7.9	Probabilities of byte errors at the input of RS-ECC (non-interleaved) decoder, with various constrained PC codes, $\tilde{\Omega}_u = 0.43$, and user SNR=15 dB.	140
7.10	Effect of interleaving on ECC failure rate. (a) user SNR = 14.5 dB; (b) user SNR = 15 dB.	141
7.11	Probabilities of byte errors within each interleave, with $\tilde{\Omega}_u = 0.43$ and user SNR=15 dB. (a) rate 9/13 code; (b) rate 135/198 code; (c) rate 277/406 code.	144
7.12	Failure rates of RS-ECC with different interleaving degrees, for various constrained PC codes and $\tilde{\Omega}_u = 0.43$. (a) $\alpha = 5$; (b) $\alpha = 17$; (c) $\alpha = 36$	146
7.13	Probabilities of byte errors within each interleave, with $\tilde{\Omega}_u = 0.43$ and user SNR=15 dB. (a) $\alpha = 5$; (b) $\alpha = 17$; (c) $\alpha = 36$	148

7.14	Failure rates of RS-ECC with optimum interleaving degrees, for various constrained PC codes and $\tilde{\Omega}_u = 0.43$.	149
C.1	A $(n \times \alpha)$ block interleaver and deinterleaver. (a) interleaver; (b) deinterleaver.	166

List of Tables

2.1	The generalized Braat-Hopkins model.	30
3.1	Comparison of theoretical and simulated error event probabilities at VD output.	45
3.2	Error detection capability of various CRC codes.	47
4.1	Basic sequences from the set of generalized Fibonacci sequences (GFS) $\{G^i(q)\}$, for $G^i(2) \leq 10$ and $0 \leq q \leq 11$	78
5.1	Distribution of codewords in the various encoder states for a rate 12/19 (1,18) parity-related constrained (PRC) code.	101
5.2	Distribution of codewords in the various encoder states for a rate 7/16 (1,18) parity-related constrained (PRC) code, Part I.	103
5.3	Distribution of codewords in the various encoder states for a rate 7/16 (1,18) parity-related constrained (PRC) code, Part II.	104
5.4	Summary of newly designed constrained PC codes.	105

Chapter 1

Introduction

In this chapter, we first give a brief introduction to optical data storage technology and its evolution. We then present a detailed overview of optical recording systems, from the coding and signal processing point of view. Thereafter, the motivation and scope of the work reported in this thesis are given. Finally, the contributions and organization of the thesis are presented.

1.1 Optical Recording Technology

The advent of the information age and the fast growth of information technology have created a tremendous demand for automated storage and retrieval of huge amounts of data. The data storage industry serves this need and is one of the most dynamic industries in the world.

Different types of media can be utilized for storage of digital data. The three main categories of storage approaches are magnetic recording, optical recording and solid state memory. The removable storage systems market is currently dominated by optical recording media. Compared with other storage technologies, the distinguishing features and key success factors of optical recording include aspects such as (1) removability of the media, (2) low-cost replicable read-only media (ROM), low-cost writable (R) and rewritable media (RW), and (3) long archival life and resistance to dust, debris and scratches.

Although optical recording dates back to the early seventies [22], the first generation optical recording system, the compact disc (CD)

system, was only launched in 1983 [78]. The capacity of a CD is 650 mega-bytes (MB) per disc. The successor of the CD standard, known as digital versatile disc (DVD), has enlarged the storage capacity to 4.7 giga-bytes (GB) [24]. The CD and DVD are based on lasers with wavelengths (λ) 780 nm and 650 nm, respectively, and objective lens with numerical apertures (NAs) 0.45 and 0.60, respectively. These wavelengths correspond to lasers in the red color range. Currently, two standards are competing to be the third generation optical recording system: the blu-ray disc (BD) [70, 17] and the high-definition digital versatile disc (HD-DVD) [45, 48]. Both standards use a blue laser with a wavelength of 405 nm. The BD format is based on a NA of 0.85 and a cover layer of 0.1 mm thickness. It achieves a capacity of 23.3, 25 or 27 GB on a single layer. The HD-DVD format is based on a NA of 0.65 and a cover layer of 0.6 mm thickness. It achieves a capacity of 15 GB for ROM and 20 GB for RW. Although the capacity of HD-DVD is lower than that of BD, it is less sensitive to dust and scratches compared with BD, due to the use of a thicker cover layer. Furthermore, the 0.6 mm cover layer fabrication process of HD-DVD is similar to the conventional DVD technology. This results in a significant reduction in fabrication cost. Meanwhile, intensive research is underway to develop high-density optical recording systems beyond the standard BD (or HD-DVD) [18, 89, 50].

There has been a steadily increasing demand for high-density and high-speed optical recording systems. The increase in capacity required from one generation to the next is mainly achieved by decreasing the wavelength, λ , of the laser and increasing the numerical aperture, NA, of the objective lens. Since the diameter of the laser spot is proportional to λ/NA , by making the optical spot smaller, the disc area illuminated by the spot becomes smaller. Therefore, the size of the recorded bits can be reduced accordingly to increase capacity. Although technological innovations in the design of optical media, objective lenses and lasers are key to achieving high-density and high-speed optical recording systems, the role of sophisticated coding and signal processing techniques for data recovery is increasingly becoming crucial in supporting and augmenting these advancements. In this thesis, we focus on coding and detection strategies for high-density blue laser disc systems. Furthermore, the developed algorithms can be easily generalized to CD and DVD systems as well.

1.2 Coding and Detection for Optical Recording

1.2.1 Optical Recording Systems

Digital recording systems can be considered as a type of digital communication systems in the sense that while communication systems transmit the information from one place to another, the recording systems transmit the information from one time to another. The principles of information and communication theory apply equally well to both these systems [90].

In optical recording, the data bits are imprinted onto the optical discs in the form of marks of various lengths. The primary electronic signal for data recovery is generated by scanning the disc surface by a beam of light from a semiconductor laser, and detecting the reflected light using a photodetector [5]. Various physical principles are used to induce variations in the reflected light beam. Read-only systems, such as CD-ROM, employ a pattern of pits and lands to write information on the disc. When the laser beam is focused on the disc, the pits, due to their low reflectivity, result in weak signals at the photodetector. The lands, however, have high reflectivity and hence result in strong signals at the photodetector. In this way, pits and lands can be distinguished and information can be read out from the disc. In rewritable systems, phase changes due to local differences in material structure are used to represent information [1]. The amorphous state has low reflectivity, while the crystalline state has high reflectivity. Therefore, these states can then be used in a manner analogous to the pit and land system.

The block diagram of a digital optical recording system is shown in Figure 1.1. It consists of three parts, namely, the write channel, recording channel (*i.e.* optical head/media), and read channel. The write channel and read channel behave like the transmitter and receiver, respectively, in a communication system. The write channel accepts the input binary data and converts it into a form suitable for writing onto the storage media. The read channel recovers the original data by processing the output of the read head, namely, the optical pick-up or light pen, in accordance with certain algorithms. The functions of each block shown in Figure 1.1 are briefly described below.

The user data is first passed through an error correction code

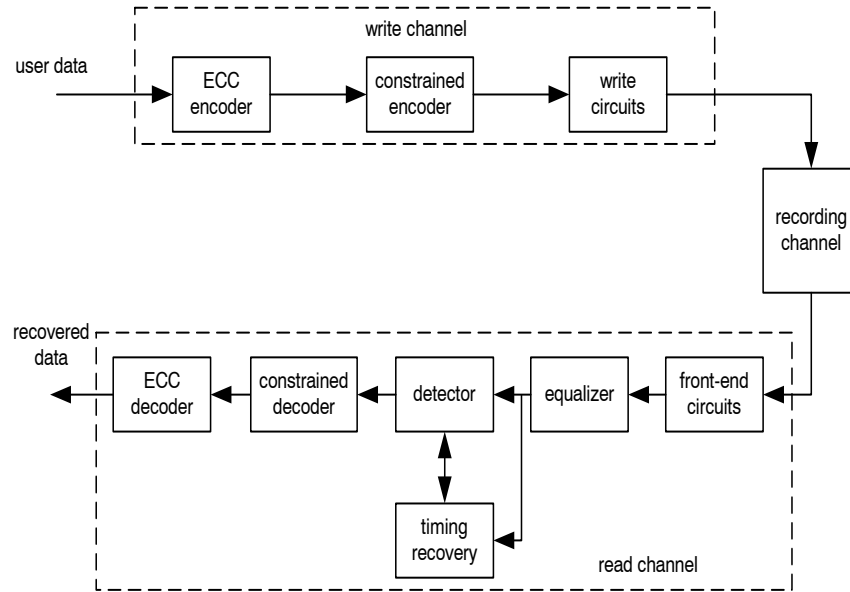


Figure 1.1: Block diagram of digital optical recording system.

(ECC) encoder, which adds extra symbols to the data stream. These extra symbols enable the ECC decoder to detect and correct some of the errors in the detected data stream. There are many different types of ECC codes [57, 101], among which the Reed-Solomon (RS) codes [84] are widely used in data storage systems. RS codes are non-binary linear block codes. They are often denoted RS $[n, k]$ with m -bit symbols. The encoder takes k information symbols and adds parity-check (PC) symbols to make a n -symbol codeword. The minimum distance d_{min} of a linear block code is the minimum weight (*i.e.* number of nonzero components) of its nonzero codewords. It is a measure of the error correction capability of the code. A linear block code can correct any combination of t symbol errors if and only if $d_{min} \geq 2t + 1$. For RS codes, $d_{min} = n - k + 1$. Therefore, RS codes correct up to t errors in a codeword, with $t = \frac{n-k}{2}$. For a symbol size m , the maximum codeword length is $n = 2^m - 1$. A technique known as “shortening” can produce a smaller code of any desired n from a larger code [57, 101].

Modulation codes [90, 37, 38], also known as constrained codes, on the other hand, are used to match the data to the recording channel characteristics, to improve the detector performance and to help in the operation of control loops (*e.g.* timing/gain loops) at the receiver. They are usually characterized by the so-called (d,k) constraints, or runlength constraints. Here, runlength refers to the length of time expressed in channel bits between consecutive transitions (*i.e.* changes in the state of the medium) along the recording track. The d constraint stipulates the minimum runlength to be $d + 1$, which helps to increase the minimum spacing between transitions in the data recorded on the medium (when $d > 0$). This, in turn, has a bearing on the linear and nonlinear interferences and distortions present in the readback signal. The k constraint stipulates the maximum runlength to be $k + 1$, which ensures adequate frequency of transitions for timing recovery. For optical recording, modulation codes often also need to have the dc-free property [81, 73, 19], *i.e.* they should have almost no content at very low frequencies. The dc-free constraint reduces interference between data and servo signals, and also facilitates filtering of low-frequency disk noise, such as finger marks on the disc surface.

Both ECC and modulation codes are referred to as channel codes. The major difference between them is that the ECC is concerned with how the different codewords relate to each other (*e.g.* in how many symbols must any two distinct codewords differ), while the modulation codes are concerned with properties of the individual codewords. The benefits of these codes are obtained at the cost of added redundancy in the codewords. The amount of redundancy is quantified by a parameter known as ‘code rate’. The code rate of a channel code is defined as

$$R = \frac{p}{q}, \quad (1.1)$$

specifying that a p -symbol information word at the encoder input is converted into a q -symbol channel codeword. Since only a limited number of q -symbol sequences can be used as channel codewords, the rate of a channel code is necessarily less than unity. Furthermore, the stricter the code constraints imposed (*e.g.* larger d constraint and smaller k constraint in modulation codes, or larger minimum distance, d_{min} , of ECC), the lower the code rate, and vice versa. The main disadvantages of a low code rate are the increase of noise bandwidth

and channel density, which lead to reduced signal-to-noise ratio (SNR) and poor performance (see Chapter 2 for details). Therefore, it is very important to design codes with the maximum possible code rate, while satisfying the required code constraints.

The write circuits convert the constrained coded data into a write-current waveform, which is used for writing the data on the storage medium. During the readout process, extraction of information is achieved through modulation of the reflected light emitted from a scanning laser spot. The photodetector inside the optical pick-up collects the reflected light and converts it to a replay signal. Although such a transformation is nonlinear in nature, the associated nonlinearities tend to be small. Therefore, optical readout can normally be modeled as a linear process [6], with sufficient accuracy.

The electrical signal generated by the photodetector is then processed by the front-end circuits which condition the replay signal (*e.g.* amplify, limit noise bandwidth, *etc.*) prior to equalization. The equalizer shapes the signal according to certain criteria so that the detector is able to recover the binary data from the equalized signal. The task of timing recovery is to recover an accurate sampling clock by extracting the timing information from the received signals and adjusting the receiver clock accordingly. The constrained decoder and ECC decoder operate on the output of the detector to provide an estimate of the original user data that was input to the recording system.

1.2.2 Imperfections in Optical Recording Channels

Retrieving the stored data from optical recording systems would be effortless if the output of the recording channel were an undistorted replica of the input. Unfortunately, readback signals are corrupted by various noises, interferences and nonlinear distortions, all of which increase with recording density. The major imperfections of optical recording channels are as follows.

Intersymbol Interference (ISI)

In optical recording, the bandwidth limitation of the system causes the channel symbol response to be of long duration. Therefore, responses due to successive bits interfere with each other, resulting in ISI [29]. Clearly, ISI increases with increase in recording density. However, this

interference is a deterministic function of the recorded data pattern, and may be accounted for accurately to any desired degree of precision in the reception strategy.

Noise Sources

Dominant noise sources in optical recording systems include the photodetector, preamplifier, laser, and storage medium [96, 28]. The resulting noises are, in general, mutually uncorrelated. The noises generated by the photodetector and preamplifier constitute the electronics noise, and can be modeled to a first-order approximation as additive white Gaussian (AWGN) random process. The fluctuations of the laser beam intensity causes shot noise, whose noise level has typically been found to be lower than that of electronics noise [6]. Media noise, which arises from irregularities and imperfections of the medium, is another major noise source that degrades the performance of high-density optical recording systems [79, 1]. Unlike electronics noise, media noise is correlated, data-dependent, non-stationary and non-additive in nature.

Asymmetry

As described earlier, the optical readback process is essentially linear. For read-only systems, the principal nonlinearities arise during the writing process, and are caused by differences in the effective sizes of pits and lands, as they are supposed to be of the same nominal size. This phenomenon is known as domain bloom or asymmetry [6, 82], and is due, among other factors, to under- or over-etching during the mastering process. Domain bloom causes asymmetry in the eye pattern of the replay signal. In CD and DVD systems, the use of modulation codes with $d = 2$ constraint [39, 40], which makes the minimum mark length to be 3 on the disc, helps to considerably reduce the impact of domain bloom on the replay signal. For rewritable systems, asymmetry is much less significant than for ROM systems [82]. This is because the rewritable systems contain a so-called write strategy [92], which has a fine control of the laser driver so that nonlinearities are small.

Disc Tilt and Cross-Talk

Disc tilt and cross-talk are two other major imperfections in optical recording systems [105, 75, 6]. Disc tilt can come from many sources, such as the mechanical misalignment, or the imperfect flatness of the disc. When the disc is tilted, the optical beam is distorted and optical aberrations appear [105], leading to degradation of the replay signal. In particular, tangential tilt distorts the laser beam in the tangential direction (*i.e.* along the track), and therefore increases ISI. Radial tilt distorts the laser beam in the radial direction (*i.e.* across the track), which results in increased interference to the adjacent tracks [105]. Cross-talk refers to the interference between replay signals of neighboring tracks, due to a small track pitch, radial tilt, defocus of laser spot, *etc.* Both disc tilt and cross-talk can be reduced effectively by using appropriate compensation techniques [105, 75]. In particular, tangential tilt can be well compensated by a sufficiently powerful adaptive equalizer, and cross-talk and radial tilt can be suppressed to a large extent by using a cross-talk canceller [105, 75].

Since asymmetry, disc tilt, and cross-talk can be effectively controlled/ compensated by advanced read/write strategies, in this thesis, we focus on the coding and detection techniques for optical recording channels corrupted by ISI, electronics noise, and media noise.

1.2.3 Overview of Coding and Detection Techniques

Detection Techniques

The detectors that have been used for data storage systems can be classified into two categories: symbol-by-symbol (SBS) detectors and sequence detectors [83]. SBS detectors map the channel output signals into binary detected bits, through a memoryless mapping. They require appropriate precoding and equalization schemes [97, 7, 8, 61]. Sequence detectors make a decision on a sequence of symbols based on observation of channel outputs over many symbol intervals. They can significantly outperform threshold detectors in combating noise and ISI, at the cost of a decision delay and high complexity. In particular, the maximum-likelihood sequence detector (MLSD) yields optimum detection in the presence of ISI [29]. When the channel noise is additive, white and Gaussian, MLSD can be implemented using the computationally efficient Viterbi algorithm [30]. We remark that by

changing the branch metric computation in the Viterbi detector (VD) in view of the data-dependent nature of media noise, the VD can be modified to combat media noise as well (see Section 6.2.2 for details).

The traditional detectors used for CD and DVD are SBS threshold detectors. A common reception scheme for CD includes a fixed prefilter for noise suppression, and a memoryless slicer for bit detection [93]. To achieve improved performance, the use of a nonlinear equalizer called “limit equalizer” [64] and post-processing to correct the dominant errors in the raw output of the threshold detector [44] have been proposed. These additional mechanisms equip the receiver with greater robustness to handle ISI and other artifacts, such as media noise. In the latest blue laser disc systems, the threshold detectors have given way to more powerful Viterbi-like sequence detection approaches [75]. The VD is invariably preceded by a partial response (PR) equalizer. This combination technique is referred to as partial response maximum-likelihood (PRML) detection [95, 14, 54, 55], and is well suited to combat the severe ISI at high recording densities.

PR equalization typically uses a linear filter to shape the original channel bit response into a predefined and short response that is referred to as the PR target [46]. Bit detection then involves a sequence detector (*e.g.* VD) that is matched to the PR target. The PR target should be chosen such that it gives a good spectral match to the unequalized channel bit response to minimize mis-equalization and noise enhancement. Furthermore, since noise must be white and Gaussian for the VD to be optimum, the target design should aim to limit noise correlation at the PR equalizer output. In addition, the PR target should be short enough to keep the detector complexity acceptable, since detector complexity grows exponentially with the length of the PR target.

A widely used method of designing PR target is to jointly optimize the target and equalizer based on a minimum mean square error (MMSE) criterion, which minimizes the total power of residual ISI and noise at the equalizer output [95, 14, 54]. To avoid trivial solutions, some constraint is imposed on the target [55, 65]. Among different investigated constraints, the monic constraint (*i.e.* first tap of the target should be unity) outperforms other constraints [65]. The advantage of monic constrained MMSE arises from its noise whitening capability, since it results in an equalizer that is equivalent to

the forward equalizer of the MMSE based decision feedback equalizer [8, 62, 61]. Yet another approach to whiten the correlated noise is to use a noise predictor at the output of the PR equalizer. This gives rise to noise-predictive maximum-likelihood (NPML) detection [21].

At high recording densities, media noise becomes dominant in optical recording systems. Due to the correlated, data dependant, and non-stationary nature of media noise, conventional channel detectors designed for AWGN are no longer optimum, and will have severe performance degradation. Although several approaches have been proposed to improve detection performance in the presence of media noise [65, 47, 66, 51] for magnetic recording systems, relatively little work has been done for optical recording.

Constrained Codes

Constrained codes have been widely and successfully applied in data storage systems. As introduced in Section 1.2.1, two major constraints for optical recording systems are runlength constraints (*i.e.* (d,k) constraints) and the dc-free constraint. In CD systems, an eight-to-fourteen modulation (EFM) code [39] is used, with $d = 2$ and $k = 10$ constraints. The EFM code is a block code which maps each 8-bit information word into a 14-bit codeword. The 14-bit codewords are cascaded using 3 merging bits to ensure that the runlength constraints continue to be satisfied when codewords are cascaded, and to achieve effective suppression of the low-frequency content in the channel bit-stream,. This reduces the code rate to 8/17. The strategy for designing modulation codes for DVD is more refined than the original EFM code, without changing the runlength constraints. In particular, the ‘Adler-Coppersmith-Hassner’ (ACH) algorithm, also known as *state-splitting algorithm* [2, 58], has been used in the code design, and the resulting so-called EFMPlus code [40] is a rate 8/16 sliding block code with four encoder states. The dc-control is performed using the surplus codewords that pertain to each encoder state.

In blue laser disc systems, the minimum runlength constraint has been reduced from $d = 2$ to $d = 1$, since the latter constraint permits a higher code rate. Furthermore, compared with $d = 2$ codes, $d = 1$ codes permit the use of larger channel bit length for the same user bit length. This provides better tolerance against writing jitter [70, 17]. For BD, the modulation code used is called 17PP code [70], which is

a rate $2/3$ variable-length code with $d = 1$ and $k = 7$ constraints, and with the parity-preserve property. The information word may have a length of 2, 4, 6 or 8 bits, and the corresponding codewords have a length of 3, 6, 9 and 12 bits, respectively. The parity-preserve property means that the *parity* of the information word is always equal to that of the corresponding codeword. Here, the parity, P , of a n -bit word (x_1, x_2, \dots, x_n) is defined by [38]

$$P = \sum_{i=1}^n x_i \bmod 2. \quad (1.2)$$

By multiplexing dc-control bits with user data, and by making use of the parity-preserve property of the code, the parity of the channel bit-stream can be controlled. This results in effective dc-control. For HD-DVD, the recently proposed modulation code, namely eight-to-twelve modulation (ETM) code, is a rate $8/12$, $d = 1$ and $k = 10$ code [48] with dc suppression. Furthermore, a repeated minimum transition runlength (RMTR) constraint has been adopted by both BD and HD-DVD, which limits the number of consecutive minimum runlengths to 6 with the 17PP code [70], and to 4 or 5 with the ETM code [48]. It has been found that the RMTR constraint can help to improve system tolerances, especially against tangential tilt [70], and to prevent the quasi-catastrophic error patterns that occur in certain PRML systems [48].

Error Correction Codes (ECC)

In data storage systems, without ECC, the bit error rate (BER) is around 10^{-4} , while the use of ECC brings this down to the order of 10^{-12} or less [85, 75]. In optical recording, random byte errors are mainly due to media defects, while burst errors are caused by contaminations on the disc surface, such as finger-prints and scratches. In CD systems, a cross-interleaved RS code (CIRC) [13] is used, which consists of two RS codes (a $[32, 28]$ code and a $[28, 24]$ code) separated by a cross-interleaver. The DVD system adopts a product code for error correction [24]. The user data is split into clusters of 32 kilo-bytes (KB), and each cluster is stored in a 192×172 matrix for encoding purpose. Each row of data is first encoded into a $[182, 172]$ RS code, which is designed for correcting random errors and for indicating the

locations of burst errors. Then the vertical code, being a [208, 192] RS code, uses erasure decoding to correct these bursts.

Compared to DVD, the spot size in blue laser disc systems (*e.g.* BD) is significantly small. This results in increased sensitivity to dust and scratches on the disc surface. A so-called picket code [70, 17] is the new error detection and correction method used for BD. The picket code consists of a long distance code (LDC) combined with a burst indicator subcode (BIS). The BIS is used for indicating the locations of bursts. An errors-and-erasures decoder of the LDC corrects these bursts together with random errors. The LDC has 304 [248, 216] RS codewords containing 64 KB user data, while the BIS has 24 [62, 30] RS codewords. Further, all the above RS codes are based on 8-bit symbols (*i.e.* bytes).

1.2.4 Performance Measures

The performance of coding and detection schemes for data storage systems can be assessed and quantified using the three criteria described below.

Bit Error Rate (BER)

Referring to Figure 1.1, the bit error rate (BER) is usually measured at the output of the channel detector. It reflects the frequency of occurrence of bit errors at the detector output. The BER is given by

$$\text{BER} = \frac{\text{number of error bits at channel detector output}}{\text{number of bits at constrained encoder output}}. \quad (1.3)$$

Byte Error Rate (ByER)

The byte (symbol) error rate (ByER) is measured at the output of the constrained decoder, and it reflects the average ECC byte (symbol) errors at the ECC decoder input. Compared to BER, the ByER further accounts for the impact of the inverse-precoder (see Section 2.1 for details) and the constrained decoder on the system performance. It is defined as

$$\text{ByER} = \frac{\text{number of error bytes at constrained decoder output}}{\text{number of bytes at constrained encoder input}}. \quad (1.4)$$

ECC Failure Rate (EFR)

The ECC failure rate (EFR) is the ultimate performance criterion of a data storage system. An ECC decoder (without interleaving) typically fails when the number of error symbols, N_s , that occur within one codeword is greater than the error correction capability, t , of the code. Therefore, EFR is defined as

$$\text{EFR} = \Pr\{N_s > t\}. \quad (1.5)$$

1.3 Motivation and Scope of the Current Work

The main features expected from a data storage system are high recording density, high data transfer rate, good reliability and low cost. However, as density increases, the readback signal suffers from increase in noise and linear/nonlinear distortions. This makes data recovery increasingly difficult, and requires increasingly powerful coding and signal processing techniques. At the same time, complexity of these techniques should be kept to a minimum to support high-speed and low-cost implementation of the system. The complexity issue is especially important for optical recording systems, whose major application lies in the consumer electronics area.

To improve the performance under aggressive recording conditions, currently, there is a great deal of activity exploring the viability of turbo coding and soft iterative decoding for future storage products [91, 63, 56]. Iteratively decodable codes, such as turbo codes or low density parity-check (LDPC) codes, are promising soft ECC schemes with the potential to approach channel capacity. However, due to their extremely high implementation complexity and large latency, these schemes have so far not found their way to commercial systems.

In recent years, there has also been a lot of interest in constrained codes that have error correction properties [23, 27, 15]. The reason is that unlike the situation in conventional read channels, where ECC is expected to correct all the errors that remain at the output of the constrained decoder, dominant short error events can be corrected by the application of low-redundancy PC codes in conjunction with a post-processor. This reduces the loss in error correction capability of ECC due to random short errors and results in a simple and efficient solution to improve the overall performance. Compared with

iteratively decodable codes, the PC-code-based approach has found wide acceptance since the performance-complexity trade-off offered by these codes is very attractive and affordable. This approach has been widely studied for magnetic recording channels [23, 27, 86, 49]. However, relatively little work has been done for optical recording channels [20]. Therefore, the focus of this thesis is to develop PC-code-based recording systems to achieve higher recording capacity with low implementation complexity, for high-density blue laser disc systems such as the advanced versions of BD or HD-DVD systems.

Since optical recording systems have completely different channel characteristics, code constraints, and media noise and other linear/nonlinear distortions compared to magnetic recording systems, different PC coding and post-processing schemes need to be developed. In this work, we make a thorough study on development and analysis of PC-code-based optical recording systems. In particular, most of the key constituents of the system shown in Figure 1.1 are designed and optimized in this thesis for optical recording channels with different recording densities and different proportions of white noise and media noise. Our study comprises the following steps.

- Channel model: From the conventional Braat-Hopkins model to a generalized Braat-Hopkins model, to facilitate fair comparison for different coding schemes and recording densities in a common framework;
- Constrained codes: From the rate $2/3$ $d = 1$ code, whose code rate equals that of the standard codes for BD and HD-DVD, to capacity-approaching $d = 1$ codes, to reduce the code rate loss as well as the encoder/decoder complexity;
- Constrained PC codes: From modeling the PC codes by assuming the parity-check result for each channel data segment to be known at the receiver (*i.e.* computing the syndrome in a data-aided mode without constructing explicit constrained PC encoder), to designing capacity-approaching combined constrained PC codes, to provide a general and systematic way to efficiently combine constrained codes with PC codes;
- Noise: From a channel with AWGN only to a channel with both media noise and AWGN, to develop novel channel detection, PC

coding and post-processing scheme to combat media noise, which is correlated, data dependant, and non-stationary in nature;

- Recording density: From nominal density to high density, to evaluate the system's performance over different recording densities;
- Performance criteria: From BER at the channel detector output, to ByER at the constrained decoder output, and to EFR at the ECC decoder output, to provide various assessments of the PC-code-based systems.

1.4 Contributions of the Thesis

The thesis makes contributions and reports original work on PC-code-based optical recording systems. The main contributions include six parts.

- The first part proposes a generalized Braat-Hopkins model for optical recording channels, which provides a fair basis for the performance comparison of detection schemes over different coding schemes and recording densities, for channels with white Gaussian noise and media noise.
- The second part investigates the performance bounds for bit error rates and error event probabilities, different PC codes, simple and efficient post-processors, and impact of error events that are split across data block boundaries and the corresponding remedy.
- The third part derives analytically the relationship between the number of encoder states and the maximum size of certain capacity-approaching constrained codes, and obtains the minimum number of encoder states.
- The fourth part proposes a general and systematic code design technique for constructing capacity-approaching constrained PC codes, which can detect any dominant error event or error event combinations.
- The fifth part proposes novel detection, PC coding and post-processing schemes to combat media noise.

- The sixth part analyzes the failure rates of ECC for the developed PC-code-based systems.

1.5 Organization of the Thesis

The rest of the thesis is organized as follows.

In Chapter 2, we propose a generalized Braat-Hopkins model for optical recording channels, which facilitates fair comparison for different coding schemes and recording densities in a common framework. The investigation on various basic issues associated with PC-code-based systems is presented in Chapter 3. In Chapter 4, we analytically investigate the relationship between the number of encoder states and the maximum size of certain capacity-approaching constrained codes, and obtain the corresponding minimum number of encoder states. A general and systematic code design technique to efficiently combine constrained codes with PC codes is proposed in Chapter 5. In Chapter 6, we propose two novel modifications to the channel detector to combat media noise. We also develop a PC coding and a data-dependent post-processing scheme for error detection and correction. In Chapter 7, we develop semi-analytical approaches for estimating the failure rates of ECC, and analyze the EFR for the developed PC-code-based systems. Finally, in Chapter 8, we present a summary of the thesis. Future research issues are also mentioned in this chapter.

Chapter 2

Channel Model

In this chapter, a generalized Braat-Hopkins model is proposed for read-only and re-writable optical recording channels, and it is used throughout this thesis. In this model, the quantities signal-to-noise ratio (SNR), optical cut-off frequency, and media noise are defined with respect to user bit rate. We also show how these quantities can be used to arrive at various parameters of the channel (*e.g.* noise variance, symbol response) for given code rate and cut-off frequency. In Section 2.1, we start with an overview of the conventional Braat-Hopkins model. The generalized model specification for channels with white noise is developed in Section 2.2 and that for channels with white noise and media noise is developed in Section 2.3. In Section 2.4, we present a summary of the generalized model and illustrate the effect of code rate on certain key quantities in the channel model. Finally, concluding remarks are given in Section 2.5.

2.1 Conventional Braat-Hopkins Model

Future optical recording channels are likely to be severely hampered by electronics noise and media noise. During the development of reception schemes to ensure satisfactory data detection performance on such channels, it becomes necessary to investigate the system's performance with different coding schemes and recording densities. The purpose of this chapter is to develop a generalized description of the recording channel model that will facilitate fair comparison for different coding schemes and densities in a common framework, for channels

with additive, white and Gaussian noise (AWGN) and media noise. During the development, we assume that there are no timing uncertainties due to clock drift or channel delay, so that no timing recovery is necessary. This simplifies the study.

Figure 2.1 shows the basic continuous-time model of the optical recording channel. The user bits, at the rate $1/T_u$ bits/second, are coded by a channel encoder, which in general, consists of an ECC encoder and a constrained encoder. There are two formats to denote the channel bit-stream, namely, the non-return-to-zero-inverse (NRZI) and non-return-to-zero (NRZ) formats. In NRZI format, whereas a change in the state of the medium corresponds to a channel bit ‘1’, no change corresponds to a channel bit ‘0’. In NRZ format, whereas one state of the medium corresponds to a channel bit ‘1’, the other state corresponds to a channel bit ‘0’. Usually, the input data is first encoded by the constrained encoder using NRZI format [38] and then transformed into NRZ format before being sent to the write circuit. This transformation is known as ‘NRZI to NRZ precoding’, which is illustrated by Figure 2.2. The constrained coded NRZI sequence $\{b_k\}$ is first passed through a precoder whose transfer function is given by $1/(1 \oplus D)$, where ‘ \oplus ’ is the Boolean XOR operator and ‘ D ’ is the 1-bit delay operator. Therefore, the precoder output sequence $\{\hat{a}_k\}$ in NRZ format is given by

$$\hat{a}_k = b_k \oplus \hat{a}_{k-1}. \quad (2.1)$$

By assigning $+1$ to ‘1’ and -1 to ‘0’ in $\{\hat{a}_k\}$, we obtain the sequence $\{a_k\}$, which indicates the polarity of the write current.

The resulting channel bits $a_k \in \{-1, +1\}$ are at the rate $1/T$, where $T = RT_u$, with R being the code rate of the channel code. A

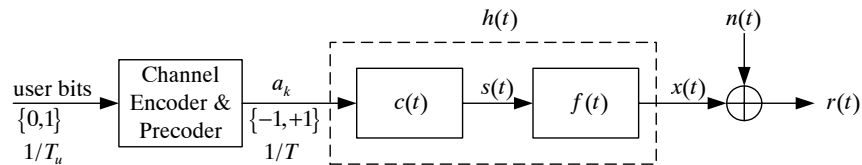


Figure 2.1: *Continuous-time model of optical recording channel with additive noise.*

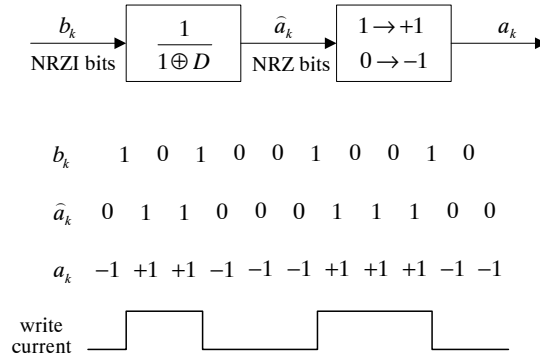


Figure 2.2: *Illustration of NRZI to NRZ precoding.*

linear pulse modulator transforms the channel coded data sequence a_k into a binary write signal $s(t)$. The impulse response $c(t)$ of the linear pulse modulator is given by

$$c(t) = \begin{cases} 1, & |t/T| < 0.5, \\ 0, & \text{else,} \end{cases} \quad (2.2)$$

and in frequency domain,

$$C(\Omega) = T \frac{\sin(\pi\Omega)}{\pi\Omega}, \quad (2.3)$$

where Ω is the frequency normalized by the channel bit rate $1/T$.

The replay signal $r(t)$ is assumed to be a linearly filtered and noisy version of the write signal according to

$$r(t) = (s \otimes f)(t) + n(t). \quad (2.4)$$

Here, $f(t)$ denotes the impulse response of the channel, ' \otimes ' denotes convolution, and $n(t)$ is the additive noise, which is assumed to be white Gaussian with power spectrum density (two sided) $N_0/2$. Media noise is ignored in this section and will be considered in Section 2.3.

The Braat-Hopkins model [11] is a linear model for the channel impulse response $f(t)$, applicable for read-only and re-writable optical channels. According to this model, the Fourier transform of $f(t)$ is

given by

$$F(\Omega) = \begin{cases} \frac{2}{\pi} \left[\arccos\left(\left|\frac{\Omega}{\Omega_c}\right|\right) - \left|\frac{\Omega}{\Omega_c}\right| \sqrt{1 - \left(\frac{\Omega}{\Omega_c}\right)^2} \right], & \left|\frac{\Omega}{\Omega_c}\right| < 1, \\ 0, & \left|\frac{\Omega}{\Omega_c}\right| \geq 1. \end{cases} \quad (2.5)$$

The quantity $\Omega_c = f_c T$, which is the optical cut-off frequency (f_c) normalized to the channel bit rate, is a measure of the recording density. The smaller the cut-off frequency is, the higher is the recording density, and vice versa. For an optical recording system using a laser diode with wavelength λ and a lens with numerical aperture NA, the normalized cut-off frequency is given by $\Omega_c = \frac{2NA}{\lambda} L_c$, where $L_c = \nu T$ is the physical length of one channel bit, and ν is the media velocity. For BD systems, with $\lambda=405$ nm, NA=0.85 and $L_c=75$ nm, we get $\Omega_c \approx 0.315$.

The channel symbol response $h(t)$ is obtained by convolving $f(t)$ with $c(t)$. In frequency domain, we have

$$H(\Omega) = \begin{cases} \frac{2T}{\pi} \frac{\sin(\pi\Omega)}{\pi\Omega} \left[\arccos\left(\left|\frac{\Omega}{\Omega_c}\right|\right) - \left|\frac{\Omega}{\Omega_c}\right| \sqrt{1 - \left(\frac{\Omega}{\Omega_c}\right)^2} \right], & \left|\frac{\Omega}{\Omega_c}\right| < 1, \\ 0, & \left|\frac{\Omega}{\Omega_c}\right| \geq 1. \end{cases} \quad (2.6)$$

Because $h(t)$ is bandlimited to well within $[-1/(2T), 1/(2T)]$ for recording densities of practical interest, we can replace the continuous-time model in Figure 2.1 with the discrete-time model in Figure 2.3. In Figure 2.3, h_k is the sampled version of $h(t)$, n_k is a discrete-time white Gaussian noise with power spectrum density (two sided) $N_0/2$, and r_k denotes the readback signal samples, all at the rate of $1/T$ samples/second. Equivalently, r_k can be obtained by passing the

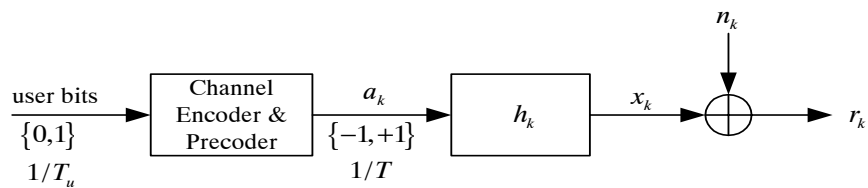


Figure 2.3: Discrete-time model of optical recording channel with additive noise.

continuous-time readback signal $r(t)$ through an ideal low-pass filter with cut-off frequency $1/(2T)$ and sampling the output at the rate $1/T$ samples/second.

Conventionally, the SNR is defined in the sense of the matched-filter bound [83]. That is,

$$\text{SNR(dB)} = 10 \log_{10} \left(\frac{\sum h_k^2}{\sigma_n^2} \right), \quad (2.7)$$

where $\sum h_k^2$ is the energy in the symbol response h_k and σ_n^2 is the noise variance in the channel bandwidth $1/T$, given by

$$\sigma_n^2 = \frac{N_0}{2T}. \quad (2.8)$$

In summary, in the conventional Braat-Hopkins model, quantities such as SNR and optical cut-off frequency are all defined with respect to channel bit rate. Because the channel bit rate is not independent of the code rate, these quantities are not able to explicitly bring out the impact of different code rates on the system performance, which is extremely important for the design and performance evaluation of different coding schemes.

In the following sections, a generalized Braat-Hopkins model is developed. In this model, we consider the user as an absolute point of reference and define the quantities SNR, optical cut-off frequency, and media noise with respect to user bit rate rather than channel bit rate. Because the user bit rate does not include the effect of code rate, defining the above quantities with respect to user bit rate is preferable when we need to study the performance of various coding and detection schemes. In addition, we also show how these user-defined quantities can be used to arrive at various parameters of the channel (*e.g.* noise variance, symbol response).

2.2 Generalized Model for Channel with White Noise

Optical cut-off frequency: In the generalized model, the optical cut-off frequency is specified in terms of the user bit rate. That is, the normalized cut-off frequency is given by $\tilde{\Omega}_u = f_c T_u$. Thus, $\tilde{\Omega}_u$ is an

indication of the recording density at the user side. The smaller the value of $\tilde{\Omega}_u$ is, the higher is the recording density, and vice versa. At the channel side, $T = RT_u$ and $\Omega_c = R\tilde{\Omega}_u$, with R being net code rate arising from all the channel codes, *i.e.* constrained codes, parity-check (PC) codes, and ECC. In this way, we keep the information density (in terms of $\tilde{\Omega}_u$) the same, and adjust the channel bit density (in terms of Ω_c) to account for the influence of different code rates.

ECC is always present in data storage systems (see Figure 1.1). However, when it is not considered during the development of coding and detection schemes, a common practice, for the sake of convenience, is to ignore its code rate while defining the user density [85, 62, 69]. In Chapters 3 to 6, our study does not include ECC, and hence we do not include the code rate of ECC while defining R and user density. The resulting user normalized optical cut-off frequency without ECC consideration (*i.e.* the normalized frequency defined at the input of the constrained encoder) is denoted by Ω_u . In Chapter 7, the ECC has been included in the study, and hence its code rate has been taken into account while defining R and $\tilde{\Omega}_u$. Similar definitions can be found in [85].

According to the current standards of BD [70], at the channel side, cut-off frequencies $\Omega_c = 0.33$ and $\Omega_c = 0.25$ represent recording systems with nominal density and high density, respectively. Considering only the standard rate $2/3$ $d = 1$ constrained code without ECC, the corresponding user normalized cut-off frequencies are given by $\Omega_u = 0.5$ and $\Omega_u = 0.375$, respectively. They are used in the study of Chapters 3 to 6. In Chapter 7, an ECC with code rate R_{ECC} is included in the study and hence we obtain $\tilde{\Omega}_u = 0.5/R_{\text{ECC}}$ and $\tilde{\Omega}_u = 0.375/R_{\text{ECC}}$, for the nominal density and high density, respectively.

Impulse response: According to the Braat-Hopkins model, the Fourier transform of the channel impulse response, with $\Omega_c = R\tilde{\Omega}_u$, is given by

$$F(\Omega) = \begin{cases} \frac{2}{\pi} \left[\arccos\left(\left|\frac{\Omega}{R\tilde{\Omega}_u}\right|\right) - \left|\frac{\Omega}{R\tilde{\Omega}_u}\right| \sqrt{1 - \left(\frac{\Omega}{R\tilde{\Omega}_u}\right)^2} \right], & \left|\frac{\Omega}{R\tilde{\Omega}_u}\right| < 1, \\ 0, & \left|\frac{\Omega}{R\tilde{\Omega}_u}\right| \geq 1. \end{cases} \quad (2.9)$$

The channel impulse response can be obtained as

$$\begin{aligned} f(t) &= \frac{1}{RT_u} \int_{-R\tilde{\Omega}_u}^{R\tilde{\Omega}_u} F(\Omega) e^{j2\pi\Omega t/RT_u} d\Omega \\ &= \frac{2}{RT_u} \int_0^{R\tilde{\Omega}_u} F(\Omega) \cos(2\pi\Omega t/RT_u) d\Omega, \end{aligned} \quad (2.10)$$

since $F(\Omega)$ is an even symmetric function. Substituting for $F(\Omega)$ from (2.9) in (2.10) and with $\omega_c = 2\pi f_c = 2\pi\tilde{\Omega}_u/T_u$, we get

$$\begin{aligned} f(t) &= \frac{2\omega_c}{\pi^2} \int_0^1 [\arccos(x) - x\sqrt{1-x^2}] \cos(\omega_c t x) dx \\ &= \frac{4}{\pi^2 t} \int_0^1 \sqrt{1-x^2} \sin(\omega_c t x) dx \quad \text{for } t \neq 0 \end{aligned} \quad (2.11)$$

$$\text{with } f(0) = \frac{4\omega_c}{\pi^2} \int_0^1 x\sqrt{1-x^2} dx = \frac{8\tilde{\Omega}_u}{3\pi T_u} = \frac{8f_c}{3\pi}. \quad (2.12)$$

Sampling $f(t)$ at $t = kT = kRT_u$, we get the discrete-time channel impulse response f_k as

$$f_k = \frac{4}{\pi^2 RT_u k} \int_0^1 \sqrt{1-x^2} \sin(2\pi\tilde{\Omega}_u Rkx) dx \quad \text{for } k \neq 0 \quad (2.13)$$

$$\text{with } f_0 = f(0) = \frac{8\tilde{\Omega}_u}{3\pi T_u} = \frac{8f_c}{3\pi}. \quad (2.14)$$

The impulse response is obtained by numerically evaluating the integral in (2.11) or (2.13), since it does not have a closed-form expression.

Transition response: If the recorded data contains an isolated transition, when the laser spot scans the leading edge of a recorded domain, the replay signal exhibits an upgoing transition $g(t)$ that is the transition response of the channel. Noting that the isolated transition response $g(t)$ of the channel is two times the step response, we get from (2.11)

$$\begin{aligned} g(t) &= 2 \int_{-\infty}^t f(y) dy \\ &= \frac{8}{\pi^2} \int_{-\infty}^t \int_0^1 \sqrt{1-x^2} \frac{\sin(\omega_u y x)}{y} dx dy \end{aligned} \quad (2.15)$$

which can be evaluated numerically.

Symbol response: Fourier transform of the channel symbol response is given by

$$H(\Omega) = C(\Omega)F(\Omega) = \begin{cases} \frac{2RT_u}{\pi} \frac{\sin(\pi\Omega)}{\pi\Omega} \left[\arccos\left(\left|\frac{\Omega}{R\tilde{\Omega}_u}\right|\right) - \left|\frac{\Omega}{R\tilde{\Omega}_u}\right| \sqrt{1 - \left(\frac{\Omega}{R\tilde{\Omega}_u}\right)^2} \right], & \left|\frac{\Omega}{R\tilde{\Omega}_u}\right| < 1, \\ 0, & \left|\frac{\Omega}{R\tilde{\Omega}_u}\right| \geq 1. \end{cases}$$

Therefore, we can obtain the channel symbol response as

$$h(t) = \frac{1}{RT_u} \int_{-R\tilde{\Omega}_u}^{R\tilde{\Omega}_u} H(\Omega) e^{j2\pi\Omega t/T} d\Omega, \quad (2.16)$$

$$h_k = h(t)|_{t=kT} = \frac{1}{RT_u} \int_{-R\tilde{\Omega}_u}^{R\tilde{\Omega}_u} H(\Omega) e^{j2\pi\Omega k} d\Omega. \quad (2.17)$$

As before, we have to resort to a numerical approach to evaluate (2.16) or (2.17).

User SNR: To develop a SNR that is independent of the code rate, we define a ‘user SNR’ as the channel SNR for the case of $R = 1$. Thus, we get

$$\text{SNR}_u(\text{dB}) = 10 \log_{10} \left(\frac{\sum h_{ku}^2}{\sigma_u^2} \right), \quad (2.18)$$

where

$$\sigma_u^2 = \frac{N_0}{2T_u} \quad (2.19)$$

is the white noise power in the user bandwidth $1/T_u$ and h_{ku} is the channel symbol response for $R = 1$, resulting in

$$h_{ku} = \frac{1}{T_u} \int_{-\tilde{\Omega}_u}^{\tilde{\Omega}_u} H(\Omega) e^{j2\pi\Omega k} d\Omega. \quad (2.20)$$

When studying the performance over different user densities, the reference signal power in the user SNR needs to be independent of density. For this, h_{ku} in (2.20) is evaluated for a particular user density, *e.g.* $\tilde{\Omega}_u = 0.33$, which is independent of the densities at which the channel and receiver are tested.

At the channel side, due to coding, the system bandwidth increases by a factor of $1/R$. Thus, the noise variance in the channel bandwidth is given by (see (2.8), (2.19))

$$\sigma_n^2 = \frac{1}{R}\sigma_u^2. \quad (2.21)$$

2.3 Generalized Model for Channel with Media Noise

2.3.1 Media Noise in Rewritable Optical Recording Systems

The media noise [96, 79, 71, 94] is a physical property of the storage media, and hence it exhibits different characteristics for different systems. For example, in magnetic recording systems, the media noise is mainly caused by transition jitter, which causes the edges of magnetic transitions to be randomly displaced [71, 74].

In rewritable optical recording systems, the active layer of the disc is first initialized to the crystalline phase. During the writing process, a high power laser beam heats the material locally and makes the material melt and lose its polycrystalline structure. When the material solidifies, amorphous marks are formed [79, 1]. Previously written amorphous marks are erased by heating the material to a temperature between crystallization and melting temperature using a medium power laser. This reverts the material back from the amorphous state to the crystalline state, and enables direct overwrite. The noise in this type of medium arise from roughness of medium surface and groove profiles, graininess of the polycrystalline recording layer, substrate material, and thin-film coatings. In particular, the media noise is mainly caused by scattering of light at grain boundaries and variation of material refractive index due to the random orientation of crystallites. Experimental study shows that these noises cause fluctuations in the reflectivity of the disc [79, 1], which further lead to random fluctuations in the amplitude of the replay signal (or photodetector voltage).

Moreover, the media noise in the crystalline state has been found to be significantly higher than that in the amorphous state [79, 1]. This is due to the fact that the polycrystalline grain noise is absent in the amorphous marks, and the influence of crystal orientation variations on the disc reflectivity is more significant in the crystalline state

than that in the amorphous state [79]. Therefore, the media noise for rewritable systems is mainly caused by random fluctuations in the amplitude of the replay signal from the crystalline state. The fluctuations of the replay signal from the amorphous state are relatively weak, and hence can be ignored. In this thesis, we focus on combating this type of media noise. In the study, without loss of generality, we associate the crystalline state with $a_k = +1$ and the amorphous state with $a_k = -1$.

2.3.2 Channel Modeling for Media Noise

For simplicity we first consider only the erased track, which corresponds to the all ‘+1’ data sequence. We will then extend the study to the recorded track (*i.e.* arbitrary data sequence). Figure 2.4 depicts the continuous-time channel model with media noise, for an erased track. For an all-polycrystalline track, the continuous-time noise caused by fluctuations of the reflectivity of the disc can be represented as a stationary random signal $u(t)$. We assume that $u(t)$ is white Gaussian with power spectrum density U_0 . The value of U_0 depends on various properties of the laser and system, such as the laser wavelength λ , the numerical aperture NA, the effective incident laser power, the optical efficiency of the system, as well as the sensitivity of the photodetector [79]. The noise component $y(t)$ of the channel readback signal $z(t)$ is obtained by filtering $u(t)$ with the impulse response $f(t)$ of the channel. This is because media noise is generated at the disc and read circuit side, rather than the write circuit side. Note

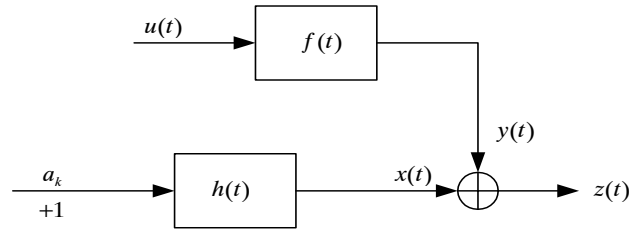


Figure 2.4: *Continuous-time model of the channel with media noise, for an erased track.*

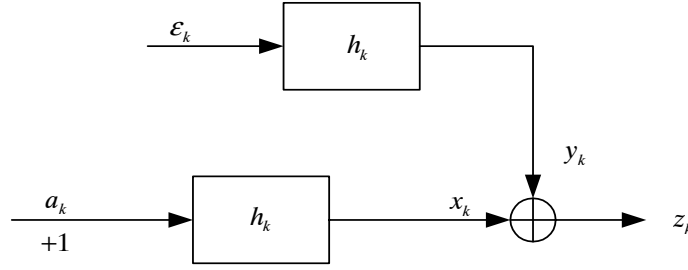


Figure 2.5: Discrete-time counterpart of the channel model of Figure 2.4.

that for the all ‘+1’ data sequence, $y(t)$ is stationary. Furthermore, for given media and system, $y(t)$ is independent of the channel bit interval T .

Figure 2.5 shows the discrete-time counterpart of the channel model of Figure 2.4. It is obtained as follows. As mentioned in Section 2.1, we can convert $x(t)$ into a discrete-time sequence x_k of data rate $1/T$, as depicted by the lower path of Figure 2.5. Furthermore, the Fourier transform of h_k is given by

$$H_d(e^{j2\pi\Omega}) = \frac{1}{T} \sum_{n=-\infty}^{n=+\infty} C(\Omega + n)F(\Omega + n). \quad (2.22)$$

In the absence of excess bandwidth [6], (2.22) simplifies to

$$H_d(e^{j2\pi\Omega}) = \frac{1}{T}C(\Omega)F(\Omega), \quad \text{for } |\Omega| \leq 0.5. \quad (2.23)$$

For recording densities of practical interest, $F(\Omega)$ has no excess bandwidth, and $C(\Omega) \approx T$ for Ω such that $F(\Omega)$ is significant. Hence we have the approximation

$$H_d(e^{j2\pi\Omega}) \approx F(\Omega). \quad (2.24)$$

Similarly, by sampling $y(t)$ at rate $1/T$, we obtain the sampled noise sequence y_k , whose power spectrum density is given by

$$P_d(e^{j2\pi\Omega}) = \frac{1}{T} \sum_{n=-\infty}^{n=+\infty} U_0|F(\Omega + n)|^2. \quad (2.25)$$

In the case of no excess bandwidth, (2.25) simplifies to

$$P_d(e^{j2\pi\Omega}) = \frac{U_0}{T} |F(\Omega)|^2, \quad \text{for } |\Omega| \leq 0.5. \quad (2.26)$$

In the discrete-time channel model, it is desirable to model y_k by passing a white Gaussian discrete-time random sequence ε_k through a linear filter q_k . In that case, y_k has power spectrum density

$$P_d(e^{j2\pi\Omega}) = \sigma_\varepsilon^2 |Q_d(e^{j2\pi\Omega})|^2, \quad (2.27)$$

where σ_ε^2 is the variance of ε_k , and $Q_d(e^{j2\pi\Omega})$ is the Fourier transform of q_k . A logical way to equate (2.26) and (2.27) is to choose

$$\sigma_\varepsilon^2 = \frac{U_0}{T}, \quad \text{and} \quad Q_d(e^{j2\pi\Omega}) = F(\Omega). \quad (2.28)$$

In view of (2.24), we obtain the approximate equivalent discrete-time model for the noise part, depicted by the upper path of Figure 2.5.

Let us now consider the recorded track, which corresponds to an arbitrary data sequence. Since media noise is mainly caused by fluctuations in the reflectivity of the crystalline marks (*i.e.* ‘+1’s), and no such fluctuations arise from the amorphous marks (*i.e.* ‘-1’s), we can extend the above derived model to arbitrary data. The resulting discrete-time channel model with media noise and AWGN is shown in Figure 2.6. As illustrated by Figure 2.6, we model the fluctuations in the reflectivity of the disc as a white Gaussian random process $\{\varepsilon_k\}$ with variance σ_ε^2 . We model the data-dependent nature of the media noise by multiplying ε_k with $\frac{1+a_k}{2}$. That is, the input data a_k is additively corrupted by ε_k only when $a_k = +1$, which corresponds to the crystalline state. The readback signal can then be written as

$$\begin{aligned} r_k &= \sum_{i=-\infty}^{\infty} \left(a_i + \frac{1+a_i}{2} \varepsilon_i \right) h_{k-i} + n_k \\ &= \sum_{i=-\infty}^{\infty} a_i h_{k-i} + \sum_{i=-\infty}^{\infty} m_i h_{k-i} + n_k, \end{aligned} \quad (2.29)$$

with $m_k = \frac{1+a_k}{2} \varepsilon_k$. Therefore, the media noise results in an additive signal-dependent noise component $\sum_{i=-\infty}^{\infty} m_i h_{k-i}$.

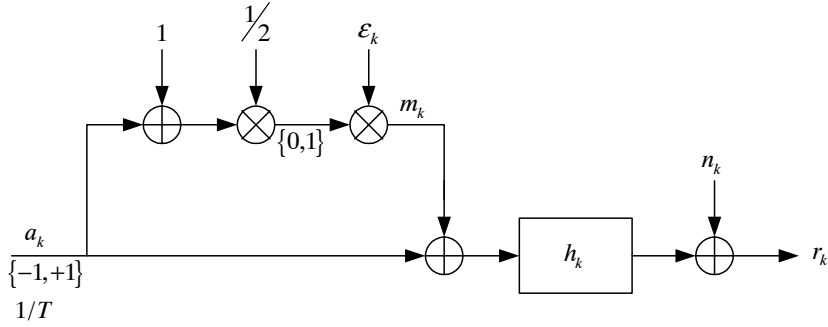


Figure 2.6: Discrete-time model of the channel with media noise and AWGN.

The variance σ_ε^2 of ε_k , according to (2.28), is given by $\sigma_\varepsilon^2 = U_0/T$ at the channel side. At the user side, the corresponding noise power in the user bandwidth is given by $\tilde{\sigma}_\varepsilon^2 = U_0/T_u$. With $T = RT_u$, we get $\sigma_\varepsilon^2 = \tilde{\sigma}_\varepsilon^2/R$. Moreover, we wish to further normalize $\tilde{\sigma}_\varepsilon^2$ into a dimensionless quantity. For simplicity, we choose to specify $\tilde{\sigma}_\varepsilon^2$ with respect to the power of the channel input data, *i.e.* to $E[a_k^2] = 1$. We thus get

$$\tilde{\Delta}_\varepsilon = \frac{\tilde{\sigma}_\varepsilon^2}{E[a_k^2]} = \tilde{\sigma}_\varepsilon^2. \quad (2.30)$$

At the channel side, we have $\Delta_\varepsilon = \frac{\sigma_\varepsilon^2}{E[a_k^2]} = \tilde{\Delta}_\varepsilon/R$.

For the channel with both media noise and AWGN, we still use the SNR defined by (2.18) to (2.21). Thus, the SNR definition includes the electronics noise only. The influence of media noise is indicated separately by the quantity $\tilde{\Delta}_\varepsilon$. Similar definitions can be found in [51]. The SNR definition we adopted here is unlike that proposed in [66, 67], where the total noise power is defined as the sum of the powers of the electronics noise and media noise. We do this because of the following. Since these two noises have different characteristics, they introduce different effects on the system performance. For example, the performance with a noise composition 10:90 may be quite different from that with a composition of 90:10. Therefore, defining SNR using the sum of the powers of the two noises may lead to inconsistent results.

2.4 Model Summary and Numerical Results

In the generalized Braat-Hopkins model, the channel is completely defined by specifying the SNR, optical cut-off frequency and media noise with respect to user bit rate, and the code rate. Table 2.1 summarizes the generalized model, showing how to obtain the various channel parameters from the above quantities.

Using the generalized model, we now illustrate the effect of the code rate on channel SNR and symbol response h_k . In this section, for the sake of simplicity, we temporarily ignore ECC. Figure 2.7 shows the channel SNR for cut-off frequencies $\Omega_u = 0.5, 0.375$ and 0.33 , for a

Table 2.1: *The generalized Braat-Hopkins model.*

Model specification ($T_u =$ user bit period)
1) Code rate, R
2) Optical cut-off frequency (normalized by user bit rate), $\tilde{\Omega}_u$
3) User SNR, $\text{SNR}_u(\text{dB}) = 10 \log_{10} \left(\frac{\sum h_{ku}^2}{\sigma_u^2} \right)$, h_{ku} is the T_u -spaced symbol response with $\tilde{\Omega}_u = 0.33$ (see eqn.(2.20))
4) Power of disc reflectivity fluctuations ε_k in the user bandwidth $1/T_u$ (normalized by $E[a_k^2] = 1$), $\tilde{\Delta}_\varepsilon = \tilde{\sigma}_\varepsilon^2$
Channel parameters ($T =$ channel bit period)
1) Optical cut-off frequency (normalized by channel bit rate), $\Omega_c = R\tilde{\Omega}_u$
2) Symbol response (T -spaced), h_k (see eqn.(2.17))
3) Impulse response (T -spaced), f_k (see eqns.(2.13),(2.14))
4) Electronics noise variance, $\sigma_n^2 = \sigma_u^2/R$
5) Channel SNR, $\text{SNR}(\text{dB}) = 10 \log_{10} \left(\frac{\sum h_k^2}{\sigma_n^2} \right)$
6) Power of disc reflectivity fluctuations ε_k in the channel bandwidth $1/T$ (normalized by $E[a_k^2] = 1$), $\Delta_\varepsilon = \sigma_\varepsilon^2 = \tilde{\Delta}_\varepsilon/R$

given user SNR of 20 dB. The code rate ranges from $R = 8/17$, which corresponds to the EFM code, to the maximum of $R = 1$. As expected, the channel SNR increases as code rate increases for a given Ω_u . For a given code rate, the channel SNR decreases as Ω_u decreases. Recall from (2.7) that the channel SNR is the ratio of the energy in h_k to the noise power in the bandwidth $1/T$. In other words, for given user SNR and user density, the channel SNR shows the effect of decrease in signal energy and increase in noise bandwidth due to decrease in code rate, and vice versa. Further, since the channel SNR is defined as the the matched-filter bound (see (2.7)), it specifies the maximum achievable detection SNR in the absence of distortions. Thus, it is the minimum SNR that is required to achieve a given BER level.

The channel symbol response h_k for different R with $\Omega_u = 0.5$ and 0.375 are shown in Figure 2.8. The chosen code rates correspond to no code ($R = 1$), 17PP code ($R = 2/3$), EFM code ($R = 8/17$), and EFMPlus code ($R = 8/16$). Observe that for a given Ω_u , the energy and bandwidth of h_k decrease as R decreases. For a given R , the energy and bandwidth of h_k decrease and ISI increases as Ω_u decreases.

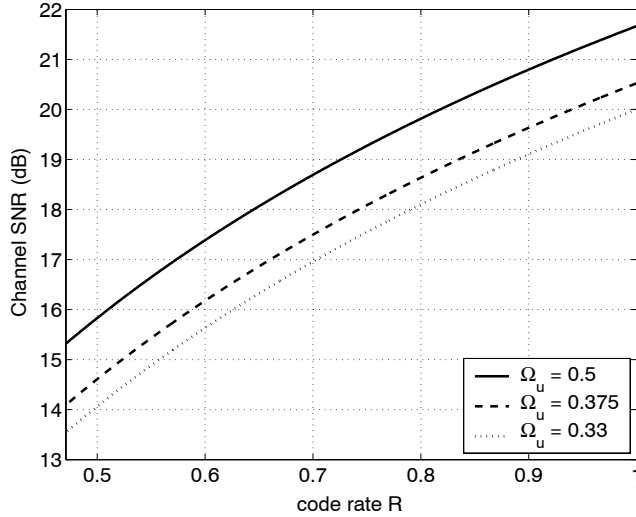


Figure 2.7: Channel SNR versus code rate for 20 dB user SNR.

2.5 Conclusion

A generalized Braat-Hopkins model for optical recording channels has been proposed. Important channel parameters such as signal to noise

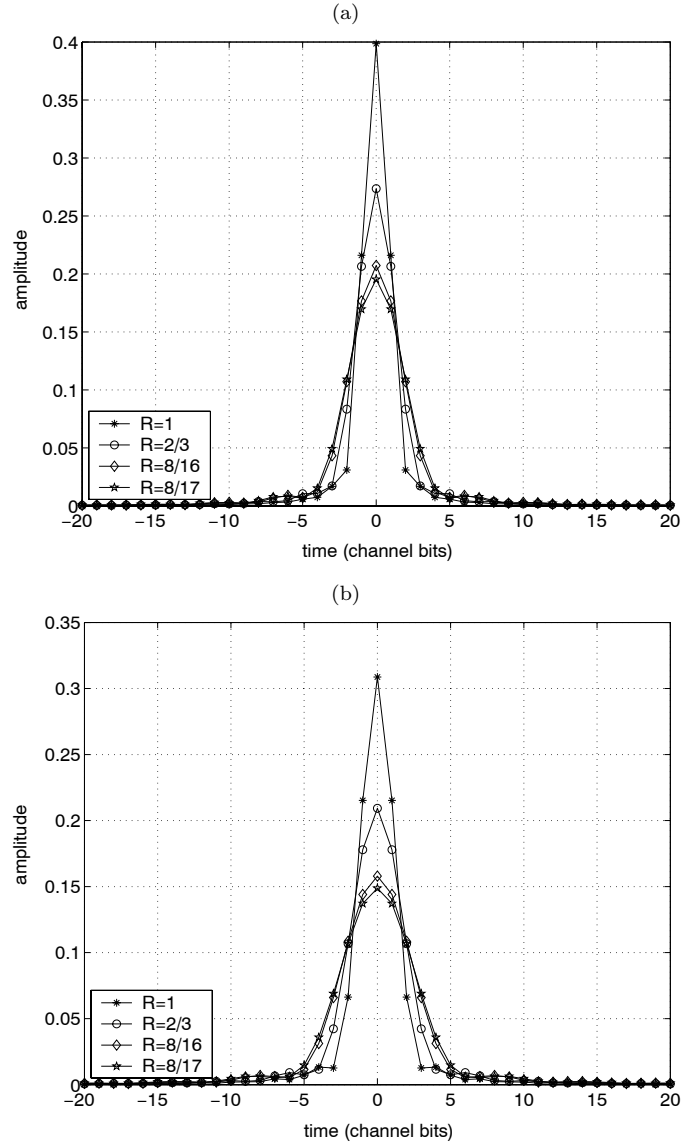


Figure 2.8: Discrete-time channel symbol response h_k with (a) $\Omega_u = 0.5$; (b) $\Omega_u = 0.375$.

ratio, optical cut-off frequency and media noise are defined with respect to user bit rate. These definitions are independent of the code rate and recording density, and therefore provide a fair basis for the performance comparison of detection schemes over different coding schemes and recording densities, for channels with AWGN and media noise.

Chapter 3

Parity-Check Coded Channels Based on Rate 2/3 Code

Compared with magnetic recording channels, optical recording channels have completely different channel characteristics, code constraints, as well as media noise and other linear/nonlinear distortions. Therefore, different parity-check (PC) codes and post-processors need to be designed to detect and correct the specific dominant error events (*i.e.* the most likely error events that can occur) at the output of the Viterbi detector (VD). The objective of this chapter is to investigate various basic issues associated with PC-code-based optical recording systems. This includes: the performance bounds for bit error rates and error event probabilities, the dominant error events at the VD output, different PC codes, simple and efficient post-processors, impact of error events that are split across data block boundaries and the corresponding remedy, as well as the effects of code rates and recording densities. In this chapter, for the sake of simplicity, the investigation is based on the rate 2/3 (1,7) code [100], whose code rate equals that of the standard $d = 1$ codes used in BD and HD-DVD [70, 48]. Further, the study does not include an explicit constrained PC encoder. Instead, we model the PC codes by assuming that the parity-check result for each data segment is known at the receiver, and by taking into account the corresponding code rate loss (see Section 3.2.1 for details). In addition, the channel noise is assumed to be additive, white and

Gaussian (AWGN). The performance of the system is evaluated in terms of the bit error rate (BER).

In Section 3.1, we compute performance bounds on bit error rates and error event probabilities, with and without PC codes. Based on these analyses, dominant error events have been found. Various PC codes are discussed in Section 3.2. The study of post-processing schemes is presented in Section 3.3. In Section 3.4, we analyze the impact of boundary error events and propose a novel remedy scheme. In Section 3.5, we first study the choice of codeword length for the PC codes under study. Comparison of theoretical and simulated BER bounds of these PC codes is then presented to corroborate the BER analysis. Thereafter, BER performances of various PC codes with post-processing are presented, at both nominal density and high density. Finally, Section 3.6 concludes this chapter.

The original contributions in this chapter are as follows.

- Analysis of BER and error event probabilities of VD based $d = 1$ channels, and derivation of analytical bounds on bit error rates, achievable by various PC codes.
- Analytical approach to compute the probability of $d = 1$ sequences that support given error events at the VD output.
- The design of various PC coding schemes using existing as well as new codes, and the associated optimum codeword lengths.
- The development of a multiple-error-event correction matched-filtering type post-processor.
- The analysis of the boundary error events and the development of a remedy scheme for post-processing.

3.1 Bit Error Rate Analysis

3.1.1 Introduction

Analytical approaches for evaluating the performance of PRML detection systems for the uncoded (*i.e.* $d = 0$ constrained) case are available in [29, 31, 33, 68]. But, similar approaches for coded channels are not available. Moreover, the published BER performances

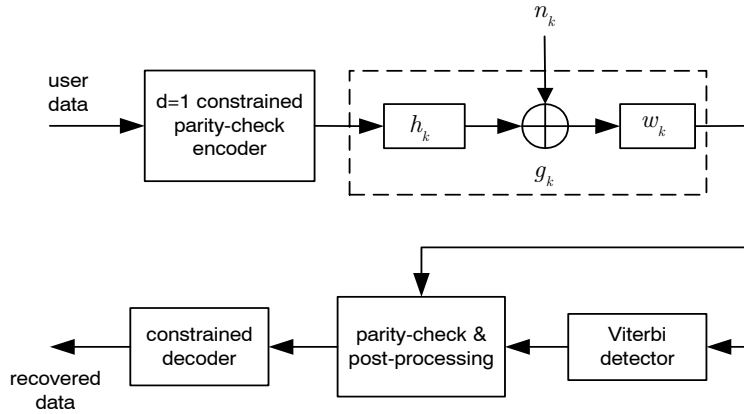


Figure 3.1: Block diagram of $d = 1$ channel with parity-check (PC) code and post-processing. The channel noise is assumed to be AWGN.

for optical recording with PC codes and post-processing are based on Monte Carlo simulations, and no BER bounds have been developed for such systems. Due to a number of reasons, there is a need to develop analytical methods to evaluate the BER performance of optical recording channels with the $d = 1$ constraint and PC codes. Such analyses enable predicting the system's BER performance in very low error probability regions, and/or with different codes. Furthermore, they also provide valuable guidelines for the development of PC codes and post-processing schemes.

Figure 3.1 shows the block diagram of a $d = 1$ channel with PC codes and post-processing. The $d = 1$ constrained PC encoder adds the $d = 1$ constraint as well as PC constraints on fixed-length segments of user data. Violation of the PC constraints in the detected bit sequence enables error detection. The task of locating the exact positions of the errors is done by a post-processor. Since this chapter focuses on BER of the PC coded system, the error-correction code (ECC) encoder/decoder is not included in this study.

In Figure 3.1, h_k is the channel symbol response, which is defined in Chapter 2. Furthermore, w_k is the impulse response of the equalizer,

which is designed using the minimum mean-squared error (MMSE) criterion [35] for a given choice of the partial response (PR) target g_k , $k = 0, 1, \dots, L_g - 1$. The PR target used in this chapter is a 7-tap symmetric filter whose coefficients are taken directly from the central part of the channel symbol response h_k . Our study shows that the VD with the above 7-tap PR target achieves similar performance to that of both the monic-constrained (*i.e.* $g_0 = 1$) MMSE equalization [65] and the noise-predictive (NP) PR equalization [21], implying that the 7-tap target is close to optimum. This can be attributed to the very good match between the 7-tap target and the channel response. The channel noise n_k is assumed to be AWGN.

3.1.2 Performance Bounds

Performance Without Parity-Check (PC) Codes

We define the transmitted non-return-to-zero (NRZ) data sequence of length N at the output of the $d = 1$ constrained encoder by codeword $\mathbf{a} = (a_1, a_2, \dots, a_N)$, with $a_i \in \{-1, +1\}$, and $1 \leq i \leq N$. Hence, $\mathbf{a} \in \mathcal{A}$, where \mathcal{A} denotes the code book of the $d = 1$ constrained codewords of length N . The corresponding detected sequence is denoted by $\hat{\mathbf{a}} = (\hat{a}_1, \hat{a}_2, \dots, \hat{a}_N)$. If this decision is incorrect, it will produce error(s) $e_i = a_i - \hat{a}_i$. An error event \mathbf{e} is said to extend from time i_1 to i_2 if

- there exist finite integers $1 \leq i_1 \leq i_2 \leq N$ such that $e_{i_1-i} = 0$ and $e_{i_2+i} = 0$, for $0 < i \leq L_d$, with $e_{i_1} \neq 0$ and $e_{i_2} \neq 0$, and
- there exists no integer j , with $i_1 \leq j \leq i_2 - L_d$, such that $e_j = e_{j+1} = \dots = e_{j+L_d-1} = 0$,

where L_d is the memory length of the channel¹. A given error event \mathbf{e} can be supported by only certain, if any, admissible codewords. The codeword \mathbf{a} is said to be admissible if the erroneous word $\hat{\mathbf{a}} = \mathbf{a} - \mathbf{e}$ also satisfies the $d = 1$ constraint. We do not include the k constraint in this analysis, since its impact on the performance is negligible.

Let \mathbf{E} denote the set of error events at the detector output. Since these error events are mutually exclusive, the probability that an error

¹In the presence of colored noise at the VD input, L_d must exceed the channel memory length to account for the memory induced by noise correlation.

event starts at time i_1 is given by

$$P_{event} = \sum_{\mathbf{e} \in \mathbf{E}} \Pr(\mathbf{e}), \quad (3.1)$$

where $\Pr(\mathbf{e})$ is the probability of an error event \mathbf{e} . Furthermore,

$$\Pr(\mathbf{e}) = \sum_{\mathbf{a} \in \mathcal{U}(\mathbf{e})} \Pr(\mathbf{a}) \Pr(\mathbf{e}|\mathbf{a}), \quad (3.2)$$

with $\mathcal{U}(\mathbf{e}) \subset \mathcal{A}$ is the set of all admissible codewords for a given \mathbf{e} . It is difficult to find exact expressions for $\Pr(\mathbf{e}|\mathbf{a})$ because of the complicated shapes of decision regions in the N -dimensional signal space. Instead, we can derive an upper bound, $P_{ub}(\mathbf{e})$, which ignores the possibility of any detected sequence other than $\hat{\mathbf{a}}$. That is, the bound $P_{ub}(\mathbf{e})$ is essentially the probability that the detector input is closer to the input signal estimated using $\hat{\mathbf{a}}$ than it is to the signal estimated using the correct sequence \mathbf{a} [53].

For Viterbi detection with equalization target g_k of length L_g , and by taking into account the noise correlation at the detector input, $P_{ub}(\mathbf{e})$ can be expressed as [33, 68]

$$P_{ub}(\mathbf{e}) = Q \left(\frac{\sum_{i=0}^{L_e+L_g-2} e_{y,k+i}^2}{2\sqrt{\sum_{i=0}^{L_e+L_g-2} \sum_{j=0}^{L_e+L_g-2} e_{y,k+i} e_{y,k+j} \phi(i-j)}} \right), \quad (3.3)$$

where $e_{y,k} = \sum_{l=0}^{L_g-1} g_l e_{k-i}$, $\phi(k)$ denotes k^{th} lag of the noise autocorrelation at the detector input, and $L_e = i_2 - i_1 + 1$ is the length of event \mathbf{e} . Here, $Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-t^2/2} dt$.

Therefore, we may rewrite $\Pr(\mathbf{e})$ as

$$\Pr(\mathbf{e}) \leq U(\mathbf{e}) P_{ub}(\mathbf{e}), \quad (3.4)$$

where $U(\mathbf{e}) = \sum_{\mathbf{a} \in \mathcal{U}(\mathbf{e})} \Pr(\mathbf{a})$ is the probability of selecting an admissible data sequence $\mathbf{a} \in \mathcal{A}$ that supports \mathbf{e} . This probability will be computed in Section 3.1.2. Hence, the error event probability P_{event} is bounded by

$$P_{event} \leq \sum_{\mathbf{e} \in \mathbf{E}} U(\mathbf{e}) P_{ub}(\mathbf{e}). \quad (3.5)$$

Similarly, the BER can be bounded as

$$P_{ber} \leq \sum_{\mathbf{e} \in \mathcal{E}} W(\mathbf{e})U(\mathbf{e})P_{ub}(\mathbf{e}), \quad (3.6)$$

where $W(\mathbf{e})$ is the number of nonzero elements in \mathbf{e} .

Performance With Parity-Check (PC) Codes

The $d = 1$ constrained PC encoder contains a (1,7) encoder combined with the PC encoder. The PC encoder provides codewords with the “zero syndrome” property [57, 101]. We consider p parity-checks defined by the vectors $\mathbf{c}_l = (c_{l,0}, c_{l,1}, \dots, c_{l,N-1})$ (*i.e.* the l^{th} row of the corresponding parity-check matrix), where $c_{l,i} \in \{0, 1\}$, and $0 \leq l \leq p - 1$. It follows from the definition of a codeword that any uncorrectable error event, considered as a ternary vector, should also satisfy the zero-syndrome property

$$\left\{ \sum_{i=0}^{N-1} c_{l,i} \cdot \bar{e}_i \right\} \bmod 2 = 0, \quad 0 \leq l \leq p - 1, \quad (3.7)$$

where $\bar{e}_i = 1$ if $e_i \neq 0$, and $\bar{e}_i = 0$ if $e_i = 0$.

To compute the performance bound with PC code, we assume that the post-processor is able to correct all dominant error events (*i.e.* the most likely error events that can occur) at VD output with nonzero syndromes without mis-correction. Dominant error events with zero syndromes as well as non-dominant error events are left uncorrected. We define \mathcal{E} to be the set of dominant error events at the detector output with nonzero syndromes. Proceeding in similar lines as we did for the case without PC codes, we can obtain the BER bound with a given PC code as

$$P_{ber.c} \leq \sum_{\mathbf{e} \in \mathcal{E}} W(\mathbf{e})U(\mathbf{e})P_{ub}(\mathbf{e}). \quad (3.8)$$

Computation of Probabilities $U(\mathbf{e})$

To compute $U(\mathbf{e})$, a maxentropic approach is adopted. A simple maxentropic Markov chain, describing $d = 1$ constrained sequences in NRZ format, is shown in Figure 3.2 [52]. It has 4 states, which are labeled by the four different combinations of 2-bit NRZ symbols

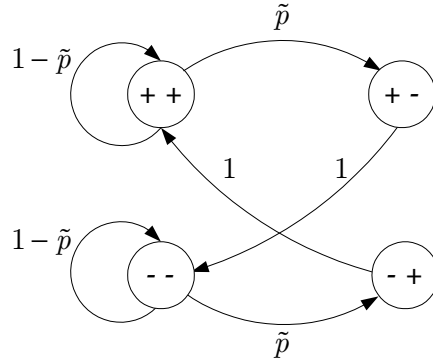


Figure 3.2: State transition diagram for $d = 1$ codes in NRZ format.

$\{++, +-, -, -\}$. The states are connected by arrows which represent allowed transitions from one state to the other, and the corresponding probabilities are indicated on the edges. The transition probability matrix is given by

$$\mathbf{T}_{NRZ} = \begin{bmatrix} 1 - \tilde{p} & \tilde{p} & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & \tilde{p} & 1 - \tilde{p} \end{bmatrix}, \quad (3.9)$$

where $\tilde{p} = \frac{3-\sqrt{5}}{2} \approx 0.382$ is the probability of transition from state $\{++\}$ to state $\{+-\}$, or from state $\{--\}$ to state $\{-+\}$ [38]. This chain has the steady state probabilities $\pi(++) = \pi(--)=\frac{1}{2(1+\tilde{p})}$ and $\pi(+-)=\pi(-+)=\frac{\tilde{p}}{2(1+\tilde{p})}$. We consider the 16 states formed by all the combinations $S_i = \begin{pmatrix} a_{i-1} & a_i \\ \hat{a}_{i-1} & \hat{a}_i \end{pmatrix}$ of 2 successive bits in \mathbf{a} and $\hat{\mathbf{a}}$, presented in the following order:

$$\begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ \begin{pmatrix} ++ \\ ++ \end{pmatrix} & \begin{pmatrix} ++ \\ +- \end{pmatrix} & \begin{pmatrix} ++ \\ -- \end{pmatrix} & \begin{pmatrix} ++ \\ -+ \end{pmatrix} & \begin{pmatrix} +- \\ ++ \end{pmatrix} & \begin{pmatrix} +- \\ +- \end{pmatrix} & \begin{pmatrix} +- \\ -+ \end{pmatrix} & \begin{pmatrix} +- \\ -- \end{pmatrix} \\ 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 \\ \begin{pmatrix} -+ \\ ++ \end{pmatrix} & \begin{pmatrix} -+ \\ +- \end{pmatrix} & \begin{pmatrix} -+ \\ -- \end{pmatrix} & \begin{pmatrix} -+ \\ -+ \end{pmatrix} & \begin{pmatrix} -- \\ ++ \end{pmatrix} & \begin{pmatrix} -- \\ +- \end{pmatrix} & \begin{pmatrix} -- \\ -+ \end{pmatrix} & \begin{pmatrix} -- \\ -- \end{pmatrix} \end{array}.$$

The states S_{i-1} and S_i are used to map the rows and columns in three 16×16 matrices \mathbf{T}_{e_i} , $e_i \in \{0, +2, -2\}$. The elements of these matrices will be denoted by $T_0[k, l]$, $T_{+2}[k, l]$ and $T_{-2}[k, l]$, respectively.

Each matrix \mathbf{T}_{e_i} describes the transitions that support e_i . An element on the intersection of the row and column corresponding to the states S_{i-1} and S_i , respectively, in \mathbf{T}_{e_i} is zero if either $\{a_{i-2}, a_{i-1}, a_i\}$ or $\{\hat{a}_{i-2}, \hat{a}_{i-1}, \hat{a}_i\}$ violates the $d = 1$ constraint, or if $e_i \neq a_i - \hat{a}_i$. Otherwise, this element represents a nonzero probability of transition from the input state $\{a_{i-2}, a_{i-1}\}$ to the new state $\{a_{i-1}, a_i\}$. By inspecting all possible states, matrices \mathbf{T}_{e_i} can be derived. For example, the matrix \mathbf{T}_0 is given by

$$\mathbf{T}_0 = \begin{bmatrix} 1-\tilde{p} & 0 & 0 & 0 & 0 & \tilde{p} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \tilde{p} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1-\tilde{p} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1-\tilde{p} & 0 & 0 & 0 & 0 & \tilde{p} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \tilde{p} & 0 & 0 & 0 & 0 & 1-\tilde{p} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1-\tilde{p} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \tilde{p} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \tilde{p} & 0 & 0 & 0 & 0 & 1-\tilde{p} \end{bmatrix} \quad (3.10)$$

Similarly, the nonzero elements of \mathbf{T}_{+2} and \mathbf{T}_{-2} can be derived as

$$\begin{aligned} T_{+2}[13, 10] &= T_{+2}[14, 12] = T_{+2}[16, 12] = \tilde{p}; \\ T_{+2}[1, 2] &= T_{+2}[2, 4] = T_{+2}[4, 4] = 1 - \tilde{p}; \\ T_{+2}[9, 2] &= T_{+2}[10, 4] = T_{+2}[12, 4] = 1; \\ T_{-2}[1, 5] &= T_{-2}[3, 5] = T_{-2}[4, 7] = \tilde{p}; \\ T_{-2}[13, 13] &= T_{-2}[15, 13] = T_{-2}[16, 15] = 1 - \tilde{p}; \\ T_{-2}[5, 13] &= T_{-2}[7, 13] = T_{-2}[8, 15] = 1. \end{aligned} \quad (3.11)$$

Further, we define a vector \mathbf{b} which contains the stationary probabilities of states at the beginning of the error event. The nonzero elements of \mathbf{b} are given by

$$b[1] = b[16] = \frac{1}{2(1+\tilde{p})}, \quad b[6] = b[11] = \frac{\tilde{p}}{2(1+\tilde{p})}. \quad (3.12)$$

In terms of these quantities, we get

$$U(\mathbf{e}) = \mathbf{b}^T \cdot \left\{ \prod_{i=1}^{L_e} \mathbf{T}_{e_i} \right\} \cdot \mathbf{T}_0^2 \cdot \mathbf{1}, \quad (3.13)$$

where ‘ T ’ denotes matrix transpose, $\mathbf{1}$ is an all-ones vector of length 16, and \mathbf{T}_0^2 is used to ensure that the error event \mathbf{e} ends without violating the $d = 1$ constraint, since two more NRZ bits after the ending of the error event are needed to examine the $d = 1$ constraint. We remark here that the above analytical approach for computing $U(e)$ can also be generalized for modulation codes with other constraints, such as the $d = 2$ codes or the maximum transition run (MTR) codes [69].

3.1.3 Comparison of Numerical and Simulation Results, without Parity-Check (PC) Codes

In this section, we present numerical calculations and computer simulation results to corroborate the above analyses, without including the PC codes. The performance evaluation with PC codes will be presented in Section 3.5.2.

Figures 3.3(a) and (b) compare the BER bounds obtained from (3.6) with computer simulations, at nominal density (with $\Omega_u = 0.5$) and high density (with $\Omega_u = 0.375$), respectively. As Figure 3.3 shows, inclusion of more error events in the summation on the right side of (3.6) results in reducing the difference between the theoretical and simulated results. With eight dominant error events, the difference becomes negligible, thus corroborating the BER analysis in Section 3.1.2, for the case without PC codes.

Table 3.1 compares the theoretical and simulated error event probabilities at VD output. In particular, the probability bounds of the first four dominant error events (*i.e.* $\pm\{2\}$, $\pm\{2, 0, -2\}$, $\pm\{2, 0, -2, 0, 2\}$ and $\pm\{2, 0, -2, 0, 2, 0, -2\}$) obtained from (3.4), as well as the error event rate bound obtained by including eight dominant error events in the summation on the right side of (3.5) are compared with simulation results. Using the cases of a user signal-to-noise ratio (SNR) of 13 dB at $\Omega_u = 0.5$, and 16 dB at $\Omega_u = 0.375$ as examples, Table 3.1 verifies that the theoretical estimated error event probabilities closely match those obtained by simulations. Similar observations are obtained with other SNRs as well.

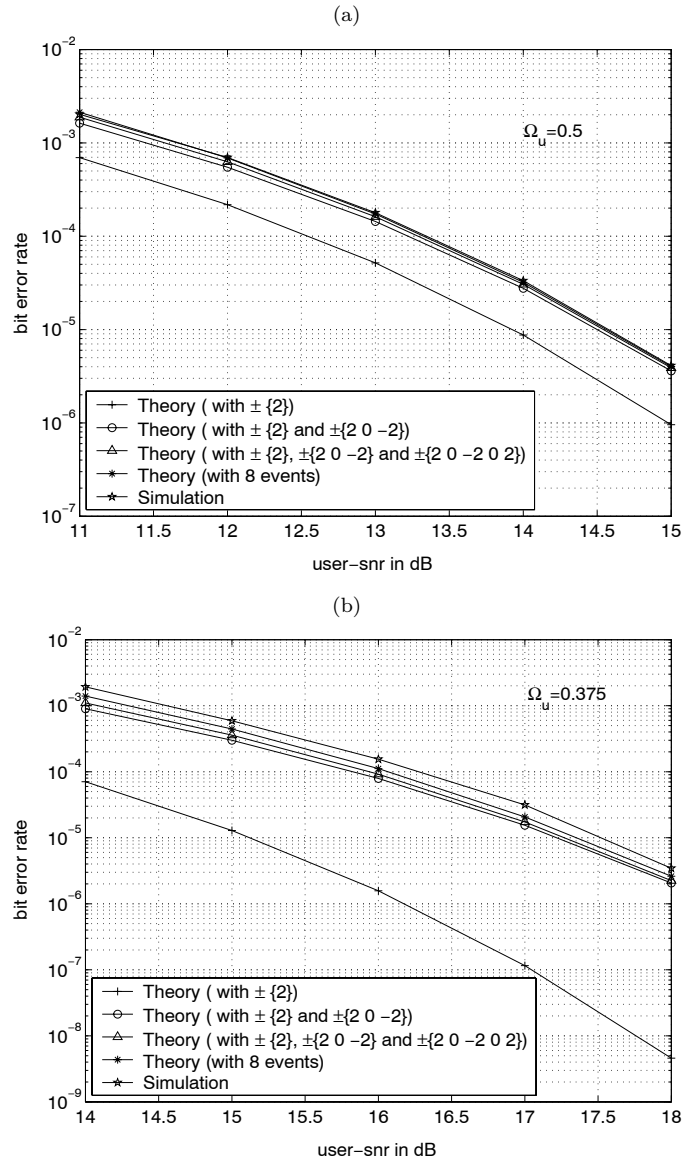


Figure 3.3: Comparison of theoretical and simulated BER performances, without parity-check (PC) codes. (a) $\Omega_u = 0.5$; (b) $\Omega_u = 0.375$.

Table 3.1: Comparison of theoretical and simulated error event probabilities at VD output.

(i) $\Omega_u = 0.5$, user SNR = 13 dB

	$\pm\{2\}$	$\pm\{2, 0, -2\}$	$\pm\{2, 0, -2, 0, 2\}$	$\pm\{2, 0, -2, 0, 2, 0, -2\}$	P_{event}
Theoretical probability	5.1633×10^{-5}	4.5974×10^{-5}	5.9091×10^{-6}	1.8186×10^{-6}	1.0701×10^{-4}
Simulated probability	5.9510×10^{-5}	4.0962×10^{-5}	5.2220×10^{-6}	1.4995×10^{-6}	1.1633×10^{-4}

(ii) $\Omega_u = 0.375$, user SNR = 16 dB

	$\pm\{2\}$	$\pm\{2, 0, -2\}$	$\pm\{2, 0, -2, 0, 2\}$	$\pm\{2, 0, -2, 0, 2, 0, -2\}$	P_{event}
Theoretical probability	1.5179×10^{-6}	3.7453×10^{-5}	4.0262×10^{-6}	3.2071×10^{-6}	4.9441×10^{-5}
Simulated probability	2.0791×10^{-6}	4.4517×10^{-5}	3.6568×10^{-6}	2.8068×10^{-6}	6.1150×10^{-5}

The pre-requisite knowledge required for the design of PC codes is the distribution of dominant error events. From Section 3.1.2, we know that by applying (3.4) for all possible error events \mathbf{e} , we can get the error event distribution. Figure 3.4 shows the histograms of dominant error events at the detector output. The probabilities of the first four dominant error events, normalized by the error event rate, are particularly indicated for various SNR values. These histograms were also verified by computer simulations. We found that the theoretical calculations match the simulations very well.

We note that the error event distributions vary with SNR. This is because, as derived in Section 3.1.2, the probability of a specific error event \mathbf{e} is approximated as $U(\mathbf{e})P_{ub}(\mathbf{e})$. Although $P_{ub}(\mathbf{e})$ is monotonous in SNR due to the nature of the $Q(\cdot)$ function, the presence of $U(\mathbf{e})$ causes the dominance to change with SNR. Thus, the distribution of error events with a given code constraint depends on both $P_{ub}(\mathbf{e})$ and $U(\mathbf{e})$.

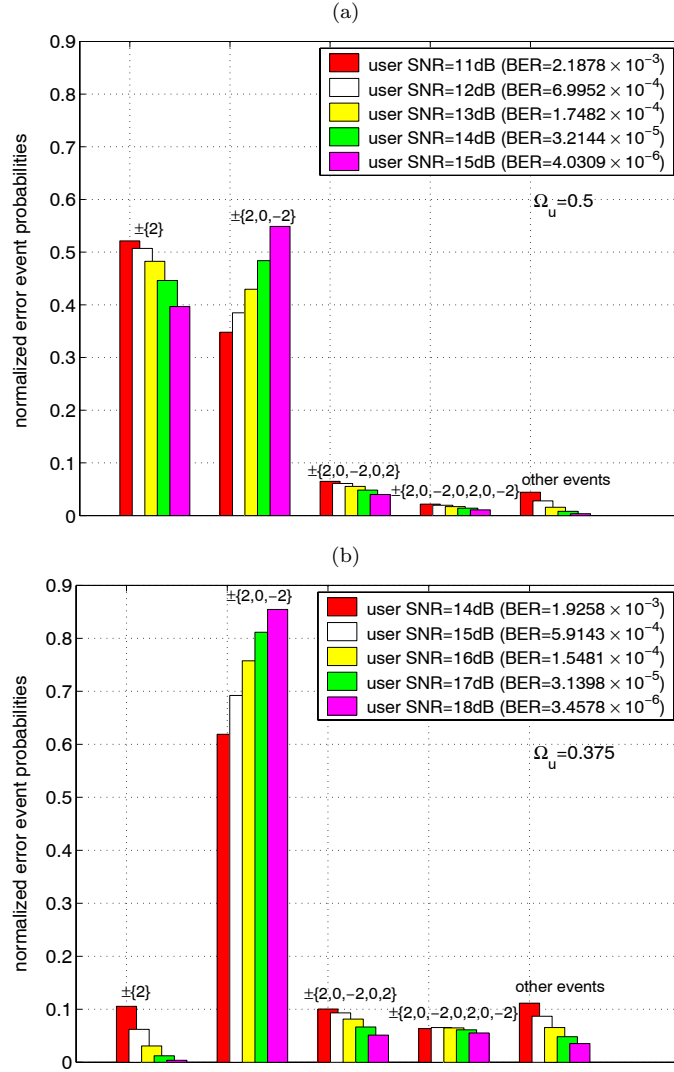


Figure 3.4: Histogram of dominant error events obtained from theoretical analysis, without parity-check (PC) codes. (a) $\Omega_u = 0.5$; (b) $\Omega_u = 0.375$.

Table 3.2: *Error detection capability of various CRC codes.*

Dominant error events	CRC codes				
	$g(x) = 1 + x$	$g(x) = 1 + x + x^2$	$g(x) = 1 + x + x^3$	$g(x) = 1 + x + x^4$	$g(x) = 1 + x^7$
$\pm\{2, 0, -2\}$	×	√	√	√	√
$\pm\{2\}$	√	√	√	√	√
$\pm\{2, 0, -2, 0, 2\}$	√	×	√	√	√
$\pm\{2, 0, -2, 0, 2, 0, -2\}$	×	√	√	√	√
$\pm\{2, 0, 0, -2\}$	×	×	√	√	√
$\pm\{2, 0, -2, 0, 2, 0, -2, 0, 2\}$	√	√	√	√	√

3.2 Parity-Check Codes

3.2.1 Cyclic Redundancy Check (CRC) Codes

As effective and simple error detecting codes, cyclic redundancy check (CRC) codes [57] are highly suitable for imposing PC constraints on constrained coded data sequences. CRC codes can be classified into two categories: single-bit PC codes and multiple-bit PC codes. These codes are defined by their generator polynomials $g(x)$ [57]. The degree of $g(x)$ specifies the number of parity bits added to the data.

The suitability and efficiency of different PC codes for a given channel and detector are determined by the error event distribution at the detector output. Through theoretical analysis as described in Section 3.1.2, we obtained the dominant error events at the output of the detector, at $\Omega_u = 0.5$ and $\Omega_u = 0.375$. Table 3.2 lists the first six dominant error events and the capabilities of selected CRC codes in detecting these error events. In the table, a “√” indicates that the error event can be detected by the corresponding PC code, and a “×” indicates that the error event cannot be detected by the corresponding PC code.

In Table 3.2, the PC code with $g(x) = 1 + x$ is the conventional single-bit even PC code, which can detect error events with an odd number of bit errors. The remaining codes in Table 3.2 are multiple-bit PC codes. Obviously, the codes with more parity bits can detect more types of error events. By using 3 or more parity bits, we can detect all the dominant error events listed in Table 3.2.

For the sake of simplicity, our study does not include an explicit

constrained PC encoder. Instead, we model the PC codes by assuming that the parity bits for each codeword are known at the receiver. This is done by generating the syndrome for each codeword in a “data-aided” mode. A general and systematic technique for designing efficient constrained PC codes will be presented in Chapter 5.

When the PC constraints are imposed on the data, the modulation constraints should be satisfied simultaneously. This will result in additional code rate loss. The minimum overhead is one user bit per parity bit. Equivalently, $\frac{1}{C_R}$ channel bits are needed per parity bit, where C_R is the capacity of the constrained code [20]. For the rate 2/3 $d = 1$ codes studied in this chapter, the minimum feasible overhead is 1.5 channel bits per parity bit. Therefore, we assume that each parity bit will require the minimum overhead of 1.5 channel bits. Let there be p parity bits per codeword of length N . Then, the overall code rate with PC coding is

$$R = \frac{N - 1.5p}{1.5N} = \frac{2}{3} - \frac{p}{N}. \quad (3.14)$$

3.2.2 A New Single-Bit Parity-Check (PC) Code

PC coding focuses on the dominant error events at the detector output. Figure 3.4 shows that $\pm\{2\}$ and $\pm\{2, 0, -2\}$ are the first two dominant error events. Clearly, the event $\pm\{2, 0, -2\}$ cannot be detected by the conventional single-bit even PC code. Therefore, we propose a new single-bit PC code with PC matrix given by $\mathbf{H} = [1, 1, 0, 0, 1, 1, 0, 0, \dots, 1]$. It is easy to verify that this code can detect the event $\pm\{2, 0, -2\}$ occurring at any position in the data sequence. Moreover, it can also detect the events $\pm\{2\}$, $\pm\{2, 0, -2, 0, 2\}$, $\pm\{2, 0, 0, -2\}$ and $\pm\{2, 0, -2, 0, 2, 0, -2, 0, 2\}$ listed in Table 3.2, at specific positions of the data sequence. The proposed new single-bit PC code also facilitates the development of a 2-level hierarchical PC coding scheme.

3.2.3 Hierarchical Scheme for Parity-Check (PC) Coding

To enhance error detection and correction beyond the single-bit PC codes, we need to use multiple-bit PC codes. In this chapter, we model the multiple-bit constrained PC codes by assuming a minimum code rate loss of 1.5 channel bits per parity bit. However, when it is

difficult to construct such efficient constrained PC codes, a hierarchical PC coding scheme may be used [20]. In this section, we investigate the performance of the hierarchical scheme for PC coding.

We propose a 2-level hierarchical scheme. It arranges the data sequence into 2 data levels, according to the probabilities of the first and second dominant error events. Each data level is protected by a PC code. Since the probability of the first dominant error event is larger than the second dominant error event, the PC constraint for the first event is imposed on a shorter data segment while the PC constraint for the second event is imposed on a larger data segment. Such a 2-level hierarchical scheme is shown in Figure 3.5. The overall code rate is given by (assuming 1.5 channel bits per parity bit)

$$\begin{aligned} R_h &= \frac{N_2 - 1.5 \left(\frac{p_1(N_2 - 1.5p_2)}{N_1} + p_2 \right)}{1.5N_2} \\ &= \frac{2}{3} - \frac{p_1}{N_1} - \frac{p_2}{N_2} + \frac{1.5p_1p_2}{N_1N_2}, \end{aligned} \quad (3.15)$$

where p_i and N_i are the number of parity bits and data block length, respectively, for the i^{th} data level, $i = 1, 2$. Usually, $N_2 \gg p_2$ and $N_1N_2 \gg p_1p_2$, and hence the code rate loss introduced by the parity-check for the second data level is very small. Therefore, using the hierarchical scheme, more of the dominant error events can be detected, without incurring much rate loss.

In this work, we choose the proposed new single-bit PC code, denoted by *PC1*, to protect the first data level, since it can detect the event $\pm\{2, 0, -2\}$ which is the first dominant error event at high SNR or high recording density. The single-bit even PC code, de-

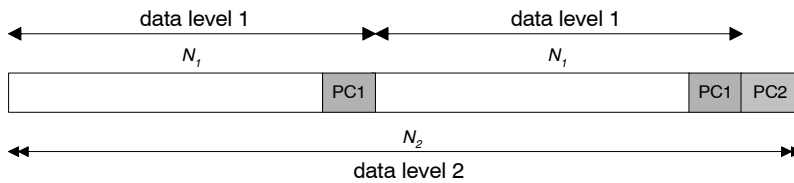


Figure 3.5: A 2-level hierarchical parity-check (PC) coding scheme.

noted by $PC2$, is chosen to protect the second data level, since it can detect most of the remaining dominant error events, such as $\pm\{2\}$, $\pm\{2, 0, -2, 0, 2\}$ and $\pm\{2, 0, -2, 0, 2, 0, -2, 0, 2\}$. Here, notice that the $\pm\{2, 0, -2, 0, 2, 0, -2\}$ type error events are left out, since it cannot be detected by any of the two single-bit PC codes. While decoding, we first carry out error correction with $PC1$ on each segment of Level 1. Subsequently, we compute the syndrome of $PC2$ and carry out error correction on Level 2. Note that the proposed component codes are single-bit PC codes. Compared with multiple-bit PC codes, they are easier to be combined effectively with modulation constraints, since they have less stringent PC constraints (*i.e.* fewer parity bits). Further, the 2-level hierarchical scheme could be extended to multi-level schemes.

3.3 Post-Processing Schemes

3.3.1 Analysis of Parity-Check (PC) based Post-Processing

The optimum detection for a PC coded channel may be performed by using Viterbi based sequence detection on the combined trellis of the channel and PC code. But, this may be computationally very expensive since the number of trellis states grows exponentially with the length of channel PR target and the number of PC bits. In general, for a channel PR target of length L_g and a PC code with p parity bits, a 2^{L_g-1+p} -state combined trellis is required. For the 7-tap PR target studied in this chapter, and a 4-bit PC code that can detect all the dominant error events of the system, the total number of states in the combined trellis is $2^{10} = 1024$, while the number of trellis states of the current channel detector is only up to 32 [34].

On the other hand, PC based post-processing schemes have been shown to be simple and efficient soft-decision decoders [23, 27, 15, 86, 20], even for a large bank of matched-filters. These schemes have been widely accepted by current magnetic recording systems [34]. Therefore, in this thesis, we focus on this type of schemes for error detection and correction

In PC based post-processing, the purpose of PC codes is to detect the presence of errors. The post-processor handles the task of locating the error positions. The inputs of the post processor are the VD

output bits $\hat{\mathbf{a}} = [\hat{a}_1, \hat{a}_2, \dots, \hat{a}_N]^T$, the corresponding detector input samples $\mathbf{q} = [q_1, q_2, \dots, q_N]^T$ as well as the parity-check result (*i.e.* the syndrome of the received codeword). The post-processor needs to maximize the probabilities of correct decisions for the transmitted data sequence $\mathbf{a} = [a_1, a_2, \dots, a_N]^T$ based on the above input information. This leads to the maximum *a posteriori* (MAP)-based decision rule for post-processing [25]. That is,

$$\hat{\mathbf{a}} = \arg \max_{\tilde{\mathbf{a}}_i^j} \Pr(\tilde{\mathbf{a}}_i^j | \mathbf{q}, \hat{\mathbf{a}}, \mathbf{s}), \quad (3.16)$$

where \mathbf{s} is the nonzero syndrome obtained from PC. We may write

$$\tilde{\mathbf{a}}_i^j = \hat{\mathbf{a}} + \mathbf{e}_i^j, \quad (3.17)$$

where $\tilde{\mathbf{a}}_i^j$ is the re-estimation of the transmitted data sequence \mathbf{a} based on the detected bits $\hat{\mathbf{a}}$ as well as an assumed error event (or error event combination) \mathbf{e}_i^j of length L_e^j and type j , starting at position i , with $i = 1, 2, \dots, N - L_e^j + 1$. The syndrome \mathbf{s} and the $d = 1$ constraint help to reduce the number of possible candidates for \mathbf{e}_i^j . This is described in detail in Section 3.3.2. Since the posterior probabilities $\Pr(\tilde{\mathbf{a}}_i^j | \mathbf{q}, \hat{\mathbf{a}}, \mathbf{s})$ are equivalent to the probabilities of $\Pr(\mathbf{e}_i^j | \mathbf{q}, \hat{\mathbf{a}})$, using Bayes' rule, the MAP criterion for post-processing can be re-expressed as

$$\hat{\mathbf{e}} = \arg \max_{\mathbf{e}_i^j} p(\mathbf{q} | \mathbf{e}_i^j, \hat{\mathbf{a}}) \Pr(\mathbf{e}_i^j, \hat{\mathbf{a}}), \quad (3.18)$$

where $p(\mathbf{q} | \mathbf{e}_i^j, \hat{\mathbf{a}})$ is the conditional probability density of the VD input samples and $P_j \triangleq \Pr(\mathbf{e}_i^j, \hat{\mathbf{a}})$ is the *a priori probability* of \mathbf{e}_i^j associated with data pattern $\hat{\mathbf{a}}$. Therefore, the post-processor according to (3.18) is essentially a MAP based error event(s) detector. Note that the value of $\Pr(\mathbf{e}_i^j, \hat{\mathbf{a}})$ for given j and $\hat{\mathbf{a}}$ is a constant for all i . The value of P_j can be computed analytically using $U(\mathbf{e})P_{ub}(\mathbf{e})$, according to Section 3.1.2.

Appendix A simplifies (3.18) to

$$\hat{\mathbf{e}} = \arg \max_{\mathbf{e}_i^j} \left\{ \sum_{k=1}^{N_t} e_{q,k} \left(\sum_{l=0}^{L_g-1} g_l e_{i,k-l}^j \right) - \eta_j \right\}, \quad (3.19)$$

where $e_{q,k} = q_k - \hat{q}_k$, with $\hat{q}_k = \sum_{l=0}^{L_g-1} g_l \hat{a}_{k-l}$ and $N_t = N + L_e^j + L_g - 2$. Here, η_j is an offset associated with the error event, given by

$$\eta_j = \frac{1}{2} \sum_{k=1}^{N_t} \left(\sum_{l=0}^{L_g-1} g_l e_{i,k-l}^j \right)^2 - \sigma_v^2 \ln(P_j), \quad (3.20)$$

where σ_v^2 is the variance of the combined channel noise and residual intersymbol interference (ISI) at the VD input.

From (3.19), we see that the MAP post-processor can be implemented by passing the error signal $e_{q,k}$ through a bank of filters which are matched to the assumed error event(s) and normalizing the filter outputs by subtracting a set of offsets. The matched-filter for a given error event is the time-reversed version of the convolution between the error event and the channel target response g_k . Furthermore, by dropping the term $2\sigma_v^2 \ln(P_j)$ in (3.20), we can obtain the maximum-likelihood (ML) based post-processor. In principle, the MAP post-processor is superior to the ML post-processor, when the *a priori probabilities* P_j are significantly different for each dominant error event. However, for the systems studied in this thesis, no significant performance difference has been found between the MAP and ML post-processors. Therefore, for the case with AWGN, we use the above deduced matched-filters based implementation of the ML post-processor, due to its simplicity. Note that in the case with media noise, this post-processor is no longer optimum, due to the correlated, data-dependent, and non-stationary nature of the media noise. In Section 6.3.2, a data dependent post-processing scheme is developed for channels with media noise.

3.3.2 Multiple-Error-Event Correction Post-Processor

Most of the PC-based post-processor schemes can only correct a single error event in one PC codeword. That is, the post-processor assumes that only one error event is present in a detected codeword [23, 27, 15, 49, 20]. Obviously, these schemes may make mis-correction when multiple error events occur within one codeword. Therefore, in this part, a multiple-error-event correction post-processing scheme is developed. Further, compared with the $d = 0$ constraint used in magnetic recording systems [72], the optical recording system has the

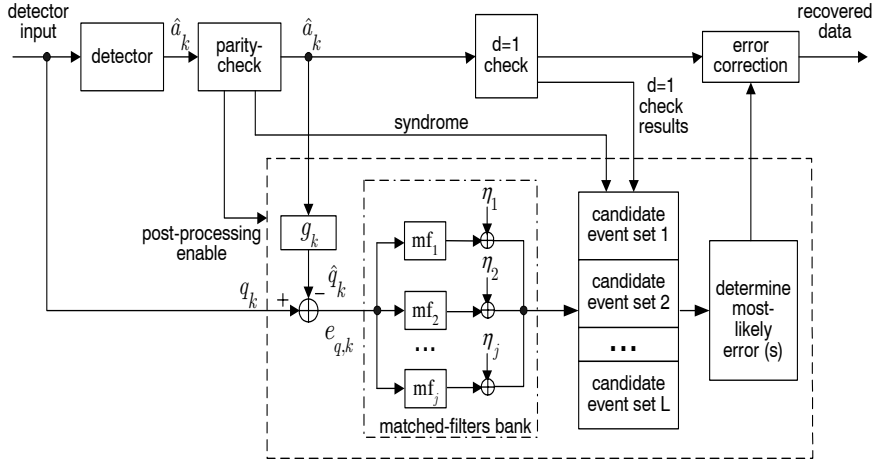


Figure 3.6: Block scheme of the receiver for parity-check (PC) coded channels with matched-filtering type multiple-error-event correction post-processor.

more stringent $d = 1$ constraint. Therefore, we also make use of this constraint in post-processing to minimize mis-correction.

Figure 3.6 shows the block diagram of the receiver for parity coded channels with matched-filtering type post-processor. When the parity-check detects an error event, it activates the post-processor and also passes the nonzero syndrome of the received codeword to the post-processor. The post-processor first calculates the error signal $e_{q,k}$, which is then passed to the bank of matched-filters corresponding to the dominant error events. The outputs of the matched-filters are normalized by subtracting the offsets η_j . We apply two additional criteria on these normalized outputs to minimize mis-correction. First, the detected error event should produce the same nonzero syndrome as the parity-check result. We call this the *syndrome criterion*. Second, the corrected version of the detected bits should not violate the $d = 1$ constraint. We call this the $d = 1$ *criterion*². In single-error-event correction post-processing, the normalized maximum outputs of all the filters that satisfy the two criteria are compared and the largest

²We do not include the k constraint examination in post-processing since its impact on performance is negligible.

one is used to determine the error event type and position.

To perform multiple-error-event correction, the straightforward way is to use a bank of filters that are matched to the dominant error events as well as their various combinations, since multiple error events occurring within one PC codeword are mainly combinations of various single dominant error events of the system. However, this is computationally prohibitive, since even for a limited number of dominant error events, there may be numerous combinations of these events, corresponding to all possible starting positions of the events. Hence, we propose a low-complexity multiple-error-event correction post-processing scheme. The key idea is to first use matched-filtering to obtain the possible types and locations of the single-error events, and then to compute the squared Euclidean distance of each possible error event or error event combination to determine the most-likely event(s). The proposed post-processing scheme is summarized as follows.

(1) In the post-processor, we use a bank of filters that are matched to the individual dominant error events of the system. For each matched-filter, the output samples are sorted in descending order of magnitude. Several large samples that satisfy the $d = 1$ constraint are selected to indicate the candidate error events.

(2) From the candidate error events obtained in Step (1), we select single error events as well as various combinations of multiple error events that produce the same nonzero syndrome as the parity-check result. Furthermore, each pair of error events should be separated by an interval that is larger than the error-free interval, which is associated with the effective length of the channel memory L_d . Several sets of candidate error events can be obtained, *e.g.* Set 1 contains the single-error events, Set 2 contains various combinations of double-error events, and so forth.

(3) Compute $\sum_{k=1}^{N_i} (q_k - \tilde{q}_{i,k}^j)^2$, with $\tilde{q}_{i,k}^j = \sum_{l=0}^{L_g-1} g_l \tilde{a}_{i,k-l}^j$, the squared Euclidean distance between the detector input samples and their reconstructed versions based on each error event or error event combination in the candidate event sets. The error event(s) with the minimum distance is(are) determined as the most likely event(s). Since the number of candidate error events obtained from Step (2) is limited after the first-round screening performed via the matched-filters in Step (1), the complexity for correcting multiple error events is limited.

The computational complexity of the proposed post-processing scheme

can be adjusted according to desired performance/complexity trade-off. In this work, we find that the double-error-event correction post-processor provides sufficiently good performance for the system under study.

3.4 Study of Boundary Error Events

The impact of error events that are split across the boundary of two codewords is an issue that is of much concern for error correction and in particular, for the post-processor. In this section, we investigate this issue and propose a remedy.

3.4.1 Boundary Error Event Analysis

The probability that error events occur at the boundary obeys

$$P_{b_event} \leq \sum_{\mathbf{e} \in \mathbf{E}} \frac{L_e - 1}{N} U(\mathbf{e}) P_{ub}(\mathbf{e}). \quad (3.21)$$

Following Section 3.1.2, we know that $U(\mathbf{e})P_{ub}(\mathbf{e})$ approximates the probability of occurrence of error event \mathbf{e} . Therefore, each individual term in the summation in (3.21) (*i.e.* $\frac{L_e - 1}{N} U(\mathbf{e})P_{ub}(\mathbf{e})$) denotes the probability that the event \mathbf{e} happens in the boundary region of a data block of length N .

The incremental contribution of the boundary error events to the BER, compared to the performance of the ideal post-processor (3.8) in Section 3.1.2, can be bounded as

$$P_{b_ber} \leq \sum_{\mathbf{e} \in \mathbf{E}} \frac{L_e - 1}{N} W(\mathbf{e}) U(\mathbf{e}) P_{ub}(\mathbf{e}). \quad (3.22)$$

Using the case of $N = 100$ and $\Omega_c = 0.33$ as an example, we verify the above analysis via computer simulations. The results (Curves 1 and 2 of Figure 3.7) show that the theoretical calculations match the simulations very well. Using (3.22), the BER increase (P_{b_ber}) due to the boundary error events can be calculated as a function of data block length. The numerical results are shown by Curves 2 to 5 of Figure 3.7. Here, to avoid the influence of code rate loss, we do the calculations/simulations based on channel SNR and channel bit

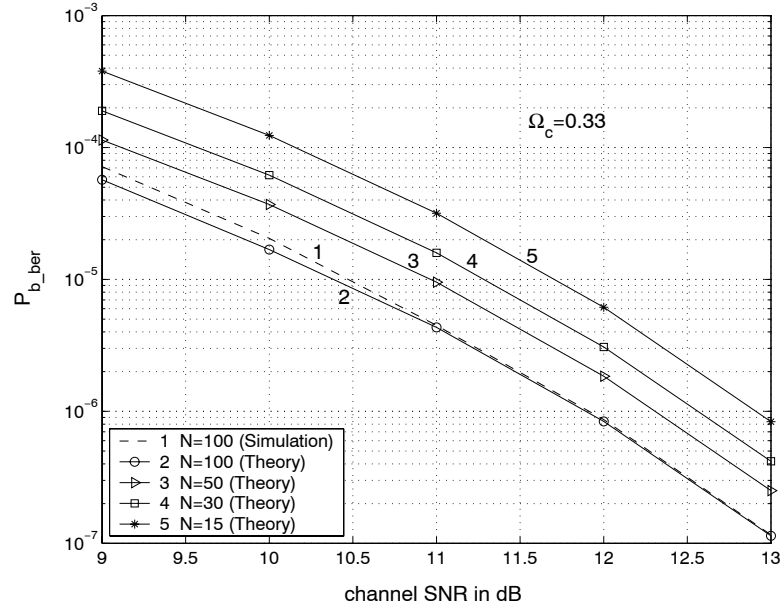


Figure 3.7: BER increase due to boundary error events for different block lengths (theoretical).

rate normalized cut-off frequency. Observe that as the block length decreases, the BER increase becomes larger, as we might intuitively expect. Similar observations are also obtained with $\Omega_c = 0.25$.

3.4.2 A Novel Remedy Scheme

To minimize the impact of boundary error events on the overall BER, we have developed a remedy scheme. We divide each boundary error event into 2 parts (see Figure 3.8): the *leading part* that lies at the end of the current data block, and the *trailing part* that lies in the beginning of the next data block. For example, if the error event length is 5, the *leading part* and the *trailing part* may last 4 bits at most.

We note that the error patterns obtained by splitting the dominant error events in Table 3.2 are still in the list of error events given in this table. As described in Section 3.2.1, by adding 3 or more parity bits, all these error events can be detected. Therefore, the PC for

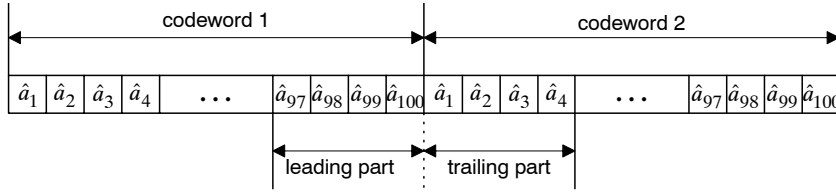


Figure 3.8: Analysis of boundary error events.

each data block is still carried out from time index 1 to N , for each data block. Thus, the *leading* and *trailing* parts of the boundary error event are detected by the PC codes of the two consecutive data blocks separately.

For error correction, we realized that if we simply treat the *leading part* and the *trailing part* as two separate error events, the erroneous bits in the *trailing part* may affect the correction of the *leading part*, and *vice versa*. Therefore, in the proposed remedy scheme, we correct the *leading part* and the *trailing part* separately. However, during error correction, we extend the current data block by a few bits (*e.g.* 4 bits). This helps to obtain the normalized outputs of matched-filters corresponding to the boundary error event, and to locate its type and position. Moreover, whenever we find that the detected error event extends across the block boundary, the $d = 1$ *criterion* in the post-processor needs to be checked through the extended data block, while the *syndrome criterion* is checked within the original data block. Based on this, the erroneous bits in the *leading part* are corrected. For the *trailing part* of the boundary event, since the *leading part* has already been corrected, we treat it as a separate error event and implement error correction in the next data block.

3.5 BER Performance and Discussion

In this section, we first discuss the choice of the codeword length of PC codes. We then present comparison of theoretical and simulated BER bounds to corroborate the BER analysis in Section 3.1.2, for the case with PC codes. Thereafter, BER performance evaluations of various

PC codes with post-processing will be presented.

3.5.1 Choice of Codeword Length

For a given PC code, the choice of codeword length is a compromise between rate loss and error correction capability. This can be seen as follows.

1. As described in Section 3.3.2, the post-processor assumes that only one or a few dominant error events are present in a codeword. Obviously, the shorter the codeword length N is, the more likely it is that this requirement is satisfied. In addition, short data blocks decrease the probability of error in locating the error event as well as the probability that multiple error events produce a zero syndrome.
2. On the other hand, according to (3.14), decreasing the codeword length will increase the code rate loss. The resulting increase in noise bandwidth and channel density lead to reduced SNR and poor performance.

The effect of code rate loss stated in Item 2 can be clearly seen from Figure 3.9, which shows the BER performance at VD output as a function of the codeword length per parity bit (*i.e.* $\frac{N}{p}$ in (3.14)). For each user SNR, the flat trace shows the BER without PC codes, and the curved trace corresponds to the BER for PC codes with different codeword lengths. Note that these curved traces, being the BER at VD output, do not show the performance gain from PC based post-processing. In other words, what this figure shows is the rate loss as a function of the codeword length. All these curves are computed theoretically according to (3.6). Figure 3.9 shows that the performance loss resulting from rate loss due to the use of PC codes decreases with increase in codeword length. When the codeword length per parity bit is 200, the loss becomes really small. Hence, we conclude that there is no advantage in choosing the codeword lengths (per parity bit) exceeding 200, since this will only increase the probability of multiple error events to occur within one codeword.

To study the effect of codeword length N on the overall performance, we show in Figure 3.10 the BER performance at the post-processor output, as a function of codeword length (per parity bit),

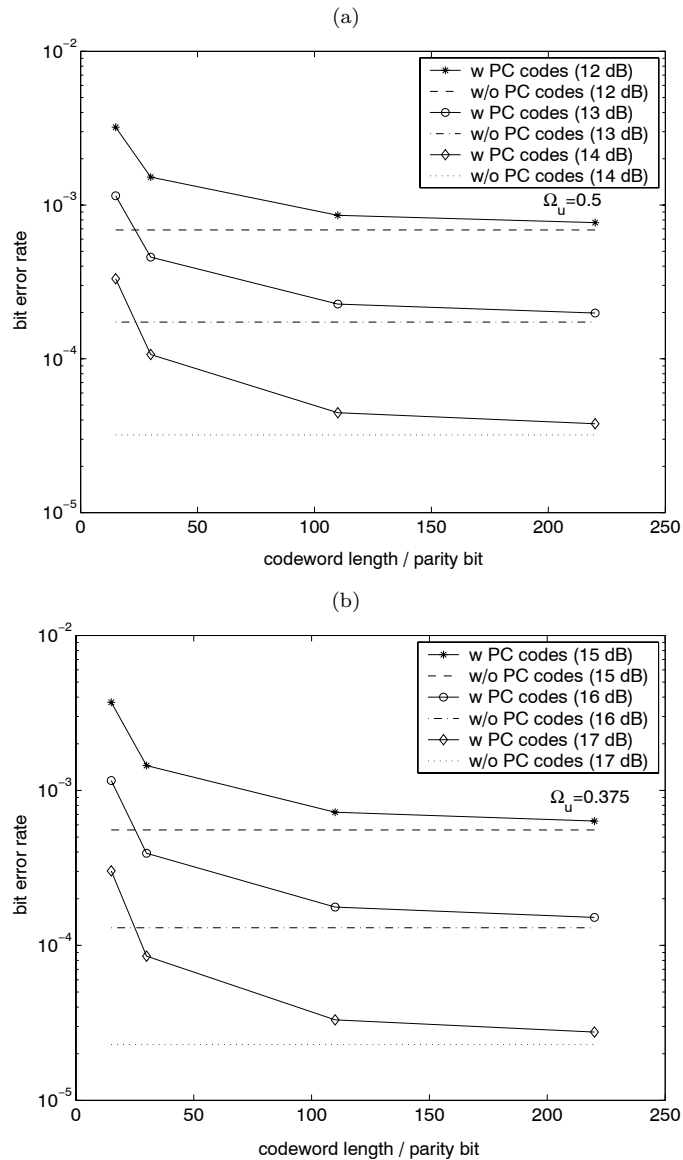


Figure 3.9: BER performance (theoretical) at Viterbi output, taking into account the code rate loss for different codeword lengths per parity bit. The ‘user SNR’ for each plot is shown in the brackets. (a) $\Omega_u = 0.5$; (b) $\Omega_u = 0.375$.

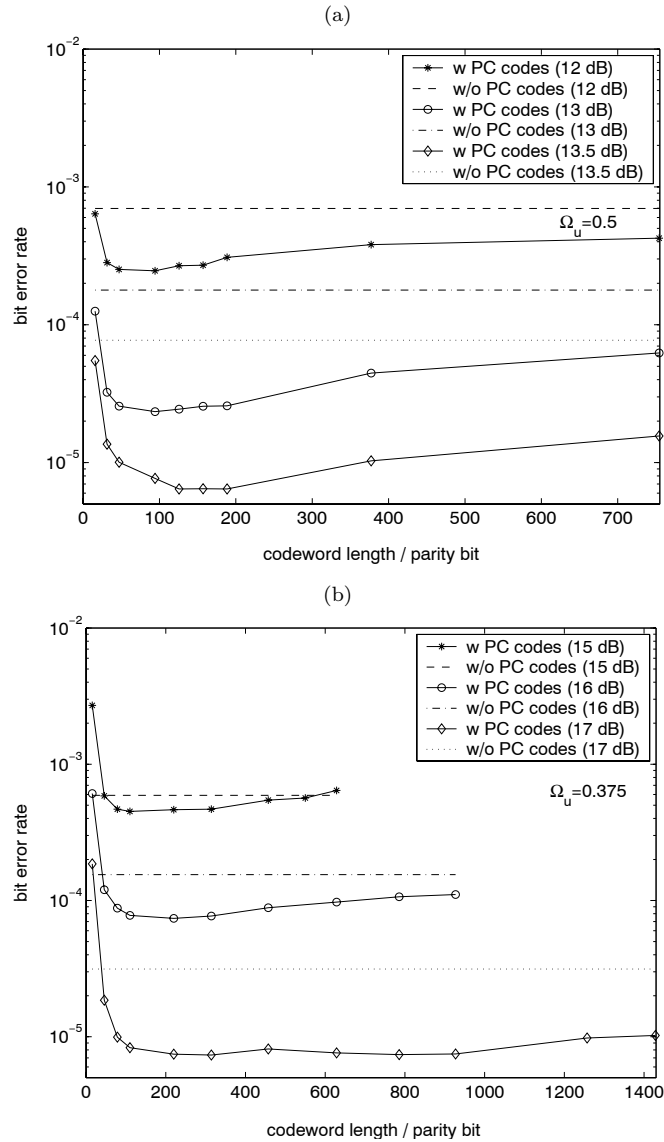


Figure 3.10: BER performance at the post-processor output, as a function of codeword length per parity bit. The PC code corresponds to $g(x) = 1 + x + x^4$. The ‘user SNR’ for each plot is shown in the brackets. (a) $\Omega_u = 0.5$; (b) $\Omega_u = 0.375$.

which is obtained using simulations. The BER performances without PC codes are calculated theoretically. We use the PC code with $g(x) = 1 + x + x^4$ as an example. Ideally, for a given density, the optimum codeword length should be a function of SNR. At low SNR, the codeword length should be shorter to ensure that only one or a few error events are present in a codeword. At high SNR, the codeword length can be larger. However, from Figure 3.10, we find that for each SNR, the minimum BER is obtained over a certain range of codeword lengths. These ranges of codeword lengths for different SNRs have an overlapping region, which is approximately 100 to 200 bits per parity bit. This also agrees with the theoretical analysis illustrated in Figure 3.9. Therefore, at a given density, instead of varying the codeword length for each SNR, we can simply take a fixed codeword length from the overlapping region. Similar observations were obtained for other PC codes also.

Since a shorter codeword length will simplify the complexity of the post-processor, we choose to use a codeword length of $N' = 100$ bits per parity bit. The corresponding code rate loss is 1.5%. Along the same lines, suitable codeword lengths for the proposed 2-level hierarchical scheme have also been found, which are $N_1 = 100$ bits for the first data level, and $N_2 = 200$ bits for the second data level.

3.5.2 Comparison of Theoretical BER Bounds with Simulations, with Parity-Check (PC) Codes

To validate the bound of (3.8), we choose three PC codes as examples. They correspond to $g(x) = 1 + x$, $g(x) = 1 + x + x^2$, and $g(x) = 1 + x + x^3$, respectively. As described in the previous section, their codeword lengths are chosen to be 100, 200, and 300, respectively. The comparison between theoretical and simulated BER bounds for these codes is illustrated in Figure 3.11, with the corresponding code rate loss taken into consideration. The optical cut-off frequency, normalized by user bit rate, is $\Omega_u = 0.5$. The simulation bounds are obtained by removing the dominant error events that have nonzero syndromes with respect to the corresponding PC codes. Observe that the theoretically calculated bounds match the simulation results very well. Further, Figure 3.11 clearly shows that adding more parity bits enables detection and correction of more types of error events, and therefore, improves the performance. Similar observations are also

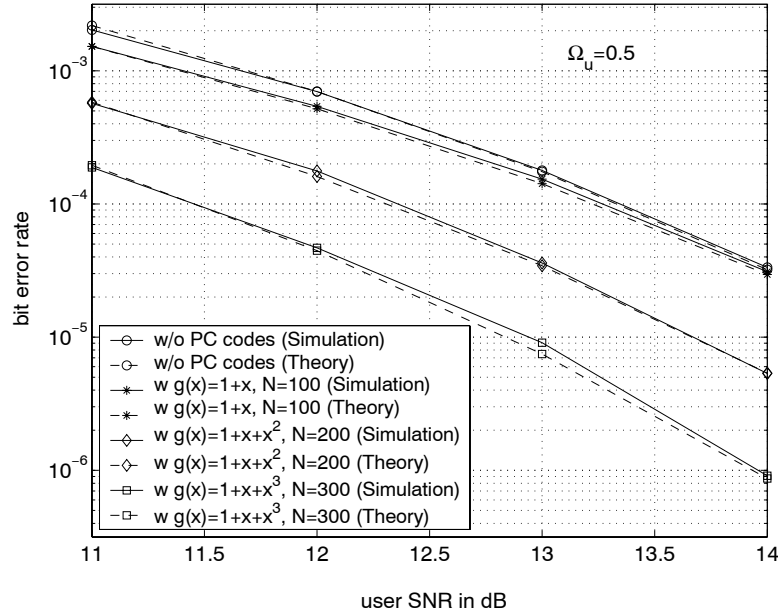


Figure 3.11: Comparison of BER bounds obtained from theory and simulations, with parity-check (PC) codes, $\Omega_u = 0.5$.

obtained with $\Omega_u = 0.375$.

3.5.3 Simulation Results

In this section, we present the BER performance with various PC codes and post-processing, obtained using simulations. The constrained code used in the simulations is the rate 2/3 (1,7) code. The syndrome of the PC code is obtained in a “data aided” mode, and the corresponding code rate loss has been taken into account. The channel noise is assumed to be AWGN. The resulting performances are summarized in Figures 3.12 (a) and (b), for nominal density (with $\Omega_u = 0.5$) and high density (with $\Omega_u = 0.375$), respectively.

In Figure 3.12, Curves 2 to 6 illustrate the performance of five typical PC codes. The BER performance of the system without PC codes is plotted in Curve 1 as a reference. The PC codes with $g(x) = 1 + x + x^3$, $g(x) = 1 + x + x^4$ and $g(x) = 1 + x^7$ can detect all the dominant error events listed in Table 3.2. The corresponding analytical bound for these three codes, obtained from (3.8), is illustrated

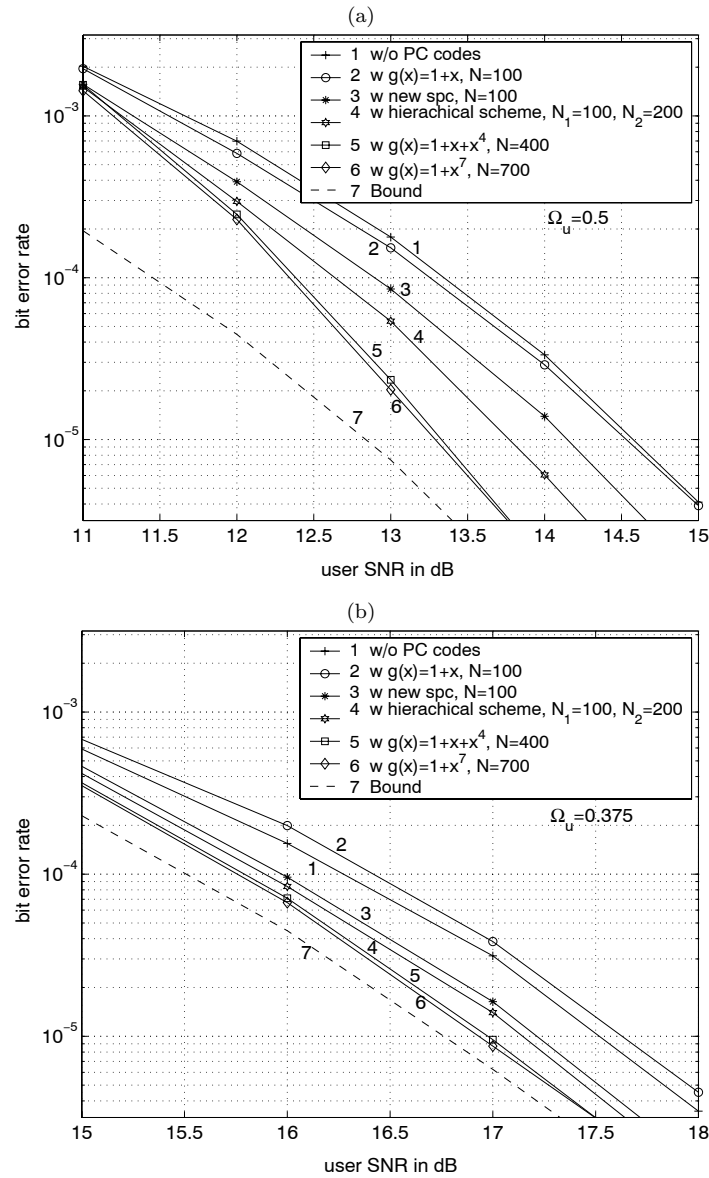


Figure 3.12: BER performance of various parity-check (PC) codes in conjunction with rate 2/3 code. (a) $\Omega_u = 0.5$; (b) $\Omega_u = 0.375$.

by Curve 7. In the post-processor, according to the dominant error events that can be detected by the above PC codes (see Table 3.2), two filters matched to the events $\pm\{2\}$ and $\pm\{2, 0, -2, 0, 2\}$ are used for the conventional single-bit even PC code, three filters matched to the events $\pm\{2\}$, $\pm\{2, 0, -2\}$ and $\pm\{2, 0, -2, 0, 2, 0, -2\}$ are used for the dual-bit PC code, four filters matched to the events $\pm\{2\}$, $\pm\{2, 0, -2\}$, $\pm\{2, 0, -2, 0, 2\}$ and $\pm\{2, 0, 0, -2\}$ are used for the new single-bit PC code and the 2-level hierarchical scheme, and five filters matched to the events $\pm\{2\}$, $\pm\{2, 0, -2\}$, $\pm\{2, 0, -2, 0, 2\}$, $\pm\{2, 0, -2, 0, 2, 0, -2\}$ and $\pm\{2, 0, 0, -2\}$ are used for PC codes with 3 or more parity bits.

From Figure 3.12(a), we observe that compared with the performance of the system without PC code, the conventional single-bit even PC code provides almost no performance gain, since it cannot detect the dominant error event $\pm\{2, 0, -2\}$. The proposed new single-bit code and the 2-level hierarchical scheme provide about 0.4 dB and 0.8 dB gain in performance, respectively, at $\text{BER} = 10^{-5.5}$. The performance of the 2-level hierarchical scheme is superior to that of its two components codes, and is similar to that of the PC code with $g(x) = 1 + x + x^2$. The reason is that the two codes have similar error detection capabilities as well as similar code rate loss³. The PC codes with $g(x) = 1 + x + x^4$ and $g(x) = 1 + x^7$ achieve similar and much larger performance gains, since they can also detect all the dominant error events listed in Table 3.2. The performance is 1.4 dB better than the rate 2/3 code without PC, and is 0.3 dB behind the performance bound. In addition, all the performances with PC codes are obtained using the proposed remedy scheme of Section 3.4.2. It brings around 0.2 dB gain over the performances without remedy, which are not shown here for the sake of brevity.

At $\Omega_u = 0.375$, observe that compared with the results at nominal density, the performance gains are modest for most of the codes. On the other hand, the performance gaps of these codes from the corresponding bound are also reduced. In Figure 3.12(b), the PC code with $g(x) = 1 + x + x^4$ achieves the best performance of 0.7 dB gain compared to that without PC code. It lags behind the bound by around 0.2 dB. The rather modest performance gains observed at high den-

³We do not show the performances of PC codes with $g(x) = 1 + x + x^2$ and $g(x) = 1 + x + x^3$, since they are found to be very close to that of the 2-level hierarchical scheme and the PC code with $g(x) = 1 + x + x^4$, respectively.

sity are due to the specific error event distribution at high density. From Figure 3.4, we observe that there are more non-dominant error events at $\Omega_u = 0.375$ than at $\Omega_u = 0.5$. Therefore, the performance bounds at $\Omega_u = 0.375$ are closer to the performance without parity-check codes. These non-dominant error events are long error events with small probabilities and are mostly caused by consecutive minimum distance transitions (*i.e.* $\pm\{1, 1, -1, -1, \dots\}$). To detect them, more parity bits are needed. They are also difficult to correct, since mis-correction of these long events will introduce many more errors. Using appropriate coding techniques (*e.g.* repeated minimum transition runlength (RMTR) codes) may eliminate the underlying data patterns that support these events and improve the performance of PC codes and post-processing. This is beyond the scope of the thesis.

3.6 Conclusion

In this chapter, we have analyzed bit error rates and error event probabilities of $d = 1$ optical recording channels with Viterbi detection, and derived the analytical bounds on bit error rates, achievable by various PC codes. We proposed an analytical approach to compute the probability of $d = 1$ sequences that support the error events at the VD output. Computer simulations demonstrate that the derived bounds are tight over a broad range of SNRs. The bounds are useful to predict the BER performance of the system, thus saving time-consuming simulations. They further provide helpful guidelines for the development of PC codes and post-processing schemes.

We have investigated various PC codes, including single-bit PC codes and multiple-bit PC codes. Furthermore, a new single-bit PC code and a 2-level hierarchical PC coding scheme have also been proposed. Our study showed that the multi-level hierarchical scheme can be used when efficient multiple-bit PC codes are not available. For error correction, we started with the MAP decision rule and derived a matched-filters based implementation. We further developed a multiple-error-event correction post-processor. We investigated the impact of error events that are split across codeword boundaries, and developed a novel remedy scheme to minimize the mis-correction of such events. Furthermore, we explored the choice of codeword length for a given PC code. The optimum codeword length is around 100

bits per parity bit, for the system under study.

Computer simulations have been carried out to evaluate the overall performance of various PC codes and post-processing schemes, with the corresponding code rate loss taken into consideration. Simulation results showed that the PC code that can detect all the dominant error events of the system, *i.e.* the 4-bit code with $g(x) = 1+x+x^4$, achieves the best performance. The corresponding performances are very close to the performance bounds, at both nominal and high densities. There is no gain with further increase in the number of parity bits.

Chapter 4

Capacity-Approaching $d = 1$ Codes

Constrained codes serve as a key component in the read-write channel of data storage systems. In Chapter 3, for the sake of simplicity, we have used the rate $2/3$ (1,7) code to investigate basic issues associated with parity-check (PC) coded channels. The rate is equal to that of the standard $d = 1$ codes used in the blu-ray disc (BD) and the high-definition digital versatile disc (HD-DVD) [70, 48]. However, the efficiency of these codes is only 96.034% with respect to the theoretical maximal rate of $d = 1$ codes [38], revealing that there is potential for increasing the code rate further. In this chapter, we focus on the design of certain $d = 1$ codes, whose rates are higher than $2/3$ and approach the theoretical maximum to within a few tenths of a percent. We use these codes to study PC-code-based reception schemes in the following chapters of the thesis.

The original contributions in this chapter are as follows.

- We derive analytically the relationship between the number of encoder states and the probable size of certain high efficiency $d = 1$ codes.
- By associating the number of encoder states with (generalized) Fibonacci numbers [99, 36], the minimum number of encoder states is obtained, which maximizes the rate of the designed code, irrespective of the codeword length.
- Our analysis provides direct guidelines for the design of high

efficiency codes with $d = 1$ constraint, as well as other constraints that may be desired for future systems.

An introduction to constrained codes for blue laser disc systems, the prevailing code design techniques, as well as the motivation of this chapter's work are given in Section 4.1. Section 4.2 presents the techniques to design efficient $d = 1$ codes. In Section 4.3, the definition of Fibonacci and generalized Fibonacci numbers, which is used to search for the minimum number of encoder states is given. The analysis on the relationship between the number of encoder states and the probable size of the code is then presented. Thereafter, the minimum number of encoder states is determined. Finally, concluding remarks are given in Section 4.4.

4.1 Introduction

4.1.1 Constrained Codes for Blue Laser Disc Systems

As we mentioned in Chapter 1, the objectives for the use of constrained codes are reduced intersymbol interference (ISI), reduced nonlinear distortions, higher recording density, satisfactory timing recovery, better resistance to low-frequency distortions, and improved detection signal-to-noise ratio (SNR). The ability of constraints to support these objectives depends strongly on the specific characteristics of the recording channel and the receiver under consideration. To achieve the best data recovery performance, it is important to select constraints that are suitable for the system under consideration and to design the constrained encoder and decoder accordingly.

As described in Chapter 1, in read-only optical recording systems, the use of $d = 2$ codes helps to considerably reduce the the impact of domain bloom on the replay signal, by making the minimum mark length equal to 3 bits on the disc. In rewritable blue laser disc systems (*i.e.* rewritable BD and HD-DVD), constrained codes with $d = 1$ constraint¹ have been adopted due to the following reasons. In phase-change rewritable systems, the user data is recorded in terms of crystalline and amorphous marks. When overwriting the old data, differ-

¹In addition to the $d = 1$ constraint, an appropriate k constraint as well as a dc-free constraint are imposed in the blue laser disc systems. For simplicity, they are ignored in this chapter.

ences in optical absorption and thermal properties between the crystalline and amorphous marks due to the old data cause distortions in the newly written data, leading to variations in the effective mark positions. This phenomenon is referred to as overwriting jitter [70, 17], and has been found to be severe at high data rates. For the same user bit length, the length of the channel bit with $d = 1$ codes is larger than that with $d = 2$ codes, since the latter has a lower code rate. Experimental studies show that the increase in channel bit length provides larger timing tolerance against overwriting jitter [70, 17]. In addition, the higher code rates of $d = 1$ codes help to decrease the noise bandwidth and channel density to achieve a better channel SNR (as shown in Chapter 2). Therefore, in this thesis, we focus on developing PC-code-based advanced reception techniques for $d = 1$ optical recording channels. In other words, we aim at stretching the limits with $d = 1$ codes. We remark here that due to the drive for high code rates, in the future, $d = 0$ codes may be considered as a possibility. The algorithms we develop for $d = 1$ codes can be easily generalized to $d = 0$ codes.

For given code constraint(s), the achievable maximum code rate was derived by Shannon and is called the Shannon Capacity [88]. For example, the capacity of (1,7) codes is 0.6793 [38]. The efficiency of a code, referred to as *code efficiency*, is defined as the ratio of the code rate to the Shannon Capacity. Constrained codes whose rates are very close to the capacity (*e.g.* with efficiencies higher than 99%) are called capacity-approaching codes.

In Chapter 3, we focused on the rate $2/3$ (1,7) code, whose rate equals that of the constrained codes used in BD and HD-DVD [70, 48]. The efficiency of the code is $(2/3)/0.6793 = 98.1\%$, revealing that the maximum achievable gain in code rate is 1.9%. Furthermore, significant improvements in timing recovery techniques in recent years enable us to use codes with a larger k constraint. The efficiency of the rate $2/3$ codes is $(2/3)/0.6942 = 96.034\%$ with respect to the capacity of $(1, \infty)$ constraint, showing that there is a maximum of 3.97% gain in coding efficiency that can be explored. Since increase in code rate will bring definite gains in SNR and density, in this chapter, we focus on the design of certain $d = 1$ codes, whose rates are higher than $2/3$ and approach the Shannon Capacity.

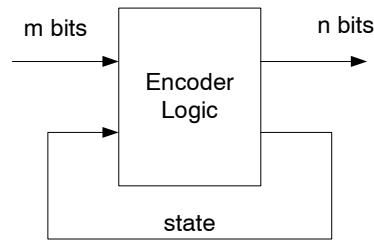


Figure 4.1: *Finite-state encoder.*

4.1.2 Constrained Code Design Techniques

While designing a code with given constraints, the following features are desired: code rate close to the capacity of the constrained system; simple implementation, and nil or minimum error propagation during decoding [38, 59]. There are, in general, no explicit rules for the design of constrained codes. An encoder may be state-dependent or state-independent. In the state-dependent case, the codewords that map to a given user word are functions of the encoder states, while in the state-independent case, this mapping is independent of the encoder states. State-dependent encoders can be represented by a finite-state machine (FSM), which is illustrated in Figure 4.1. In the figure, m and n are the number of bits of the user words and codewords, respectively.

Block encoders [77, 41] are structurally the simplest type of encoders and have played an important role in digital storage systems. They can be considered as state-dependent codes with only one state. Since block encoders do not permit reuse of codewords (*i.e.* mapping the same codeword to more than one user word to achieve high coding efficiency), they are less efficient than state-dependent codes with multiple states.

The decoders of constrained codes are preferred to be state-independent. This is because detection errors caused by the channel detector may make a state-dependent decoder lose track of the encoder state. This causes error propagation during subsequent decoding. The main drawback of state-independent codes lies in the fact that attempts to increase the coding efficiency result in significantly increased codeword lengths and encoder/decoder complexity. Codes

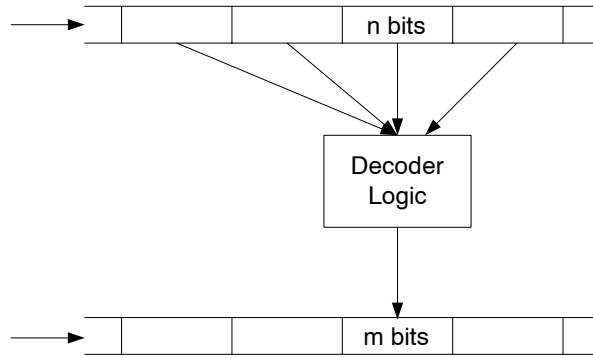


Figure 4.2: *Sliding-block encoder, with $v = 2$ and $u = 1$.*

that can be decoded with sliding-block decoders [58, 4] have been found to be have high efficiencies, simple hardware, and limited error propagation. As shown in Figure 4.2, sliding-block decoders need v preceding codewords and u following codewords to decode the current codeword. Therefore, error propagation is limited to the decoder window of $v + u + 1$ data blocks. Therefore, in general, the problem of constrained code design is to construct efficient finite-state encoders with sliding-block decoders of affordable complexity [37, 59].

Various techniques have been proposed to construct efficient finite-state constrained codes. Among these, the ‘Adler-Coppersmith-Hassner’ (ACH) algorithm, also known as *state-splitting algorithm* [2, 58], provides a systematic way for designing finite-state codes that are sliding-block-decodable. Starting with a finite directed graph (referred to as *labeled graph*) to represent the given constrained system, the state-splitting algorithm uses the *approximate eigenvector equation* [58] to guide the state-splitting during the encoder design. This algorithm gives a constructive proof of the finite-state coding theorem [58] and yields many codes of practical interest [48, 40, 100]. However, there are still many uncertainties in the algorithm, for example, the choice of approximate eigenvector, the state splitting and merging processes, the assignment of codewords to the encoder states to yield optimal encoder/decoder. Considerable potential exists for further research in this direction, or for the development of other code design approaches.

In [42], Immink *et al.* introduced a new family of efficient finite-state constrained codes with $d = 1$ or $d = 2$ constraints, with rates very close to the Shannon capacity. Unlike the *state-splitting method* starting with the *labeled graph*, they [42] propose a simple and efficient finite-state encoding method, which directly specifies the encoding/decoding principles for $d = 1$ and $d = 2$ codes. The rates of the designed codes are only a few tenths of a percent below the capacity. In this chapter, we focus on the investigation of the optimum number of encoder states for this type of capacity-approaching codes.

4.1.3 Motivation for the Current Work

For the design of finite-state constrained codes, the number of encoder states is a key consideration since it directly affects the coding efficiency as well as the encoding/decoding complexity. For finite-state constrained codes, there is not yet a definite solution on how to determine the minimum number of encoder states to maximize the code rate. The *approximate eigenvector equation* guides a variety of code constructions, such as the *state-splitting method*. The sum of the components of the *approximate eigenvector* gives a loose upper bound on the number of encoder states, depending on the code constraints and designed rate [58, 60].

In [42], an efficient FSM is proposed to design capacity-approaching codes with $d = 1$ constraint. Starting with this FSM, a relationship between the probable size of the code and the number of encoder states can be derived. This guides the code design. To obtain the optimum number of encoder states for the designed codes, a brute-force computer exhaustive search approach has been used in [42]. Thus, for each desired codeword length, the optimum number of encoder states is searched separately. Such an approach is time consuming. Furthermore, the obtained number of encoder states may not always be optimum. In this chapter, we derive this relationship by using an analytical approach. We also obtain the minimum number of encoder states, which maximizes the probable size of the designed code, for any desired codeword length. These encoder states help to assign codewords to various encoder states to maximize the code rate. Therefore, our analysis provides direct guidelines for choosing the number of encoder states (or equivalently, for splitting the states of the encoder) for high efficiency codes with $d = 1$ constraint, as well as for

other constraints that may be desired for future recording systems, such as the PC constraint and the repeated minimum transition run-length (RMTR) constraint. The results of this chapter will be used in Chapter 5 to design efficient combined constrained PC codes.

4.2 Design of Capacity-Approaching $d = 1$ Codes

The operation of the finite-state encoder of these capacity-approaching $d = 1$ codes can be represented by a FSM, which is defined by the input set, the output set, the state set, and two logical functions: the output function and the next-state function. The principles of encoding/decoding can be described as follows.

The input set B consists of m -bit user words, with size $|B| = 2^m$. The t^{th} user word in an input sequence is denoted by b_t , where t is an integer, denoting time. The output set X consists of n -bit codewords. The set of codewords, X , is divided into four subsets X_{00} , X_{01} , X_{10} and X_{11} such that the codewords in X_{00} start and end with a '0', codewords in X_{01} start with a '0' and end with a '1', *etc.* The t^{th} codeword in an output sequence is denoted by x_t .

The state set Σ consists of a total of $|\Sigma| = r$ encoder states. It is divided into two state subsets of a first and second type, denoted by Σ_1 and Σ_2 , respectively. The encoder has $|\Sigma_1| = r_1$ states of the first type and $|\Sigma_2| = r_2 = r - r_1$ states of the second type. All codewords in the state subset Σ_1 must start with a '0', while codewords in the state subset Σ_2 can start with either a '0' or a '1'. Here, the state at the time b_t is encoded is denoted by s_t .

The output function h has domain $\Sigma \times B$ and range X . It specifies the following mapping

$$x_t = h(s_t, b_t). \quad (4.1)$$

The output function h can be defined by a simple look-up table, based on the input user word b_t and the encoder state s_t provided by the previous codeword. According to the definitions of the state subsets Σ_1 and Σ_2 , we have

$$x_t \in \begin{cases} \{X_{00}^1 \cup X_{01}^1\} & \text{if } s_t \in \Sigma_1, \\ \{X_{10} \cup X_{11} \cup X_{00}^2 \cup X_{01}^2\} & \text{if } s_t \in \Sigma_2, \end{cases} \quad (4.2)$$

where the sets X_{ab}^c are such that $\{X_{00}^1 \cup X_{01}^1\}$ and $\{X_{00}^2 \cup X_{01}^2\}$ denote the sets of codewords starting with a '0' that are assigned to the first and second state subsets, respectively, and $\{X_{00}^2 \cup X_{01}^2\} = \{X_{00} \cup X_{01}\} \setminus \{X_{00}^1 \cup X_{01}^1\}$. As will become apparent in the following paragraphs, the same codeword cannot be assigned to different states.

The next-state function f has domain $\Sigma \times B$ and range Σ , and it specifies the state of the encoder after transmitting the current codeword. Thus,

$$s_{t+1} = f(s_t, b_t). \quad (4.3)$$

To facilitate reuse of codewords, *i.e.* mapping the same codeword to more than one user word to achieve a high coding efficiency, each codeword may enter more than one encoder state. In particular, codewords that end with a '0', *i.e.* codewords in subsets X_{00} and X_{10} , may enter any of the r encoder states. Codewords that end with a '1' may enter the r_1 states of the first state set only. This prohibits a codeword ending with '1' from entering states of the second type. Hence,

$$s_{t+1} \in \begin{cases} \Sigma & \text{if } x_t \in \{X_{00} \cup X_{10}\}, \\ \Sigma_1 & \text{if } x_t \in \{X_{01} \cup X_{11}\}. \end{cases} \quad (4.4)$$

Due to the reuse of codewords in encoding, to ensure unique decodability, the set of codewords that belongs to a given state must be disjoint. This attribute implies that any codeword can be unambiguously related to the state from which it emerged. During decoding, by observing both the current and the next codewords, the decoder can uniquely decide which user word was actually transmitted. Thus, the output function h is chosen such that

$$b_t = h^{-1}(x_t, s_{t+1}) = h^{-1}(x_t, x_{t+1}). \quad (4.5)$$

Obviously, the corresponding decoders are sliding-block decoders with zero memory and one codeword look-ahead.

4.3 Optimum Number of Encoder States

4.3.1 Fibonacci and Generalized Fibonacci Numbers

Fibonacci numbers [99] play an important role in the study of constrained sequences. They satisfy the following recurrence relation:

$$F(q) \equiv F(q-1) + F(q-2), \quad q \geq 2, \quad F(0) = 0, \quad F(1) = 1. \quad (4.6)$$

The first few Fibonacci numbers are 0, 1, 1, 2, 3, 5, 8, 13, 21, It has been found that the number of distinct $d = 1$ sequences as a function of the sequence length, denoted by $\{N_1(q)\}$, is a Fibonacci sequence (FS) satisfying $N_1(q) = F(q + 2)$ for $q \geq 0$ [38]. There are various types of generalized Fibonacci numbers. Here, we illustrate one type of numbers proposed by Horadam [36], which is related to the FSM studied here. It is defined by

$$G(q) \equiv G(q - 1) + G(q - 2), \quad q \geq 2. \quad (4.7)$$

Obviously, sequences defined by (4.7) are a generalization of the FS defined by (4.6), with arbitrary seeds $G(0)$ and $G(1)$. For instance, with $G(0) = 2$ and $G(1) = 1$, the numbers generated are called Lucas numbers, which are denoted by $L(q)$. Note that the FS can be considered as a special case of the above defined generalized Fibonacci sequence (GFS).

4.3.2 Relationship Between Encoder States and Code Size

To determine the minimum number of encoder states for the designed codes, we start with specific conditions for the code construction, which are derived from the FSM described in Section 4.2. The conditions are given by

$$r |X_{00}| + r_1 |X_{01}| \geq r_1 M, \quad (4.8)$$

$$r (|X_{00}| + |X_{10}|) + r_1 (|X_{01}| + |X_{11}|) \geq r M, \quad (4.9)$$

where M is referred to as the probable size of the code, which is essentially the maximum number of user words that the encoder may accommodate, associated with a given number of encoder states. The above inequalities specify that for a fixed-length code with probable size M , the number of codewords leaving a state set, counting multiplicity, should be at least M times the number of states within the state set. Therefore, they are equivalent to the *approximate eigenvector equation*, and are necessary conditions for code construction. Thus, for a desired code rate and number of encoder states, if (4.8) or (4.9) fail, a code cannot be constructed. If (4.8) and (4.9) both hold, we can proceed to allocate the valid codewords to various encoder states. To do this, as described in Section 4.2, the valid codewords

(counting multiplicity) should be assigned to the encoder states such that there is no overlapping of codewords between different states. The code construction will fail if such an allocation of codewords is not possible. In such cases, the actual number of user words that the encoder can accommodate will be smaller than M .

To facilitate successful allocation of codewords and achieve high coding efficiency, a large value of M , associated with a small number of encoder states, is highly desirable. This is due to the following reason. In general, the larger the value of M and/or the smaller the associated number of encoder states are/is, the more easy it will be to successfully allocate the codewords without incurring the overlap of codewords between different encoder states, and vice versa. In addition, a large value of M may also help to impose other modulation constraints, such as a k constraint, dc-free constraint, maximum transition run (MTR) constraint [69, 72, 102] and PC constraint [12, 103]. Therefore, in the following sections, we focus on searching the minimum number of encoder states, which maximize the value of M , for any given codeword length.

Note that in (4.8) and (4.9), the choice of the number of encoder states r and r_1 determines the value of M , for a given codeword length n . Therefore, in this section, we explore all possible choices for r and r_1 , and derive common properties for these choices.

Proposition 1: For given positive integers r , r_1 and $r_2 = r - r_1$, there always exist a unique GFS $\{G(l)\}$ and a unique integer $q \geq 2$ with the following properties:

- 1) $G(q) = r$, $G(q - 1) = r_1$, and $G(q - 2) = r_2$;
- 2) $G(0) \geq G(1) > 0$.

Proof: Due to the two-term recurrence nature of the GFS (see (4.7)), for any given r and r_1 , we can always define $G(q) = r$ and $G(q - 1) = r_1$, and thereby generate a GFS based on these two numbers.

To expose the effect of various choices of r and r_1 on M , a judicious selection of indices for the corresponding GFS is crucial. Since r , r_1 and r_2 are all positive integers, without loss of any generality, we define $G(0) > 0$, $G(1) > 0$ and $G(-1) \leq 0$ for all the associated GFSs. Thus, we have $G(0) \geq G(1) > 0$. In each of these GFSs, with $q \geq 2$, any consecutive three numbers $[r_2 = G(q - 2), r_1 = G(q - 1), r = G(q)]$ represent a possible encoder state combination. \square

Remarks:

- In Proposition 1, we consider r_1 and r_2 as positive integers, and the obtained codes are sliding block codes. In principle, we could also set r_1 or r_2 to zero. In such cases, the encoder has only one type of state, and the resulting codes are block codes. With $r_1 = 0$, conditions (4.8) and (4.9) reduce to

$$|X_{00}| + |X_{10}| \geq M. \quad (4.10)$$

The associated codewords are free to start with either ‘0’ or ‘1’, but must end with a ‘0’. Similarly, with $r_2 = 0$, we obtain the condition

$$|X_{00}| + |X_{01}| \geq M. \quad (4.11)$$

The corresponding codewords start with a ‘0’, and end with either ‘0’ or ‘1’. Obviously, the efficiency this type of codes will be lower than that of the sliding block codes obtained with $r_1 > 0$ and $r_2 > 0$, since we do not reuse codewords during the code construction. The disadvantage of setting $r_1 = 0$ or $r_2 = 0$ on M will be further shown in the next section.

We use $\{G^i(q)\}$ to denote the i^{th} sequence in the set of all GFSs, \mathbf{G} , defined by Proposition 1, with $i = 1, 2, \dots, K$ and $K = |\mathbf{G}|$. Note that K is finite, since the number of encoder states cannot be infinite. Note also that, hereafter, the GFSs under consideration are all within the sequence set \mathbf{G} . The basic GFSs, with $G^i(2) \leq 10$ and $0 \leq q \leq 11$, are shown in Table 4.1. Here, sequences that are integer multiples of the basic GFSs shown in Table 4.1 are not included since, as proved by Corollary 1 of the next section, they produce the same M as the corresponding basic GFSs. Note that the first sequence $\{G^1(q)\}$ corresponds to the shifted Fibonacci numbers $\{F'(q)\}$, with $F'(q) = F(q+1)$. The second sequence $\{G^2(q)\}$ corresponds to the Lucas numbers $\{L(q)\}$.

Lemma 1: For the GFS $\{G^i(q)\}$ associated with any given r and r_1 as defined by Proposition 1, there exist $G^i(2)$ and $G^i(1)$ such that $G^i(2) \geq 2G^i(1) > 0$ for $i = 1, 2, \dots, K$.

Proof: According to Proposition 1, $G^i(0) \geq G^i(1) > 0$, for any $i = 1, 2, \dots, K$. Further, $G^i(2) = G^i(0) + G^i(1)$. Therefore, we obtain $G^i(2) \geq 2G^i(1) > 0$. \square

Table 4.1: Basic sequences from the set of generalized Fibonacci sequences (GFS) $\{G^i(q)\}$, for $G^i(2) \leq 10$ and $0 \leq q \leq 11$.

$\begin{matrix} q \\ \{G^i(q)\} \end{matrix}$	0	1	2	3	4	5	6	7	8	9	10	11
$\{G^1(q)\}$	1	1	2	3	5	8	13	21	34	55	89	144
$\{G^2(q)\}$	2	1	3	4	7	11	18	29	47	76	123	199
$\{G^3(q)\}$	3	1	4	5	9	14	23	37	60	97	157	254
$\{G^4(q)\}$	4	1	5	6	11	17	28	45	73	118	191	309
$\{G^5(q)\}$	5	1	6	7	13	20	33	53	86	139	225	364
$\{G^6(q)\}$	6	1	7	8	15	23	38	61	99	160	259	419
$\{G^7(q)\}$	7	1	8	9	17	26	43	69	112	181	293	474
$\{G^8(q)\}$	8	1	9	10	19	29	48	77	125	202	327	529
$\{G^9(q)\}$	9	1	10	11	21	32	53	85	138	223	361	584
$\{G^{10}(q)\}$	3	2	5	7	12	19	31	50	81	131	212	343
$\{G^{11}(q)\}$	5	2	7	9	16	25	41	66	107	173	280	453
$\{G^{12}(q)\}$	7	2	9	11	20	31	51	82	133	215	348	563
$\{G^{13}(q)\}$	4	3	7	10	17	27	44	71	115	186	301	487
$\{G^{14}(q)\}$	5	3	8	11	19	30	49	79	128	207	335	542
$\{G^{15}(q)\}$	7	3	10	13	23	36	59	95	154	249	403	652
$\{G^{16}(q)\}$	5	4	9	13	22	35	57	92	149	241	390	631

Theorem 1: For given numbers of encoder states r and r_1 , and the associated $G^i(q) = r$ and $G^i(q-1) = r_1$ with $q \geq 2$, as defined by Proposition 1,

$$M = \begin{cases} \left\lfloor \frac{G^i(n+q)}{G^i(q)} \right\rfloor, & \text{if } q \text{ is even,} \\ \left\lfloor \frac{G^i(n+q-1)}{G^i(q-1)} \right\rfloor, & \text{if } q \text{ is odd.} \end{cases} \quad (4.12)$$

Proof: It can be easily shown [38] that

$$|X_{00}| = N_1(n-2), \quad (4.13)$$

$$|X_{01}| = |X_{10}| = N_1(n-3), \quad (4.14)$$

$$|X_{11}| = N_1(n-4). \quad (4.15)$$

Further, with $G^i(q) = r$ and $G^i(q-1) = r_1$, we can rewrite (4.8) and (4.9) as

$$G^i(q)N_1(n-2) + G^i(q-1)N_1(n-3) \geq G^i(q-1)M,$$

$$G^i(q)N_1(n-1) + G^i(q-1)N_1(n-2) \geq G^i(q)M.$$

By induction, it can be shown that [98]

$$G^i(q)N_1(n-1) + G^i(q-1)N_1(n-2) = G^i(n+q). \quad (4.16)$$

Combining (4.16), (4.16) and (4.16), we get

$$M = \min \left(\left\lfloor \frac{G^i(n+q-1)}{G^i(q-1)} \right\rfloor, \left\lfloor \frac{G^i(n+q)}{G^i(q)} \right\rfloor \right). \quad (4.17)$$

To compare the values of $\frac{G^i(n+q-1)}{G^i(q-1)}$ and $\frac{G^i(n+q)}{G^i(q)}$ for any given $q \geq 2$ and $n \geq 1$, we define

$$\begin{aligned} \zeta_q &\equiv G^i(n+q-1)G^i(q) - G^i(n+q)G^i(q-1) \\ &= G^i(q-2)G^i(n+q-3) \\ &\quad + (G^i(q) - 2G^i(q-1))G^i(n+q-2). \end{aligned} \quad (4.18)$$

When $q = 2$, we have

$$\begin{aligned} \zeta_2 &= G^i(n+1)G^i(2) - G^i(n+2)G^i(1) \\ &= G^i(0)G^i(n-1) + (G^i(2) - 2G^i(1))G^i(n). \end{aligned} \quad (4.19)$$

From Lemma 1, we get $G^i(2) \geq 2G^i(1)$ for the GFS associated with any given r and r_1 . Further, $G^i(n-1) > 0$ and $G^i(n) > 0$, for $n \geq 1$. Therefore, we obtain

$$\zeta_2 > 0, \quad \text{for } n \geq 1. \quad (4.20)$$

Furthermore, since

$$\begin{aligned} \zeta_{q+1} &= G^i(n+q)G^i(q+1) - G^i(n+q+1)G^i(q) \\ &= - (G^i(n+q-1)G^i(q) - G^i(n+q)G^i(q-1)), \end{aligned} \quad (4.21)$$

we obtain

$$\zeta_{q+1} = -\zeta_q, \quad \text{for } q \geq 2. \quad (4.22)$$

From (4.20) and (4.22), we get $\zeta_3 < 0$, $\zeta_4 > 0$, and so on. Therefore, we conclude that

$$\begin{cases} \zeta_q > 0, & \text{if } q \text{ is even,} \\ \zeta_q < 0, & \text{if } q \text{ is odd,} \end{cases} \quad (4.23)$$

for any $n \geq 1$. This proves (4.12). \square

In particular, choosing the number of encoder states to be the Fibonacci numbers $r = F'(q)$ and $r_1 = F'(q - 1)$, we can rewrite (4.18) as

$$\zeta'_q = F'(n + q - 1)F'(q) - F'(n + q)F'(q - 1). \quad (4.24)$$

Furthermore, according to d'Ocagne's identity [98], we have

$$F'(n + q - 1)F'(q) - F'(n + q)F'(q - 1) = (-1)^q F'(n - 1). \quad (4.25)$$

Obviously, (4.25) coincides with (4.23). This provides another proof of (4.12), for M associated with the FS $\{F'(q)\}$.

4.3.3 Minimum Number of Encoder States

In this section, we search the minimum numbers of encoder states r and r_1 that maximize M , for any desired codeword length. For ease of derivation, we use $M^i(q, n) = \min\left(\frac{G^i(n+q-1)}{G^i(q-1)}, \frac{G^i(n+q)}{G^i(q)}\right)$ to denote M in fractional format (*i.e.* without applying the floor operator $\lfloor \cdot \rfloor$ in (4.17)) associated with the i^{th} GFS in the sequence set \mathbf{G} , and for any given q and n .

From Theorem 1, we conclude the following.

Corollary 1: While comparing the values of M generated by different choices of the number of encoder states r and r_1 , it is sufficient to use $n = 1$. The same trend in M exists for other codeword lengths also.

Proof: From Theorem 1, we know that for given $G^i(q) = r$ and $G^i(q - 1) = r_1$, depending on whether q is even or odd, $M^i(q, n)$ is either $\frac{G^i(n+q)}{G^i(q)}$ or $\frac{G^i(n+q-1)}{G^i(q-1)}$. Noting that $M^i(q, 0) = 1$ for any q , we further have

$$M^i(q, n) = M^i(q, n - 1) + M^i(q, n - 2), \quad \text{for } n \geq 2. \quad (4.26)$$

Thus, for a given q , the sequence $M^i(q, n)$ can be viewed as a GFS in the fractional format, with seeds $M^i(q, 0) = 1$ and $M^i(q, 1)$. Therefore, to compare the M generated by different choices of r and r_1 , it is sufficient to compare the corresponding $M^i(q, 1)$. \square

From Corollary 1, we can also conclude that for any given q and n , with the same $M^i(q, 1)$, integer multiples of the basic GFSs produce the same M as the corresponding basic sequences.

Corollary 2: With a given number of encoder states $r = r_1 + r_2$, by choosing $r_1 \geq r_2$, we always obtain a larger M than that with $r_1 < r_2$, for any codeword length n .

Proof: Assume two positive integers a and b with $a \geq b$. Without loss of generality, we prove Corollary 2 by comparing the M of the encoder having $[r_2 = G^i(q_1 - 2) = a, r_1 = G^i(q_1 - 1) = b, r = G^i(q_1) = a + b]$ with that of the encoder having $[r_2 = G^j(q_2 - 2) = b, r_1 = G^j(q_2 - 1) = a, r = G^j(q_2) = a + b]$, for $q_1, q_2 \geq 2$, $i, j = 1, 2, \dots, K$, and $i \neq j$.

For the encoder with $[r_2 = a, r_1 = b, r = a + b]$, we obtain $r \geq 2r_1$. Thus, $G^i(q_1) \geq 2G^i(q_1 - 1)$. By (4.18), we obtain $\zeta_{q_1} > 0$. Therefore,

$$M^i(q_1, 1) = \frac{G^i(n + q_1)}{G^i(q_1)} \Big|_{n=1} = \frac{a + b + b}{a + b} = 1 + \frac{b}{a + b}. \quad (4.27)$$

For the encoder with $[r_2 = b, r_1 = a, r = a + b]$, from Theorem 1, we obtain either

$$M^j(q_2, 1) = \frac{G^i(n + q_2 - 1)}{G^i(q_2 - 1)} \Big|_{n=1} = \frac{a + b}{a} = 1 + \frac{b}{a}, \quad (4.28)$$

or

$$M^j(q_2, 1) = \frac{G^i(n + q_2)}{G^i(q_2)} \Big|_{n=1} = \frac{a + b + a}{a + b} = 1 + \frac{a}{a + b}. \quad (4.29)$$

Comparing (4.27) with (4.28) and (4.29) shows that $M^j(q_2, 1) \geq M^i(q_1, 1)$. The equality holds only when $a = b$. Therefore, Corollary 2 is proved for the case of $n = 1$. Furthermore, using Corollary 1, we conclude that Corollary 2 is also true for any $n \geq 1$. \square

Corollary 3: For a GFS $\{G^i(q)\}$ defined by $G^i(q) = r$ and $G^i(q - 1) = r_1$, and with any codeword length n , increasing q from an even integer $q = 2p$, with $p \geq 1$, to an odd integer $q = 2p + 1$, results in the same M .

Proof: From (4.12), it is straightforward to get

$$M^i(2p, n) = M^i(2p + 1, n), \quad (4.30)$$

for all integers $p \geq 1$ and $n \geq 1$. \square

Therefore, in each GFS shown in Table 4.1, setting r to be the two numbers within each pair of columns (with $q \geq 2$) results in

the same M , irrespective of the codeword length. That is, increasing the encoder complexity from $r = G^i(2p)$ to $r = G^i(2p + 1)$ does not increase M .

Corollary 4: For all GFSs $\{G^i(q)\}$ defined by $G^i(q) = r$ and $G^i(q - 1) = r_1$, and with even integers $q = 2p$ and $p \geq 1$, M increases with increase in p , whether or not the new encoder states are still within the original GFSs, and irrespective of the codeword length n .

Proof: What we want to prove is

$$M^i(2p + 2, n) > M^j(2p, n), \quad (4.31)$$

for all integers $p \geq 1$ and $n \geq 1$, and with any $i, j = 1, 2, \dots, K$. We first prove (4.31) for the case $n = 1$ by induction. For $p = 1$, by Theorem 1, we have

$$M^i(2p + 2, 1) \big|_{p=1} = \frac{G^i(5)}{G^i(4)} = 1 + \frac{1}{1 + \frac{1}{\frac{G^i(3)}{G^i(2)}}}, \quad (4.32)$$

and

$$M^j(2p, 1) \big|_{p=1} = \frac{G^j(3)}{G^j(2)} = 1 + \frac{1}{1 + \frac{1}{\frac{G^j(1)}{G^j(0)}}}. \quad (4.33)$$

According to Proposition 1, $\frac{G^j(1)}{G^j(0)} \leq 1$. We further have $\frac{G^i(3)}{G^i(2)} > 1$, since $G^i(3) = G^i(2) + G^i(1)$, and $G^i(1) > 0$. We thus obtain

$$\frac{G^i(3)}{G^i(2)} > \frac{G^j(1)}{G^j(0)}. \quad (4.34)$$

Combining (4.32), (4.33) and (4.34), we get

$$M^i(2p + 2, 1) > M^j(2p, 1), \quad \text{for } p = 1. \quad (4.35)$$

Now, for $p = l$, suppose

$$M^i(2p + 2, 1) > M^j(2p, 1), \quad \text{with } l \geq 1. \quad (4.36)$$

Then, we get

$$\frac{G^i(2l + 3)}{G^i(2l + 2)} > \frac{G^j(2l + 1)}{G^j(2l)}. \quad (4.37)$$

Further, for $p = l + 1$, we have

$$M^i(2p + 2, 1) \big|_{p=l+1} = \frac{G^i(2l + 5)}{G^i(2l + 4)} = 1 + \frac{1}{1 + \frac{1}{\frac{G^i(2l+3)}{G^i(2l+2)}}}, \quad (4.38)$$

and

$$M^j(2p, 1) \big|_{p=l+1} = \frac{G^j(2l + 3)}{G^j(2l + 2)} = 1 + \frac{1}{1 + \frac{1}{\frac{G^j(2l+1)}{G^j(2l)}}}. \quad (4.39)$$

Combining (4.37), (4.38) and (4.39), we obtain

$$M^i(2p + 2, 1) > M^j(2p, 1), \quad \text{for } p = l + 1. \quad (4.40)$$

Thus, we prove Corollary 4 for $n = 1$. According to Corollary 1, we conclude that the statement is true for other codeword lengths also. \square

Corollary 5: By choosing the number of encoder states as $r = F'(q)$ and $r_1 = F'(q-1)$, we always obtain a larger M than that with r and r_1 being the q^{th} and $(q-1)^{\text{th}}$ elements of other GFSs, for any $q \geq 2$ and any codeword length n . Furthermore, the corresponding number of encoder states is always smaller than that resulting from other GFSs.

Proof: To prove Corollary 5, we use the following properties of the FS $\{F'(q)\}$.

- *Property 1:* $\frac{F'(1)}{F'(0)} > \frac{G^{i \neq 1}(1)}{G^{i \neq 1}(0)}$.

Proof: According to Proposition 1, in the set \mathbf{G} of GFSs, only the FS $\{F'(q)\}$ and its integer multiples satisfy $F'(0) = F'(1)$. All the other GFSs result in $G^{i \neq 1}(0) > G^{i \neq 1}(1)$. \square

- *Property 2:* $F'(q) < G^{i \neq 1}(q)$, with $q \geq 2$.

Proof: This is due to the reason that the seeds, *i.e.* $F'(0) = 1$ and $F'(1) = 1$, are the smallest among all positive integers that can be used as seeds for the GFSs in the set \mathbf{G} . According to the recurrence relation (4.7) for GFS, the q^{th} element of the FS is always smaller than the q^{th} elements of other GFSs in \mathbf{G} , for $q \geq 2$. \square

We first prove

$$M^1(q, n) > M^{i \neq 1}(q, n), \quad (4.41)$$

for any $q \geq 2$ and $n \geq 1$. We start with $n = 1$ and even q , and use induction. For $q = 2$, by Theorem 1, we have

$$M^1(q, 1) |_{q=2} = \frac{F'(3)}{F'(2)} = 1 + \frac{1}{1 + \frac{1}{\frac{F'(1)}{F'(0)}}}, \quad (4.42)$$

and

$$M^{i \neq 1}(q, 1) |_{q=2} = \frac{G^{i \neq 1}(3)}{G^{i \neq 1}(2)} = 1 + \frac{1}{1 + \frac{1}{\frac{G^{i \neq 1}(1)}{G^{i \neq 1}(0)}}}. \quad (4.43)$$

According to Property 1, we get

$$M^1(q, 1) > M^{i \neq 1}(q, 1), \quad \text{for } q = 2. \quad (4.44)$$

Now, for $q = 2l$, assume that

$$M^1(q, 1) > M^{i \neq 1}(q, 1), \quad \text{with } l \geq 1. \quad (4.45)$$

Then, we get

$$\frac{F'(2l+1)}{F'(2l)} > \frac{G^{i \neq 1}(2l+1)}{G^{i \neq 1}(2l)}. \quad (4.46)$$

For $q = 2l + 2$, we have

$$M^1(q, 1) |_{q=2l+2} = \frac{F'(2l+3)}{F'(2l+2)} = 1 + \frac{1}{1 + \frac{1}{\frac{F'(2l+1)}{F'(2l)}}}, \quad (4.47)$$

and

$$M^{i \neq 1}(q, 1) |_{q=2l+2} = \frac{G^{i \neq 1}(2l+3)}{G^{i \neq 1}(2l+2)} = 1 + \frac{1}{1 + \frac{1}{\frac{G^{i \neq 1}(2l+1)}{G^{i \neq 1}(2l)}}}. \quad (4.48)$$

Combining (4.46), (4.47) and (4.48), we obtain

$$M^1(q, 1) > M^{i \neq 1}(q, 1), \quad \text{for } q = 2l + 2. \quad (4.49)$$

Thus, we prove (4.41) for the case of $n = 1$ and even integers $q = 2l$. It also holds for odd integers $q = 2l + 1$, since $M^i(2l, 1) = M^i(2l + 1, 1)$ according to Corollary 3. Further, using Corollary 1, we conclude that (4.41) is also true for other codeword lengths. Combining (4.41) and Property 2, we thus prove Corollary 5. \square

Corollary 6: For given $r = F'(q_1)$, choosing $r_1 = F'(q_1 - 1)$ results in a larger M than that with r_1 being an element of other GFSs (*i.e.* $r = G^{i \neq 1}(q_2) = F'(q_1)$ and $r_1 = G^{i \neq 1}(q_2 - 1)$, for any $q_1, q_2 \geq 2$), for any codeword length n .

Proof: With $r = G^{i \neq 1}(q_2) = F'(q_1)$, Property 2 of Corollary 5 gives $q_1 > q_2$. According to Corollaries 3 and 4, $M^1(q_1, n) \geq M^1(q_2, n)$, with $q_1 > q_2$. Furthermore, by Corollary 5, we have $M^1(q_2, n) > M^{i \neq 1}(q_2, n)$. Therefore, we obtain $M^1(q_1, n) > M^{i \neq 1}(q_2, n)$ for any $q_1, q_2 \geq 2$ and $n \geq 1$. \square

Remarks:

- In the case of $r_1 = 0$ or $r_2 = 0$, the associated basic GFS $\{F''(q)\}$ can be defined as

$$\begin{array}{cccccccc} q & 0 & 1 & 2 & 3 & 4 & 5 & 6 & \dots \\ \{F''(q)\} & 1 & 0 & 1 & 1 & 2 & 3 & 5 & \dots \end{array}$$

This is again a shifted FS defined by $F''(q) = F(q - 1)$. The encoder state combinations associated with this FS for $r_1 = 0$ and $r_2 = 0$ are given by $[r_2 = F''(0), r_1 = F''(1), r = F''(2)]$ and $[r_2 = F''(1), r_1 = F''(2), r = F''(3)]$, respectively. Theorem 1 still holds for this sequence since $F''(2) > 2F''(1)$. Therefore, according to Corollary 3, we obtain

$$M''(2, n) = M''(3, n), \quad (4.50)$$

with $M''(q, n) = \min\left(\frac{F''(n+q-1)}{F''(q-1)}, \frac{F''(n+q)}{F''(q)}\right)$. Further, since

$$\frac{G^i(1)}{G^i(0)} > \frac{F''(1)}{F''(0)}, \quad \text{for any } i = 1, 2, \dots, K, \quad (4.51)$$

following a derivation similar to the proof of Corollary 5, we get

$$M^i(2, n) > M''(2, n), \quad \text{for any } n \geq 1. \quad (4.52)$$

In addition, from Corollaries 3 and 4, we have

$$M^i(q, n) \geq M^j(2, n), \quad (4.53)$$

for all integers $q > 2$, $n \geq 1$, and with any $i, j = 1, 2, \dots, K$. Combining (4.50), (4.52) and (4.53), we conclude that choosing $r_1 = 0$ or $r_2 = 0$ results in the same M , which is smaller than that with r_1 and r_2 being positive integers, irrespective of the codeword length n .

- Corollaries 5 and 6 show the properties of M associated with the FS $\{F'(q)\}$. It may also be interesting to point out that among other GFSs $\{G^{i \neq 1}(q)\}$ in the sequence set \mathbf{G} , and for a given $q_1 = 2p + 2$ and $p \geq 1$, the Lucas sequence $\{L(q)\}$ provides a M larger than all those associated with $q_2 = 2p$, with the fewest encoder states. This is because, according to Corollary 4, we have $M^{i \neq 1}(2p + 2, n) > M^j(2p, n)$. On the other hand, the seeds of the Lucas sequence, *i.e.* $L(0) = 2$ and $L(1) = 1$, are the smallest among all positive integers that can be used as the seeds for the GFSs $\{G^{i \neq 1}(q)\}$ in \mathbf{G} . Due to the recurrence relation (4.7) of the GFS, we obtain $L(q) < G^{i \neq 1, 2}(q)$ for any $q \geq 2$.

The above corollaries are combined into the following Theorem.

Theorem 2: For any given codeword length n , choosing the number of encoder states to be the Fibonacci numbers $r = F'(q)$ and $r_1 = F'(q - 1)$, where q is a positive even integer, always gives a locally maximum M with the minimum number of encoder states. The value of M increases with increase in q , and saturates at the global maximum $\lfloor 2^{nC(1, \infty)} \rfloor$.

Proof: From Corollary 2, we know that for a given r , we should choose $r_1 \geq r_2 = r - r_1$ to get a larger M . From Corollary 3, we know that for all GFSs defined by Proposition 1 in the sequence set \mathbf{G} , choosing the sequence index q such that $q = 2p$ and $p \geq 1$ results in the same M as that with $q = 2p + 1$, and choosing $q = 2p$ will result in fewer number of states. Furthermore, Corollary 4 shows that the value of M increases with increase in p , for all GFSs in \mathbf{G} . From Corollary 5, we further know that by choosing the number of encoder states to be the Fibonacci numbers with $r = F'(q)$ and $r_1 = F'(q - 1)$, we always obtain a larger M than that with r and r_1 being the q^{th} and $(q - 1)^{\text{th}}$ elements of other GFSs in \mathbf{G} , for any $q \geq 2$. The associated number

of encoder states is always smaller than that chosen from other GFSs. Finally, Corollary 6 shows that with the same number of encoder states r , which is a Fibonacci number $F'(q)$, choosing r_1 to be the adjacent number $F'(q - 1)$ of the same FS will result in a larger M than that with r_1 being an element of other GFSs in \mathbf{G} . In addition, all these statements are true irrespective of the codeword length. Therefore, to achieve the maximum M (either locally or globally) with minimum number of encoder states, we should choose the number of encoder states as $r = F'(2p)$ and $r_1 = F'(2p - 1)$, with $p \geq 1$, for any desired codeword length.

The global maximum of M is given by

$$M_{max} = \left\lfloor \lim_{q \rightarrow \infty} \frac{F'(n+q)}{F'(q)} \right\rfloor. \quad (4.54)$$

For the FS, we have [98]

$$\lim_{q \rightarrow \infty} \frac{F'(q+1)}{F'(q)} = \tau, \quad (4.55)$$

where $\tau = \frac{1+\sqrt{5}}{2}$ is known as the golden ratio [98]. Following (4.55), we get

$$\lim_{q \rightarrow \infty} \frac{F'(n+q)}{F'(q)} = \lim_{q \rightarrow \infty} \frac{F'(q+1)}{F'(q)} \frac{F'(q+2)}{F'(q+1)} \cdots \frac{F'(q+n)}{F'(q+n-1)} = \tau^n.$$

With $C(1, \infty) = \log_2\left(\frac{1+\sqrt{5}}{2}\right)$ [38], we then obtain

$$M_{max} = \left\lfloor \left(\frac{1+\sqrt{5}}{2} \right)^n \right\rfloor = \lfloor 2^{nC(1, \infty)} \rfloor. \quad (4.56)$$

Note that in (4.56), $\lfloor 2^{nC(1, \infty)} \rfloor$ is indeed the theoretical limit of the size of $d = 1$ codes with length n . \square

In summary, we conclude that choosing r and r_1 as Fibonacci numbers with $[r_2 = 1, r_1 = 1, r = 2]$, $[r_2 = 2, r_1 = 3, r = 5]$, $[r_2 = 5, r_1 = 8, r = 13]$, *etc*, gives a locally maximum M , with minimum number of encoder states. The value of M increases with increase in r , until it saturates at the global maximum $M_{max} = \lfloor 2^{nC(1, \infty)} \rfloor$. These choices of encoder states help to allocate the valid codewords to the encoder states so as to maximize the code rate. The final choice

of r depends on the desired code rate, the code constraints, and the affordable implementation complexity. For example, based on the above analysis, we find that with a codeword length of $n = 13$, a 5-state encoder with $[r_2 = 2, r_1 = 3, r = 5]$ provides $M = \lfloor M^1(4, 13) \rfloor = 516$. It can be verified that these states enable an effective allocation of codewords to accommodate $|B| = M = 516$ user words. As a result, a rate $9/13$ (1,18) code can be designed, whose rate is 3.85% higher than that of the rate $2/3$ $d = 1$ codes used for BD and HD-DVD. Furthermore, a 13-state encoder with $[r_2 = 5, r_1 = 8, r = 13]$ generates $M = \lfloor M^1(6, 13) \rfloor = 520$. This results in a code with size $|B| = M = 520$, which approaches the theoretical limit of $\lfloor 2^{13C(1, \infty)} \rfloor = 521$. The excess codewords can be used to reduce the k constraint of the rate $9/13$ code to $k = 14$. Since these $d = 1$ codes achieve a 99.73% code efficiency, they are used to investigate the PC-code-based optical recording systems in the remaining chapters of the thesis.

We remark here that by sacrificing a certain amount of code rate with respect to the rate $9/13$, a RMTR constraint that is stricter than that of the 17PP and ETM codes may further be imposed on the $d = 1$ coded channel bit-stream. The final error rate performance is a compromise between the gain from the stricter RMTR constraint and additional code rate loss. The investigation on the RMTR constraint is beyond the scope of this thesis.

4.4 Conclusion

In this chapter, we have analytically investigated the relationship between the number of encoder states and the probable size of certain capacity-approaching $d = 1$ codes. We have found that the number of encoder states can always be associated with generalized Fibonacci numbers. Choosing the number of encoder states to be specific Fibonacci numbers maximizes the probable size of the designed code with the minimum number of states, for any desired codeword length. These states, in general, also enable successful allocation of codewords to the encoder states to maximize the code rate. Our analysis provides direct guidelines for the design of capacity-approaching constrained codes with $d = 1$ constraint. This analysis can be generalized to other finite-state constrained codes as well.

Chapter 5

Capacity-Approaching Constrained Parity-Check Codes

As mentioned in Chapter 3, the purpose of parity-check (PC) codes is to detect the presence of errors and to provide localization in error type and position. This is done by imposing specific PC constraints (in addition to the modulation constraints), thereby incurring additional code rate loss. Therefore, the design of efficient constrained PC codes (*i.e.* codes with modulation constraints and PC constraints) is key to the development of PC-code-based recording systems. Although several attempts have been reported in recent years to efficiently combine constrained codes and PC codes [80, 32, 20, 16], the systematic design of efficient constrained PC codes is still an open problem. In particular, for optical recording systems, it is difficult to design efficient constrained PC codes that satisfy multiple-bit PC constraints. In this chapter, we solve this code design bottle-neck and design novel capacity-approaching constrained PC codes. In particular, a general and systematic code design approach is proposed to efficiently combine constrained codes with various PC codes. Performance of the newly designed codes is evaluated using bit error rate (BER) simulations.

The original contributions in this chapter are as follows.

- We propose a general and systematic way to efficiently combine constrained codes with PC codes for optical recording. The proposed constrained PC codes include two component codes: a

normal constrained (NC) code and a parity-related constrained (PRC) code, and are designed based on the same finite state machine (FSM). The code rates are only a few tenths of a percent below the theoretical maximum. The parity-check constraint can be chosen to detect any type of dominant error events in the system.

- Approaches are proposed for designing the code in the non-return-to-zero-inverse (NRZI) format and the non-return-to-zero (NRZ) format. Designing the codes in NRZ format may reduce the number of parity bits required for error detection and may simplify post-processing for error correction.

The chapter is organized as follows. A survey of prior art coding schemes is presented in Section 5.1. The general principle of the new code design is presented in Section 5.2. Methods for designing codes in NRZI and NRZ formats are presented in Section 5.3 and Section 5.4, respectively. In Section 5.5, examples of several newly designed codes are illustrated. Their BER performances are presented in Section 5.6. Conclusions are presented in Section 5.7.

5.1 Introduction

There have been several attempts in recent years to efficiently combine constrained codes with PC codes. For example, in the scheme described by Perry *et al.* [80], a constrained data sequence is parsed into shorter blocks of equal length, and a parity data block is inserted between each pair of these blocks. The data and parity blocks are connected such that the modulation constraints are not violated. The major disadvantage of this scheme is that it can only correct specific mixed-type errors.

Gopaldaswamy and Bergmans [32] proposed concatenated coding to construct modulation codes with error detection capability. In this scheme, the parity-check information is first calculated for each constrained data block. This information is then encoded by a standard constrained encoder and appended to the end of the corresponding data block. In this way, the proposed scheme achieves high coding efficiency. For the rate $2/3$ (1,7) code, a parity bit requires 1.5 channel bits. However, in this scheme, the channel bit-stream corresponding

to the parity bits is not protected by parity-checks. Therefore, errors occurring in this portion may cause further errors during decoding. This results in error propagation.

A combi-code scheme proposed by Coene *et al.* [20] achieves high efficiency similar to [32], without introducing the parity-bit related error propagation. In this scheme, the constrained PC code consists of two sliding block codes, which are designed to detect single-bit transition shift errors. Because the two constituent codes are based on the same FSM, no additional channel bits are needed for stitching the two codes together. By using this scheme, efficient PC codes with $d = 2$ constraint, which achieve 2 channel bits per parity bit, have been designed. However, efficient combi-codes with $d = 1$ constraint are not available. Furthermore, this scheme is not general enough to detect any arbitrary error event or error event combinations. The efficiency of this approach can be further improved by using the method proposed in [42].

In this chapter, a novel coding scheme is proposed that overcomes the drawbacks of the prior art schemes. In this scheme, the parity-check constraint is defined by the generator matrix or generator polynomial of a linear binary PC code [57, 76], which can be chosen to detect any error event or error event combinations. This provides a systematic way for defining the error detection criteria for constrained PC codes. The modulation constraint can be either $d = 1$ or $d = 2$. The rates of the designed codes are only a few tenths of a percent below the theoretical maximum. Furthermore, error propagation due to parity bits is avoided, since errors are corrected equally well over the entire constrained PC codeword. While most of the prior art schemes design codes in the NRZI format [80, 32, 20, 16], we propose approaches to design constrained PC codes either in NRZI format or in NRZ format. Designing the codes in NRZ format is found to be preferable for the PC codes and post-processing based detection approach. Note that although this chapter focuses on designing codes for optical recording, the proposed design technique is general, and can encompass other recording channels, such as the magnetic recording channels. This technique can also be generalized to combine constrained codes with other types of error correction codes (ECCs) (*e.g.* the Reed-Solomon (RS) codes), which, however, is beyond the scope of this thesis.

5.2 General Principle of the New Code Design

The general principle of the new code design is as follows. A segment of user data, which is typically the binary output of a RS-ECC encoder, is partitioned into several data words. All the data words except the last one are encoded by any suitable finite state constrained encoder, such as that proposed in Chapter 4. The resulting codewords are referred to as “normal constrained (NC) codewords”. The last data word is encoded by a parity-related constrained encoder, and the resulting codeword is referred to as the “parity-related constrained (PRC) codeword”. In particular, the PRC encoder maps the last data word into a specific codeword chosen from a candidate codeword set, so that a certain parity-check constraint is realized over the combined codeword, which is a concatenation of the sequence of NC codewords and the PRC codeword. This parity-check constraint corresponds to a predetermined generator matrix, which is defined to detect any type of error events in the system. For ease in imposing the modulation constraints, the generator matrix needs to be designed to generate a systematic PC code. This code design principle is based on the following proposition.

Proposition 1: Consider the encoder for a $[l, n]$ systematic linear binary PC code C , which transforms a n -bit information word into a l -bit codeword, with $p = l - n$ being the number of parity bits. Let \mathbf{u}_1 and \mathbf{u}_2 , respectively, denote row vectors with n_1 bits and $n_2 = n - n_1$ bits consisting of a sequence of NC codewords and a PRC codeword. If the parity bits of $[\mathbf{u}_1 \mid \underbrace{0, \dots, 0}_{1 \times n_2}]$ and $[\underbrace{0, \dots, 0}_{1 \times n_1} \mid \mathbf{u}_2]$ are equal, then the combined constrained codeword $[\mathbf{u}_1 \mid \mathbf{u}_2]$, with p bits of zeros appended, generates a codeword of C .

Proof: Let $\mathbf{G} = [\mathbf{I} \ \mathbf{P}]$ be a generator matrix that describes the encoder of C , where \mathbf{I} is the $n \times n$ identity matrix, and \mathbf{P} is a $n \times p$ matrix. The parity bits of $[\mathbf{u}_1 \mid \underbrace{0, \dots, 0}_{1 \times n_2}]$ and $[\underbrace{0, \dots, 0}_{1 \times n_1} \mid \mathbf{u}_2]$ are computed as

$$\mathbf{p}_1 = [\mathbf{u}_1 \mid \underbrace{0, \dots, 0}_{1 \times n_2}] \mathbf{P} \quad \text{and} \quad \mathbf{p}_2 = [\underbrace{0, \dots, 0}_{1 \times n_1} \mid \mathbf{u}_2] \mathbf{P}.$$

If $\mathbf{p}_1 = \mathbf{p}_2$, we get

$$[\mathbf{u}_1 \mid \mathbf{u}_2] \mathbf{P} = [\underbrace{0, \dots, 0}_{1 \times p}]. \quad (5.1)$$

Thus, $[\mathbf{u}_1 \mid \mathbf{u}_2 \mid \underbrace{0, \dots, 0}_{1 \times p}]$ is a codeword of C . \square

The structure of our constrained PC code, thus, includes two component codes: the NC code and the PRC code. Both codes serve as information words of the PC code C . The NC codewords are first constructed and connected. The parity bits of the sequence of NC codewords (with n_2 trailing zeros appended) are then computed. After that, a specific PRC codeword, which produces the same parity bits when n_1 leading zeros are appended, is selected from a candidate codeword set and concatenated directly with the NC codewords, thus forming the combined constrained PC codeword. The combined codeword is transmitted over the channel without appending its parity bits $[\underbrace{0, \dots, 0}_{1 \times p}]$, since the latter is fixed and known to the receiver. At

the detector output, by checking the parity bits reconstructed from the received constrained PC codeword according to (5.1), which are equal to the syndrome of the received codeword (with p bits of zeros appended), we can detect errors in the received codeword that are within the error detection capability of the corresponding PC code C . Note that, in principle, we could always choose $\mathbf{p}_1 + \mathbf{p}_2 = \mathbf{a}$, where \mathbf{a} is an arbitrary p bits row vector, and generate a codeword of C in terms of $[\mathbf{u}_1 \mid \mathbf{u}_2 \mid \mathbf{a}]$. At the detector output, errors can be detected by checking the syndrome of the received codeword with the parity bits \mathbf{a} appended. Choosing $\mathbf{a} = \mathbf{0}$ makes the syndrome equal to the parity bits reconstructed from the received codeword, and simplifies decoding.

Both the NC code and the PRC code are finite-state constrained codes designed based on the same FSM. This enables the two component codes to be connected in any order without violating the modulation constraints, and also facilitates simpler hardware implementation of the encoder/decoder. In principle, any efficient FSM can be used in conjunction with the proposed code design approach. Here, we choose to use the FSM discussed in Chapter 4, since capacity approaching codes can thereby be obtained. Furthermore, since the PRC code

is also protected by parity-checks, error propagation due to the PRC code is avoided. In addition, by applying the Guided Scrambling (GS) scheme [38, 42] to the NC code, whose codewords occupy the major portion of the constrained PC code, as shown in [42], satisfactory dc-free performance can be achieved.

The rate of our constrained PC code is given by

$$\begin{aligned} R &= \frac{m}{n} = \frac{n_1 R_1 + n_2 R_2}{n} \\ &= R_1 - \frac{n_2}{n} (R_1 - R_2), \end{aligned} \quad (5.2)$$

where m and n are the lengths of the segment of user data and the combined constrained PC codeword, respectively, and R_1 and R_2 are the rates of the NC code and the PRC code, respectively. The choice of n , as shown in Section 3.5.1, depends on the specific recording system and is a compromise between the code rate loss due to parity-check and the error correction capability of the post-processor. The optimum codeword length has been found to be around 100 channel bits per parity bit, for $d = 1$ coded optical recording channels. The corresponding details are given in Section 5.6.

5.3 Code Design in NRZI Format

5.3.1 Encoder Description

In this section, the codes are designed in NRZI format. Figure 5.1 is a block schematic for encoding a constrained PC code in NRZI format. As illustrated, a m -bit segment of user data is partitioned into $K + 1$ data words. The K leading data words are individually encoded into the first component codewords by the NC encoder¹. Therefore, we have $K = \frac{n_1 R_1}{m_{NC}}$, with m_{NC} being the length of each input data word of the NC encoder. The $(K + 1)^{\text{th}}$ data word is encoded into the second component codeword by the PRC encoder. During encoding, the next-state information (obtained from a code table) is passed from each codeword to the next. It indicates the next state from which to select a codeword for encoding the next data word. The encoder also

¹Guided scrambling needs to be applied to the NC codewords to impose the dc-free constraint. This is not shown in the figure for the sake of simplicity.

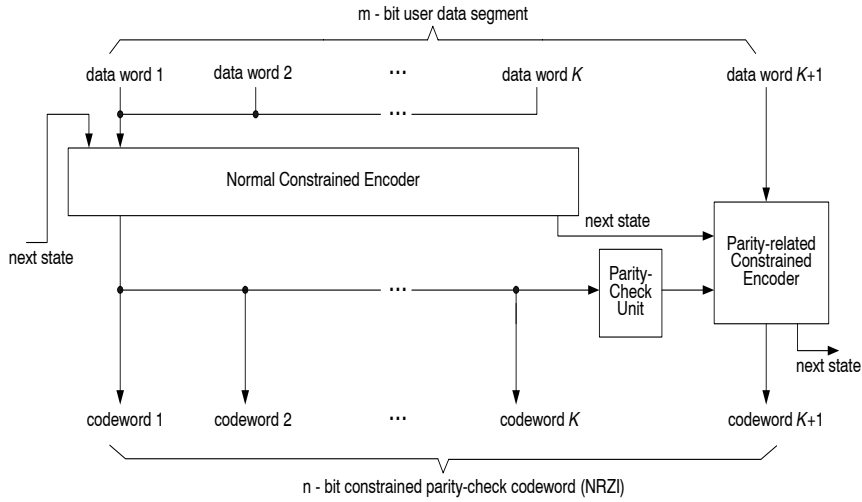


Figure 5.1: Block diagram for encoding a constrained parity-check (PC) code in NRZI format.

includes a parity-check unit, which calculates the parity bits of the sequence of the leading K NC codewords (appended with n_2 trailing bits of zeros). The parity bits are then passed to the PRC encoder, and used to guide the encoding of the $(K + 1)^{\text{th}}$ data word into the PRC codeword. Concatenating the NC codewords and the PRC codeword together, results in the combined constrained PC codeword in NRZI format. The NRZI codewords are then converted into NRZ format by a precoder, which is not shown in the figure, before they are transmitted over the channel.

5.3.2 Design of the Component Codes

To design the NC code, we use the FSM discussed in Chapter 4, since rates of the resulting constrained codes are very close to the capacity. The optimum number of encoder states is obtained using the analytical approach proposed in Chapter 4. To achieve high encoding efficiency, one data word is mapped into one codeword only in each state of the FSM.

To design the PRC code, we propose a novel approach to design

sets of codewords with distinct parity bits, based on the same FSM of the NC code. These parity bits correspond to a predetermined generator matrix. We first propose a set of criteria that guides the design of the PRC code. With a given parity-check constraint, these criteria indicate how to choose the number of encoder states and assign valid codewords to these states to maximize the code rate.

To design a PRC code with m_2 user data bits and p parity bits, the number of codewords leaving a state set should be at least 2^{m_2+p} times the number of states within the state set. Based on the FSM of $d = 1$ codes discussed in Chapter 4, we can obtain the following criteria:

$$r|X_{00}| + r_1|X_{01}| \geq r_1 2^{m_2+p}, \quad (5.3)$$

$$r(|X_{00}| + |X_{10}|) + r_1(|X_{01}| + |X_{11}|) \geq r 2^{m_2+p}, \quad (5.4)$$

where X_{ab} denotes the set of codewords starting with a ‘ a ’ and ending with a ‘ b ’, and $|X_{ab}|$ denotes the size of X_{ab} , where $a, b \in \{0, 1\}$. The encoder has r states, which are divided into two state subsets of a first and second type. The encoder has r_1 states of the first type and $r_2 = r - r_1$ states of the second type. All codewords in states of the first type must start with a ‘0’, while codewords in states of the second type start with either a ‘0’ or a ‘1’.

Furthermore, for each set of codewords with the same parity bits, the number of codewords leaving a state set should be at least 2^{m_2} times the number of states within the state set. The criteria that guide the design of each set of codewords are therefore given by

$$r|\tilde{X}_{00}| + r_1|\tilde{X}_{01}| \geq r_1 2^{m_2}, \quad (5.5)$$

$$r(|\tilde{X}_{00}| + |\tilde{X}_{10}|) + r_1(|\tilde{X}_{01}| + |\tilde{X}_{11}|) \geq r 2^{m_2}, \quad (5.6)$$

where \tilde{X}_{ab} denotes the set of codewords with the same parity bits that start with a ‘ a ’ and end with a ‘ b ’. In each set of the codewords with the same parity bits, each codeword has an assigned next state. A codeword that ends with ‘0’ (*i.e.* codewords in \tilde{X}_{00} and \tilde{X}_{10}) can be assigned up to r different next states in both the first and second state sets, and therefore can be used to map to r different user data words. A codeword that ends with ‘1’ (*i.e.* codewords in \tilde{X}_{01} and \tilde{X}_{11}) can only be assigned up to r_1 next states in the first state set, and therefore can be used to map to r_1 different user data words. The particular

mapping of the codeword to the data word is a matter of design choice, and is not critical to the operation of the system. However, to ensure unique decodability, the sets of codewords that belong to a given state must be disjoint.

Similar criteria for designing PRC codes with $d = 2$ constraint can be derived. They are given in Appendix B. Note that the above inequalities are equivalent to the *approximate eigenvector equation*, and they are necessary conditions for code construction. Following these criteria, and by using either computer search or analytical approaches similar to that proposed in Chapter 4, we can determine the optimum number of encoder states to maximize the rate of the PRC code. The corresponding code rate is given by

$$R_2 = m_2/n_2. \quad (5.7)$$

The main steps for the design of the PRC code are as follows.

(1) For a PRC code with m_2 user data bits and p parity bits, use the criteria described above (*i.e.* (5.3) to (5.6)) to determine the codeword length n_2 and the optimum number of encoder states. Note that at this step, the maximum runlength constraint k is temporarily relaxed (*e.g.* larger than $k = 7$ for $d = 1$ codes, and larger than $k = 10$ for $d = 2$ codes).

(2) Enumerate all the valid d constrained codewords of length n_2 . Based on the given generator matrix, compute the parity bits of each codeword (with n_1 leading zeros appended) and distribute them into a group of codeword sets. A total of 2^p codeword sets are obtained.

(3) For each set of codewords with the same parity bits, allocate the codewords to various encoder states by following the FSM of the NC code. This results in a set of 2^p sub-tables.

(4) Concatenate the 2^p sub-tables, and form a code table for encoding/decoding the PRC code. Compared with the code table of the NC code, the PRC code table is enlarged by a factor of 2^p . In each state of the FSM, there is a set of 2^p codewords potentially mapped to one user data word. During encoding, as illustrated in Figure 5.1, the parity bits associated with the sequence of NC codewords are first calculated. The PRC codeword having the same parity bits is selected from the codeword set and assigned to the user data word.

(5) Tighten the k constraint of the designed PRC code by optimizing the code table obtained from Step (4) by deleting codewords

that start or end with long runs of ‘0’s, or by increasing the number of states of the FSM.

5.3.3 Decoder Description

Based on the same FSM, the operation of the PRC decoder is generally the same as that of the NC decoder (as described in Section 4.2), but with the code tables being different. Both decoders are sliding-block decoders with a look ahead of one codeword. However, unlike the NC decoders which are based on only one code table, the PRC decoder is based on two code tables. One is the NC code table, which is used to determine the state of the next codeword, while the other is the PRC code table, which is used to decode the current codeword using the obtained state information of the next codeword.

5.4 Code Design in NRZ Format

In this section, we present an approach to design constrained PC codes in NRZ format. For PC codes and post-processing based detection approaches, it is preferable to encode the data in NRZ format due to the following reasons. In the NRZI case, error detection and post-processing have to be done at the output of the ‘NRZ to NRZI inverse precoder’. The process of inverse precoding will cause error propagation and thus increase the length of error events. For example, a single bit error in NRZ format will be converted into a transition shift error of 2 bits in NRZI format. As a result, the number of parity bits required for detecting errors may increase. Furthermore, carrying out post-processing at the detector output is more straightforward than doing it at the inverse precoder output.

The conventional approach for detection and correction of errors in NRZ format is to use a concatenation of a modulation encoder with a precoder, followed by a PC encoder [15, 86]. However, this approach will considerably weaken the modulation constraint of the encoded channel data stream. In [16], Cideciyan *et al.* proposed the cascade of a modulation encoder with a PC encoder followed by a precoder. In this approach, before precoding, the user data is first encoded into a constrained PC code in NRZI format, which can detect and correct NRZ errors. This is done by translating the parity-check matrix at the

output of the precoder into that at the input of the precoder, under the condition that the PC code at the output of the precoder must contain the all-one codeword.

We now present a new approach to design the constrained PC code in NRZ format, without parity-check matrix transformation and without the specific requirement on the PC code. In our approach, the code table of the NC code remains the same as that of the NC code in NRZI format. However, the code table for the PRC code is designed in a different way. The details are as follows.

(1) For a PRC code with m_2 user data bits and p parity bits, determine the codeword length n_2 and the optimum number of encoder states. The criteria that guide the design are similar to those in the NRZI case. The only difference is that the parity bits of each codeword are computed in the NRZ format, rather than in the NRZI format, based on an assumed initial NRZ bit. To do this, ‘0’ and ‘1’ are used to denote NRZ bits ‘ -1 ’ and ‘ $+1$ ’, respectively.

(2) Enumerate all the valid d constrained codewords of length n_2 in NRZI format. Compute the parity bits of the codewords in NRZ format with an assumed initial NRZ bit.

(3) Distribute each set of NRZI codewords with the same NRZ parity bits obtained from Step (2) into different encoder states, and form a set of 2^p sub-tables.

(4) Concatenate the 2^p sub-tables together to form the code table for encoding/decoding of the PRC code in NRZ format. For the two different initial NRZ bits (*i.e.* ‘ $+1$ ’ and ‘ -1 ’), we use the same code table to simplify encoding/decoding. However, the order of codeword sets with the same parity bits may need to be adjusted according to the initial NRZ bit.

To do encoding, as shown in Figure 5.2, the NC codewords are first constructed and connected as in the NRZI case. The resulting codewords are then converted into NRZ format by a precoder, and the associated parity bits are computed. Based on these parity bits as well as the last bit of the NRZ sequence, the PRC codeword in NRZI format that has the same NRZ parity bits is selected from the codeword set. The PRC codeword needs to be converted into NRZ format before concatenating with the NRZ format NC codewords. During decoding, the detected NRZ data sequence is first converted into NRZI format through an inverse precoder, and the resulting NRZI sequence is then

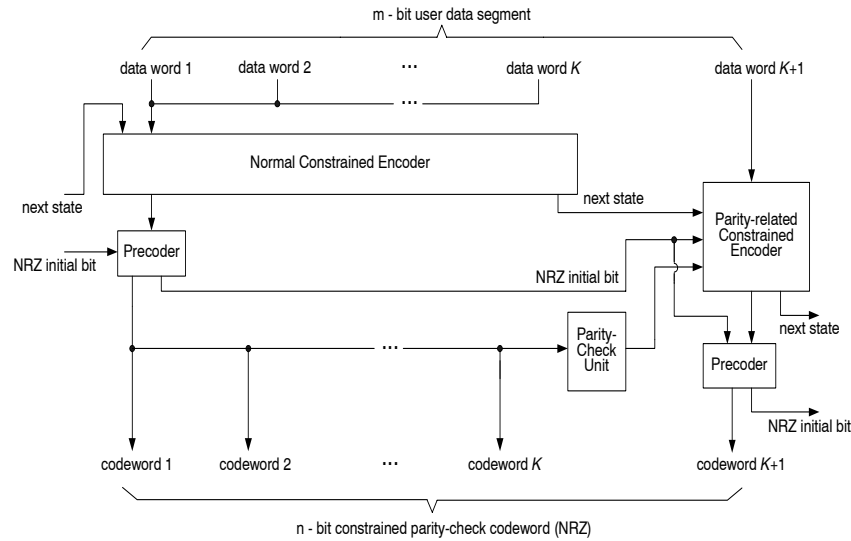


Figure 5.2: Block diagram for encoding a constrained parity-check (PC) code in NRZ format.

decoded based on the code tables of the NC code and the PRC code, along the same lines as those described in Section 5.3.3.

5.5 Examples of New Codes

In this section, we present several new constrained PC codes, designed in NRZ format, using the above code design method.

First of all, a new (1,18) constrained single-bit even PC code is designed. The rate $9/13$ (1,18) code with 5 states (*i.e.* $r = 5$, $r_1 = 3$, $r_2 = 2$) FSM discussed in Chapter 4 is used as the NC code, since its rate is 3.85% higher than that of the rate $2/3$ $d = 1$ codes used in BD and HD-DVD systems. A new rate $12/19$ (1,18) code with 5 states is designed as the PRC code, which requires only 1 channel bit per parity bit with respect to the rate $2/3$ $d = 1$ codes.

Table 5.1 shows the distribution of codewords in $r = 5$ encoder states, which are denoted by S_1 to S_5 , for the rate $12/19$ PRC code. Through enumeration, we find that among the total 10946 valid $d = 1$

Table 5.1: *Distribution of codewords in the various encoder states for a rate 12/19 (1,18) parity-related constrained (PRC) code.*

(i) Parity even for initial NRZ bit '-1' (Parity odd for '+1')					(ii) Parity odd for initial NRZ bit '-1' (Parity even for '+1')				
	\tilde{X}_{00}	\tilde{X}_{01}	\tilde{X}_{10}	\tilde{X}_{11}		\tilde{X}_{00}	\tilde{X}_{01}	\tilde{X}_{10}	\tilde{X}_{11}
Size	2135	1275	1275	805	Size	2046	1309	1309	792
S_1	605	358	0	0	S_1	589	384	0	0
S_2	599	369	0	0	S_2	599	368	0	0
S_3	603	362	0	0	S_3	594	378	0	0
S_4	0	0	589	384	S_4	0	0	605	357
S_5	0	0	600	368	S_5	0	0	598	370

codewords of length 19, there are 5490 codewords having even parity with an assumed initial NRZ bit of '-1' (or odd parity with an initial NRZ bit of '+1'). Among these codewords, we further find $|\tilde{X}_{00}| = 2135$, $|\tilde{X}_{01}| = |\tilde{X}_{10}| = 1275$ and $|\tilde{X}_{11}| = 805$. We also find that there are 5456 codewords having odd parity with an assumed initial NRZ bit of '-1' (or even parity with an initial NRZ bit of '+1'), among which we have $|\tilde{X}_{00}| = 2046$, $|\tilde{X}_{01}| = |\tilde{X}_{10}| = 1309$ and $|\tilde{X}_{11}| = 792$. Each sub-table in Table 5.1 illustrates the distribution of codewords with the same parity bit among the $r = 5$ states.

We take Table 5.1 (i) as an example, which contains all the codewords having even parity with an assumed initial NRZ bit of '-1'. Observe that the set \tilde{X}_{00} has 605 codewords allocated in State 1, 599 codewords in State 2, and 603 codewords in State 3. The total number of assigned codewords is $605 + 599 + 603 = 1807$, which is smaller than the set size 2135. Similarly, for each of the other codeword set, the total number of assigned codewords is smaller than the size of the set. On the other hand, in each state, the codewords are distributed according to the restrictions that a codeword ending with a '0' can be assigned to up to $r = 5$ different user data words, while a codeword that ends with a '1' can only be assigned to up to $r_1 = 3$ different user data words. Therefore, for State 1, the total number of assigned codewords is $605 \times 5 + 358 \times 3 = 4099$, which is sufficient to map $2^{12} = 4096$ user data words. Similarly, it can be verified that from any of the $r = 5$ encoder states, there are at least 4096 codewords that can

be assigned to the user data words. This means that 12-bit user data words can be encoded. In the same manner, codewords having odd parity with an assumed initial NRZ bit of ‘-1’ are distributed as shown in Table 5.1 (ii), which also shows that 12-bit user data words can be supported. Hence, following Table 5.1, a rate 12/19 (1,18) PRC code can be constructed. We remark that the distribution of codewords given above may not be unique.

As a second example, using the same rate 9/13 code as the NC code, we design new constrained 2-bit and 4-bit PC codes. They are defined by generator polynomials $g(x) = 1 + x + x^2$ and $g(x) = 1 + x + x^4$, respectively. The corresponding PRC codes are a rate 9/16 (1,18) code and a rate 7/16 (1,18) code, respectively. With respect to the rate $2/3$ $d = 1$ codes, these PRC codes achieve 1.5 and 1.375 channel bits per parity bit, respectively. Tables 5.2 and 5.3 show the distribution of codewords in the various encoder states for the rate 7/16 PRC code, in the same fashion as Table 5.1. Note that with 4-bit parity-check, there are in total 16 sub-tables illustrated in Tables 5.2 and 5.3, corresponding to the various parity bits of the codewords. In computing the parity bits of the codewords in the tables, following the guideline of 100 channel bits per parity bit, we use a codeword length of $n = 406$.

As a third example, we consider $d = 2$ codes. With the rate 6/11 (2,15), 9-state (*i.e.* $r = 9, r_1 = 4, r_2 = 2, r_3 = 3$) code proposed in [42] as the NC code, whose rate is 2.27% higher than that of the rate 8/15 (2,10) EFM-like codes [38] used in DVD systems, we have designed a new constrained single-bit even PC code and a new constrained 4-bit PC code defined by $g(x) = 1 + x + x^4$. The PRC codes are a 9-state rate 10/20 (2,15) code and a 9-state rate 8/22 (2,15) code, respectively. With respect to the EFM-like codes, these PRC codes achieve 1.25 and 1.75 channel bits per parity bit, respectively.

The above examples of newly designed codes are summarized in Table 5.4. The codeword length, n , is chosen such that the number of channel bits per parity bit is around 100. As can be seen, the new codes achieve minimum parity overhead, and the efficiency of most of the new codes is only a few tenths of a percent below capacity².

²Following the discussions of Section 3.2.1, the capacity of constrained PC codes is given by $C_{pc} = R_1 - p/n$. For example, for the $(1, \infty)$ code, the capacity is $C_{pc(1, \infty)} = R_{(1, \infty)} - 1/100 = 0.6842$, with 100 channel bits per parity bit.

Table 5.2: *Distribution of codewords in the various encoder states for a rate 7/16 (1,18) parity-related constrained (PRC) code, Part I.*

(i) Parity bits [0 0 0 0] for initial NRZ bit '-1' ([1 1 0 0] for '+1')

	\tilde{X}_{00}	\tilde{X}_{01}	\tilde{X}_{10}	\tilde{X}_{11}
Size	67	42	39	22
S_1	19	11	0	0
S_2	18	13	0	0
S_3	19	13	0	0
S_4	0	0	19	11
S_5	0	0	19	11

(ii) Parity bits [0 0 0 1] for initial NRZ bit '-1' ([1 1 0 1] for '+1')

	\tilde{X}_{00}	\tilde{X}_{01}	\tilde{X}_{10}	\tilde{X}_{11}
Size	58	45	39	20
S_1	17	15	0	0
S_2	19	12	0	0
S_3	18	15	0	0
S_4	0	0	20	10
S_5	0	1	19	10

(iii) Parity bits [0 0 1 0] for initial NRZ bit '-1' ([1 1 1 0] for '+1')

	\tilde{X}_{00}	\tilde{X}_{01}	\tilde{X}_{10}	\tilde{X}_{11}
Size	64	34	37	24
S_1	19	11	0	0
S_2	21	9	0	0
S_3	19	12	0	0
S_4	1	0	18	12
S_5	0	0	19	12

(iv) Parity bits [0 0 1 1] for initial NRZ bit '-1' ([1 1 1 1] for '+1')

	\tilde{X}_{00}	\tilde{X}_{01}	\tilde{X}_{10}	\tilde{X}_{11}
Size	63	35	35	27
S_1	20	10	0	0
S_2	19	12	0	0
S_3	20	10	0	0
S_4	1	0	18	12
S_5	0	0	17	15

(v) Parity bits [0 1 0 0] for initial NRZ bit '-1' ([1 0 0 0] for '+1')

	\tilde{X}_{00}	\tilde{X}_{01}	\tilde{X}_{10}	\tilde{X}_{11}
Size	58	36	38	25
S_1	18	13	0	0
S_2	20	11	0	0
S_3	20	12	0	0
S_4	0	0	19	11
S_5	0	0	18	13

(vi) Parity bits [0 1 0 1] for initial NRZ bit '-1' ([1 0 0 1] for '+1')

	\tilde{X}_{00}	\tilde{X}_{01}	\tilde{X}_{10}	\tilde{X}_{11}
Size	64	36	37	25
S_1	19	12	0	0
S_2	21	9	0	0
S_3	19	12	0	0
S_4	0	0	19	11
S_5	0	0	18	14

(vii) Parity bits [0 1 1 0] for initial NRZ bit '-1' ([1 0 1 0] for '+1')

	\tilde{X}_{00}	\tilde{X}_{01}	\tilde{X}_{10}	\tilde{X}_{11}
Size	60	41	39	23
S_1	18	14	0	0
S_2	20	11	0	0
S_3	18	14	0	0
S_4	0	0	19	11
S_5	0	0	19	11

(viii) Parity bits [0 1 1 1] for initial NRZ bit '-1' ([1 0 1 1] for '+1')

	\tilde{X}_{00}	\tilde{X}_{01}	\tilde{X}_{10}	\tilde{X}_{11}
Size	61	37	40	22
S_1	20	10	0	0
S_2	18	13	0	0
S_3	20	11	0	0
S_4	0	1	18	12
S_5	0	0	20	10

It should be noted that for the above new codes, the sizes of input symbols of most of the component codes are not 8 bits. As a result, error propagation due to the mismatch of symbol sizes between the constrained code and the byte-oriented RS-ECC may arise. However, this error propagation can be avoided by using the 'modified concatenation' scheme [9, 43, 26]. Alternatively, a non-byte-oriented RS-ECC can be used to eliminate this error propagation. For example, when

Table 5.3: *Distribution of codewords in the various encoder states for a rate 7/16 (1,18) parity-related constrained (PRC) code, Part II.*

(ix) Parity bits [1 0 0 0] for initial NRZ bit '-1' ([0 1 0 0] for '+1')	(x) Parity bits [1 0 0 1] for initial NRZ bit '-1' ([0 1 0 1] for '+1')	(xi) Parity bits [1 0 1 0] for initial NRZ bit '-1' ([0 1 1 0] for '+1')																																																																																																									
<table border="1" style="border-collapse: collapse; width: 100%; text-align: center;"> <thead> <tr> <th></th> <th>\tilde{X}_{00}</th> <th>\tilde{X}_{01}</th> <th>\tilde{X}_{10}</th> <th>\tilde{X}_{11}</th> </tr> </thead> <tbody> <tr> <td>Size</td> <td>60</td> <td>39</td> <td>36</td> <td>22</td> </tr> <tr> <td>S_1</td> <td>18</td> <td>13</td> <td>0</td> <td>0</td> </tr> <tr> <td>S_2</td> <td>19</td> <td>12</td> <td>0</td> <td>0</td> </tr> <tr> <td>S_3</td> <td>19</td> <td>12</td> <td>0</td> <td>0</td> </tr> <tr> <td>S_4</td> <td>1</td> <td>0</td> <td>18</td> <td>12</td> </tr> <tr> <td>S_5</td> <td>2</td> <td>0</td> <td>18</td> <td>10</td> </tr> </tbody> </table>		\tilde{X}_{00}	\tilde{X}_{01}	\tilde{X}_{10}	\tilde{X}_{11}	Size	60	39	36	22	S_1	18	13	0	0	S_2	19	12	0	0	S_3	19	12	0	0	S_4	1	0	18	12	S_5	2	0	18	10	<table border="1" style="border-collapse: collapse; width: 100%; text-align: center;"> <thead> <tr> <th></th> <th>\tilde{X}_{00}</th> <th>\tilde{X}_{01}</th> <th>\tilde{X}_{10}</th> <th>\tilde{X}_{11}</th> </tr> </thead> <tbody> <tr> <td>Size</td> <td>60</td> <td>39</td> <td>39</td> <td>21</td> </tr> <tr> <td>S_1</td> <td>20</td> <td>10</td> <td>0</td> <td>0</td> </tr> <tr> <td>S_2</td> <td>18</td> <td>14</td> <td>0</td> <td>0</td> </tr> <tr> <td>S_3</td> <td>19</td> <td>11</td> <td>0</td> <td>0</td> </tr> <tr> <td>S_4</td> <td>0</td> <td>0</td> <td>19</td> <td>11</td> </tr> <tr> <td>S_5</td> <td>0</td> <td>0</td> <td>20</td> <td>10</td> </tr> </tbody> </table>		\tilde{X}_{00}	\tilde{X}_{01}	\tilde{X}_{10}	\tilde{X}_{11}	Size	60	39	39	21	S_1	20	10	0	0	S_2	18	14	0	0	S_3	19	11	0	0	S_4	0	0	19	11	S_5	0	0	20	10	<table border="1" style="border-collapse: collapse; width: 100%; text-align: center;"> <thead> <tr> <th></th> <th>\tilde{X}_{00}</th> <th>\tilde{X}_{01}</th> <th>\tilde{X}_{10}</th> <th>\tilde{X}_{11}</th> </tr> </thead> <tbody> <tr> <td>Size</td> <td>62</td> <td>38</td> <td>37</td> <td>25</td> </tr> <tr> <td>S_1</td> <td>19</td> <td>11</td> <td>0</td> <td>0</td> </tr> <tr> <td>S_2</td> <td>19</td> <td>11</td> <td>0</td> <td>0</td> </tr> <tr> <td>S_3</td> <td>19</td> <td>11</td> <td>0</td> <td>0</td> </tr> <tr> <td>S_4</td> <td>0</td> <td>1</td> <td>17</td> <td>14</td> </tr> <tr> <td>S_5</td> <td>0</td> <td>0</td> <td>20</td> <td>11</td> </tr> </tbody> </table>		\tilde{X}_{00}	\tilde{X}_{01}	\tilde{X}_{10}	\tilde{X}_{11}	Size	62	38	37	25	S_1	19	11	0	0	S_2	19	11	0	0	S_3	19	11	0	0	S_4	0	1	17	14	S_5	0	0	20	11
	\tilde{X}_{00}	\tilde{X}_{01}	\tilde{X}_{10}	\tilde{X}_{11}																																																																																																							
Size	60	39	36	22																																																																																																							
S_1	18	13	0	0																																																																																																							
S_2	19	12	0	0																																																																																																							
S_3	19	12	0	0																																																																																																							
S_4	1	0	18	12																																																																																																							
S_5	2	0	18	10																																																																																																							
	\tilde{X}_{00}	\tilde{X}_{01}	\tilde{X}_{10}	\tilde{X}_{11}																																																																																																							
Size	60	39	39	21																																																																																																							
S_1	20	10	0	0																																																																																																							
S_2	18	14	0	0																																																																																																							
S_3	19	11	0	0																																																																																																							
S_4	0	0	19	11																																																																																																							
S_5	0	0	20	10																																																																																																							
	\tilde{X}_{00}	\tilde{X}_{01}	\tilde{X}_{10}	\tilde{X}_{11}																																																																																																							
Size	62	38	37	25																																																																																																							
S_1	19	11	0	0																																																																																																							
S_2	19	11	0	0																																																																																																							
S_3	19	11	0	0																																																																																																							
S_4	0	1	17	14																																																																																																							
S_5	0	0	20	11																																																																																																							
(xii) Parity bits [1 0 1 1] for initial NRZ bit '-1' ([0 1 1 1] for '+1')	(xiii) Parity bits [1 1 0 0] for initial NRZ bit '-1' ([0 0 0 0] for '+1')	(xiv) Parity bits [1 1 0 1] for initial NRZ bit '-1' ([0 0 0 1] for '+1')																																																																																																									
<table border="1" style="border-collapse: collapse; width: 100%; text-align: center;"> <thead> <tr> <th></th> <th>\tilde{X}_{00}</th> <th>\tilde{X}_{01}</th> <th>\tilde{X}_{10}</th> <th>\tilde{X}_{11}</th> </tr> </thead> <tbody> <tr> <td>Size</td> <td>63</td> <td>35</td> <td>37</td> <td>24</td> </tr> <tr> <td>S_1</td> <td>18</td> <td>13</td> <td>0</td> <td>0</td> </tr> <tr> <td>S_2</td> <td>20</td> <td>10</td> <td>0</td> <td>0</td> </tr> <tr> <td>S_3</td> <td>19</td> <td>11</td> <td>0</td> <td>0</td> </tr> <tr> <td>S_4</td> <td>0</td> <td>0</td> <td>19</td> <td>11</td> </tr> <tr> <td>S_5</td> <td>0</td> <td>0</td> <td>18</td> <td>13</td> </tr> </tbody> </table>		\tilde{X}_{00}	\tilde{X}_{01}	\tilde{X}_{10}	\tilde{X}_{11}	Size	63	35	37	24	S_1	18	13	0	0	S_2	20	10	0	0	S_3	19	11	0	0	S_4	0	0	19	11	S_5	0	0	18	13	<table border="1" style="border-collapse: collapse; width: 100%; text-align: center;"> <thead> <tr> <th></th> <th>\tilde{X}_{00}</th> <th>\tilde{X}_{01}</th> <th>\tilde{X}_{10}</th> <th>\tilde{X}_{11}</th> </tr> </thead> <tbody> <tr> <td>Size</td> <td>63</td> <td>36</td> <td>41</td> <td>27</td> </tr> <tr> <td>S_1</td> <td>20</td> <td>10</td> <td>0</td> <td>0</td> </tr> <tr> <td>S_2</td> <td>19</td> <td>11</td> <td>0</td> <td>0</td> </tr> <tr> <td>S_3</td> <td>19</td> <td>11</td> <td>0</td> <td>0</td> </tr> <tr> <td>S_4</td> <td>0</td> <td>0</td> <td>19</td> <td>11</td> </tr> <tr> <td>S_5</td> <td>0</td> <td>0</td> <td>19</td> <td>12</td> </tr> </tbody> </table>		\tilde{X}_{00}	\tilde{X}_{01}	\tilde{X}_{10}	\tilde{X}_{11}	Size	63	36	41	27	S_1	20	10	0	0	S_2	19	11	0	0	S_3	19	11	0	0	S_4	0	0	19	11	S_5	0	0	19	12	<table border="1" style="border-collapse: collapse; width: 100%; text-align: center;"> <thead> <tr> <th></th> <th>\tilde{X}_{00}</th> <th>\tilde{X}_{01}</th> <th>\tilde{X}_{10}</th> <th>\tilde{X}_{11}</th> </tr> </thead> <tbody> <tr> <td>Size</td> <td>65</td> <td>33</td> <td>37</td> <td>27</td> </tr> <tr> <td>S_1</td> <td>19</td> <td>11</td> <td>0</td> <td>0</td> </tr> <tr> <td>S_2</td> <td>20</td> <td>10</td> <td>0</td> <td>0</td> </tr> <tr> <td>S_3</td> <td>20</td> <td>11</td> <td>0</td> <td>0</td> </tr> <tr> <td>S_4</td> <td>0</td> <td>1</td> <td>17</td> <td>14</td> </tr> <tr> <td>S_5</td> <td>0</td> <td>0</td> <td>19</td> <td>12</td> </tr> </tbody> </table>		\tilde{X}_{00}	\tilde{X}_{01}	\tilde{X}_{10}	\tilde{X}_{11}	Size	65	33	37	27	S_1	19	11	0	0	S_2	20	10	0	0	S_3	20	11	0	0	S_4	0	1	17	14	S_5	0	0	19	12
	\tilde{X}_{00}	\tilde{X}_{01}	\tilde{X}_{10}	\tilde{X}_{11}																																																																																																							
Size	63	35	37	24																																																																																																							
S_1	18	13	0	0																																																																																																							
S_2	20	10	0	0																																																																																																							
S_3	19	11	0	0																																																																																																							
S_4	0	0	19	11																																																																																																							
S_5	0	0	18	13																																																																																																							
	\tilde{X}_{00}	\tilde{X}_{01}	\tilde{X}_{10}	\tilde{X}_{11}																																																																																																							
Size	63	36	41	27																																																																																																							
S_1	20	10	0	0																																																																																																							
S_2	19	11	0	0																																																																																																							
S_3	19	11	0	0																																																																																																							
S_4	0	0	19	11																																																																																																							
S_5	0	0	19	12																																																																																																							
	\tilde{X}_{00}	\tilde{X}_{01}	\tilde{X}_{10}	\tilde{X}_{11}																																																																																																							
Size	65	33	37	27																																																																																																							
S_1	19	11	0	0																																																																																																							
S_2	20	10	0	0																																																																																																							
S_3	20	11	0	0																																																																																																							
S_4	0	1	17	14																																																																																																							
S_5	0	0	19	12																																																																																																							
(xv) Parity bits [1 1 1 0] for initial NRZ bit '-1' ([0 0 1 0] for '+1')	(xvi) Parity bits [1 1 1 1] for initial NRZ bit '-1' ([0 0 1 1] for '+1')																																																																																																										
<table border="1" style="border-collapse: collapse; width: 100%; text-align: center;"> <thead> <tr> <th></th> <th>\tilde{X}_{00}</th> <th>\tilde{X}_{01}</th> <th>\tilde{X}_{10}</th> <th>\tilde{X}_{11}</th> </tr> </thead> <tbody> <tr> <td>Size</td> <td>61</td> <td>40</td> <td>39</td> <td>21</td> </tr> <tr> <td>S_1</td> <td>19</td> <td>12</td> <td>0</td> <td>0</td> </tr> <tr> <td>S_2</td> <td>19</td> <td>12</td> <td>0</td> <td>0</td> </tr> <tr> <td>S_3</td> <td>18</td> <td>13</td> <td>0</td> <td>0</td> </tr> <tr> <td>S_4</td> <td>0</td> <td>0</td> <td>19</td> <td>11</td> </tr> <tr> <td>S_5</td> <td>0</td> <td>0</td> <td>20</td> <td>10</td> </tr> </tbody> </table>		\tilde{X}_{00}	\tilde{X}_{01}	\tilde{X}_{10}	\tilde{X}_{11}	Size	61	40	39	21	S_1	19	12	0	0	S_2	19	12	0	0	S_3	18	13	0	0	S_4	0	0	19	11	S_5	0	0	20	10	<table border="1" style="border-collapse: collapse; width: 100%; text-align: center;"> <thead> <tr> <th></th> <th>\tilde{X}_{00}</th> <th>\tilde{X}_{01}</th> <th>\tilde{X}_{10}</th> <th>\tilde{X}_{11}</th> </tr> </thead> <tbody> <tr> <td>Size</td> <td>58</td> <td>44</td> <td>40</td> <td>22</td> </tr> <tr> <td>S_1</td> <td>17</td> <td>15</td> <td>0</td> <td>0</td> </tr> <tr> <td>S_2</td> <td>18</td> <td>13</td> <td>0</td> <td>0</td> </tr> <tr> <td>S_3</td> <td>18</td> <td>14</td> <td>0</td> <td>0</td> </tr> <tr> <td>S_4</td> <td>0</td> <td>0</td> <td>20</td> <td>10</td> </tr> <tr> <td>S_5</td> <td>0</td> <td>0</td> <td>19</td> <td>12</td> </tr> </tbody> </table>		\tilde{X}_{00}	\tilde{X}_{01}	\tilde{X}_{10}	\tilde{X}_{11}	Size	58	44	40	22	S_1	17	15	0	0	S_2	18	13	0	0	S_3	18	14	0	0	S_4	0	0	20	10	S_5	0	0	19	12																																				
	\tilde{X}_{00}	\tilde{X}_{01}	\tilde{X}_{10}	\tilde{X}_{11}																																																																																																							
Size	61	40	39	21																																																																																																							
S_1	19	12	0	0																																																																																																							
S_2	19	12	0	0																																																																																																							
S_3	18	13	0	0																																																																																																							
S_4	0	0	19	11																																																																																																							
S_5	0	0	20	10																																																																																																							
	\tilde{X}_{00}	\tilde{X}_{01}	\tilde{X}_{10}	\tilde{X}_{11}																																																																																																							
Size	58	44	40	22																																																																																																							
S_1	17	15	0	0																																																																																																							
S_2	18	13	0	0																																																																																																							
S_3	18	14	0	0																																																																																																							
S_4	0	0	20	10																																																																																																							
S_5	0	0	19	12																																																																																																							

the rate 135/198 2-bit constrained PC code is used in conjunction with a 9-bit/symbol RS-ECC, error propagation is avoided since the sizes of the input symbols for both the NC and PRC codes are 9 bits. Finally, we remark here that it is possible to impose stricter k constraint and the repeated minimum transition runlength (RMTR) constraint [70, 48] on the designed codes, by increasing the number of states of the FSM, and/or by applying the GS scheme.

Table 5.4: Summary of newly designed constrained PC codes.

Code type	Generator polynomial	(d, k)	Parity overhead (channel bits / parity bit)	R	$\eta = R / C_{pc}$
9/13 code (NC) & 12/19 code (PRC)	$g(x) = 1 + x$	(1,18)	1	$66/97 = 0.6804$	0.9945
9/13 code (NC) & 9/16 code (PRC)	$g(x) = 1 + x + x^2$	(1,18)	1.5	$135/198 = 0.6818$	0.9965
9/13 code (NC) & 7/16 code (PRC)	$g(x) = 1 + x + x^4$	(1,18)	1.375	$277/406 = 0.6823$	0.9972
6/11 code (NC) & 10/20 code (PRC)	$g(x) = 1 + x$	(2,15)	1.25	$52/97 = 0.5361$	0.9900
6/11 code (NC) & 5/17 code (PRC)	$g(x) = 1 + x + x^4$	(2,15)	1.75	$215/402 = 0.5348$	0.9877

5.6 Performance Evaluation

In this section, we present BER performances of the newly designed constrained PC codes with $d = 1$ constraint, obtained using simulations. In the simulations, the channel noise is assumed to be additive white Gaussian (AWGN). A Viterbi detector (VD) that is matched to a 7-tap partial response (PR) target is used as the detector. As in Chapter 3, the PR target consists of 7 symmetric taps taken directly from the middle of the channel symbol response. The dominant error events at the VD output turn out to be very similar to that with the rate 2/3 code. Therefore, Table 3.2 can be still be used to determine the error detection capability of the newly designed codes. At the VD output, a matched-filtering type post-processor is used, which can correct both single and double error events that occur in each detected codeword.

According to Table 3.2, the new single-bit even PC code can only detect error events with odd number of errors, and therefore cannot detect the error events $\pm\{2, 0, -2\}$, $\pm\{2, 0, -2, 0, 2, 0, -2\}$ and $\pm\{2, 0, 0, -2\}$. The new 2-bit PC codes, defined by the generator polynomial $g(x) = 1 + x + x^2$, can detect most of the dominant error events, except $\pm\{2, 0, -2, 0, 2\}$ and $\pm\{2, 0, 0, -2\}$. The new 4-bit PC codes, with $g(x) = 1 + x + x^4$, however, can detect all the error events listed in Table 3.2. In the simulations, by using an approach similar to that proposed in Section 3.5.1, the codeword length (n) is chosen

such that the number of channel bits per parity bit is around 100. For real-life channels, the dominant error events may differ from those illustrated above. However, following the code design method described in the above sections, we can easily define a generator matrix that detects all the required error events, and design the constrained PC code accordingly.

Figures 5.3 and 5.4 illustrate the BER performance of the system with the rate 2/3 code, rate 9/13 code, and the new constrained PC codes, at nominal density and high density, respectively. From Figure 5.3, we observe that compared to the performance of the system with the rate 2/3 code and without parity (Curve 1), the rate 9/13 code without parity (Curve 2) gives a gain of 0.5 dB at $\text{BER} = 10^{-5}$, due to its higher code rate. Compared to the rate 9/13 code without parity, the new single-bit PC code (Curve 3) gives no significant gain, since it cannot detect the error event $\pm\{2, 0, -2\}$. The new 2-bit PC code (Curve 4), however, achieves a gain of 1 dB over the rate 9/13 code, since it can detect the error event $\pm\{2, 0, -2\}$. Using the new 4-bit PC code (Curve 5), around 0.5 dB gain is obtained over the 2-bit PC code.

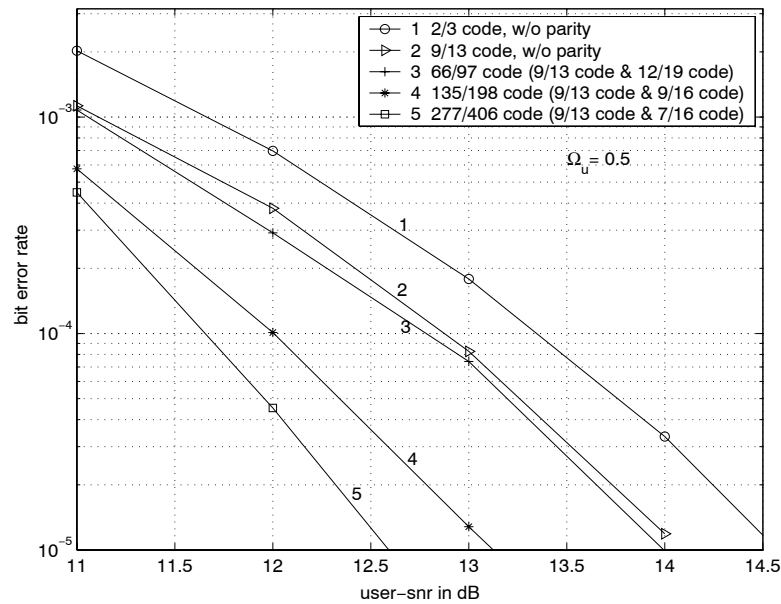


Figure 5.3: BER performance with various codes, $\Omega_u = 0.5$.

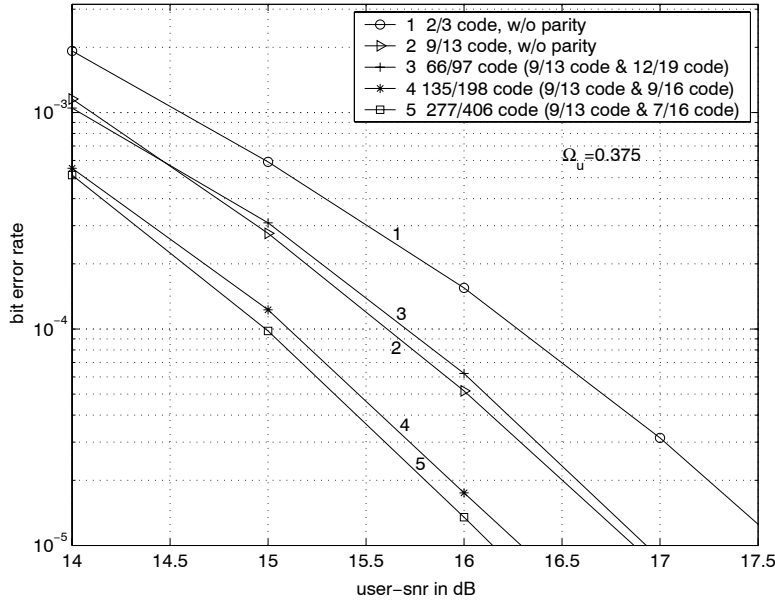


Figure 5.4: BER performance with various codes, $\Omega_u = 0.375$.

The reason is that it can detect all the dominant error events. Overall, the new constrained 4-bit PC code gains 2 dB over the system with the rate 2/3 code and without parity. At $\Omega_u = 0.375$, observe from Figures 5.4 that compared with the results at nominal density, the performance gains of PC codes are more modest. As indicated in Section 3.5.3, this is due to the reason that at high density there are many non-dominant error events, which are long events with small probabilities. These events cause difficulties for both error detection and correction. According to Figure 5.4, the new 4-bit PC code achieves an overall performance gain of 1.5 dB at high density.

5.7 Conclusions

In this chapter, a general and systematic code design technique has been proposed for constructing capacity-approaching constrained PC codes, which can detect any type of dominant error events or error event combinations in optical recording systems. The parity-check constraint corresponds to linear systematic binary PC codes. The

modulation constraint can be any practical d constraint (*i.e.* $d = 1$ and $d = 2$). Approaches have been proposed to design the code in NRZI format and NRZ format. Designing the codes in NRZ format is found to be preferable. Using the proposed method, various new codes for different optical recording systems can be designed. Application of this technique to other recording systems is straightforward. Examples of several new codes have been illustrated, and their BER performances have been evaluated. Simulation results show that the new $d = 1$ constrained 4-bit PC code can detect all the dominant error events. Compared to the rate $2/3$ code without parity, it achieves a performance gain of 2 dB at nominal density, and 1.5 dB at high density, at $\text{BER} = 10^{-5}$.

Chapter 6

Parity-Check Coded Channels with Media Noise

Media noise is the dominant noise in high-density optical recording systems [79, 1]. Being correlated, data-dependent, and non-stationary in nature, the media noise seriously degrades the performance of channel detectors and post-processors that are designed for stationary additive, white and Gaussian noise (AWGN) [65, 106]. In this chapter, our study is focused on equalization and detection, parity-check (PC) coding and post-processing for high-density optical recording systems with media noise.

The original contributions in this chapter are as follows.

- We first propose two novel modifications to the channel detector to combat media noise: i) monic constrained minimum mean square error (MMSE) based equalization with media noise considerations, and ii) a modified Viterbi detector (VD) accounting for data-dependent noise variance. The proposed modifications provide significant performance gain over the conventional detection approaches that are designed for AWGN. Moreover, they are simple and can be easily incorporated into existing partial response maximum likelihood (PRML) systems.
- A data-dependent post-processing scheme is further proposed for error correction, which has been found to provide better performance than the conventional matched-filtering type post-processors [23, 27, 15, 86, 49, 16].

- Simulation results show that the developed PC coding and post-processing scheme achieves an overall performance gain of more than 11 dB at high media noise levels.

This chapter is organized as follows. In Section 6.1, we start with an overview of existing detection schemes to combat media noise. The proposed modifications to the channel detector are presented in Section 6.2. The PC coding and data-dependent post-processing schemes for media noise channels are presented in Section 6.3. The chapter is concluded in Section 6.4.

6.1 Introduction

Several approaches have been proposed in the literature to deal with the degradation in the detector's performance due to media noise. In [65], the effect of various constraints on the PR target is studied for channels with jitter media noise, and the superiority of the monic constraint over other constraints is reported. The maximum-likelihood sequence detector (MLSD) for media noise channels is derived in [47], under the assumption that the noise can be modeled as a data-dependent finite-order Markov process¹. The resulting detector is essentially a VD that incorporates data-dependent noise prediction in its branch metric calculation. In [66], the same detector structure has been derived from the data-dependent noise prediction viewpoint. The data-dependent branch metric calculation has also been used by the soft-output post-processor proposed in [87], to correct dominant error events at the output of a conventional VD. These methods have been shown to significantly improve the performance, however, at the expense of a significant increase of computational complexity. This is because the conditional statistics of media noise depend on a large span of the input data, which leads the requirement of a large bank of noise-whitening filters whose coefficients depend on the input data and a considerable increase in the size of VD trellis.

In summary, most of the reported detection approaches are proposed for magnetic recording systems, and very little work has been done for the design of detectors for optical recording channels with

¹In a strict sense, the media noise cannot be described as a Markov process [66], and the detection approach proposed in [47] is still sub-optimal.

media noise, whose channel characteristics and code constraints are significantly different from those of magnetic recording systems. There is also no reported work on PC coding and post-processing for optical recording channels with media noise.

In this chapter, we develop advanced reception schemes for PC coded optical recording channels with media noise. As the first step, two novel modifications to the channel detector are proposed to combat media noise. Based on the media noise model proposed in Section 2.3, we first derive an analytical approach for jointly designing the PR target and equalizer, based on the MMSE criterion with monic constraint, taking into account the data-dependence property of the media noise. Simulation results show that this approach provides substantial performance gain compared to the case where the equalizer and target are designed without considering the media noise. To further account for the data-dependent nature of the media noise, while keeping the increase in complexity minimal, we propose a modified VD which changes the noise variance in its branch metrics depending on the input data. Simulation results show that the modified VD provides better performance than the conventional VD.

As the second step, we develop PC coding and post-processing schemes based on the proposed detection approach. In particular, the dominant error events at the output of the channel detector are characterized, suitable constrained PC codes are chosen for error detection, and a data-dependent post-processing scheme is developed for error correction.

6.2 Detection for Media Noise Channels

6.2.1 Monic Constrained MMSE Equalization

In this section, we present an analytical approach for jointly designing the channel PR target and equalizer, based on the media noise model presented in Section 2.3. We choose to use the monic constrained MMSE design criterion, since the monic constraint has good noise whitening ability.

Figure 6.1 shows the block diagram of an optical recording channel with electronic noise n_k and media noise γ_k , and the PRML receiver.

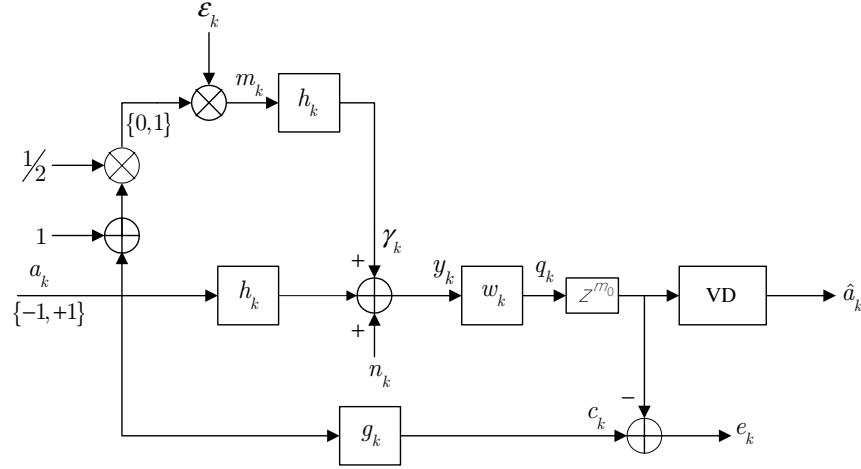


Figure 6.1: Optical recording channel with electronics noise and media noise, and the PRML receiver.

The media noise γ_k can be given as

$$\gamma_k = \sum_{i=0}^{L_h-1} h_i m_{k-i} = \mathbf{h}^T \mathbf{m}_k, \quad (6.1)$$

where $\mathbf{h} = [h_0, h_1, \dots, h_{L_h-1}]^T$ is the channel symbol response (as defined in Chapter 2) of length L_h , and $\mathbf{m}_k = [m_k, m_{k-1}, \dots, m_{k-L_h+1}]^T$. Here, $m_k = \frac{1+a_k}{2} \varepsilon_k$ corresponds to the fluctuations in the reflectivity of the crystalline marks as defined in Section 2.3. Since a_k and ε_k are mutually uncorrelated, the autocorrelation of m_k can be given by

$$\begin{aligned} E[m_k m_l] &= E \left[\left(\frac{1+a_k}{2} \varepsilon_k \right) \left(\frac{1+a_l}{2} \varepsilon_l \right) \right] \\ &= \frac{1}{4} \{ E[\varepsilon_k \varepsilon_l] + E[a_k a_l] E[\varepsilon_k \varepsilon_l] \} \\ &= \begin{cases} \frac{1}{2} \sigma_\varepsilon^2, & k = l \\ 0, & k \neq l. \end{cases} \end{aligned} \quad (6.2)$$

The equalizer output q_k , PR target output c_k , and error signal e_k

are given by

$$q_k = \sum_{i=0}^{L_w-1} w_i y_{k-i} = \mathbf{w}^T \mathbf{y}_k, \quad (6.3)$$

$$c_k = \sum_{i=0}^{L_g-1} g_i a_{k-i} = \mathbf{g}^T \mathbf{a}_k, \quad (6.4)$$

$$e_k = q_{k+m_0} - c_k = \mathbf{w}^T \mathbf{y}_{k+m_0} - c_k, \quad (6.5)$$

where y_k are equalizer input samples, $\mathbf{g} = [g_0, g_1, \dots, g_{L_g-1}]^T$ is the PR target of length L_g , $\mathbf{w} = [w_0, w_1, \dots, w_{L_w-1}]^T$ is the equalizer of length L_w , $\mathbf{a}_k = [a_k, a_{k-1}, \dots, a_{k-L_g+1}]^T$, $\mathbf{y}_k = [y_k, y_{k-1}, \dots, y_{k-L_w+1}]^T$, and m_0 is the delay introduced by the channel and equalizer. The mean square error (MSE) is thus given by

$$E[e_k^2] = \mathbf{w}^T \mathbf{R} \mathbf{w} + \mathbf{g}^T \mathbf{A} \mathbf{g} - 2\mathbf{w}^T \mathbf{P} \mathbf{g}, \quad (6.6)$$

where $\mathbf{R} = E[\mathbf{y}_{k+m_0} \mathbf{y}_{k+m_0}^T]$ is a $L_w \times L_w$ autocorrelation matrix of y_k , $\mathbf{A} = E[\mathbf{a}_k \mathbf{a}_k^T]$ is a $L_g \times L_g$ autocorrelation matrix of a_k , and $\mathbf{P} = E[\mathbf{y}_{k+m_0} \mathbf{a}_k^T]$ is a $L_w \times L_g$ cross-correlation matrix between y_k and a_k . Since a_k , γ_k and n_k are mutually uncorrelated, the (i, j) th elements of these matrices can be calculated as

$$\begin{aligned} R_{i,j} &= E[y_{k+m_0-i} y_{k+m_0-j}] = E[y_{k-i} y_{k-j}] \\ &= E[(\mathbf{h}^T \mathbf{a}_{k-i} + \gamma_{k-i} + n_{k-i}) (\mathbf{a}_{k-j}^T \mathbf{h} + \gamma_{k-j} + n_{k-j})] \\ &= \mathbf{h}^T E[\mathbf{a}_{k-i} \mathbf{a}_{k-j}^T] \mathbf{h} + \mathbf{h}^T E[\mathbf{m}_{k-i} \mathbf{m}_{k-j}^T] \mathbf{h} \\ &\quad + E[n_{k-i} n_{k-j}], \end{aligned} \quad (6.7)$$

$$A_{i,j} = E[a_{k-i} a_{k-j}], \quad (6.8)$$

$$\begin{aligned} P_{i,j} &= E[y_{k+m_0-i} a_{k-j}] \\ &= E[(\mathbf{h}^T \mathbf{a}_{k+m_0-i} + \gamma_{k+m_0-i} + n_{k+m_0-i}) a_{k-j}] \\ &= \mathbf{h}^T E[\mathbf{a}_{k+m_0-i} a_{k-j}]. \end{aligned} \quad (6.9)$$

Note that compared to the case of monic constrained MMSE equalization with AWGN only, the media noise results in the extra term $\mathbf{h}^T E[\mathbf{m}_{k-i} \mathbf{m}_{k-j}^T] \mathbf{h}$ in $R_{i,j}$, while matrices \mathbf{A} and \mathbf{P} remain the same.

The monic constraint of PR target can be specified as

$$\mathbf{g}^T \mathbf{i} = 1, \quad \mathbf{i} = [1, 0, \dots, 0]^T. \quad (6.10)$$

By applying the Lagrange multiplier method [3], we obtain the optimum PR target and equalizer as

$$\mathbf{g}_{\text{opt}} = \frac{(\mathbf{A} - \mathbf{P}^T \mathbf{R}^{-1} \mathbf{P})^{-1} \mathbf{i}}{\mathbf{i}^T (\mathbf{A} - \mathbf{P}^T \mathbf{R}^{-1} \mathbf{P})^{-1} \mathbf{i}}, \quad (6.11)$$

$$\mathbf{w}_{\text{opt}} = \mathbf{R}^{-1} \mathbf{P} \mathbf{g}_{\text{opt}}, \quad (6.12)$$

$$\xi_{\text{min}} = \frac{1}{\mathbf{i}^T (\mathbf{A} - \mathbf{P}^T \mathbf{R}^{-1} \mathbf{P})^{-1} \mathbf{i}}, \quad (6.13)$$

with ξ_{min} being the resulting minimum MSE.

6.2.2 Viterbi Detection with Data-Dependent Noise Variance

In the previous section, in designing the optimum equalizer and target, we accounted for the presence of media noise by modifying the underlying correlations as in (6.7)~(6.9). In this section, we modify the VD to further account for the data-dependent nature of the media noise. This is done by changing the branch-metric computation in VD in view of the fact that the noise variance at the VD input changes with data patterns for media noise channels.

The principle of the modified VD is as follows. We assume that the total noise $\{e_k\}$ at the input of VD, which consists of the equalized electronics noise, equalized media noise and residual intersymbol interference (ISI), to be a sequence of independent Gaussian random variables. Furthermore, the variance $\sigma_e^2(\mathbf{d}_k)$ of each e_k depends on the input data pattern $\mathbf{d}_k = [a_{k+I_2}, a_{k+I_2-1}, \dots, a_{k-I_1}]^T$, with I_1 and I_2 being nonnegative integers. Therefore, the joint probability density function (pdf) of the VD input samples $\mathbf{q} = [q_1, q_2, \dots, q_{N+L_g-1}]^T$, conditioned on the input bit sequence $\mathbf{a} = [a_1, a_2, \dots, a_N]^T$, is given by

$$p_q(\mathbf{q}|\mathbf{a}) = \prod_{k=1}^{N+L_g-1} \frac{1}{\sqrt{2\pi\sigma_e(\mathbf{d}_k)}} \exp\left(-\frac{(q_k - \tilde{q}_k)^2}{2\sigma_e^2(\mathbf{d}_k)}\right), \quad (6.14)$$

where $\tilde{q}_k = \sum_{i=0}^{L_g-1} g_i a_{k-i}$ is the reconstruction of the signal part of q_k based on the assumed data sequence \mathbf{a} . According to the MLSD criterion [29], we need to find a particular \mathbf{a} such that $p_q(\mathbf{q}|\mathbf{a})$ is maximized

over all possible data sequences. Maximizing (6.14) is equivalent to minimizing $\sum_{k=1}^{N+L_g-1} \left(\ln \sigma_e(\mathbf{d}_k) + \frac{(q_k - \tilde{q}_k)^2}{2\sigma_e^2(\mathbf{d}_k)} \right)$. Thus, the branch metric of the modified VD is given by $\ln \sigma_e(\mathbf{d}_k) + \frac{(q_k - \tilde{q}_k)^2}{2\sigma_e^2(\mathbf{d}_k)}$, instead of $(q_k - \tilde{q}_k)^2$ in the conventional VD.

To implement the modified VD, the data-dependent noise variance $\sigma_e^2(\mathbf{d}_k)$ needs to be computed. For the system shown in Figure 6.1, we have

$$\begin{aligned}
e_k &= q_{k+m_0} - c_k \\
&= \sum_{i=0}^{L_p-1} p_i m_{k+m_0-i} + \sum_{i=0}^{L_w-1} w_i n_{k+m_0-i} + \sum_{i=0}^{L_p-1} p_i a_{k+m_0-i} - \sum_{i=0}^{L_g-1} g_i a_{k-i} \\
&= \underbrace{\mathbf{p}^T \mathbf{m}_{k+m_0}}_{\text{equalized media noise}} + \underbrace{\mathbf{w}^T \mathbf{n}_{k+m_0}}_{\text{equalized white noise}} + \underbrace{\mathbf{u}^T \mathbf{a}_{k+m_0}}_{\text{residual ISI}}, \quad (6.15)
\end{aligned}$$

where $p_k = \sum_{i=0}^{L_h-1} h_i w_{k-i}$ is the equalized channel response with length $L_p = L_h + L_w - 1$, and $u_k = p_k - g'_k$ is the residual ISI channel with

$$g'_k = \begin{cases} 0, & 0 \leq k \leq m_0 - 1 \\ g_k, & m_0 \leq k \leq m_0 + L_g - 1 \\ 0, & m_0 + L_g \leq k \leq L_p - 1. \end{cases} \quad (6.16)$$

Therefore, the noise variance is given by

$$\sigma_e^2(\mathbf{d}_k) = E[e_k^2 | \mathbf{d}_k] - (E[e_k | \mathbf{d}_k])^2, \quad (6.17)$$

where

$$\begin{aligned}
E[e_k^2 | \mathbf{d}_k] &= E \left[(\mathbf{p}^T \mathbf{m}_{k+m_0} + \mathbf{w}^T \mathbf{n}_{k+m_0} + \mathbf{u}^T \mathbf{a}_{k+m_0})^2 | \mathbf{d}_k \right] \\
&= \mathbf{p}^T E[\mathbf{m}_{k+m_0} \mathbf{m}_{k+m_0}^T | \mathbf{d}_k] \mathbf{p} + \mathbf{w}^T E[\mathbf{n}_{k+m_0} \mathbf{n}_{k+m_0}^T] \mathbf{w} \\
&\quad + \mathbf{u}^T E[\mathbf{a}_{k+m_0} \mathbf{a}_{k+m_0}^T | \mathbf{d}_k] \mathbf{u}, \quad (6.18)
\end{aligned}$$

$$\begin{aligned}
E[e_k | \mathbf{d}_k] &= E[\mathbf{p}^T \mathbf{m}_{k+m_0} + \mathbf{w}^T \mathbf{n}_{k+m_0} + \mathbf{u}^T \mathbf{a}_{k+m_0} | \mathbf{d}_k] \\
&= \mathbf{u}^T E[\mathbf{a}_{k+m_0} | \mathbf{d}_k]. \quad (6.19)
\end{aligned}$$

All the above conditional expectations can be calculated from the knowledge of the data pattern \mathbf{d}_k , the autocorrelation function of the

input data a_k , and the autocorrelation and conditional autocorrelation functions of the media noise m_k . In particular, the conditional autocorrelation of $E[m_i m_j | a_i, a_j]$, with $k - I_1 \leq i, j \leq k + I_2$, is given by

$$\begin{aligned} E[m_i m_j | a_i, a_j] &= E \left[\left(\frac{1 + a_i}{2} \varepsilon_i \right) \left(\frac{1 + a_j}{2} \varepsilon_j \right) \mid a_i, a_j \right] \quad (6.20) \\ &= \frac{1}{4} (1 + a_i)(1 + a_j) E[\varepsilon_i \varepsilon_j] \\ &= \begin{cases} \frac{1}{4} (1 + a_i)^2 \sigma_\varepsilon^2, & i = j, \\ 0, & i \neq j. \end{cases} \end{aligned}$$

The choice of I_1 and I_2 for defining the data pattern \mathbf{d}_k is a trade-off between the performance gain and computation complexity. In this work, we set $I_1 = L_g - 1$ and $I_2 = 0$. This ensures that the modified branch-metric can be easily incorporated into conventional VD without increasing the trellis size.

6.2.3 BER Simulation Results

In this part, we illustrate the improvement in BER performance achieved due to the detector modifications presented in Sections 6.2.1 and 6.2.2. The conventional detection approaches selected for comparison are designed without considering media noise, *i.e.* MMSE equalization with the central taps of the channel symbol response taken as the PR target (legend: central 5-tap MMSE) or MMSE equalization with monic constraint (legend: conv. monic MMSE), with the detector being the conventional VD which assumes the total noise e_k at its input as white Gaussian and data-independent. The target length is 5 for all the approaches. The new approaches are denoted ‘prop. monic MMSE’ (*i.e.* MMSE equalization accounting for media noise [Section 6.2.1] and conventional VD), and ‘modified VD’ (*i.e.* monic MMSE equalization of Section 6.2.1 combined with the modified branch-metric VD of Section 6.2.2).

We take two different normalized variances of disc reflectivity fluctuations (in the user bandwidth), *i.e.* $\tilde{\Delta}_\varepsilon = 2\%$ and $\tilde{\Delta}_\varepsilon = 3\%$, as examples to study the performance. The normalized cut-off frequency is $\Omega_u = 0.375$. The corresponding BER performances are illustrated in Figure 6.2.

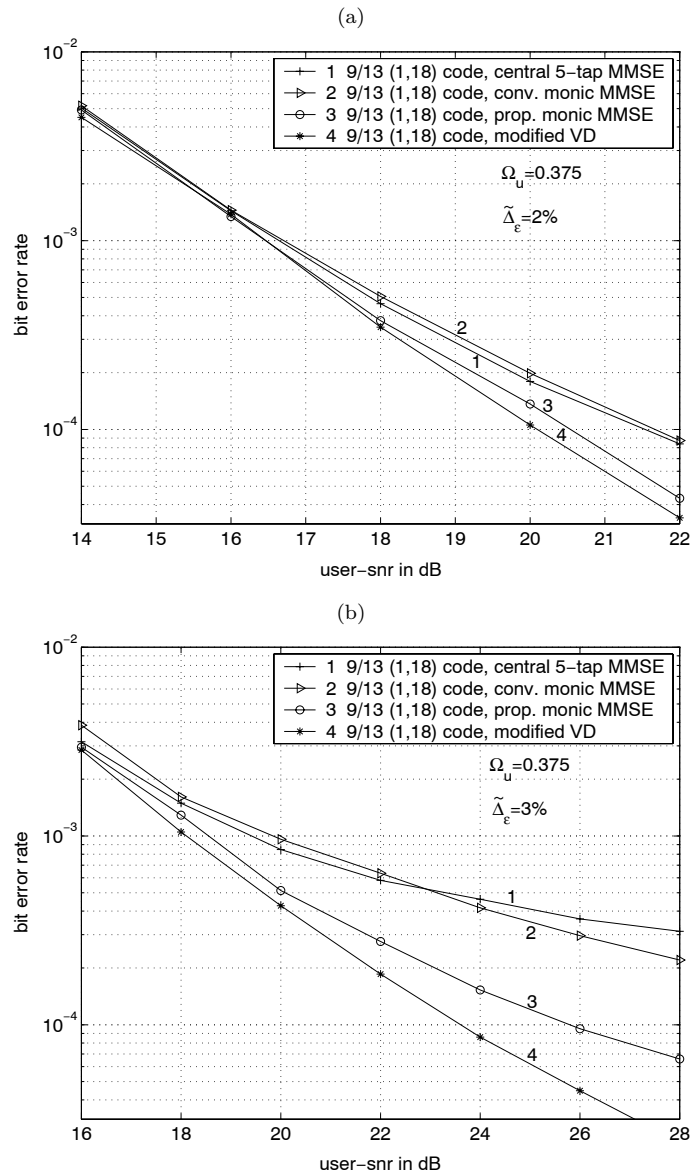


Figure 6.2: BER performance comparison of various detection approaches, without parity-check (PC) codes, at $\Omega_u = 0.375$. (a) $\tilde{\Delta}_\varepsilon = 2\%$; (b) $\tilde{\Delta}_\varepsilon = 3\%$.

From Figure 6.2, we observe that the new detection approaches provide significant performance gain over the conventional detection approaches. Furthermore, the gain increases with increase in media noise power. In particular, with $\tilde{\Delta}_\varepsilon = 2\%$, the ‘prop. monic MMSE’ outperforms the ‘conv. monic MMSE’ by more than 1 dB at $\text{BER} = 10^{-4}$. The modified VD provides an additional 0.5 dB gain. The performance gains become very significant in the case of $\tilde{\Delta}_\varepsilon = 3\%$. In this case, the ‘prop. monic MMSE’ achieves a gain of more than 5 dB compared to ‘conv. monic MMSE’, at $\text{BER} = 10^{-4}$. The additional gain provided by modified VD is more than 2 dB. Thus, as the amount of media noise increases, the proposed detector modifications help to significantly delay the error floor caused by media noise.

6.3 PC Codes and Post-processing for Media Noise Channels

In this section, we develop PC codes and post-processors for high-density optical recording channels with media noise. The PC coding and post-processing are implemented based on the modified VD approach, which outperforms other detection approaches as shown in the previous section.

6.3.1 Parity-Check Codes

Prior to designing PC codes and post-processors, the dominant error events at the output of VD need to be determined. For the modified VD approach, the dominant error events turn out to be $\pm\{2\}$, $\pm\{2, 0, -2\}$ and $\pm\{2, 0, -2, 0, 2, 0, -2\}$. The histograms of error event distributions with $\tilde{\Delta}_\varepsilon = 2\%$ and $\tilde{\Delta}_\varepsilon = 3\%$, obtained from simulations based on a total of 10^4 error events, are illustrated in Figures 6.3(a) and (b), respectively. As expected, the error event distribution for the low media noise case of $\tilde{\Delta}_\varepsilon = 2\%$ is more similar to that of the AWGN case shown in Figure 3.4(b). For the case with a high media noise power of $\tilde{\Delta}_\varepsilon = 3\%$, the probability of the single-bit error event is much larger, and the probability of the event $\pm\{2, 0, -2\}$ is much smaller than that with AWGN. Furthermore, the probability of the other error events is also smaller. These characteristics enable less mis-corrections of the post-processor and much higher performance

6.3 PC Codes and Post-processing for Media Noise Channels 119

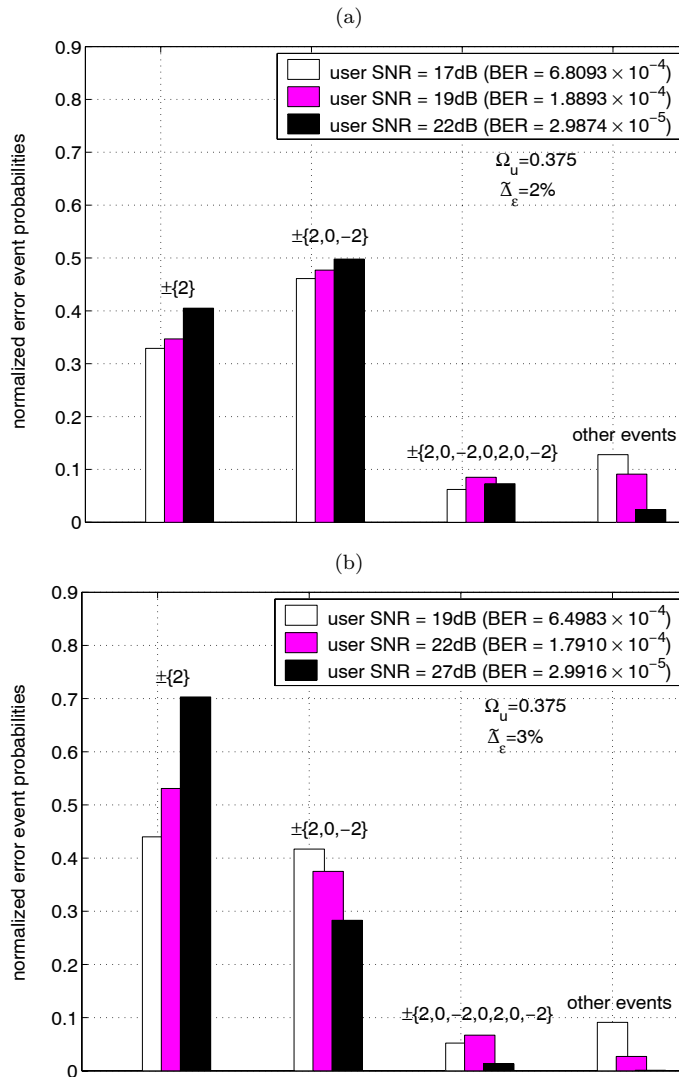


Figure 6.3: Histogram of dominant error events, without parity-check (PC) codes, at $\Omega_u = 0.375$. (a) $\tilde{\Delta}_\varepsilon = 2\%$; (b) $\tilde{\Delta}_\varepsilon = 3\%$.

gain from PC coding and post-processing. In addition, for both cases, there are lesser types of dominant error events. Referring to Table 3.2 of Chapter 3, we conclude that by applying 2 or more parity bits, we

can detect all the dominant error events.

6.3.2 Data-Dependent Post-Processing

We have proved in Chapter 3 that for channels with AWGN, both the MAP and ML based post-processing can be implemented by using a simple matched-filtering type post-processor. However, this post-processor is no longer optimum in the presence of media noise. In this section, we propose a ML based post-processing scheme to account for the data-dependence of the noise variance.

According to Section 3.3.1, the ML based decision rule for post-processing can be expressed as

$$\hat{\mathbf{e}} = \arg \max_{\mathbf{e}_i^j} p(\mathbf{q} | \mathbf{e}_i^j, \hat{\mathbf{a}}), \quad (6.21)$$

where $p(\mathbf{q} | \mathbf{e}_i^j, \hat{\mathbf{a}})$ is the pdf of the VD input samples \mathbf{q} conditioned on the detected bits $\hat{\mathbf{a}} = [\hat{a}_1, \hat{a}_2, \dots, \hat{a}_N]^T$ and an assumed error event \mathbf{e}_i^j of length L_e^j and type j , starting at position i , with $i = 1, 2, \dots, N - L_e^j + 1$. Using the same assumptions as in Section 6.2.2 for the total noise e_k at the VD input, we can rewrite (6.21) as

$$\begin{aligned} \hat{\mathbf{e}} &= \arg \max_{\mathbf{e}_i^j} \prod_{k=1}^{N+L_e^j+L_g-2} \frac{1}{\sqrt{2\pi}\sigma_e(\tilde{\mathbf{d}}_{k,i}^j)} \exp\left(-\frac{(q_k - \tilde{q}_{i,k}^j)^2}{2\sigma_e^2(\tilde{\mathbf{d}}_{k,i}^j)}\right), \\ &= \arg \min_{\mathbf{e}_i^j} \sum_{k=1}^{N+L_e^j+L_g-2} \left(\ln \sigma_e(\tilde{\mathbf{d}}_{k,i}^j) + \frac{(q_k - \tilde{q}_{i,k}^j)^2}{2\sigma_e^2(\tilde{\mathbf{d}}_{k,i}^j)} \right), \end{aligned} \quad (6.22)$$

where $\tilde{q}_{i,k}^j = \sum_{l=0}^{L_g-1} g_l(\hat{a}_{k-l} + e_{i,k-l}^j)$ are the re-constructed VD input samples based on \mathbf{e}_i^j , and $\tilde{\mathbf{d}}_{k,i}^j = \hat{\mathbf{d}}_k + \mathbf{e}_i^j$, with $\hat{\mathbf{d}}_k = [\hat{a}_{k+I_2}, \hat{a}_{k+I_2-1}, \dots, \hat{a}_{k-I_1}]^T$. In (6.22), unlike the case with AWGN, where the variance of e_k is a constant, $\sigma_e^2(\tilde{\mathbf{d}}_{k,i}^j)$ varies with $\tilde{\mathbf{d}}_{k,i}^j$ for media noise channels, and therefore cannot be dropped from the cost function.

The proposed data-dependent post-processing scheme is summarized as follows.

(1) For each of the dominant error events, determine all valid starting positions of the event so that both the *syndrome criterion* and the $d = 1$ *criterion* defined in Section 3.3.2 are satisfied. A set of candidate error events is thus obtained.

(2) Compute $\sum_{k=1}^{N+L_e^j+L_g-2} \left(\ln \sigma_e(\tilde{\mathbf{d}}_{k,i}^j) + \frac{(q_k - \tilde{q}_{i,k}^j)^2}{2\sigma_e^2(\tilde{\mathbf{d}}_{k,i}^j)} \right)$ for each error event in the set of candidate error events. The error event with the minimum distance is determined as the most likely event.

We remark here that the remedy scheme developed in Section 3.4.2 for correction of boundary error events applies to the data-dependent post-processor as well. Further, in principle, the above data-dependent post-processing scheme can also be generalized for multiple-error-event correction. However, this may result in prohibitive computational complexity. Therefore, low-complexity sub-optimum schemes have to be developed, which is beyond the scope of this thesis.

6.3.3 BER Simulation Results

We now present simulation results on media noise channels with PC codes and post-processing. We perform the simulations for $\tilde{\Delta}_\varepsilon = 2\%$ and $\tilde{\Delta}_\varepsilon = 3\%$, with $\Omega_u = 0.375$. For media noise channels, as stated in Section 6.3.1, PC codes with 2 or more parity bits can detect all the dominant error events. Therefore, we use the rate 135/198 2-bit constrained PC code, designed in Chapter 5, and the corresponding generator polynomial is $g(x) = 1 + x + x^2$. The BER performances for different media noise powers are illustrated in Figure 6.4.

In the figures, Curves 1 to 4 illustrate various detectors' performances without PC codes and post-processing. They are the same as those in Figure 6.2, and are used here as references. The PC coding and post-processing are applied only in the modified VD case, since it outperforms other detection approaches. For the ease of comparison, the performances of two types of post-processing schemes have been evaluated, which are the conventional matched-filtering type post-processor (legend: MF-PP), and the proposed post-processor with data-dependent noise variance (legend: DD-PP). The resulting performances are illustrated by Curves 5 and 6, respectively. Note that the 4-bit constrained PC code (*i.e.* the rate 277/406 code designed in Chapter 5) has been found to achieve similar performance as the 2-bit PC code, and hence its performance is not shown in the figures.

Observe that in conjunction with the 2-bit PC code, both post-processors achieve significant performance gains over the systems without PC codes. Furthermore, the data-dependent post-processor provides much better performance than the conventional matched-filtering

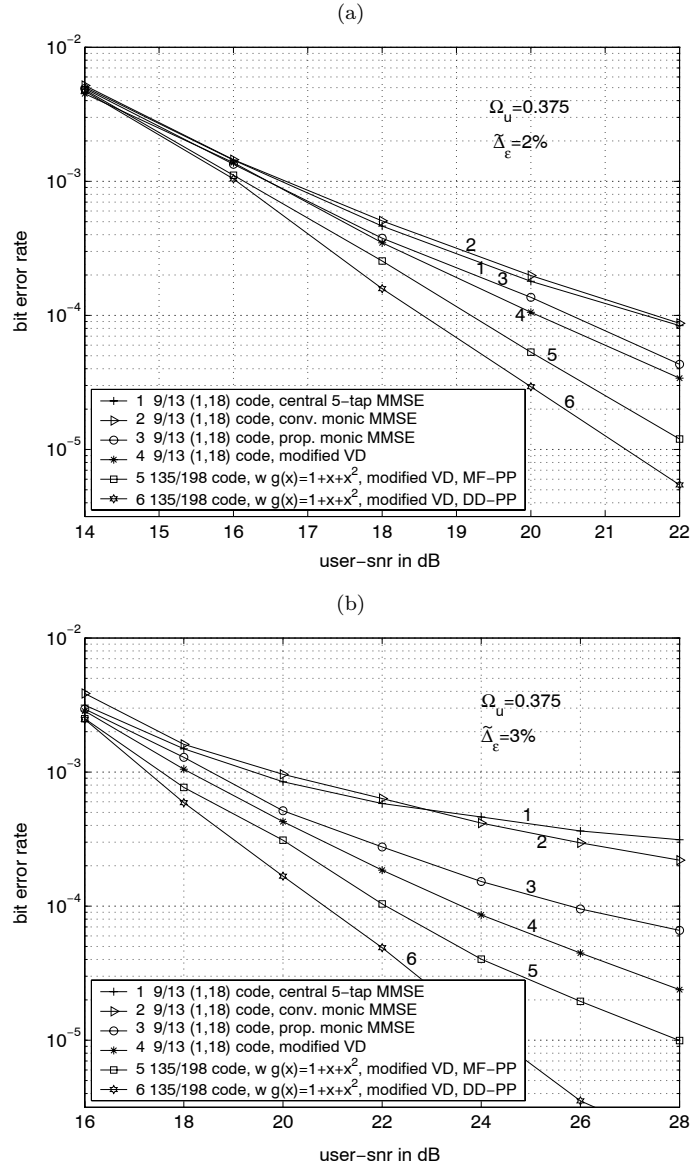


Figure 6.4: BER performance of modified VD with parity-check (PC) code and different post-processors, at $\Omega_u = 0.375$. (a) $\tilde{\Delta}_\varepsilon = 2\%$; (b) $\tilde{\Delta}_\varepsilon = 3\%$.

type post-processor. The performance gains increase with increase in media noise power. With $\tilde{\Delta}_\varepsilon = 2\%$, the PC code with the MF-PP achieves more than 1 dB improvement over the modified VD without PC codes, at $\text{BER} = 3 \times 10^{-5}$. The corresponding gain of DD-PP is more than 2 dB. The performance gains with $\tilde{\Delta}_\varepsilon = 3\%$ are significantly larger. Compared to the modified VD without PC codes, the performance improvements of the two post-processors are more than 2 dB and 4 dB, respectively, at $\text{BER} = 3 \times 10^{-5}$. If we compare with the conventional detection approaches without PC codes (*i.e.* Curves 1 and 2), the overall performance gain is larger than 11 dB, at $\text{BER} = 10^{-4}$.

6.4 Conclusion

In this chapter, we have investigated the application of PC codes and post-processing for high-density optical recording channels with media noise. As the first step, we proposed two novel modifications to the channel detector, namely, the monic constrained MMSE equalization taking into account the media noise, and the modified VD with data-dependent noise variance. Simulation results show that these approaches provide significant performance gain over the conventional detection approaches where target and equalizer are designed without considering the media noise. Furthermore, they are simple and can be easily incorporated into existing PRML systems. As the second step, the dominant error events of the proposed detection approach have been found, and suitable PC codes have been chosen for error detection. Furthermore, a data-dependent post-processing scheme has been proposed for error correction. It outperforms the conventional matched-filtering type post-processors by accounting for the data-dependence of the media noise variance. Furthermore, as the amount of media noise increases, the proposed detector modifications, PC coding and post-processing scheme help to significantly delay the error floor caused by media noise, and provide much larger performance gains. Compared to the system designed without considering media noise and without PC codes, the overall performance gain of the developed scheme is more than 11 dB at $\text{BER} = 10^{-4}$, at high media noise levels.

Chapter 7

Performance Analysis with Error Correction Code

In this chapter, our study is focused on doing the performance analysis of the developed parity-check (PC)-code-based optical recording systems, taking into account the error correction code (ECC). The byte error rate (ByER) at the input of the Reed-Solomon (RS)-ECC decoder, as well as the ECC failure rate (EFR) are evaluated.

The original contributions in this chapter are as follows.

- We generalize the multinomial and block multinomial methods, which are used for estimating the failure rate of non-interleaved RS-ECC, to the interleaved case, for constrained codes without and with parity. We demonstrate the accuracy of these semi-analytical approaches using simulation results.
- Our analysis on EFR of PC coded systems shows that in the non-interleaving case, the performance of constrained PC codes lags behind that of constrained codes without parity, due to the long error bursts caused by the mis-corrections of the post-processor.
- Interleaving can effectively spread these long error bursts into different interleaves and enable the constrained PC codes to outperform the constrained codes without parity. Optimum interleaving degrees have been found for various codes without and with parity.
- It is observed that with optimum interleaving degrees, the 4-bit

PC code outperforms the 2-bit code, since it can correct more errors on the average.

This chapter is organized as follows. An introduction to the analysis of the PC-code-based reception techniques at the ECC level, together with the description of the overall recording system model are given in Section 7.1. The ByER evaluation of the PC-code-based system is presented in Section 7.2. In Section 7.3, semi-analytical approaches for analyzing the EFR are presented. In Section 7.4, the EFRs of the system are presented. Finally, concluding remarks are given in Section 7.5.

7.1 Introduction

The performance of data storage systems is usually measured by bit error rate (BER) at the output of the channel detector (or the inverse precoder) or ByER at the output of the constrained decoder. The BER and ByER, however, do not give a complete picture of the performance while comparing different coding and detection schemes. Referring to Figure 1.1, since the user data is first encoded by the ECC encoder, the EFR serves as the ultimate measure of the data recovery performance. For data storage systems, the EFR is typically in the range of $10^{-12} \sim 10^{-15}$, and this makes the estimation of EFR by the direct count approach (*i.e.* computer simulations) impractical. Therefore, development of analytical or semi-analytical methods for estimating the EFR becomes necessary.

In recent years, there have been several papers on post-ECC analysis of the PC-code-based detection approaches for magnetic recording systems. In [27, 15, 86], a semi-analytical approach based on the multinomial model has been used to evaluate the EFR. These studies focus on a three-way interleaved RS code which can correct up to $t = 5$ symbol errors per codeword. In [27, 15], it is shown that although multiple-bit PC codes outperform single-bit PC code in BER, they perform no better or worse than the single-bit PC code in terms of EFR. A block multinomial model has been further proposed in [49], which provides better estimation of the EFR of PC coded systems without interleaving. For optical recording systems, whose ECC configurations as well as code constraints [70, 17, 45] differ significantly

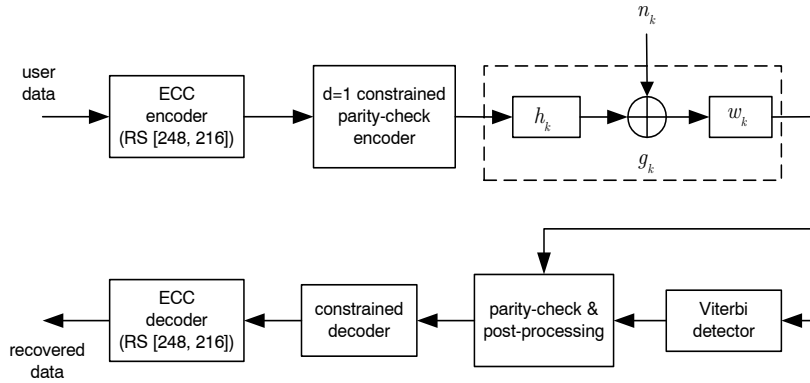


Figure 7.1: Block diagram of optical recording system with ECC and constrained parity-check (PC) code.

from those of magnetic recording systems, no report has been found on the analysis of EFR.

In Chapters 3 to 6, we evaluated the performance of the new codes and detection schemes in terms of BER, for high-density blue laser disc systems. In this chapter, we include ECC into the study, and investigate the EFR of the system.

A block diagram of a PC-code-based optical recording system is shown in Figure 7.1, and it is used for analyzing the ECC performance. It is an extension of Figure 3.1, with ECC being taken into consideration. Our study focuses on a RS [248, 216] code with 8-bit/symbol (byte), since it serves as the key constituent of the ECC configuration for BD [70]. We use channels with additive, white and Gaussian noise (AWGN) as an example to analyze the EFR of PC coded systems. Applying a similar analysis to media noise channels is straightforward. However, for the sake of keeping the presentation simple, brief and clear, we do not consider media noise in this chapter.

During the performance comparison among different constrained codes without and with parity, as what has been done in Chapters 3 to 6 for the case without ECC, we always kept the user density the same, and changed the channel density to account for the different code rates. However, as described in Section 2.2, when ECC is included in the

study, its code rate needs to be included while defining R and $\tilde{\Omega}_u$. In this chapter, the study focuses on a fixed RS [248, 216] code and a high recording density. Therefore, we obtain $\tilde{\Omega}_u = 0.375/(216/248) = 0.43$. In addition, by including the code rate of RS-ECC in the ‘user signal-to-noise ratio (SNR)’ defined by (2.18) and (2.21), its impact on the SNR definition has been taken into account.

7.2 Byte Error Rate of PC Coded Systems

Before analyzing the ECC performance, it is necessary to investigate the ByER of the PC coded system at the input of the RS-ECC decoder. Compared with BER at the output of the VD and post-processor, the ByER accounts for the impact of the inverse-precoder and the constrained decoder on the system performance.

Figure 7.2 illustrates the ByER of various constrained PC codes designed in Chapter 5, at the input of the [248, 216] RS decoder. The

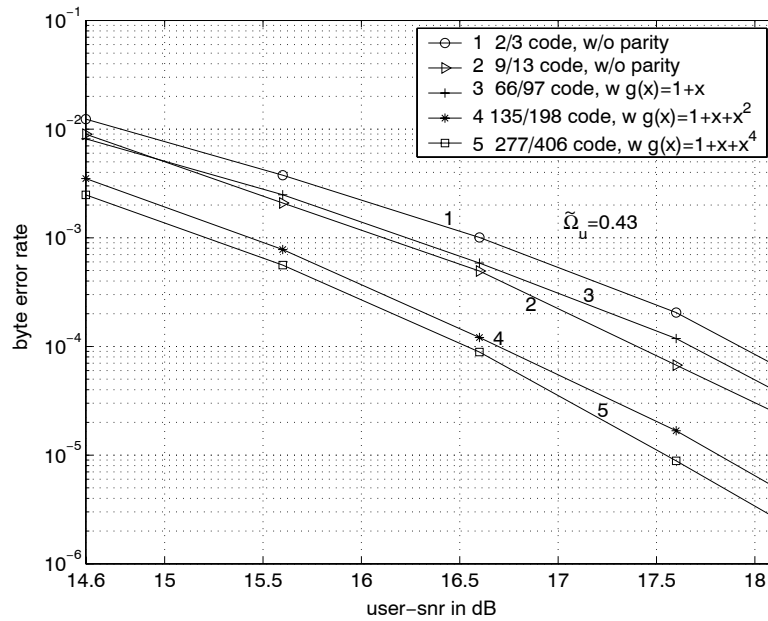


Figure 7.2: *ByER performance with various constrained PC codes, $\tilde{\Omega}_u = 0.43$.*

ByER is evaluated at a high recording density of $\tilde{\Omega}_u = 0.43$. Compared with the BER shown in Figure 5.4, the relative performances of various codes remain the same. This shows that the error propagation in the decoders of the designed constrained PC codes is as limited as that of the rate 9/13 code. This is due to the reason that all the codes are designed using the same finite-state encoding method, which results in a minimum look-ahead of one codeword during decoding. Note that the ByER of the 4-bit PC code (Curve 5) is smaller than that of 2-bit PC code (Curve 4), since the latter cannot detect the dominant error event $\pm\{2, 0, -2, 0, 2\}$ (see Table 3.2). Further, since the performance of the rate 9/13 code (Curve 2) is better than that of the rate 2/3 code (Curve 1) and the single-bit PC code (Curve 3), the latter two codes are not included in the following sections while evaluating the EFR. In addition, we remark that for all the constrained codes under study (*i.e.* the rate 9/13 code, the 2-bit and 4-bit PC codes), the sizes of their input symbols are not 8 (or multiple of 8) bits. The mismatch of symbol sizes between the constrained codes and RS-ECC results in error propagation. By using sufficiently large degree of interleaving, this error propagation can be minimized. Alternatively, this error propagation can be avoided by using the ‘modified concatenation’ scheme [9, 43, 26].

7.3 Semi-Analytical Approaches for Analyzing EFR

7.3.1 Multinomial Model

Several analytical approaches are described in the literature for estimating the RS-ECC’s failure rate [27, 104, 10]. Among these approaches, the multinomial model proposed in [27] has been found to be simple and efficient, and it has been used widely for RS-ECC’s failure rate evaluation [15, 86, 85]. In this model, it is assumed that consecutive byte errors at the RS decoder input are caused by independent error events. Therefore, the number of consecutive byte errors in a received RS codeword can be modeled by a multinomial distribution. The EFR is calculated by adding the probabilities that sufficient number of error events occur leading to $t + 1$ or more byte errors in a RS codeword. For the case without interleaving, the failure

rate P_f of RS-ECC can be expressed as [27]

$$P_f = \sum_{\sum_{k=1}^l kx_k \geq t+1} \frac{n!}{\prod_{k=0}^l x_k!} \prod_{k=0}^l p_k^{x_k}, \quad (7.1)$$

where n is the RS codeword size in symbols, x_k 's are positive integers such that $\sum_{k=0}^l x_k = n$, and l is the maximum length of consecutive byte errors within a RS codeword. Here, $\sum_{k=0}^l p_k = 1$, where $p_k = \Pr(x_k)$ is the probability of consecutive k byte errors (with $1 \leq k \leq l$), and $p_0 = 1 - \sum_{k=1}^l p_k$ is the probability of a byte without any error. Through computer simulations, it is easy to estimate p_k , based on which P_f can be obtained.

In [49], a computationally more efficient method has been proposed to calculate P_f . It uses a polynomial

$$p(D) = p_1 D + p_2 D^2 + \cdots + p_k D^k + \cdots + p_l D^l, \quad (7.2)$$

to characterize the multinomial model. The EFR, P_f , is then given by

$$P_f = \sum_{j \geq 1}^{\min(n, j_{\max})} p_0^{n-j} p_{j,t+1} \frac{n!}{j!(n-j)!}, \quad (7.3)$$

where $p_{j,t+1}$ is the probability of having j error events which lead to $t+1$ or more byte errors in a RS codeword. Here, $j_{\max} = 2t$ is a truncation parameter, beyond which $p_{j,t+1}$ is negligible. The value of $p_{j,t+1}$ can be computed in a recursive manner. It is illustrated by Figure 7.3. The accuracy of (7.3) is the same as (7.1), however it can be computed much faster than (7.1), especially for large l .

7.3.2 Block Multinomial Model

In the multinomial model, it is assumed that the consecutive byte errors are caused by independent error events. For systems with PC codes and post-processing, such an assumption may not be accurate, since it does not account for the dependence of byte errors due to the mis-correction of the post-processor. In [49], a block multinomial model is proposed, which defines byte errors on a PC codeword basis, and consecutive PC codewords are considered to be independent.

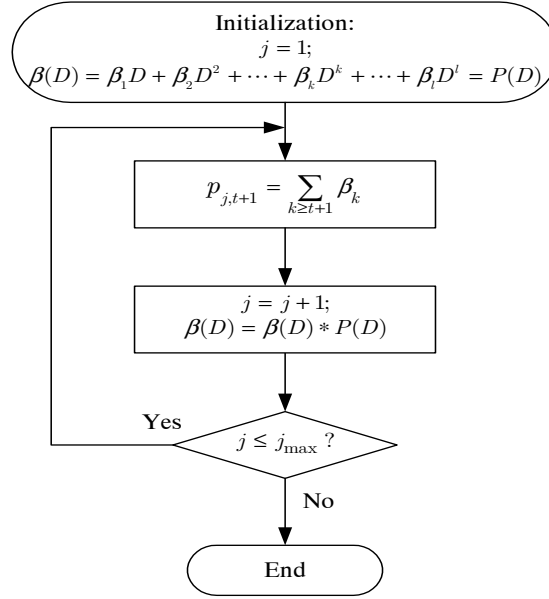


Figure 7.3: Flow chart to compute $p_{j,t+1}$.

The block multinomial model is based on the polynomial

$$q(D) = q_1 D + q_2 D^2 + \cdots + q_k D^k + \cdots + q_r D^r, \quad (7.4)$$

where q_k is the probability of receiving a PC codeword with k byte errors at the output of the constrained decoder, and r is the maximum number of byte errors within a PC codeword. The EFR P_f is then given by [49]

$$P_f = \sum_{j \geq 1}^{\min(N_{pc}, j_{\max})} q_0^{N_{pc}-j} \cdot q_{j,t+1} \cdot \frac{N_{pc}!}{j!(N_{pc}-j)!}, \quad (7.5)$$

where $N_{pc} = \lfloor \frac{n}{s} \rfloor$ is the number of PC codewords in each ECC codeword, s is the number of symbols in each PC codeword, and $q_0 = 1 - \sum_{j \geq 1}^r q_j$.

7.3.3 Accuracy Analysis for Non-Interleaved ECC

In this part, we verify the above described multinomial methods for the developed PC coded systems, with non-interleaved RS-ECC. We

found that at reasonable SNRs (*e.g.* 14 ~ 16 dB), it is not feasible to estimate the failure rate of the RS [248, 216] code ($t = 16$) using the direct count approach, since the corresponding failure rates are lower than 10^{-4} (see Figure 7.8), and obtaining such low EFRs needs excessively large simulation times. Therefore, we use a RS [248, 238] code (byte-oriented), which has a weaker error correction power of $t = 5$, to do the verification. This is based on the fact that the principle used by (7.1), (7.3) and (7.5) to compute the EFR P_f (*i.e.* P_f corresponds to the probabilities that sufficient number of error events occur leading to $t + 1$ or more byte errors in a RS codeword) applies equally well to any t . That is, for a given system, the value of t does not influence the accuracy of the EFR given by (7.1), (7.3) and (7.5). This fact has also been corroborated in [49]. For the same reason, this RS code is also used in Section 7.3.5, to analyze the accuracy of the failure rate estimation for interleaved RS-ECC. For the RS [248, 238] code, the value of $\tilde{\Omega}_u$ is taken as $\tilde{\Omega}_u = 0.375/(238/248) = 0.39$ in the performance evaluation. For a given constrained code, $\tilde{\Omega}_u = 0.39$ with the RS [248, 238] code will result in the same Ω_c as the case of $\tilde{\Omega}_u = 0.43$ with the RS [248, 216] code. Therefore, this helps to keep the error events that occur at the input of the RS [248, 238] decoder to be the same as those occur at the input of the RS [248, 216] decoder, and facilitates an easy verification of the multinomial methods. Similar approaches can be found in [49].

The comparison of EFRs obtained from the multinomial methods and from the direct count approach is shown in Figure 7.4, for without and with PC codes. For the multinomial methods, the byte error statistics (*i.e.* the probabilities p_k and q_k in (7.2) and (7.4)) are estimated based on a total of 10^4 byte errors. For the direct count approach, at least 100 ECC error codewords are collected for every estimation of the EFR. Observe that for the rate 9/13 code without parity, the EFRs estimated by the multinomial method (Curve 2) match those obtained from the direct count approach (Curve 1) very well. For the 4-bit PC code, however, the EFRs obtained from the multinomial method (Curve 4) have a bias of about 1 dB at $\text{EFR} = 5 \times 10^{-4}$. This will lead to underestimating the EFRs for the case with PC. On the other hand, by using the block multinomial method (with $s = 35$ and $N_{pc} \approx 7$) to account for the dependence of byte errors within a PC codeword, we obtain EFRs (Curve 5) which are very close to those

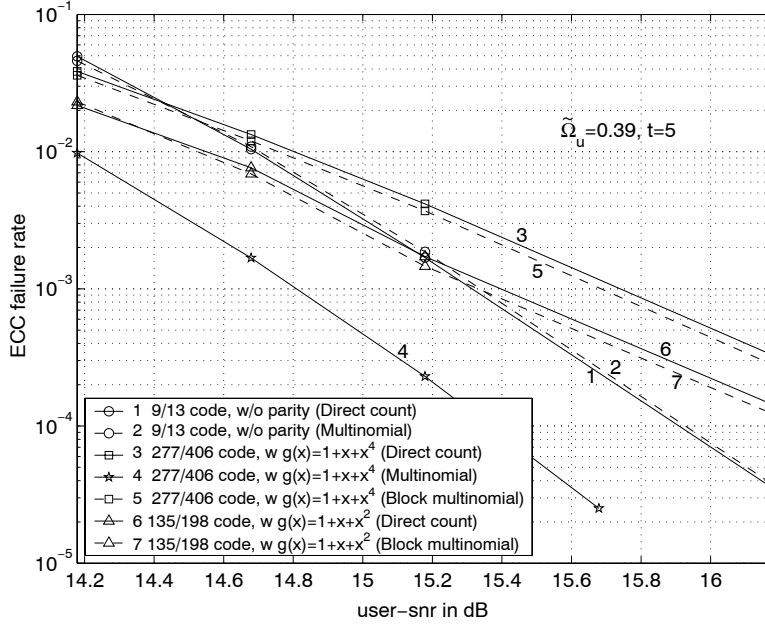


Figure 7.4: Comparison of failure rate evaluation methods for non-interleaved ECC, with RS [248, 238] code and $\tilde{\Omega}_u = 0.39$.

obtained from the direct count (Curve 3). It has been found that the block multinomial method (with $s = 17$ and $N_{pc} \approx 14$) (Curve 7) provides accurate estimation of EFRs for the 2-bit PC code (Curve 6) as well.

In addition, at SNRs that are sufficiently low (*e.g.* lower than 14 dB) so that the corresponding EFRs are larger than 10^{-4} and the direct count approach is feasible, we also verify the multinomial methods for the RS [248, 216] code with $t = 16$. In particular, we use the rate 9/13 code without parity and the 4-bit constrained PC code to verify the multinomial and block multinomial method, respectively. As shown in Figure 7.5, the multinomial method (Curve 2) and block multinomial method (Curve 4) provide accurate estimates on the EFRs for the rate 9/13 code (Curve 1) and the 4-bit constrained PC code (Curve 3), respectively.

Therefore, in the following failure rate evaluations of non-interleaved RS-ECC, we use the multinomial method for the case without PC, and use the block multinomial method for cases with PC.

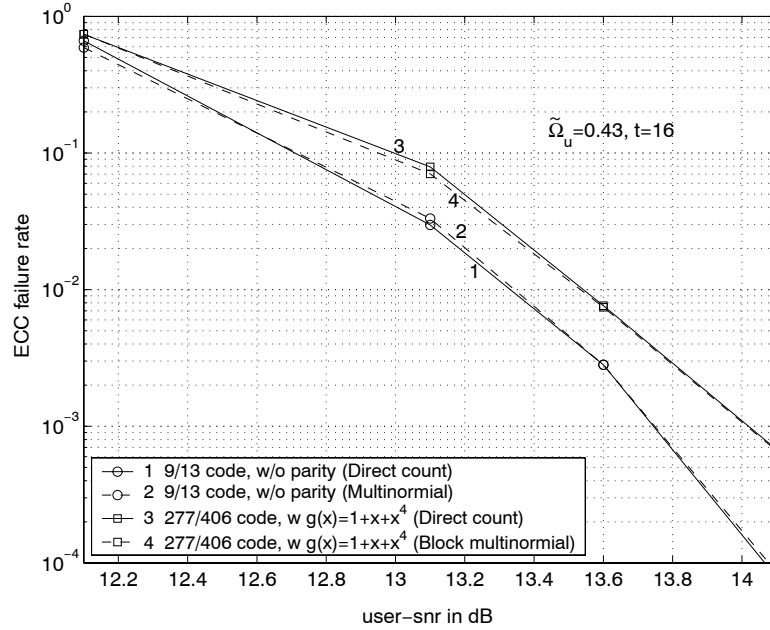


Figure 7.5: Comparison of failure rate evaluation methods for non-interleaved ECC, with RS [248, 216] code and $\tilde{\Omega}_u = 0.43$.

7.3.4 Generalized Multinomial/Block Multinomial Methods

Interleaving is a useful tool to enhance the capability of ECC for correcting burst errors [57, 101], which have been found to be dominant in high-density data storage systems. A detailed introduction to interleaving is given in Appendix C. In this section, we generalize the multinomial model and block multinomial model for the case of interleaved ECC. The failure rate of interleaved ECC corresponds to the probability that at least one of the ECC codewords in the interleaved codewords array has more than t byte errors. Let N_{s_i} denote the number of byte errors in the i^{th} interleave. In the presence of interleaving to a finite degree of α , the EFR, P_{If} , can be computed as

$$\begin{aligned}
 P_{If} &= \Pr\{(N_{s_1} > t) \cup (N_{s_2} > t) \cup \cdots (N_{s_i} > t) \cdots \\
 &\quad \cup (N_{s_\alpha} > t)\} \\
 &\leq \sum_{i=1}^{\alpha} P_f(i),
 \end{aligned} \tag{7.6}$$

with $P_f(i) = \Pr\{N_{s_i} > t\}$, for $1 \leq i \leq \alpha$. Note that in (7.6), $\sum_{i=1}^{\alpha} P_f(i)$ is an upper bound of P_{If} , with the probabilities that more than t byte errors occur simultaneously over several interleaves being ignored. With higher SNR and larger t , these probabilities will be smaller and the bound will be tighter, and vice versa [27]. As will be shown in the next section, for the systems studied in this thesis, we find that these probabilities are negligible and the bound given by (7.6) is tight.

For the case without PC, we use the multinomial method for analyzing non-interleaved RS-ECC. In the presence of interleaving, this method is used to obtain the failure probability $P_f(i)$ for the RS codeword in each interleave. Note that for the interleaved case, in the polynomial $p(D) = p_1D + p_2D^2 + \dots + p_kD^k + \dots$ used for obtaining $P_f(i)$, p_k corresponds to the probability of having k consecutive byte errors in each interleave.

To generalize the block multinomial model to the case of interleaved RS-ECC with PC, we need to estimate q_k of $q(D) = q_1D + q_2D^2 + \dots + q_kD^k + \dots$ to obtain $P_f(i)$ for the RS codeword in each interleave. However, for the case with interleaving, q_k should be the probability of receiving k byte errors in a PC codeword within each interleave, the length of which is given by $s_i = \lfloor \frac{s}{\alpha} \rfloor$. Consequently, $N_{pc} = \lfloor \frac{n}{s_i} \rfloor$ is used in (7.5) to compute $P_f(i)$. That is, compared with the case without interleaving, the generalized block multinomial method differs in its way to estimate q_k and the value of N_{pc} in computing $P_f(i)$.

7.3.5 Accuracy Analysis for Interleaved ECC

To analyze the accuracy of the proposed generalized multinomial/block multinomial methods, we compare the failure rates obtained from the generalized methods with those obtained from the direct count approach. At reasonable SNRs, since the failure rates of the interleaved RS [248, 216] code are lower than 10^{-7} (see Figure 7.14), as in the case with non-interleaved ECC, we use the RS [248, 238] code to do the verification. Note that in the case with interleaving, the EFR bound given by (7.6) will be tighter for the $t = 16$ code than for the $t = 5$ code, since the probabilities that more than t byte errors occur simultaneously over several interleaves are smaller for the $t = 16$ code.

The comparison is done for the cases of without and with PC, and a wide range of interleaving degrees. The results are shown in Figure

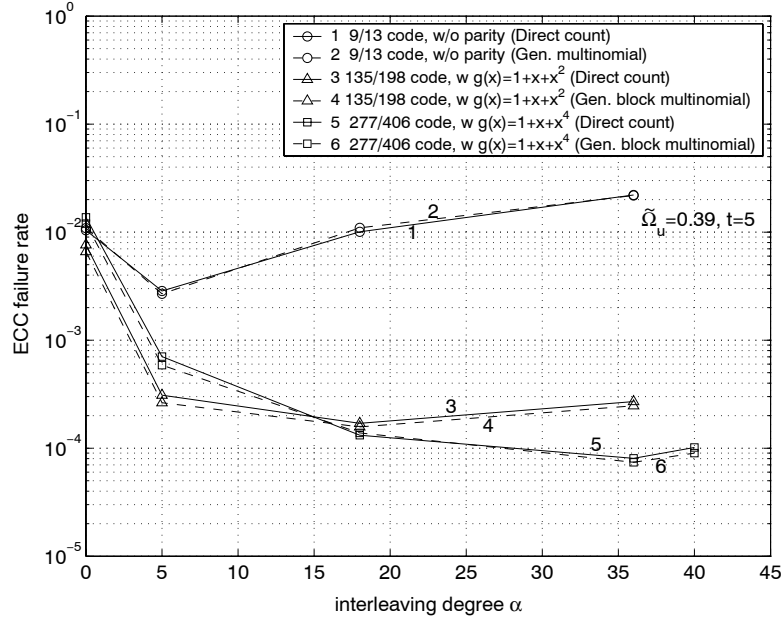


Figure 7.6: Comparison of failure rate evaluation methods for interleaved ECC, with interleaved RS [248, 238] code, $\tilde{\Omega}_u = 0.39$, and user SNR = 14.5 dB.

7.6, with a user SNR of 14.5 dB.

In Figure 7.6, Curves 1, 3 and 5 (solid) indicate the EFRs obtained from the direct count approach, associated with the rate 9/13 code, the 2-bit PC code, and the 4-bit PC code, respectively. Curve 2 shows the estimated failure rates, obtained from the generalized multinomial method for the rate 9/13 code. Curves 4 and 6 illustrate the failure rates of the 2-bit and 4-bit PC codes, estimated using the generalized block multinomial method. Observe that the generalized methods give very accurate estimation of the EFRs over various interleaving degrees, for both cases without and with PC. It has been found that the generalized methods provide similarly accurate estimation for other SNRs as well.

Furthermore, at sufficiently low SNRs and with a small interleaving degree (*e.g.* $\alpha = 5$), the failure rates of the RS [248, 216] code are larger than 10^{-4} . This enable us to evaluate the EFRs using the direct count approach, and use the results to verify the generalized methods.

We use the rate 9/13 code without parity and the 4-bit constrained PC code to verify the generalized multinomial and block multinomial method, respectively, for the RS [248, 216] code with an $\alpha = 5$ -degree interleaving. The comparison between EFRs obtained from the direct count approach and those obtained from the generalized methods are shown in Figure 7.7. Observe that with SNRs higher than 13 dB, the generalized multinomial and block multinomial methods provide very accurate estimates on the EFRs for the rate 9/13 code and the 4-bit constrained PC code, respectively. With SNRs lower than 13 dB, the EFRs obtained from the generalized methods have a bias of less than 0.1 dB, at $\text{EFR} = 10^{-1} \sim 10^0$. This is because when the SNR is very low, there exists simultaneously more than t byte errors over different interleaves. However, as SNR increases, the occurrence of such byte errors decreases significantly, and becomes negligible for SNRs higher than 13 dB.

Therefore, in the following sections, we use the generalized multinomial and block multinomial methods to estimate the failure rates of the interleaved RS [248, 216] code, for the cases without and with PC, respectively.

7.4 ECC Failure Rate of PC Coded Systems

7.4.1 Failure Rate With Non-Interleaved ECC

In Figure 7.8, we present the failure rates of the RS [248, 216] code obtained using the multinomial methods, for the cases without and with PC. Observe that the performance of the 2-bit PC code (Curve 2) lags behind the rate 9/13 code (Curve 1) by around 0.8 dB at $\text{EFR} = 10^{-14}$, and the 4-bit PC code (Curve 3) is about 0.3 dB behind the 2-bit PC code.

Recall from Figure 7.4 that, with the RS [248, 238] code, the 2-bit PC code outperforms the rate 9/13 code only at low SNRs. At high SNRs, both the 2-bit and 4-bit PC codes are inferior to the rate 9/13 code. Furthermore, the 2-bit PC code provides better performance than the 4-bit PC code.

The differences observed in the failure rate performances of the three cases are due to the difference in their byte error statistics at the input of the RS decoder. Figure 7.9 illustrates the probabilities of

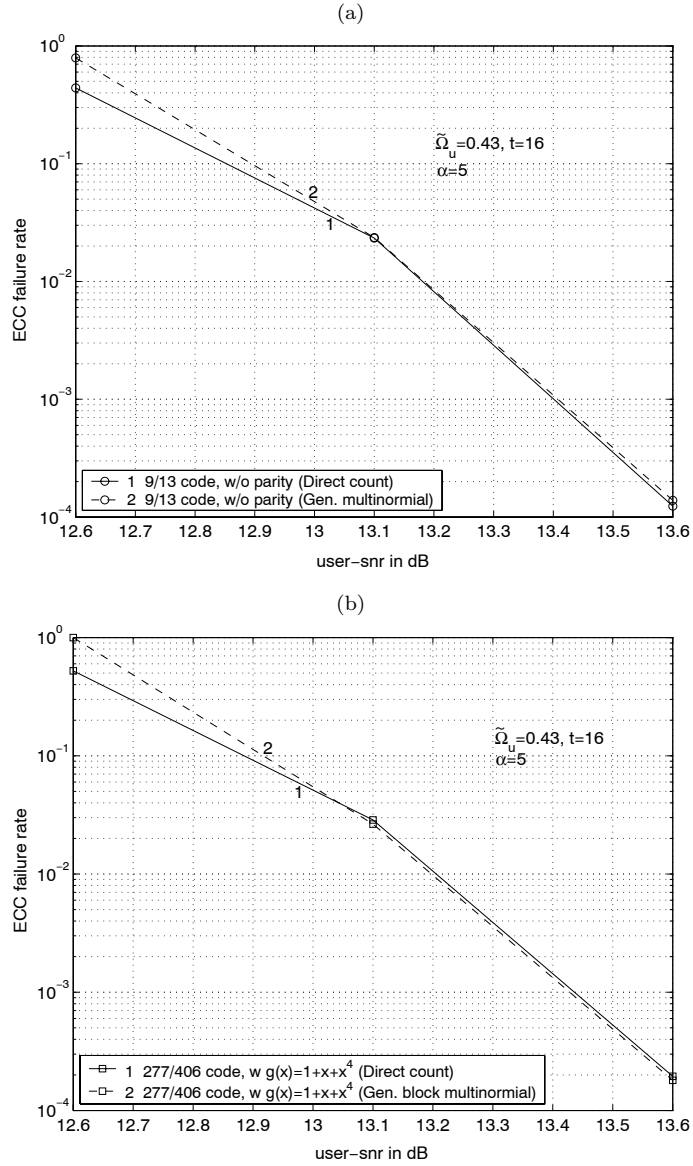


Figure 7.7: Comparison of failure rate evaluation methods for interleaved ECC, with RS $[248, 216]$ code, $\alpha = 5$, and $\tilde{\Omega}_u = 0.43$. (a) rate 9/13 code; (b) rate 277/406 code.

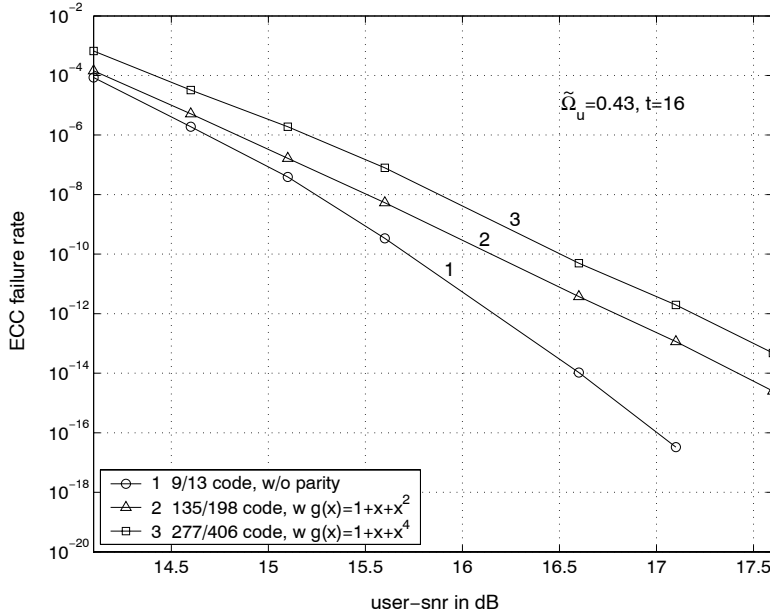


Figure 7.8: Failure rates of non-interleaved RS-ECC, with various constrained PC codes and $\tilde{\Omega}_u = 0.43$.

byte errors q_k in (7.4) for different k , for a user SNR of 15 dB. Observe that compared to the case of without PC, the probabilities of short byte errors (*i.e.* 1-3 byte errors) decrease significantly for cases with PC. This is another indication that the ByER for the 2-bit and 4-bit PC codes is smaller than that of the rate 9/13 code for a given SNR (see Figure 7.2). However, the probabilities of long byte errors (*i.e.* larger than 3 byte errors) are higher for cases with PC. These long byte errors are mainly caused by the mis-corrections of the post-processors. While short byte errors can be easily corrected by the RS codes, the existence of long byte errors severely degrades the EFR. Furthermore, the larger the error correction capability t and higher the SNR, the more severe will be the degradation in EFR due to long byte errors, and vice versa.

In addition, as illustrated by Figure 7.9, compared to the 4-bit PC code, the maximum length as well as the probabilities of long byte errors are smaller for the 2-bit PC code. Therefore, the 2-bit PC code provides better failure rate performance than the 4-bit code.

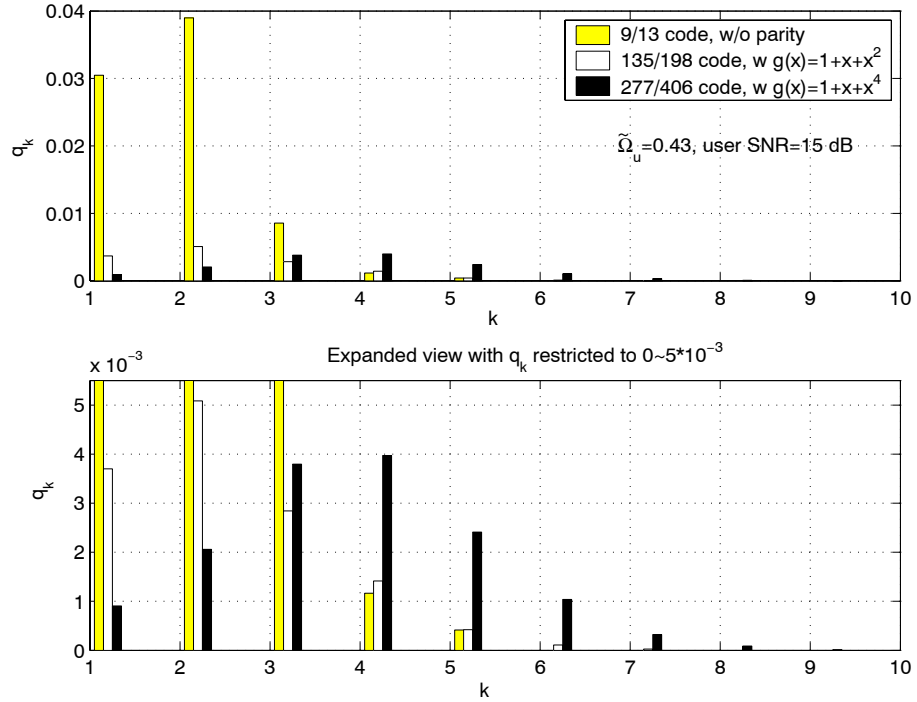


Figure 7.9: Probabilities of byte errors at the input of RS-ECC (non-interleaved) decoder, with various constrained PC codes, $\tilde{\Omega}_u = 0.43$, and user SNR=15 dB.

7.4.2 Failure Rate With Interleaved ECC

As shown in the previous section, in PC coding and post-processing based systems, the existence of long byte errors significantly degrades the EFR. Interleaving is an efficient method to distribute long byte errors into multiple ECC codewords and enhance the error correction capability of the ECC. In this section, we investigate the application of interleaving to PC coded systems to reduce the loss in error correction capability of ECC due to long byte errors.

Choice of Interleaving Degree α

To find the optimum interleaving degree α , we show in Figure 7.10 the failure rates of interleaved RS [248, 216] code, as a function of α . The

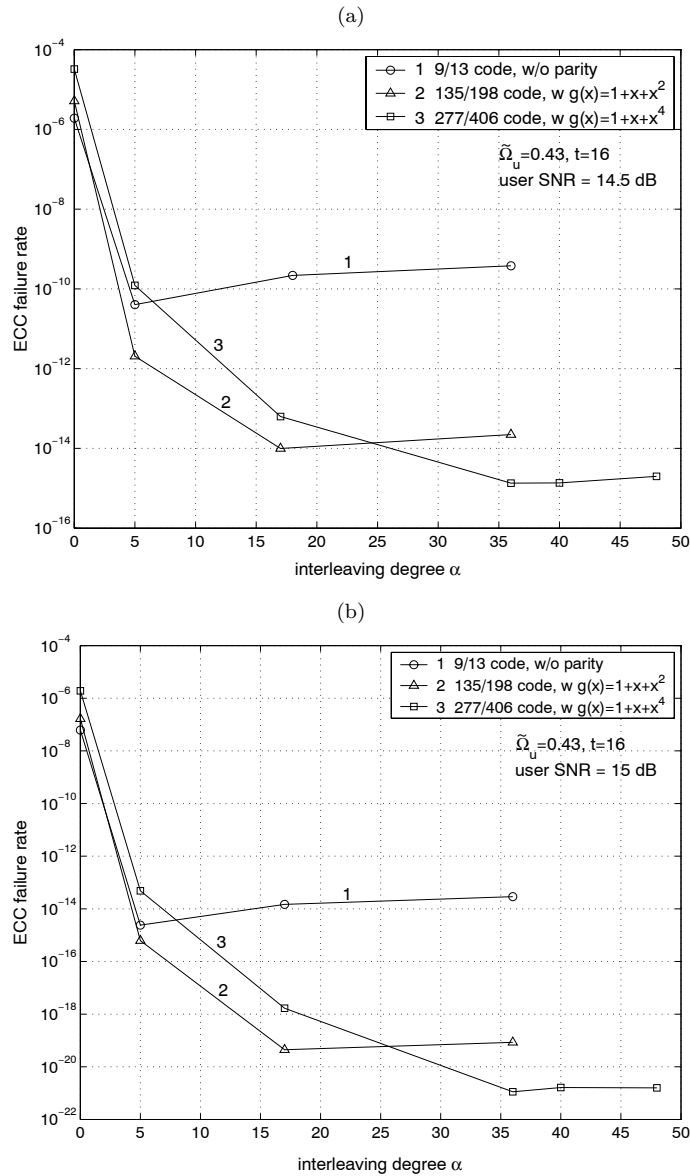


Figure 7.10: Effect of interleaving on ECC failure rate. (a) user SNR = 14.5 dB; (b) user SNR = 15 dB.

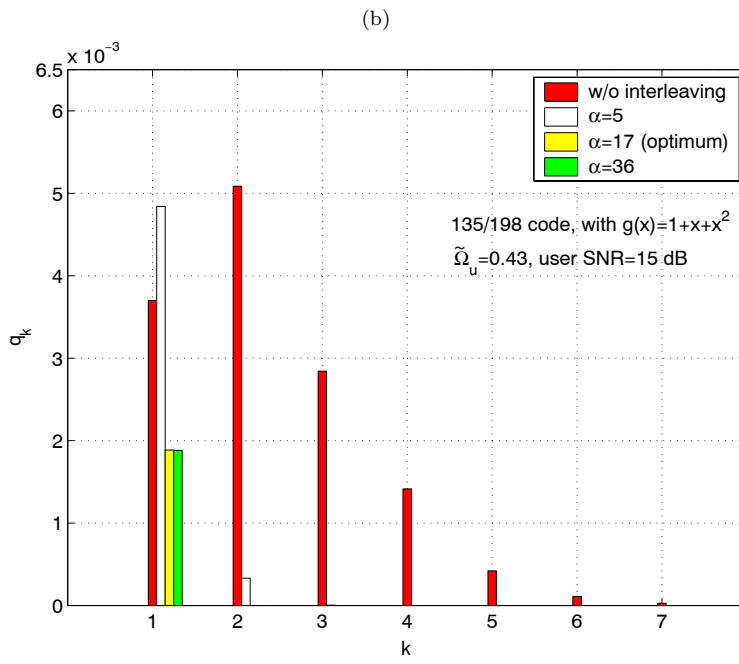
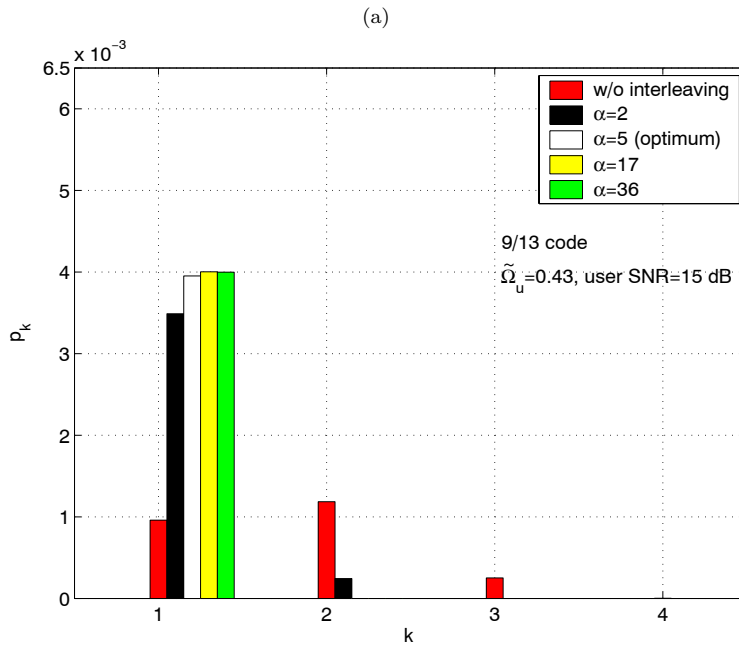
EFRs are obtained using the generalized multinomial method for the case of without PC, and by the generalized block multinomial method for the case of with PC. It can be seen that the optimum interleaving degrees are 5, 17 and 36, for the rate 9/13 code, the 2-bit PC code, and the 4-bit PC code, respectively. Use of interleaving beyond these degrees results in performance loss. Similar observations have been obtained with the interleaved RS [248, 238] code as well (see Figure 7.6).

The above results can be explained by looking at the corresponding byte error statistics within each interleave of the RS-ECC. Figures 7.11(a), (b) and (c) illustrate the probabilities of byte errors within each interleave, for the rate 9/13 code, the 2-bit and 4-bit PC codes, respectively, for a user SNR of 15 dB. For all the three codes, we observe the following. Firstly, interleaving causes the probability of the single byte errors to increase significantly, and the probabilities of more than one byte errors to decrease significantly. That is, interleaving effectively breaks long byte errors into short byte errors within each interleaved codeword. Secondly, for a sufficiently large α , all the long byte errors are converted into single byte errors. Such a value of α is close to optimum. As shown in Figure 7.11, applying interleaving beyond such an α will not make the probability of single byte errors per interleave decrease. On the other hand, this will only enlarge the size of the interleaved codewords array, and hence increase the overall failure rate of the interleaved code, according to (7.6).

Our study shows that for the case without PC, the optimum choice of α is associated with the maximum length of byte errors within each RS codeword. For the case with PC and post-processing, the optimum α can be approximated as the number of ECC-bytes per PC codeword, which is the maximum possible number of byte errors within each PC codeword.

ECC Failure Rate Evaluation

In this part, we first evaluate the failure rates of RS [248, 216] code, using a fixed interleaving degree, for the cases of without and with PC. Figures 7.12(a), (b) and (c) illustrate the failure rates for $\alpha = 5$, $\alpha = 17$ and $\alpha = 36$, respectively. The results show the following. Firstly, with $\alpha = 5$, the 2-bit PC code slightly outperforms the rate 9/13 code, while the 4-bit PC code lags behind the rate 9/13 code by



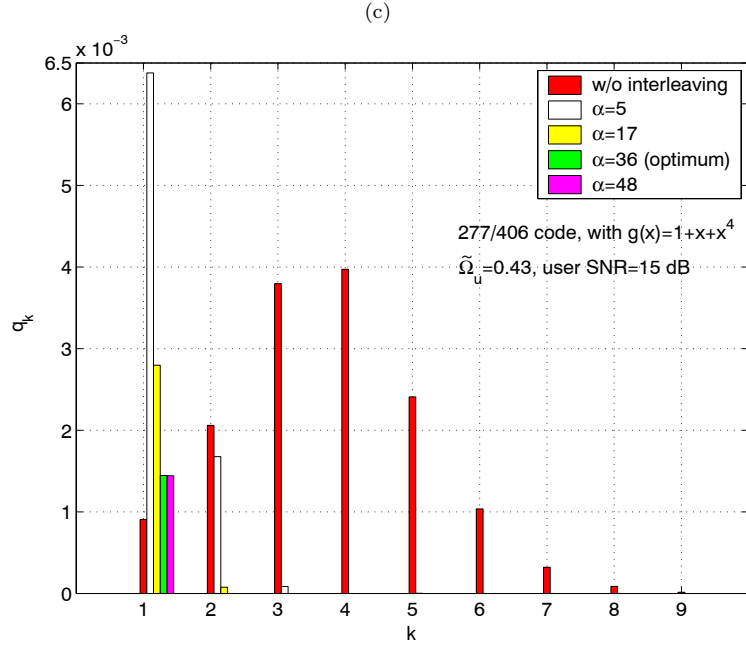
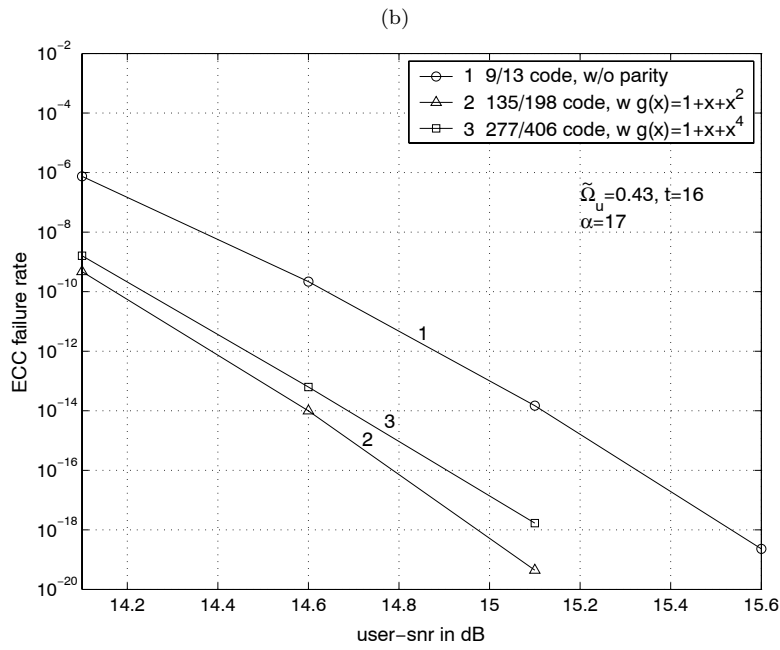
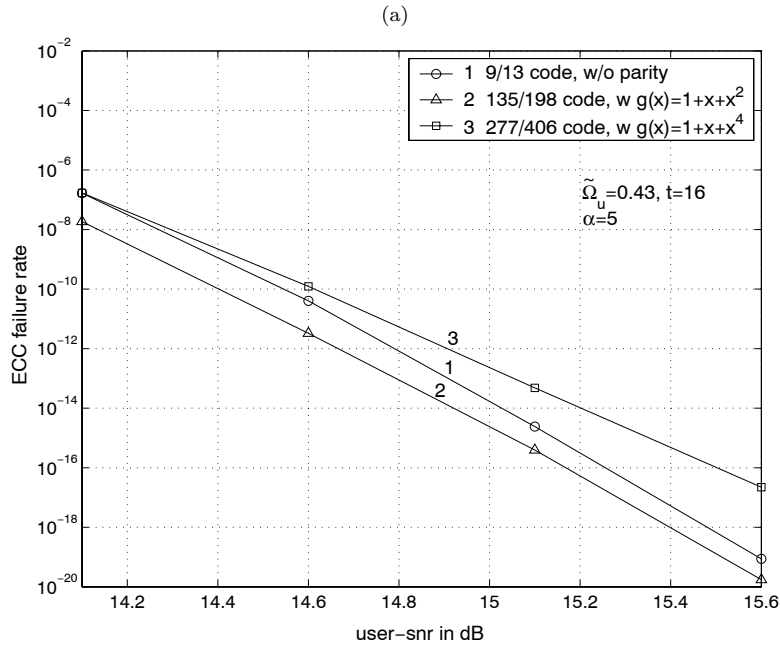


Figure 7.11: Probabilities of byte errors within each interleave, with $\tilde{\Omega}_u = 0.43$ and user SNR=15 dB. (a) rate 9/13 code; (b) rate 135/198 code; (c) rate 277/406 code.

about 0.2 dB at EFR = 10^{-16} . Secondly, with $\alpha = 17$, both PC codes outperform the rate 9/13 code. In particular, the 2-bit PC code gains 0.5 dB, and the 4-bit PC code gains 0.4 dB. Thirdly, with $\alpha = 36$, the 4-bit PC code provides better performance than the 2-bit PC code and achieves a gain of around 0.7 dB over the rate 9/13 code. The final choice of α is a compromise between the gain in EFR and the additional buffer space required and delay introduced by interleaving.

Let us now compare the EFRs associated with different PC codes. Similar to what have been found in [27, 15], our study also shows that for the case without interleaving (see Figure 7.8) or with a relatively small α (e.g. $\alpha = 5, 17$, as shown in Figures 7.12(a) and (b)), the 2-bit PC code outperforms the 4-bit PC code. However, with a sufficiently large α (e.g. $\alpha = 36$, as shown in Figure 7.12(c)), the 4-bit PC code outperforms the 2-bit PC code. The above results can be explained from the byte error statistics at the input of the RS decoder. The



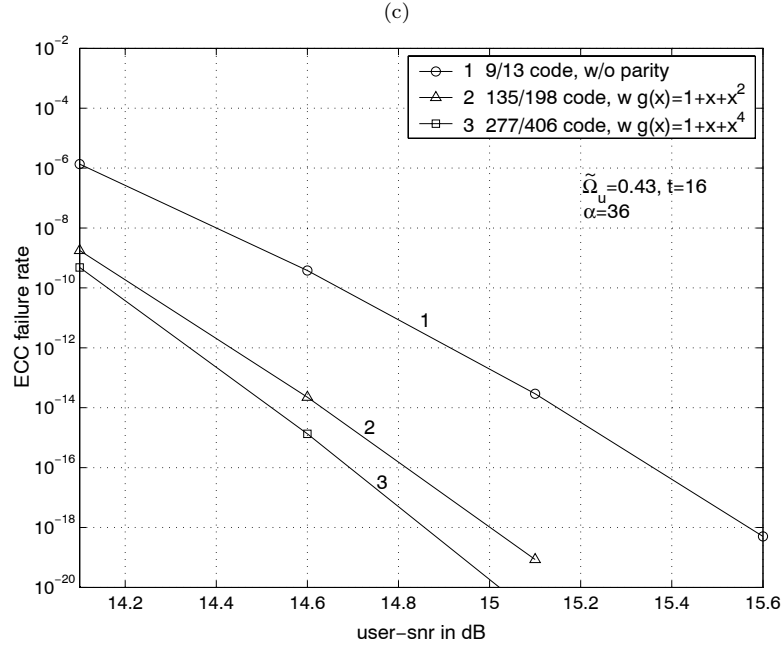
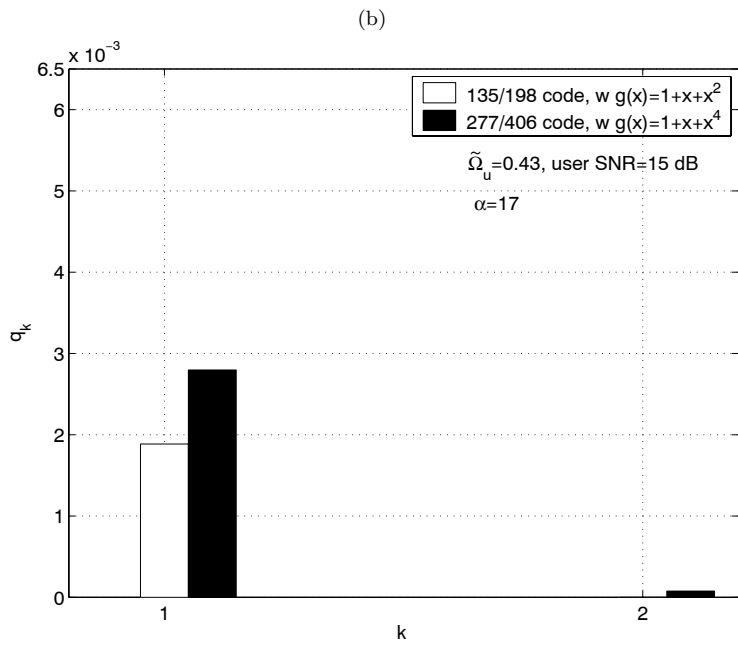
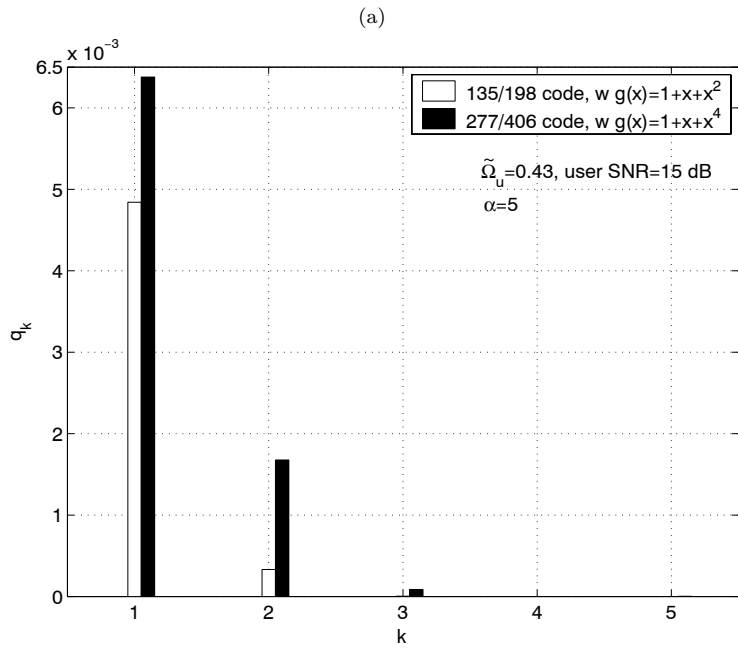


Figure 7.12: Failure rates of RS-ECC with different interleaving degrees, for various constrained PC codes and $\tilde{\Omega}_u = 0.43$. (a) $\alpha = 5$; (b) $\alpha = 17$; (c) $\alpha = 36$.

byte error statistics of the two codes, in the case without interleaving, are available in Figure 7.9. We further show in Figures 7.13(a), (b) and (c), the byte error probabilities of the two PC codes, with $\alpha = 5$, $\alpha = 17$ and $\alpha = 36$, respectively. We observe from Figure 7.9 that in the case of no interleaving, the long byte errors introduced by the 4-bit PC code are much longer and the corresponding probabilities are much larger than that of the 2-bit PC code. That is, although the 4-bit PC code corrects more errors on the average than the 2-bit PC code (as shown by Figure 7.2), the post-processor of the 4-bit code produces more long error bursts. Therefore, the 2-bit PC code provides better performance than the 4-bit code in terms of EFR. With a relatively small degree of interleaving (*e.g.* $\alpha = 5, 17$), as shown in Figures 7.13(a) and (b), the maximum length of the byte errors as well as the byte error probabilities of the 4-bit code are still larger than those of the 2-bit code, and hence its performance still lags behind the 2-bit



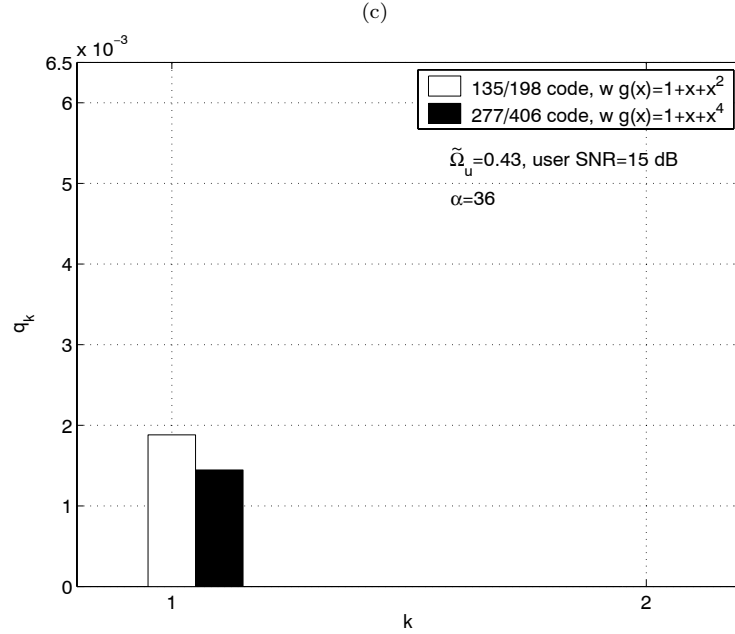


Figure 7.13: Probabilities of byte errors within each interleave, with $\tilde{\Omega}_u = 0.43$ and user SNR=15 dB. (a) $\alpha = 5$; (b) $\alpha = 17$; (c) $\alpha = 36$.

code. With a sufficiently large α (e.g. $\alpha = 36$), all the long byte errors associated with the 4-bit code are broken into single byte errors. The corresponding probability is smaller than that of the 2-bit code, since the 4-bit code corrects more errors on the average. As a result, the 4-bit PC code provides better performance than the 2-bit code.

Finally, in Figure 7.14, we show the EFRs obtained when optimum interleaving degrees are used. The interleaving degrees are taken as $\alpha = 5$, $\alpha = 17$ and $\alpha = 36$, respectively, for the rate 9/13 code (Curve 4), the 2-bit PC code (Curve 5) and 4-bit PC code (Curve 6). The failure rates of these codes for the case of without interleaving (Curves 1, 2, 3) are also included for reference purpose. The figure shows that with optimum degree of interleaving, a significant gain is obtained for both cases of without and with PC. Moreover, the interleaving gain for the case with PC is much larger than that without PC. In particular, the interleaving gain at EFR = 10^{-16} is 1.75 dB for the rate 9/13 code, and 3 dB and 3.3 dB, respectively, for the 2-bit and

4-bit PC codes. In the case of without interleaving, the performance of constrained PC codes lags behind the rate 9/13 code due to the long byte errors introduced by post-processing. Through interleaving, these long byte errors have been effectively broken into short byte errors over different interleaves. This enhances the error correction capability of the RS-ECC, and results in a larger gain for the constrained PC codes. Furthermore, the gain of the 4-bit PC code is larger than that of the 2-bit PC code, since it can correct more errors on the average. In the case of with interleaving, compared to the performance of the rate 9/13 code (with $\alpha = 5$), the 2-bit PC code (with $\alpha = 17$) achieves a gain of 0.5 dB. The 4-bit PC code (with $\alpha = 36$) outperforms the 2-bit PC code and achieves around 0.6 dB improvement over the rate 9/13 code.

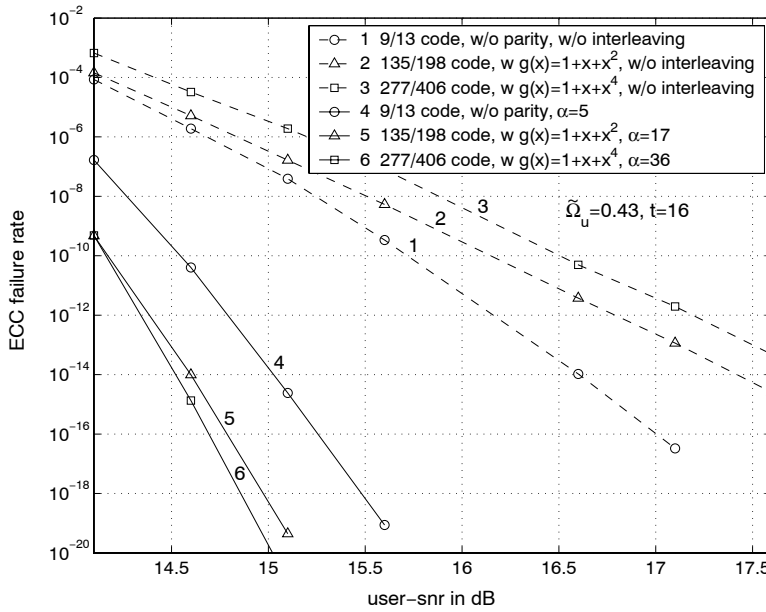


Figure 7.14: Failure rates of RS-ECC with optimum interleaving degrees, for various constrained PC codes and $\tilde{\Omega}_u = 0.43$.

7.5 Conclusion

In this chapter, we have investigated the performance of PC-code-based systems, with ECC taken into consideration. We have first evaluated the ByER at the input of the RS decoder, and then analyzed the EFR of the system. In particular, we have applied the multinomial and block multinomial methods to estimate the failure rates of the non-interleaved RS-ECC, for the cases of without and with PC, respectively. We further generalized the multinomial methods to the case of interleaved RS-ECC. The accuracy of the developed methods has been demonstrated by simulation results obtained from the direct count approach. Using the developed semi-analytical approaches, we have evaluated the EFRs of PC coded systems, for the cases of without and with interleaving. We have found that in the case of without interleaving, the performance for the constrained PC codes lags behind that for the constrained code without parity. This is mainly due to the long byte errors caused by the mis-corrections of the post-processor. By applying interleaving on the RS-ECC, these long byte errors can be effectively broken into short byte errors over different interleaves. This improves the error correction capability of the RS-ECC, and results in a larger gain for the constrained PC codes. We have found the optimum interleaving degrees for both cases of without and with PC. With the optimum interleaving degrees, the 4-bit PC code outperforms the 2-bit code, due to the reason that it can correct more errors on the average. At a high recording density, compared to the rate 9/13 code without parity, the 2-bit PC code achieves a gain of around 0.5 dB, and the 4-bit PC code gains 0.6 dB, at $\text{EFR} = 10^{-16}$.

Finally, we remark that by applying certain repeated minimum transition runlength (RMTR) constraint to the modulation code, we can prohibit the input data patterns that support some of the dominant error events (*e.g.* the $\pm\{2, 0, -2, 0, 2\}$ event) of the system. This may enable both the 2-bit and 4-bit PC codes to detect all the dominant error events. With appropriate interleaving degrees, these two codes may achieve similar EFR performances. The RMTR constraint can also eliminate some of the non-dominant error events of the system. As discussed in Section 3.5.3, this will provide further gain for the PC coded systems. However, compared to the rate 9/13 capacity-approaching $d = 1$ codes used in this thesis, an additional code rate

loss will be incurred. The final performance is a trade-off between the gain from the RMTR constraint and the code rate loss.

Chapter 8

Epilogue

8.1 Conclusions

In this thesis, we designed and analyzed parity-check (PC)-code-based recording systems to achieve higher recording capacity with low implementation complexity, for high-density blue laser disc systems. In particular, most of the key components of the PC-code-based optical recording system have been designed and optimized. The study has extended a conventional channel model to a generalized channel model; the $d = 1$ code with a code rate equal to that of standard codes, to capacity-approaching $d = 1$ codes; the constrained codes in conjunction with PC codes, which are modeled by assuming that the parity-check result for each channel data segment is known at the receiver, to capacity-approaching constrained PC codes; channel models with additive white Gaussian noise (AWGN) only, to channel models with both media noise and AWGN and reception techniques that tailored to this combination; and investigations on bit error rate (BER), to byte error rate (ByER), and eventually to error correction code (ECC) failure rate (EFR). To our knowledge, the work reported in this thesis is the first thorough and comprehensive study on PC coding and post-processing for optical recording systems.

The main contributions consist of six parts.

- Development of a generalized Braat-Hopkins model for optical recording (Chapter 2).
 - We proposed a generalized Braat-Hopkins model for optical recording channels. In this model, the channel is completely

defined by specifying the signal-to-noise ratio (SNR), optical cut-off frequency, and media noise with respect to user bit rate, and the code rate. The generalized model exposes the effect of code rate and recording density on the various parameters of the channel model, and therefore can be used to assess the receiver performance over various code rates and recording densities, for channels with AWGN and media noise.

- Investigation on various basic issues associated with PC-code-based optical recording systems (Chapter 3).
 - We analyzed BER and error event probabilities of Viterbi detector (VD) based $d = 1$ channels, and derived analytical bounds on bit error rates, achievable by various PC codes. An analytical approach has been proposed to compute the probability of $d = 1$ sequences that support given error events at the VD output.
 - We designed various PC coding schemes using existing as well as new codes, and identified the optimum codeword length of PC codes.
 - We developed a multiple-error-event correction matched-filtering type post-processor. The boundary error events were analyzed and a simple remedy scheme was proposed for post-processing.
 - Simulation results showed that the PC code that can detect all the dominant error events of the system, *i.e.* the 4-bit code with $g(x) = 1 + x + x^4$, achieves the best performance. The corresponding performances are very close to the performance bounds, at both nominal and high densities. There is no gain with further increase in the number of parity bits.
- Design of capacity-approaching $d = 1$ codes with minimum number of encoder states (Chapter 4).
 - We derived analytically the relationship between the number of encoder states and the probable size of certain capacity-approaching $d = 1$ codes.

- We found that choosing the number of encoder states to be specific Fibonacci numbers maximizes the rate of the designed code with the minimum number of states, for any desired codeword length.
- Our analysis provides direct guidelines for the design of high efficiency codes with $d = 1$ constraint, as well as other constraints that may be desired for future systems.
- Design of capacity-approaching constrained PC codes (Chapter 5).
 - We proposed a general and systematic code design technique for constructing capacity-approaching constrained PC codes, which can detect any type of dominant error events or error event combinations in optical recording systems, with the minimum code rate loss.
 - We proposed approaches to design the code in the non-return-to-zero-inverse (NRZI) format and non-return-to-zero (NRZ) format. We found that designing the codes in NRZ format may reduce the number of parity bits required for error detection and may simplify post-processing for error correction.
 - The proposed code design technique is general, and can encompass other recording channels (*e.g.* the magnetic recording channels).
 - Simulation results showed that among several newly designed codes, the $d = 1$ constrained 4-bit PC code can detect all the dominant error events. Compared to the rate $2/3$ code without parity, it achieves a performance gain of 2 dB at nominal density, and 1.5 dB at high density, at $\text{BER} = 10^{-5}$.
- Development of novel detection, PC coding and post-processing schemes to combat media noise (Chapter 6).
 - We proposed two novel modifications to the channel detector to combat media noise, namely, the monic constrained minimum mean square error (MMSE) based equalization with media noise consideration, and a modified VD with

data-dependent noise variance. The proposed modifications provide significant performance gain over the conventional detection approaches that are designed for AWGN. Moreover, they are simple and can be easily incorporated into existing partial response maximum-likelihood (PRML) systems.

- We proposed a data-dependent post-processing scheme in conjunction with PC codes for media noise channels. It outperforms the conventional matched-filtering type post-processors by accounting for the data-dependence of the media noise variance.
 - With the increase in media noise power, the proposed detector modifications, PC coding and post-processing scheme help to significantly delay the error floor caused by media noise, and provide much larger performance gains. Compared to the system designed without considering media noise and without PC codes, the overall performance gain of the developed scheme can be more than 11 dB at high media noise levels.
- Performance analyzes of ECC failure rate (EFR) of the developed PC-code-based systems (Chapter 7).
 - We developed semi-analytical approaches for estimating the EFR, and evaluated the EFRs for PC-code-based systems, for the cases of without and with interleaving.
 - We found that in the case of without interleaving, the performance for the constrained PC codes lags behind that for the constrained code without parity. This is mainly due to the long byte errors caused by the mis-corrections of the post-processor. By applying interleaving on the Reed-Solomon (RS)-ECC, these long byte errors can be effectively broken into short byte errors over different interleaves. This improves the error correction capability of the RS-ECC, and results in a larger gain for the constrained PC codes. We found the optimum interleaving degrees for both cases of without and with parity.
 - Our analysis showed that with the optimum interleaving

degrees, the 4-bit PC code outperforms the 2-bit code, due to the reason that it can correct more errors on the average.

- With the optimum interleaving degrees, compared to the rate 9/13 code without parity, the 2-bit PC code achieves a gain of around 0.5 dB, and the 4-bit PC code gains 0.6 dB, at high recording density and $\text{EFR} = 10^{-16}$.

In summary, based on the work carried out by this thesis, we conclude the following.

The developed PC coding and post-processing techniques can provide significant performance gain for optical recording systems in terms of various performance measures (*viz.* BER, ByER, and EFR), with affordable implementation complexity. Therefore, the PC-code-based reception technique shows high potential for high-density optical recording systems.

During the design of PC codes, adding more parity bits can increase the error detection capability of the codes. However, the associated probabilities of long byte errors caused by the mis-corrections of the post-processors will also increase. This will severely degrade the failure rate of RS-ECC. Therefore, capacity-approaching constrained PC codes that can detect all the dominant error events of the system with the minimum number of parity bits should be designed. The choice of the codeword length depends on the gain obtained from error correction against the performance loss due to decrease of code rate.

For a given PC code in conjunction with post-processing, the performance gain obtained from error correction is determined by the error event distribution. The larger the probability of the dominant error events that can be corrected by the PC code, *i.e.* the smaller the probability of the non-dominant error events, the larger will be the performance gain, and vice versa.

By applying novel equalization, detection, and post-processing schemes developed by this thesis, which account for the correlated, data-dependent, and non-stationary nature of media noise, significant performance gain can be obtained over the conventional approaches that are developed without considering the media noise. Furthermore, the proposed schemes are simple and can be easily incorporated into existing systems.

Interleaving can effectively break the long byte errors caused by the mis-corrections of the post-processors into short byte errors over

different interleaves, and improve the EFRs of PC-code-based systems. With the optimum degree of interleaving, the performance gain of PC-code-based systems can be maximized.

8.2 Suggestions for Future Work

We finally describe some of the possible issues that could be addressed in the future.

Our study shows that at high recording densities, more non-dominant error events with small probabilities occur, causing difficulties for both error detection and error correction. Adding specific constraints to the modulation code, such as the repeated minimum transition runlength (RMTR) constraint, can prohibit the data patterns that support these events, and will help to improve the performance of the PC-code-based systems. In addition, as described in Section 7.5, applying appropriate RMTR constraint to eliminate some of the dominant error events of the system may enable the 2-bit PC code to achieve a similar performance as that of the 4-bit PC code. However, compared to the rate $9/13$ code used in this thesis, whose rate approaches the capacity of $d = 1$ codes, an additional code rate loss has to be accepted. The final error rate performance is a compromise between the gain from the RMTR constraint and the corresponding code rate loss.

One might also consider generalizing the novel code design technique presented in Chapter 5 for combining constrained codes with RS codes. This could effectively reduce the error propagation introduced by constrained decoder. Another approach to reduce this error propagation might be the ‘reverse concatenation’ scheme, which reverses the conventional hierarchy of ECC and constrained code. This may enable us to design and apply highly efficient constrained codes with arbitrary long codewords.

Even though the modulation codes used in hard disk drives (HDDs) had $d = 2$ or $d = 1$ constraints in the early days, the use of $d = 0$ codes became universal in HDDs by the second half of the 1990s. This is because codes with strong constraints such as $d > 0$ result in large code rate loss. For this reason, a similar shift towards high rate codes in optical recording is quite natural to expect, and $d = 0$ codes may be considered as a possibility. However, the weaker the constraint becomes, the more powerful the receiver should be, to ensure reliable

data recovery. This is because some of the channel distortions which do not appear serious for $d > 0$ systems tend to become serious for $d = 0$ systems. Therefore, one might consider the development of advanced coding and signal processing techniques (*e.g.* generalizing the PC-code-based detection approaches developed in this thesis) for $d = 0$ coded optical recoding channels.

Appendix A

Simplification of the MAP Post-Processor

This appendix simplifies the maximum *a posteriori* (MAP) post-processor given by (3.18) into a matched-filtering type post-processor, which results in significant reduction in computational complexity.

For the channel described in Section 3.1.1, the Viterbi input can be expressed as $q_k = \sum_{l=0}^{L_g-1} g_l a_{k-l} + v_k$, where v_k is the sum of residual intersymbol interference (ISI) and channel noise, with zero mean and variance σ_v^2 . A good design of the equalizer and partial response (PR) target can make v_k almost white. In the post-processor, the reconstructed Viterbi input samples based on an assumed error event \mathbf{e}_i^j are given by

$$\tilde{q}_{i,k}^j = \sum_{l=0}^{L_g-1} g_l \tilde{a}_{i,k-l}^j = \sum_{l=0}^{L_g-1} g_l (\hat{a}_{k-l} + e_{i,k-l}^j). \quad (\text{A.1})$$

Assuming v_k is white and Gaussian, we may rewrite (3.18) as

$$\begin{aligned} \hat{\mathbf{e}} &= \arg \max_{\mathbf{e}_i^j} \frac{1}{(\sqrt{2\pi}\sigma_v)^{N_t}} \exp\left(-\frac{\sum_{k=1}^{N_t} (q_k - \tilde{q}_{i,k}^j)^2}{2\sigma_v^2}\right) P_j \\ &= \arg \min_{\mathbf{e}_i^j} \left\{ \sum_{k=1}^{N_t} (q_k - \tilde{q}_{i,k}^j)^2 - 2\sigma_v^2 \ln(P_j) \right\}, \end{aligned} \quad (\text{A.2})$$

with $N_t = N + L_e^j + L_g - 2$.

The computational complexity of (A.2) is still high, since for each assumed error event(s) of type j , and with every possible starting position i , the squared Euclidean distance $\sum_{k=1}^{N_t} (q_k - \tilde{q}_{i,k}^j)^2$ needs to be computed to locate the error(s). Therefore, let us simplify this term $\sum_{k=1}^{N_t} (q_k - \tilde{q}_{i,k}^j)^2$. It can be rewritten as

$$\begin{aligned}
\sum_{k=1}^{N_t} (q_k - \tilde{q}_{i,k}^j)^2 &= \sum_{k=1}^{N_t} \left(\sum_{l=0}^{L_g-1} g_l (e_{k-l} - e_{i,k-l}^j) + v_k \right)^2 \\
&= \sum_{k=1}^{N_t} \left(\sum_{l=0}^{L_g-1} g_l e_{k-l} \right)^2 + 2 \sum_{k=1}^{N_t} \left(\sum_{l=0}^{L_g-1} g_l e_{k-l} \right) v_k \\
&\quad + \sum_{k=1}^{N_t} v_k^2 - 2 \sum_{k=1}^{N_t} \left(\sum_{l=0}^{L_g-1} g_l e_{k-l} \sum_{m=0}^{L_g-1} g_m e_{i,k-m}^j \right) \\
&\quad - 2 \sum_{k=1}^{N_t} \sum_{l=0}^{L_g-1} g_l e_{i,k-l}^j v_k + \sum_{k=1}^{N_t} \left(\sum_{l=0}^{L_g-1} g_l e_{i,k-l}^j \right)^2. \quad (\text{A.3})
\end{aligned}$$

We note that the terms $\sum_{k=1}^{N_t} \left(\sum_{l=0}^{L_g-1} g_l e_{k-l} \right)^2$, $2 \sum_{k=1}^{N_t} \left(\sum_{l=0}^{L_g-1} g_l e_{k-l} \right) v_k$ and $\sum_{k=1}^{N_t} v_k^2$ in (A.3) are independent of \mathbf{e}_i^j . Therefore, these terms can be dropped while substituting (A.3) into (A.2). The resulting simplified version of (A.2) is (3.19).

Appendix B

Criteria for Designing PRC Codes with $d = 2$ Constraint

Proceeding in similar lines as in the case with $d = 1$ constraint (see Section 5.3.2), we derive criteria for designing parity-related constrained (PRC) codes with $d = 2$ constraint as

$$\begin{aligned}
 r|X_{0000}| + (r_1 + r_2)|X_{0010}| + r_1|X_{0001}| &\geq r_1 2^{m_2+p}, \\
 r|X_{0000}| + (r_1 + r_2)|X_{0010}| + r_1|X_{0001}| \\
 + r|X_{0100}| + (r_1 + r_2)|X_{0110}| + r_1|X_{0101}| &\geq (r_1 + r_2) 2^{m_2+p}, \\
 r|X_{0000}| + (r_1 + r_2)|X_{0010}| + r_1|X_{0001}| \\
 + r|X_{0100}| + (r_1 + r_2)|X_{0110}| + r_1|X_{0101}| \\
 + r|X_{1000}| + (r_1 + r_2)|X_{1010}| + r_1|X_{1001}| &\geq r 2^{m_2+p},
 \end{aligned}$$

where X_{abcd} denotes the set of codewords that start with 'ab' and end with 'cd', where $a, b, c, d \in \{0, 1\}$. For $d = 2$ codes, the encoder has r states, which are further classified into three sets of states. The first set has r_1 states and it includes codewords that start with '00'. The second set has r_2 states and it includes codewords that start with either '01' or '00'. The third set has $r_3 = r - r_1 - r_2$ states and it includes codewords that start with '10', '01' or '00'.

The criteria that guide the design of each set of codewords that

has the same parity-check bits are expressed as

$$\begin{aligned}
& r|\tilde{X}_{0000}| + (r_1 + r_2)|\tilde{X}_{0010}| + r_1|\tilde{X}_{0001}| \geq r_1 2^{m_2}, \\
& r|\tilde{X}_{0000}| + (r_1 + r_2)|\tilde{X}_{0010}| + r_1|\tilde{X}_{0001}| \\
& \quad + r|\tilde{X}_{0100}| + (r_1 + r_2)|\tilde{X}_{0110}| + r_1|\tilde{X}_{0101}| \geq (r_1 + r_2) 2^{m_2}, \\
& r|\tilde{X}_{0000}| + (r_1 + r_2)|\tilde{X}_{0010}| + r_1|\tilde{X}_{0001}| \\
& \quad + r|\tilde{X}_{0100}| + (r_1 + r_2)|\tilde{X}_{0110}| + r_1|\tilde{X}_{0101}| \\
& \quad + r|\tilde{X}_{1000}| + (r_1 + r_2)|\tilde{X}_{1010}| + r_1|\tilde{X}_{1001}| \geq r 2^{m_2},
\end{aligned}$$

where \tilde{X}_{abcd} denotes the set of codewords with the same parity bits. For each set of codewords with the same parity-check bits, a codeword that ends with '00' (*i.e.* codewords in \tilde{X}_{0000} , \tilde{X}_{1000} and \tilde{X}_{0100}) can be assigned up to r different following states, and therefore can be used to map to r different user data words. A codeword that ends with '10' (*i.e.* codewords in \tilde{X}_{0010} , \tilde{X}_{1010} and \tilde{X}_{0110}) can only be assigned up to r_1 following states in the first state set and r_2 states in the second state set, and therefore can be used to map to $r_1 + r_2$ different user data words. A codeword that ends with '01' (*i.e.* codewords in \tilde{X}_{0001} , \tilde{X}_{1001} and \tilde{X}_{0101}) can be only assigned up to r_1 different following states in the first state set, and therefore can be used to map to r_1 different user data words. In addition, different states cannot contain the same codeword.

Appendix C

Interleaving

An error burst begins and ends with an error, but may have correct symbols in the middle. These errors may affect many symbols within a codeword of an error correction code (ECC), but they occur very seldom. Since an error burst focuses several symbol errors within a small number of received codewords, while the other codewords may not be corrupted by errors at all, the error correction capability of ECC is wasted on the unaffected codewords. Interleaving is used to solve this problem.

The effect of interleaving is to spread an uncorrectable error burst into multiple ECC codewords, so that they appear to the decoder as single symbol errors or shorter correctable burst errors. Given a $[n, k]$ ECC, we can construct a $[\alpha n, \alpha k]$ interleaved code by arranging α codewords of the ECC into α rows of a rectangular array and transmitting them column by column. The parameter α is referred to as the interleaving degree. Such an interleaving scheme is referred to as block interleaving, which is one of the most commonly used interleaving schemes in practice. Figure C.1 shows a $(n \times \alpha)$ block interleaver and the corresponding deinterleaver. In this thesis, we focus on block interleaving. In particular, we explore the use of interleaving in parity-check (PC) coded systems to reduce the loss in error correction capability of the Reed-Solomon (RS)-ECC due to long symbol errors.

Interleaving does not decrease the total number of symbol errors within the interleaved codewords. Suppose the number of symbol errors within the original ECC codeword is b , then that within a α -degree interleaved codewords array is αb . However, through interleaving, we can change the error pattern within each interleaved codeword. For

example, a burst of length α will affect no more than one symbol in each row of the array, irrespective of where it starts. An appropriate choice of α will make the number of short symbol errors within each row increase significantly, while the number of long symbol errors decrease significantly. This enhances the overall error correction capability of ECC. The trade-off is that extra buffer space is needed for storing the interleaved codeword array. In addition, an additional delay is introduced during data transmission, since through interleaving, any two adjacent symbols of the input data must be separated by

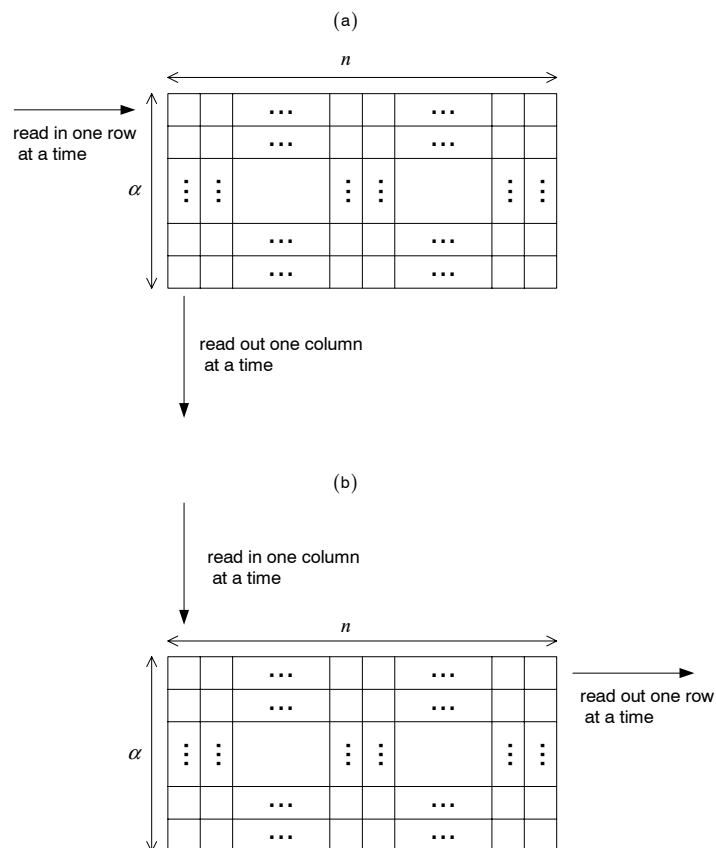


Figure C.1: A $(n \times \alpha)$ block interleaver and deinterleaver. (a) interleaver; (b) deinterleaver.

$\alpha - 1$ symbols at the output.

Samenvatting

De toenemende vraag naar optische data-opslagsystemen met hoge informatiedichtheid en hoge schrijf- en leessnelheid maakt de ontwikkeling van geavanceerde coderings- en signaalbewerkingstechnieken voor snelle en betrouwbare data-opslag steeds belangrijker. In de afgelopen jaren is de 'parity-check' (PC)-code gebaseerde data-ontvangst techniek uitvoerig bestudeerd voor magnetische data-opslagsystemen. Deze techniek lijkt ook zeer veelbelovend voor hoge-dichtheids optische data-opslag. De PC code is een foutcorrigerende code (Error Correction Code, ECC), waarmee dominante korte foutpatronen van het systeem gedetecteerd kunnen worden, gebruik makend van slechts enkele pariteits bits. Toepassing van een PC code als 'inner ECC code' ontlast de gebruikelijke 'outer ECC code' in belangrijke mate, waardoor op betrekkelijk eenvoudige wijze een belangrijke prestatie-verbetering van het opslagsysteem mogelijk wordt. Dit proefschrift behandelt het ontwerp en de analyse van PC-gebaseerde optische data-opslagsystemen, gericht op het bereiken van hogere informatiedichtheden bij een lage complexiteit, voor systemen gebruik makend van een blauwe laser. De belangrijkste onderdelen van een dergelijk systeem worden ontworpen en geoptimaliseerd voor verschillende informatiedichtheden, en voor verschillende hoeveelheden media-ruis en additieve ruis.

Bij het ontwerp van geavanceerde coderings- en data-ontvangsttechnieken, is het noodzakelijk om de prestaties van het te ontwerpen systeem te onderzoeken voor verschillende codes en informatiedichtheden. In Hoofdstuk 2 van het proefschrift wordt een veralgemening van het zogenaamde 'Braat-Hopkins' model van het optische data-opslag kanaal voorgesteld, als basis voor een eerlijke vergelijking van data-ontvangsttechnieken bij verschillende codeertechnieken en informatiedichtheden.

Diverse elementaire kwesties behorend bij het ontwerp van PC

codes worden bestudeerd in Hoofdstuk 3. Hiertoe behoren grenzen voor de bitfoutenkansen en fout-patroon waarschijnlijkheden, de dominante foutpatronen aan de uitgang van de bit-detector, eenvoudige en efficiënte 'post-processors' die gebruikt kunnen worden om fouten aan de uitgang van de bit-detector te herstellen, de invloed van foutpatronen die zich over de grenzen van data-blokken uitstrekken, en het effect van code rates en informatiedichtheden op de bereikbare prestatieniveaus. Simulatieresultaten tonen aan dat een 4-bits PC code de beste prestaties levert. De betreffende bitfoutenkansen benaderen de eerder afgeleide grenzen dicht.

Behalve foutcorrigerende codes maken optische data-opslagsystemen ook gebruik van zogenaamde 'constrained' codes. Dergelijke codes moeten idealiter een hoge code rate hebben, en een eenvoudige encoder en decoder. Hoofdstuk 4 beschrijft het ontwerp van bepaalde capaciteits-benaderende 'constrained' codes voor optische data-opslagsystemen. In het bijzonder wordt analytisch de relatie bepaald tussen het aantal toestanden van de encoder en de maximale grootte van de codes. Verder wordt het minimale aantal encoder toestanden afgeleid dat de code rate maximaliseert, voor elke voorgeschreven lengte van de codewoorden.

Het ontwerp van 'constrained' PC codes, dit wil zeggen het ontwerp van 'constrained' codes die tevens fungeren als PC codes, is cruciaal voor de ontwikkeling van efficiënte PC-gebaseerde systemen, en het systematische ontwerp van dergelijke gecombineerde codes is een uitdagend onderwerp. In hoofdstuk 5 wordt een algemene en systematische methode beschreven voor het ontwerp van capaciteits-benaderende 'constrained' PC codes, geschikt voor detectie van willekeurige voorgeschreven foutpatronen. In vergelijking met de gebruikelijke rate $2/3$ 'constrained' code zonder PC functionaliteit behaalt de nieuw ontworpen 4-bit 'constrained' PC code een 2 dB hogere marge tegen ruis bij de nominale informatiedichtheid, en een 1.5 dB hogere marge bij hoge informatiedichtheid, in beide gevallen bij een bitfoutenkans van 10^{-4} .

Bij hoge informatiedichtheden vormt media-ruis de dominante ruisbron. Dergelijke ruis heeft geheel andere karakteristieken dan de gebruikelijke additieve witte Gaussische ruis die dominant is bij lagere informatiedichtheden, en leidt tot een belangrijke prestatieverslechtering bij systemen die zijn geoptimaliseerd voor additieve witte Gaussische ruis. Hoofdstuk 6 beschrijft twee nieuwe modificaties van de bit-

detector die leiden tot een betere weerstand tegen media-ruis. Daarnaast wordt een data-afhankelijk 'post-processor' ontworpen voor fout-correctie. Ten opzichte van het systeem zonder detector-modificatie en zonder 'post-processor' heeft dit systeem een 11 dB hogere weerstand tegen media-ruis bij hoge informatie-dichtheden.

Voor data-opslag systemen wordt de zogenaamd 'ECC failure rate' (EFR) beschouwd als de ultieme prestatie-maatstaf. In hoofdstuk 7 worden semi-analytische benaderingen afgeleid voor schatting van de EFR, en worden EFRs bepaald voor de in eerdere hoofdstukken ontwikkelde systemen, voor situaties met en zonder 'interleaving'. Met een optimale graad van 'interleaving' wordt, ten opzichte van een 'constrained' code met een rate van 9/13 zonder PC functionaliteit, een 0.5 dB hogere weerstand tegen ruis bereikt met een 2-bits PC code, en 0.6 dB met een 4-bits PC code, bij hoge informatiedichtheid en $\text{EFR}=10^{-16}$.

Het proefschrift wordt besloten in Hoofdstuk 8 met enkele aanbevelingen voor toekomstig werk.

Publications Originated from This Thesis

Based on the work reported in this thesis, the author has contributed the following publications.

C.1 List of Papers

1. K. Cai, V.Y. Krachkovsky, and J.W.M. Bergmans, "Performance bounds for parity coded optical recoding channels with $d=1$ constraint," in *Proc. IEEE Intl. Conf. Commun. (ICC)*, Alaska, USA, May 2003, pp. 2914-2918.
2. K. Cai, G. Mathew, J.W.M. Bergmans, and Z. Qin, "A generalized Braat-Hopkins model for optical recording channels," in *Proc. IEEE Intl. Conf. Consumer Electronics (ICCE)*, Los Angeles, USA, Jun. 2003, pp. 324-325.
3. K.A.S. Immink and K. Cai "On the size of a type of RLL codes," in *Proc. 25th Symp. Inform. Theory in the Benelux*, Kerdrade, The Netherlands. Jun. 2004, pp. 39-46.
4. K. Cai, K.A.S. Immink, J.W.M. Bergmans, and L.P. Shi, "Novel constrained parity-check code and post-processor for advanced blue laser disk," in *Tech. Dig. Int. Symp. Optical Media and Optical Data Storage (ISOM/ODS)*, Hawaii, USA, Jul. 2005, pp. TuB5.
5. K. Cai and K.A.S. Immink, "On the number of encoder states for capacity approaching $d=1$ codes," in *Proc. IEEE Intl. Symp.*

- Inform. Theory*, (ISIT), Adelaide, Australia, Sep. 2005, pp. 1483-1487.
6. K. Cai and K.A.S. Immink, "On the design of efficient constrained parity-check codes for optical recording," in *Proc. IEEE Intl. Symp. Inform. Theory*, (ISIT), Adelaide, Australia, Sep. 2005, pp. 1473-1477.
 7. K. Cai, J.W.M. Bergmans, A. Qin, A. Padiy, and W.M.J. Coene, "Parity-check codes and post-processing for d=1 optical recording channels," *IEEE Trans. Magn.*, vol. 41, no. 11, pp. 4338-4340, Nov. 2005.
 8. K. Cai, K.A.S. Immink, J.W.M. Bergmans, and L.P. Shi, "Constrained parity-check code and post-processor for advanced blue laser disc systems," *Jpn. J. Appl. Phys.*, vol. 45, no. 2B, pp. 1071-1078, Feb. 2006.
 9. K. Cai and K.A.S. Immink, "On the number of encoder states a type of RLL codes," *IEEE Trans. Inform. Theory*, vol. 52, no. 7, pp. 3313-3319, Jul. 2006.
 10. K. Cai and K.A.S. Immink, "A general construction of constrained parity-check codes for optical recording," accepted by *IEEE Trans. Commun.*
 11. K. Cai, Y.N. Foo, and J.W.M. Bergmans, "Combating media noise for high-density optical recording," accepted by *IEEE Intl. Conf. Commun. (ICC)*, Jun. 2007.
 12. K. Cai, V.Y. Krachkovsky, and J.W.M. Bergmans, "Performance analysis of parity-check coded optical recording channels with d=1 constraint," revised version under review with *IEEE Trans. Commun.*

C.2 List of Patents

1. W.M.J. Coene, K. Cai, and J.W.M. Bergmans, "Intelligent post-processor for error-location in parity-check encoded storage system," Invention disclosure, Philips Corporation, No. 612382, Nov. 12, 2002.

-
2. A. Padiy, K. Cai, and J.W.M. Bergmans, “Novel scheme for handling cross-boundary error events in channel parity check codes,” Invention disclosure, Philips Corporation, No. 615105, Jun. 5, 2003.
 3. K. Cai and K.A.S. Immink, “Method and system for encoding and decoding information with modulation constraints and error control,” PCT patent, PCT/SG2004/000357, Oct. 26, 2004.
 4. K. Cai and K.A.S. Immink, “Method and system for encoding and decoding information with modulation constraints and error control,” TaiWan patent, No. 94137323, Oct. 25, 2005.

References

- [1] D.J. Adelerhof, "Media development for DVD+RW phase change recording," in *Proc. European Symp. Phase Change and Ovonic Science (E*PCPS)*, Balzers, Principality of liechtenstein, Sep. 2004.
- [2] R.L. Adler, D. Coppersmith, and M. Hassner, "Algorithms for sliding block codes: An application of symbolic dynamics to information theory," *IEEE Trans. Inform. Theory*, vol. 29, pp. 5-22, Jan. 1983.
- [3] G. Arfken, "Lagrange Multipliers", in *Mathematical Methods for Physicists. 3rd ed.* Orlando, FL: Academic Press, 1985, Chap. 17.
- [4] J.J. Ashley, R. Karabed, and P.H. Siegel, "Complexity and sliding-block decidability," *IEEE Trans. Inform. Theory*, vol. 42, no. 6, pp. 1925-1947, Nov. 1996.
- [5] L. Baert, L. Theunissen and G. Vergult, *Digital Audio and Compact Disc Technology*. Oxford: Butterworth-Heinemann Ltd, 1992, Chap. 2.
- [6] J.W.M. Bergmans, *Digital Baseband Transmission and Recording*. Kluwer, 1996, Chap. 2.
- [7] J.W.M. Bergmans, "SNR merits of binary modulation codes in equalized digital recording systems," *IEEE J. Select. Areas Commun.*, vol. 10, no. 1, pp. 157-161, Jan. 1992.
- [8] J.W.M. Bergmans, "Density improvements in digital magnetic recording by decision feedback equalization," *IEEE Trans. Magn.*, vol. 22, no. 3, pp. 157-162, May 1986.

- [9] W.G. Bliss, "Circuitry for performing error correction calculations on baseband encoded data to eliminate error propagation," *IBM. Tech. Discl. Bul.*, vol. 23, pp. 4633-4634, 1981.
- [10] W. Bliss, R. Karabed, K. Zhang, C. Varanasi, and J. Ashley, "Sector error rate estimation of concatenated codes in magnetic recording," in *Proc. IEEE Intl. Conf. Commun. (ICC)*, Helsinki, Finland, Jun. 2001, pp. 2726-2730.
- [11] G. Bouwhuis, J. Braat, A. Huijser, J. Pasma, G. van Rosmalen, and K.A.S. Immink, *Principles of Optical Disc Systems*. Adam Hilger Ltd, 1985, Chap. 2.
- [12] J. Campello de Souza, B.H. Marcus, R. New, and B.A. Wilson, "Constrained systems with unconstrained positions," *IEEE Trans. Information Theory*, vol. 48, no. 4, pp. 866-879, Apr. 2002.
- [13] M.G. Carasso, J.B.H. Peek, and J.P. Sinjou, "The compact disc digital audio system," *Philips Tech. Rev.*, vol. 40, no. 6, pp. 151-156, 1982.
- [14] R.D. Cideciyan, F. Dolivo, R. Hermann, W. Hirt, and W. Schott, "A PRML system for digital magnetic recording," *IEEE J. Select. Areas Commun.*, vol. 10, no. 1, pp. 38-56, Jan. 1992.
- [15] R.D. Cideciyan, J.D. Coker, E. Eleftheriou, and R.L. Galbraith, "Noise-predictive maximum likelihood detection combined with parity-based post-processing," *IEEE Trans. Magn.*, vol. 37, no. 2, pp. 714-720, Mar. 2001.
- [16] R.D. Cideciyan and E. Eleftheriou, "Codes satisfying maximum transition run and parity-check constraints," in *Proc. IEEE Intl. Conf. Commun. (ICC)*, Paris, France, Jun. 2004, pp. 635-639.
- [17] W.M.J. Coene, H. Pozidis, M.V. Dijk, J. Kahlman, R.V. Woudenberg, and B. Stek, "Channel coding and signal processing for optical recording systems beyond DVD," *IEEE Trans. Magn.*, vol. 37, no. 2, pp. 682-688, Mar. 2001.
- [18] W.M.J. Coene, "Two-dimensional optical storage," in *Tech. Dig. Int. Symp. Optical Data Storage (ODS)*, Vancouver, Canada, May 2003, pp. 90-92.

-
- [19] W.M.J. Coene, "Combi-codes for DC-free runlength-limited coding," *IEEE Trans. Cons. Electr.*, vol. 46, pp. 1082-1087, Nov. 2000.
- [20] W.M.J. Coene, H. Pozidis, and J.W.M. Bergmans, "Run-length limited parity-check coding for transition-shift errors in optical recording," in *Proc. IEEE Intl. Conf. Global Telecommun. (GLOBECOM)*, San Antonio, USA, Nov. 2001, pp. 2982-2986.
- [21] J.D. Coker, E. Eleftheriou, R.L. Galbraith, and W. Hirt, "Noise-predictive maximum likelihood (NPML) detection," *IEEE Trans. Magn.*, vol. 34, no. 1, pp. 110-117, Jan. 1998.
- [22] K. Compaan and P. Kramer, "The Philips 'VLP' system," *Philips Tech. Rev.*, vol. 33, pp. 178-180, 1973.
- [23] T. Conway, "A new target response with parity coding for high density magnetic recording channels," *IEEE Trans. Magn.*, vol. 34, no. 4, pp. 2382-2386, Jul. 1998.
- [24] DVD Specification for Read-Only Disc: Part 1, Physical Specifications ver. 1.0.
- [25] M.R. Elidrissi, "Design of efficient constrained codes and parity-check codes for perpendicular magnetic recording channels," Master thesis, The National University of Singapore, 2004, Chap. 4.
- [26] J.L. Fan and A.R. Calderbank, "A modified concatenated coding scheme, with applications to magnetic recording," *IEEE Trans. Information Theory*, vol. 45, no. 4, pp. 1566-1574, Jul. 1998.
- [27] W. Feng, A. Vityaev, G. Burd, and N. Nazari, "On the performance of parity-check codes in magnetic recording systems," in *Proc. IEEE Intl. Conf. Global Telecommun. (GLOBECOM)*, San Francisco, USA, Nov. 2000, pp. 1877-1881.
- [28] B. Finkelstein and W. Williams, "Noise sources in magneto-optical recording," *Appl. Opt.*, vol. 27, no. 4, pp. 703-709, 1988.
- [29] G.D. Forney Jr., "Maximum-likelihood sequence estimation of digital sequences in the presence of intersymbol interfer-

- ence," *IEEE Trans. Inform. Theory*, vol. 18, no. 3, pp. 363-378, May 1972.
- [30] G.D. Forney Jr., "The Viterbi algorithm," *Proc. IEEE*, vol. 61, no. 3, pp. 268-278, Mar. 1973.
- [31] G.D. Forney Jr., "Lower bounds on error probability in the presence of large intersymbol interference," *IEEE Trans. Commun.*, vol. 20, pp. 76-77, Feb. 1972.
- [32] S. Gopaldaswamy and J.W.M. Bergmans, "Modified target and concatenated coding for constrained magnetic recording channels, in *Proc. IEEE Intl. Conf. Commun. (ICC)*, New Orleans, USA, Jun. 2000, pp. 89-93.
- [33] K. Han and R. Spencer, "Comparison of different detection techniques for digital magnetic recording channels," *IEEE Trans. Magn.*, vol. 31, pp. 1128-1133, Mar. 1995.
- [34] E.F. Haratsch and Z.A. Keirn, "Digital signal processing in read channels," in *Proc. IEEE Custom Integrated Circuits Conf.*, San Jose, USA, Sep. 2005, pp. 683-690.
- [35] S. Haykin, *Adaptive Filter Theory*. Prentice-Hall Inc., 1996, Chap. 5.
- [36] A. F. Horadam, "Generating functions for powers of certain generalized sequence of numbers," *Duke Math. J.*, 32, pp. 437-446, 1965.
- [37] K.A.S. Immink, P.H. Siegel, and J.K. Wolf, "Codes for digital recorders," *IEEE Trans. Inform. Theory*, vol. 44, no. 6, pp. 2260-2299, Oct. 1998.
- [38] K.A.S. Immink, *Codes for Mass Data Storage Systems*. Shannon Foundation Publishers, 1999, Chap. 2.
- [39] K.A.S. Immink and H. Ogawa, "Method for encoding binary data," US patent 4,501,000, Feb. 1985.
- [40] K.A.S. Immink, "EFMPlus: The coding format of the multimedia compact disc," *IEEE Trans. Consumer Electronics*, vol. 41, no. 3, pp. 491-497, Aug. 1995.

-
- [41] K.A.S. Immink and A. van Wijngaarden, "Simple high-rate constrained codes," *Electronics Letters*, vol. 32, no. 20, pp. 1877, Sep. 1996.
- [42] K.A.S. Immink, J.Y. Kim, S.W. Suh, and S.K. Ahn, "Efficient DC-free RLL codes for optical recording," *IEEE Trans. Commun.*, vol. 51, pp. 326-331, Mar. 2003.
- [43] K.A.S. Immink, "A practical method for approaching the channel capacity of constrained channels," *IEEE Trans. Inform. Theory*, vol. 43, no. 5, pp. 1389-1399, Sep. 1997.
- [44] K.C. Indukumar, "A novel post-processing scheme for threshold detection on $d = 2$ optical channels," in *Proc. IEEE Intl. Conf. Global Telecommun. (GLOBECOM)*, San Antonio, USA, Nov. 2001, pp. 1479-1483.
- [45] T. Iwanaga, S. Ogkubo, M. Nakano, M. Kubota, H. Honma, T. Ide, and R. Katayama, "High-density recording systems using partial response maximum-likelihood with blue laser diode," *Jpn. J. Appl. Phys.*, vol. 42, no. 2B, part 1, pp. 1042-1043, Feb. 2003.
- [46] R. Kabal and D. Pasupathy, "Partial response signaling," *IEEE Trans. Commun.*, vol. 23, no. 9, pp. 921-934, Sep. 1988.
- [47] A. Kavčić and J.M.F. Moura, "The Viterbi algorithm and Markov noise memory," *IEEE Trans. Inform. Theory*, vol. 46, no. 1, pp. 291-301, Jan. 2000.
- [48] K. Kayanuma, C. Nota, and T. Iwanaga, "Eight to twelve modulation code for high density optical disk," in *Tech. Dig. Intl. Symp. Optical Memory (ISOM)*, Nara, Japan, Nov. 2003, pp. 160-161.
- [49] Z.A. Keirn, V.Y. Krachkovsky, E.F. Haratsch, and H. Burger, "Use of redundant bits for magnetic recording: Single-parity codes and Reed-Solomon error-correcting code," *IEEE Trans. Magn.*, vol. 40, no. 1, pp. 225-230, Jan. 2004.
- [50] J. Kim, I. Hwang, H. Kim, I. Park, and J. Tominaga, "Signal characteristics of super-resolution near-field structure disks with 100GB capacity," *Jpn. J. Appl. Phys.*, pt. 1, vol. 44, no. 5B, pp. 3609-3611, 2005.

- [51] P. Kovintavewat, I. Ozgunes, E.M. Kuitas, J.R. Barry, and S. W. McLaughlin, "Generalized partial response targets for perpendicular recording with jitter noise," *IEEE Trans. Magn.*, vol. 38, no. 5, pp. 2349-2351, Sep. 2002.
- [52] V.Y. Krachkovsky, Y.X. Lee, G. Mathew, B. Liu, M.Y. Lin, R.W. Wood, and T.M. Chew, "Error propagation evaluation for RLL-constrained DFE read channels," *IEEE Trans. Magn.*, vol. 34, pp. 147-152, Jan. 1998.
- [53] E.D. Lee and D.G. Messerschmitt, *Digital Communication*. Kluwer, 1994, Chap. 9.
- [54] I. Lee, C. Modlin, and J.M. Cioffi, "Equalized maximum-likelihood receiver in a magnetic recording channel," in *Proc. IEEE Intl. Conf. Global Telecommun. (GLOBECOM)*, Houston, USA, Nov. 1993, pp. 1970-1973.
- [55] I. Lee and J.M. Cioffi, "Equalized maximum-likelihood receiver with a unit energy constraint," *IEEE Trans. Magn.*, vol. 33, no. 1, pp. 855-862, Jan. 1997.
- [56] J. Li, K.R. Narayanan, E.M. Kurtas, and C.N. Georghiades, "On the performance of turbo product codes and LDPC codes over PR channels," *IEEE Trans. Commun.*, vol. 50, no. 5, pp. 723-734, May 2002.
- [57] S. Lin and D.J. Costello, *Error Control Coding Fundamentals and Applications*. Prentice-Hall Inc., 1983, Chaps. 4, 6.
- [58] B.H. Marcus, P.H. Siegel, and J.K. Wolf, "Finite-state modulation codes for data storage," *IEEE Trans. Commun.*, vol. 10, no. 1, pp. 5-37, Jan. 1992.
- [59] B.H. Marcus, R.M. Roth, and P.H. Siegel, "Constrained systems and coding for recording channels", in *Handbook of Coding Theory*, R. Brualdi, C. Huffman, and V. Pless, 1998, Chap. 20.
- [60] B.H. Marcus and R.M. Roth, "Bounds on the number of states in encoder graphs for input-constrained channels," *IEEE Trans. Inform. Theory*, vol. 37, no. 3, pp. 742-758, May 1991.

- [61] G. Mathew, K.C. Indukumar, Y.X. Lee, and R. Wood, "Performance comparison of a class of (1,7) DFE detectors," *IEEE Trans. Magn.*, vol. 34, pp. 1919-1921, Jul. 1998.
- [62] G. Mathew, B. Farhang-Boroujeny, and R. Wood "Equalization design techniques for MDFE detection on the magnetic recording channel," *IEEE Trans. Magn.*, vol. 33, no. 6, pp. 4528-4542, Nov. 1997.
- [63] L.L. McPheters and S.W. McLaughlin, "Turbo-coded optical recording channels with DVD minimum mark size," *IEEE Trans. Magn.*, vol. 38, no. 1, pp. 298-302, Jan. 2002.
- [64] S. Miyanabe, H. Kuribayashi, and K. Yamamoto, "New equalizer to improve signal-to-noise ratio," *Jpn. J. Appl. Phys.*, vol. 38, no. 3B, part 1, pp. 1715-1719, Mar. 1999.
- [65] J. Moon and W. Zeng, "Equalization for maximum-likelihood detector," *IEEE Trans. Magn.*, vol. 31, no. 2, pp. 1083-1088, Mar. 1995.
- [66] J. Moon and J. Park, "Pattern-dependent noise prediction in signal-dependent noise," *IEEE J. Select. Areas Commun.*, vol. 19, no. 4, pp. 730-743, Apr. 2001.
- [67] J. Moon, "Signal-to-noise ratio definition for magnetic recording channels with transition noise," *IEEE Trans. Magn.*, vol. 36, no. 5, pp. 3881-3883, Sep. 2000.
- [68] J. Moon and L.R. Carley, "Performance comparison of detection methods in magnetic recording," *IEEE Trans. Magn.*, vol. 26, no. 6, pp. 3155-3172, Nov. 1990.
- [69] J. Moon and B. Brickner, "Maximum transition run codes for data storage systems," *IEEE Trans. Magn.*, vol. 32, pp. 3992-3994, Sep. 1996.
- [70] T. Narahara, S. Kobayashi, Y. Shimpuku, G. van den Enden, J. Kahlman, M. van Dijk, and R. van Woudenberg, "Optical disc system for digital video recording," *Jpn. J. Appl. Phys.*, pt. 1, vol. 39, no. 2B, pp. 912-919, 2000.

- [71] Y. Nishida, H. Sawaguchi, A. Kuroda, H. Takano, H. Aoi, and Y. Nakamura, "Noise characteristics of double-layered perpendicular media," *J. Trans. Magn. and Magn. Mater.*, vol. 235, pp. 454-458, Oct. 2001.
- [72] T. Nishiya, K. Tsukano, T. Hirai, S. Mita, and T. Nara, "Rate 16/17 maximum transition run (3;11) code on an EEPRML channel with an error-correcting postprocessor," *IEEE Trans. Magn.*, vol. 35, no. 5, pp. 4378-4386, Sep. 1999.
- [73] M. Noda, "A new dc- and Nyquist-free trellis code for partial response class-I and class-II channels," *IEEE Trans. Magn.*, vol. 37, no. 2, pp. 756-761, Mar. 2001.
- [74] T. Oenning and J. Moon, "Modeling the Lorentzian magnetic recording channel with transition noise," *IEEE Trans. Magn.*, vol. 37, no. 1, pp. 583-591, Jan. 2001.
- [75] A. Padiy, B. Yin, C. Verschuren, R. Vlutters, T. Jansen and J. Lee, "Signal processing for 35GB on a single-layer Blu-ray disc," in *Tech. Dig. Int. Symp. Optical Data Storage (ODS)*, Monterey, USA, Apr. 2004, pp. 34-36.
- [76] A. Patapoutian, B. Shen, and P.A. McEwen, "Event error control codes and their applications," *IEEE Trans. Information Theory*, vol. 47, no. 6, pp. 2595-2603, Sep. 2001.
- [77] A.M. Patel, "Improved encoder and decoder for a byte-oriented rate 8/9, (0,3) code," *IBM Techn. Discl. Bull.*, vol. 28, pp. 1938, 1975.
- [78] J.B.H. Peek, "Communications aspect of the compact disc digital audio system," *IEEE Commun. Mag.*, vol. 23, no. 2, pp. 7-15, Feb. 1985.
- [79] C. Peng and M. Mansuripur, "Sources of noise in erasable optical disk data storage," *Appl. Opt.*, vol. 37, no. 5, pp. 921-928, Feb. 1998.
- [80] P. Perry, M.C. Lin, and Z. Zhang, "Runlength-limited codes for single error-detection and single error-correction with mixed type errors," *IEEE Trans. Inform. Theory*, vol. 44, no. 4, pp. 1588-1592, Jul. 1998.

- [81] G.L. Pierobon, "Codes for zero spectral density at zero frequency," *IEEE Trans. Inform. Theory*, vol. 30, pp. 435-439, Mar. 1984.
- [82] H. Pozidis, J.W.M. Bergmans, and W.M.J. Coene, "Modeling and compensation of asymmetry in optical recording," *IEEE Trans. Commun.*, vol. 50, no. 12, pp. 2052-2063, Dec. 2002.
- [83] J.G. Proakis, *Digital Communications*. McGraw-Hill Inc., 1995, Chap. 5.
- [84] I.S. Reed and G. Solomon, "Polynomial codes over certain finite fields," *J. Soc. Indust. Appl. Math.*, vol. 8, pp. 300-304, 1960.
- [85] K. Saeki and Z. Keirn, "Optimal combination of detection and error correction coding for magnetic recording," *IEEE Trans. Magn.*, vol. 37, no. 2, pp. 708-713, Mar. 2001.
- [86] H. Sawaguchi, S. Mita, and J.K. Wolf, "A concatenated coding technique for partial response channels," *IEEE Trans. Magn.*, vol. 37, no. 2, pp. 695-703, Mar. 2001.
- [87] H. Sawaguchi, Y. Nishida, and T. Nakagawa, "Performance analysis of soft-output post-processing detection for data-dependent media noise channels," *IEEE Trans. Magn.*, vol. 40, no. 4, pp. 3108-3110, Jul. 2004.
- [88] C.E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, pp. 379-423, Jul. 1948.
- [89] M. Shinoda, K. Saito, T. Ishimoto, A. Nakaoki, and M. Yamamoto, "High density near-field optical disc system," in *Tech. Dig. Int. Symp. Optical Media and Optical Data Storage (ISOM/ODS)*, Hawaii, USA, Jul. 2005, pp. WD1.
- [90] P.H. Siegel and J.K. Wolf, "Modulation and coding for information storage," *IEEE Commun. Mag.*, no. 12, pp. 68-86, Dec. 1991.
- [91] H. Song, B.V.K.V. Kumar, E.M. Kurtas, Y. Yuan, L.L. McPheters, and S.W. McLaughlin, "Iterative decoding for partial response (PR) equalized magneto-optical (MO) data storage channels," *IEEE J. Select. Areas Commun.*, vol. 19, no. 4, pp. 774-782, Apr. 2001.

- [92] H. Spruit, M. van der Vleuten, R. Elfrink, and D.J. Adelerhof, "High-speed DVD+RW recording," in *Tech. Dig. Int. Symp. Optical Media and Optical Data Storage (ISOM/ODS)*, Hawaii, USA, Jul. 2005, pp. TuC1.
- [93] E.F. Stikvoort and J.A.C. Rens, "An all-digital bit detector for compact disc players," *IEEE J. Select. Areas Commun.*, vol. 10, no. 1, pp. 191-200, Jan. 1992.
- [94] A. Taratorin, D. Cheng, and E. Marinero, "Media noise, nonlinear distortion, and thermal stability in high-density recording," *IEEE Trans. Magn.*, vol. 36, no. 1, pp. 80-85, Jan. 2000.
- [95] H.K. Thapar and A.M. Patel, "A class of partial response systems for increasing storage density in magnetic recording," *IEEE Trans. Magn.*, vol. 23, no. 5, pp. 2164-2166, Sep. 1987.
- [96] D. Treves and D.S. Bloomberg, "Signal, noise, and codes in optical memories," *Opt. Eng.*, vol. 25, no. 7, pp. 881-891, Jul. 1986.
- [97] D.J. Tyner and J.G. Proakis, "Partial response equalizer performance in digital magnetic recording channels," *IEEE Trans. Magn.*, vol. 29, no. 6, pp. 4194-4208, Nov. 1993.
- [98] S. Vajda, *Fibonacci and Lucas Numbers, and the Golden Section: Theory and Applications*. Halsted Press, 1989, Chap. 3.
- [99] N.N. Vorob'ev, *Fibonacci Numbers*. Heath, 1963, Chap. 1.
- [100] A.D. Weathers and J.K. Wolf, "A new rate 2/3 sliding-block code for the (1,7) constraint with the minimal number of encoder states," *IEEE Trans. Inform. Theory*, vol. 37, no. 3, pt.II, pp. 908-913, May 1991.
- [101] S.B. Wicker, *Error Control Systems for Digital Communication and Storage*. Prentice-Hall International Inc., 1995, Chap. 16.
- [102] A. van Wijngaarden and E. Soljanin, "A combinatorial technique for constructing high-rate MTR-RLL codes," *IEEE J. Select. Areas Commun.*, vol. 19, no. 4, pp. 582-588, Apr. 2001.
- [103] A. van Wijngaarden and K.A.S. Immink, "Maximum runlength-limited codes with error control capabilities," *IEEE J. Select. Areas Commun.*, vol. 19, no. 4, pp. 602-611, Apr. 2001.

-
- [104] J.K. Wolf, "ECC performance of interleaved RS codes with burst errors," *IEEE Trans. Magn.*, vol. 34, pp. 75-79, Jan. 1998.
- [105] F. Yokogawa, S. Miyanabe, M. Ogaswara, H. Kuribayashi, Y. Tomita, and K. Yamamoto, "Signal processing for 15/27 GB read-only disk system," *Jpn. J. Appl. Phys.*, vol. 39, no. 2B, pt. 1, pp. 819-823, Feb. 2000.
- [106] J-G. Zhu and X-G. Ye, "Impact of medium noise correlation on various partial response channels," *IEEE Trans. Magn.*, vol. 31, no. 6, pp. 3087-3089, Nov. 1995.

Curriculum Vitae

Kui Cai was born in Liao Ning, China, on February 14, 1970. She received the B.E. degree in information and control engineering from Shanghai Jiao Tong University, Shanghai, China, in 1992, and the M.Eng degree in electrical engineering from National University of Singapore (NUS), Singapore, in 2000. From 1992 to 1996, she was a research engineer with China Aeronautical Radio Electronics Research Institute, China, and during 1997, she was an application engineer with Siemens Shanghai Mobile Communications Ltd, China. Since 1999, she has been with the Data Storage Institute, Singapore.

During her association with DSI, she carried out algorithm development on coding and signal processing techniques for advanced data storage systems. The research covers constrained codes, error correction codes, channel equalization and detection, parity-check codes and post-processing, media noise channels' modeling, detection and post-processing.

Since May 2000, she also pursued a Ph.D. degree in the Department of Electrical and Computer Engineering of NUS, and subsequently was selected to the joint Ph.D. program between NUS and Technical University of Eindhoven (TU/e), The Netherlands. From February to May in 2003, she carried out her Ph.D. research in the Signal Processing Systems group in the Department of Electrical Engineering of TU/e.

