

UNIVERSITY OF CALIFORNIA, SAN DIEGO

**Channel Modeling, Signal Processing and Coding for Perpendicular Magnetic
Recording**

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

in

Electrical Engineering
(Communication Theory and Systems)

by

Zheng Wu

Committee in charge:

Professor Jack K. Wolf, Chair
Professor Paul H. Siegel, Co-Chair
Professor Vitaliy Lomakin
Professor Lance W. Small
Professor Alexander Vardy

2009

© Copyright
Zheng Wu, 2009
All rights reserved.

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

Co-Chair

Chair

University of California, San Diego

2009

To my parents.

CONTENTS

Signature Page	iii
Dedication	iv
Contents	v
List of Figures	viii
List of Tables	x
Acknowledgements	xi
Vita and Publications	xiii
Abstract of the Dissertation	xiv
Chapter 1 Introduction to the Magnetic Recording System	1
1.1 Perpendicular Recording Channels	3
1.1.1 Read and Write Process	3
1.1.2 Noise and Distortions in Perpendicular Recording Systems	6
1.2 Signal Processing and Coding Techniques in Read Channel	7
1.2.1 Advanced Techniques	12
1.3 Summary of Dissertation	14
Bibliography	14
Chapter 2 Design Curves and Information-Theoretic Limits for Perpendicular Recording Systems	19
2.1 Design Curves for Perpendicular Recording Systems	20
2.1.1 Perpendicular Recording System with Jitter-Noise and AWGN	20
2.1.2 Parameter Optimization and Design curves	21
2.1.3 Design Curves of Different Coding and Equalization Schemes	24
2.2 Information-Theoretic Limit of the Perpendicular Recording System	27
2.2.1 Monte-Carlo Estimation of Information Rate	29
2.2.2 Reducing computation complexity	32
2.3 Information-Theoretic Limit of the Design Curves	33

	Bibliography	40
Chapter 3	Nonlinear Transition Shift and Write Precompensation in Perpendicular Recording Systems	42
3.1	Perpendicular Recording Channel with Nonlinear Transition Shift	43
3.1.1	Nonlinear Transition Shift Model	43
3.1.2	Write Precompensation	49
3.1.3	Channel Output and Approximations	50
3.1.4	System Diagram	51
3.2	Precompensation Schemes	52
3.2.1	Dibit Precompensation	52
3.2.2	Two-level Precompensation	53
3.2.3	Multilevel Precompensation	53
3.3	Simulation Results	54
3.3.1	Dibit Precompensation	54
3.3.2	Two-level Precompensation	57
3.3.3	Higher Density and Multilevel Precompensation	61
3.4	Other Criteria	64
	Bibliography	68
Chapter 4	Analysis of the Bit-Error-Rate for Perpendicular Recording Channels with NLTS	69
4.1	Pairwise Error Probability	70
4.1.1	Order-1 channel Approximation	72
4.1.2	Order-2 Channel Approximation	73
4.2	Upper and Lower Bounds on Bit-Error-Rate	75
4.3	Simulation Results	77
4.4	Application of the Analysis	80
Chapter 5	Mean-Adjusted Pattern-Dependent Noise Prediction Detector for Perpendicular Systems with Nonlinear Transition Shift	86
5.1	Channel Model and Notations	88
5.2	Pattern-Dependent Noise Predictive Detector	89
5.3	Mean-Adjusted Pattern-Dependent Noise Predictive Detector	91
5.4	Simulation Results	93
5.5	Conclusions	95
	Bibliography	97
Chapter 6	Error-Locating Codes and Soft-Decision Decoding	99
6.1	Tensor-Product Parity Code and Soft-Decision Decoding	100
6.1.1	Introduction to Error-Locating Codes	100

6.1.2	A Tensor-Product Parity Code with Soft-Decision De-	
	coding for an ISI Channel	105
6.1.3	Simulation Results	110
6.2	Generalized Error-Locating Codes and List Decoding	112
6.2.1	Introduction to Generalized Error-Locating Codes	113
6.2.2	List Decoding Algorithm for GEL Code	118
6.3	Conclusions	119
	Bibliography	120

LIST OF FIGURES

Figure 1.1	Illustration of the writing process in perpendicular recording systems.	4
Figure 1.2	The approximations of the ideal transition response in perpendicular recording systems.	5
Figure 1.3	The dipulse response using the error function approximation of the transition response. $T_{50} = 20\text{nm}$; $B = 15\text{nm}$	6
Figure 1.4	System diagram of the perpendicular recording system.	8
Figure 2.1	The system diagram for design curve computation.	22
Figure 2.2	An example of the user density optimization.	23
Figure 2.3	Illustration of the normalized design curve and the design curve in the (T_{50}, σ_J) -plane.	24
Figure 2.4	Normalized design curves for MMSE equalizers with unit energy constraint and monic constraint.	25
Figure 2.5	The system with tensor product parity codes.	27
Figure 2.6	Normalized design curves for tensor product parity check code combined with RS code.	28
Figure 2.7	SIR curves v.s. jitter noise for $SNR_W=17\text{dB}$	34
Figure 2.8	Normalized user density curves v.s. jitter noise for $SNR_W=17\text{dB}$	35
Figure 2.9	Normalized design curves for information-theoretic limit.	36
Figure 2.10	Comparison of the design curves for optimal RS code and SIR at $SNR_W=17\text{dB}$, $B_{\text{user}}=20\text{nm}$	37
Figure 3.1	The generation of the nonlinear transition shift.	44
Figure 3.2	Illustration of the calculation of NLTS.	45
Figure 3.3	The normalized absolute contributions of the past transitions.	47
Figure 3.4	The distribution of NLTS for a random sequence.	49
Figure 3.5	The system diagram for NLTS and precompensation study.	52
Figure 3.6	Influence on BER of NLTS window length.	55
Figure 3.7	Comparison of BER for several channel approximations.	56
Figure 3.8	Best dibit precompensation levels for different jitter noise levels.	57
Figure 3.9	Best dibit precompensation levels for different AWGN levels.	58
Figure 3.10	Best dibit precompensation levels for different T_{50}	59
Figure 3.11	Surface plot of BER for the two-level precompensation scheme.	60
Figure 3.12	Comparison of simulated BER for dibit precompensation and two-level precompensation with $\sigma_J = 0.08B$	61
Figure 3.13	Comparison of simulated BER for dibit precompensation and two-level precompensation with $\sigma_J = 0.12B$	62

Figure 3.14 Comparison of the Dibit, Two-level and multilevel precompensation schemes for $B=16\text{nm}$	63
Figure 3.15 Comparison of the Dibit, Two-level and multilevel precompensation schemes for $B=14\text{nm}$	65
Figure 3.16 Normalized MSE using the dibit precompensation for different T_{50} values.	67
Figure 4.1 Illustration of error event $\epsilon = (\xi_1 \rightarrow \xi_2)$	70
Figure 4.2 The lower bound estimation for order-1 channel approximation with dibit precompensation scheme.	78
Figure 4.3 The lower bound estimation for order-2 channel approximation with dibit precompensation scheme.	79
Figure 4.4 Histogram of the pairwise error probability for a system with no NLTS and a system with optimal two-level precompensation.	82
Figure 5.1 Comparison between Viterbi, PDNP, and MA-PDNP detectors.	94
Figure 5.2 Comparison between different L_y and L_x	95
Figure 5.3 MA-PDNP detector with dibit precompensation.	96
Figure 6.1 A systematic encoding procedure for error-locating codes.	103
Figure 6.2 System diagrams using TPP code. (a) without interleaver; (b) with interleaver.	107
Figure 6.3 Hard and soft decision decoder structures for TPP code for an ISI channel.	108
Figure 6.4 Combination of the channel trellis and the accumulated parity bit trellis.	110
Figure 6.5 The graph structure of the MP-BCJR algorithm.	111
Figure 6.6 Performance comparison of several decoding algorithms for systems with and without interleaver.	112
Figure 6.7 Encoding of the GEL codes.	115
Figure 6.8 Comparison between Farhner's hard decision decoding and the list decoding of the GEL code in BSC channel.	120
Figure 6.9 Comparison between Farhner's hard decision decoding and the list decoding of the GEL code in AWGN channel.	121

LIST OF TABLES

Table 2.1	Comparing T_{50}/B_{user} of the RS-coded system and SIR.	36
Table 2.2	Comparing code rate of the RS-coded system and SIR.	37
Table 3.1	The nonlinear transition shifts for different patterns.	48
Table 3.2	The precompensation levels corresponding to the preceding 3-bit patterns.	54
Table 3.3	Error-event counts for system with no NLTS and with optimal two-level precompensation	60
Table 3.4	Optimal precompensation values for different criteria	66
Table 4.1	Comparison of the optimal precompensation values obtained from the Monte-Carlo simulation and the lower bound estimate.	80

ACKNOWLEDGEMENTS

It is my fortune and pleasure to have two great professors – professor Jack Wolf and professor Paul Siegel – to be my advisors of the PhD program in UCSD. I would like to express my great thanks to them for their guidance, encouragement and support during my work and stay in UCSD. They are superb in their areas of research, kind and patient to students. I am impressed by their comprehensive knowledge and insight into problems. I am very grateful to Professor Wolf for his wisdom and encouragement. It is an invaluable experience and training with him as a student. Professor Siegel's enthusiasm and preciseness in research is always a model for me to follow in my future research work. I learned from both of them a lot.

I would like to express my special thanks to Neal Bertram, who contributed an important part of channel modeling physics in my work.

I am thankful to my dissertation committee members Alex Vardy, Vitaliy Lomakin and Lance Small for their time, help and effort of serving in my committee.

The STAR group and Wolf group have a wonderful environment. It has been an enjoyable experience with the past and current members of the two groups. I would like to thank my officemate, Sharon Aviran, for her helpful suggestions and discussions. I would like to thank Marcus Marrow, Joseph Soriaga, Michael Cheng and Brian Kurkoski for their help on starting my research when I joined the group. I would like to thank Panu Chaichanavong, Junsheng Han, Zeinab Taghavi and Seyhan Karakulak for many inspiring discussions. I would also like to thank Houssein Taghavi, Hadi Djahanshadi, Brian Butler, Eitan Yaakobi, Hao Wang, Aravind Iyengar, Federica Garin, Zsigmond Nagy, Ori Shental and Toshio Ito for their help and friendship.

I would like to express my thanks to Robert Lynch, my mentor during the internship at Seagate Technology, and Yawshing Tang, my mentor during the internship at Samsung. They are knowledgeable and kind. I learned a lot from them about magnetic recording systems and obtained precious practical industry experiences. I would also like to thank Michael Madden from Marvell for his useful suggestion and discussion to my dissertation work.

I really appreciate the friendly and dedicated CMRR staffs, especially Betty Manoulian and Iris Villanueva for their excellent administrative work. They help establish the wonderful and pleasant environment in CMRR and make it possible for me to concentrate on the research.

I would also like to express my gratefulness to many friends, such as Yushi, Huaxin, Hongtao, Haichang, Jun, Wenyi for helping me when I just arrived at San Diego and throughout my years of stay at UCSD, my previous and current roommates Shan, Yangyang, Junwen and Chengmo for their nice and pleasant company, and many other friends who make my life in San Diego wonderful and unforgettable.

Finally, I would like to express my love and thanks to my parents, my brother and my husband. Without their love, support, tolerant and advices, I would never have gone so far.

The research was supported partly by the Information Storage Industry Consortium (INSIC) Extremely High Density Recording (EHDR) program and the Center for Magnetic Recording Research at UCSD.

Chapter 2 is partly a reprint of the materials in paper Z. Wu, P. H. Siegel, H. N. Bertram, and J. K. Wolf, “Design curves and information-theoretic limits for perpendicular recording systems”, *IEEE Trans. Magn.*, vol. 43, no. 2, pp. 721-726, Feb. 2007. Chapter 3 is partly a reprint of the materials in paper Z. Wu, H. Bertram, P. Siegel, and J. Wolf, “Nonlinear transition shift and write precompensation in perpendicular magnetic recording”, Communications, 2008. ICC’08. IEEE International Conference on, pp. 1972-1976, May 2008. Part of Chapter 3 and Chapter 4 is a reprint of the materials in paper Z. Wu, P. H. Siegel, J. K. Wolf, and H. N. Bertram, “Analysis on nonlinear transition shift and write precompensation in perpendicular magnetic recording”, submitted to JSAC, 2009. Chapter 5 is a reprint of the paper Z. Wu, P. H. Siegel, J. K. Wolf, and H. N. Bertram, “Mean-adjusted pattern dependent noise prediction for perpendicular recording channels with nonlinear transition shift”, *IEEE Trans. Magn.*, vol. 44, no. 11, pp. 3761-3764, Nov. 2008.

VITA

2000	B.S. Tsinghua University, Beijing, China
2003	M.E. Tsinghua University, Beijing, China
2009	Ph.D University of California, San Diego

PUBLICATIONS

Z. Wu, P. H. Siegel, H. N. Bertram, and J. K. Wolf, “Design curves and information-theoretic limits for perpendicular recording systems”, *TMRC 2006, the Magnetic Recording Conference*, Pittsburgh, PA, August, 2006.

Z. Wu, P. H. Siegel, H. N. Bertram, and J. K. Wolf, “Design curves and information-theoretic limits for perpendicular recording systems”, *IEEE Trans. Magn.*, vol. 43, no. 2, pp. 721-726, Feb. 2007.

Z. Wu, H. Bertram, P. Siegel, and J. Wolf, “Nonlinear transition shift and write precompensation in perpendicular magnetic recording”, *Communications*, 2008. ICC’08. IEEE International Conference on, pp. 1972-1976, May 2008.

Z. Wu, P. H. Siegel, J. K. Wolf, and H. N. Bertram, “Mean-adjusted pattern dependent noise prediction for perpendicular recording channels with nonlinear transition shift”, *INTERMAG 2008, IEEE International Magnetism Conference*, Madrid, Spain, May 2008.

Z. Wu, P. H. Siegel, J. K. Wolf, and H. N. Bertram, “Mean-adjusted pattern dependent noise prediction for perpendicular recording channels with nonlinear transition shift”, *IEEE Trans. Magn.*, vol. 44, no. 11, pp. 3761-3764, Nov. 2008.

Z. Wu, P. H. Siegel, J. K. Wolf, and H. N. Bertram, “Analysis on nonlinear transition shift and write precompensation in perpendicular magnetic recording”, submitted to *IEEE J. Sel. Areas Commun.*, 2009.

ABSTRACT OF THE DISSERTATION

Channel Modeling, Signal Processing and Coding for Perpendicular Magnetic Recording

by

Zheng Wu

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

University of California San Diego, 2009

Professor Jack K. Wolf, Chair

Professor Paul H. Siegel, Co-Chair

With the increasing areal density in magnetic recording systems, perpendicular recording has replaced longitudinal recording to overcome the superparamagnetic limit. Studies on perpendicular recording channels including aspects of channel modeling, signal processing and coding techniques are presented in this dissertation.

To optimize a high density perpendicular magnetic recording system, one needs to know the tradeoffs between various components of the system including the read/write transducers, the magnetic medium, and the read channel. We extend the work by Chaichanavong on the parameter optimization for systems via design curves. Different signal processing and coding techniques are studied. Information-theoretic tools are utilized to determine the acceptable region for the channel parameters when optimal detection and linear coding techniques are used. Our results show that a considerable gain can be achieved by the optimal detection and coding techniques.

The read-write process in perpendicular magnetic recording channels includes a number of nonlinear effects. Nonlinear transition shift (NLTS) is one of them. The signal distortion induced by NLTS can be reduced by write precompensation during data

recording. We numerically evaluate the effect of NLTS on the read-back signal and examine the effectiveness of several write precompensation schemes in combating NLTS in a channel characterized by both transition jitter noise and additive white Gaussian electronics noise. We also present an analytical method to estimate the bit-error-rate and use it to help determine the optimal write precompensation values in multi-level precompensation schemes.

We propose a mean-adjusted pattern-dependent noise predictive (PDNP) detection algorithm for use on the channel with NLTS. We show that this detector can offer significant improvements in bit-error-rate (BER) compared to conventional Viterbi and PDNP detectors. Moreover, the system performance can be further improved by combining the new detector with a simple write precompensation scheme.

Soft-decision decoding for algebraic codes can improve performance for magnetic recording systems. In this dissertation, we propose two soft-decision decoding methods for tensor-product parity codes. We also present a list decoding algorithm for generalized error locating codes.

Chapter 1

Introduction to the Magnetic Recording System

The first computer hard disk drive IBM 350 was introduced by IBM in 1956, as a part of IBM RAMAC 305 system, where RAMAC is the abbreviation for “Random Access Method of Accounting and Control”. This disk drive stored about 4.4 megabytes [1], with a rotational speed of 1200 RPM. The data transfer rate was 8,800 bits per second. However, the size and the price of this hard disk drive was huge. The disk diameter was 24 inches. Assembled with covers, the IBM 350 was 60 inches long, 68 inches high and 29 inches deep. The IBM RAMAC 305 system with IBM 350 disk storage could be leased for \$3,200 per month. Compared to this huge and costly disk drive, today’s hard disk drives are much smaller and cheaper, with much larger capacity and higher speed. In July of 2008, Seagate Technologies announced their newest 3.5 inch desktop internal hard disk drive, Barracuda®7200.11. It has up to 1.5 Terabytes (1.5×10^{12} Bytes) capacity, with rotational speed of 7200 RPM and data transfer rate of 3 Gigabits per second. Furthermore, the price is less than \$200.

Moore’s law describes the exponential growth of the number of transistors per integrated circuit over time [2]. It states that this number will double every year. The growth of the capacity of the hard disk drives has the similar trend, following Kryder’s law. The development of improved heads, servo, media and signal processing techniques

are the forces behind this fast growth.

One of the most recent significant development of the recording technologies is the introduction of the perpendicular recording technology. Before that, for over 40 years, the bits on the hard disk drives were recorded as magnetic fields horizontal to the surface of the medium, which is called longitudinal recording. When the recording density increases, the superparamagnetic effect will occur. The microscopic magnetic grains on the disk becomes so tiny that their magnetic orientations are not stable even at room temperature. Though increasing the coercivity (the “field” required to write a bit) of the media can avoid the superparamagnetic effect, the write field can not be arbitrarily large because of the materials from which the head is made [3, 4]. With perpendicular recording technology, where the orientation of the magnetic field of each bit is perpendicular to the surface of the medium, because of the geometry change in the writing process, the maximum fringing write field generated by the head of the same material is twice as big as that of the longitudinal recording. More over, unlike the longitudinal recording, the demagnetizing field in perpendicular recording decreases with higher density, thus improving thermal stability [4]. Therefore, with perpendicular recording, higher recording density can be achieved. The first commercial product using perpendicular hard disk drives was introduced in 2005 by Toshiba, followed by Hitachi Global Storage Technologies and Seagate Technology later in the same year. Today’s high capacity hard disk drives all use perpendicular recording technology.

In this dissertation, our focus will be on the read channel and the signal processing techniques used in perpendicular recording systems. In the remain of this chapter, we will give an introduction to perpendicular recording channels and the read channel architecture. A summary of the dissertation concludes the chapter.

1.1 Perpendicular Recording Channels

1.1.1 Read and Write Process

In hard disk drives, data are recorded on the disk as binary bits. The bits are written on the tracks, which are circular bands around the disk center.

In perpendicular recording systems, the polarity of the bits is represented by the upward or downward orientation of the magnetic field on the medium. Writing bits is a process of magnetizing the medium in different directions. Figure 1.1 gives an illustration of the writing process. The recording flux is controlled by the current in the write head. The amplitude of the current should be large enough to magnetize the medium to saturation and the direction of the current determines the magnetization direction. By spinning the disk, bits on the track are written sequentially, with the later bit overlapping the previous one. The rotational speed is controlled carefully to achieve the desired channel bit spacing.

The data are written on the disk sector by sector. A sector has 4096 user data bits, plus other overhead bits. Each track includes multiple sectors. However, a 4kB sector size is proposed today, so that longer error correcting codes that have better performance can be used.

When the magnetization pattern on the disk is to be read, the disk spins and the read head moves over the track to sense the magnetic field above the medium. The read-back signal can be regarded as a superposition of the isolated transition responses, at the transition positions of the magnetic fields. Mathematically, we can write the noiseless read-back signal as follows:

$$z(t) = \sum_i d_i s(t - iB) \quad (1.1)$$

where $\{d_i\}$ is the sequence of the transitions, B is the channel bit spacing, and $s(t)$ is the transition response. The “transition” d_i can be 0, +1 or -1, where 0 represents no transition at position i , and ± 1 represent transitions of different directions.

The isolated transition response could be analyzed and measured from the disk.

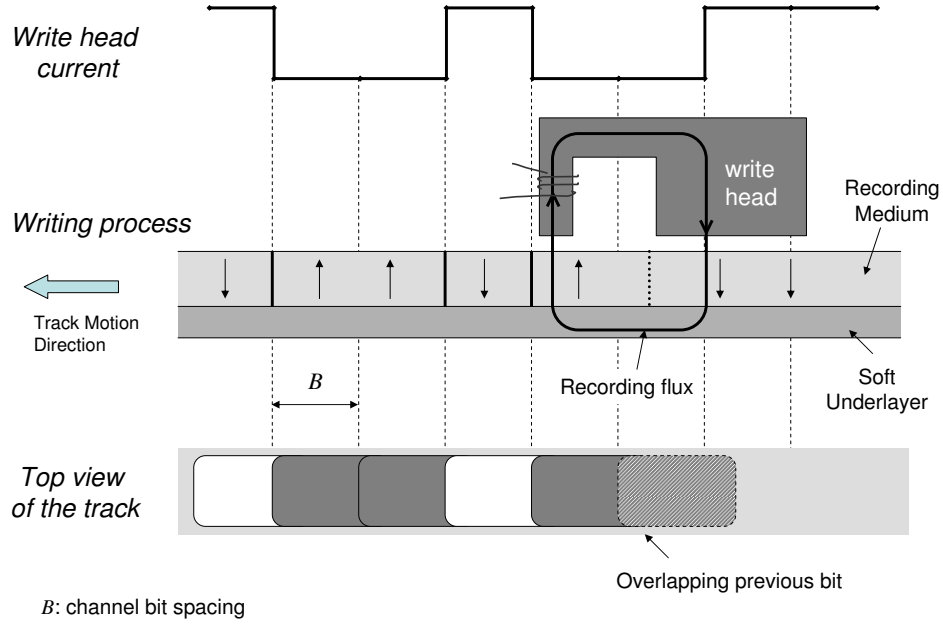


Figure 1.1 Illustration of the writing process in perpendicular recording systems.

In the literature, two approximations of the ideal transition response are used in simulations. One of them uses the tanh function, such as in [5, 6]. The transition response can be expressed as:

$$s(t) = V_{\max} \tanh \left(\frac{2t}{0.579\pi T_{50}} \right), \quad (1.2)$$

where V_{\max} is the zero-to-peak amplitude and T_{50} is the width when $s(t)$ changes from $-A/2$ to $A/2$, as shown in Figure 1.2. Since the tanh function is an odd function, T_{50} is also the value that satisfies

$$s(T_{50}/2) = A/2. \quad (1.3)$$

The other approximation for the channel transition response is the error function approximation [7, 8, 9]. The transition response is defined as

$$s(t) = V_{\max} \operatorname{erf} \left(\frac{0.954t}{T_{50}} \right), \quad (1.4)$$

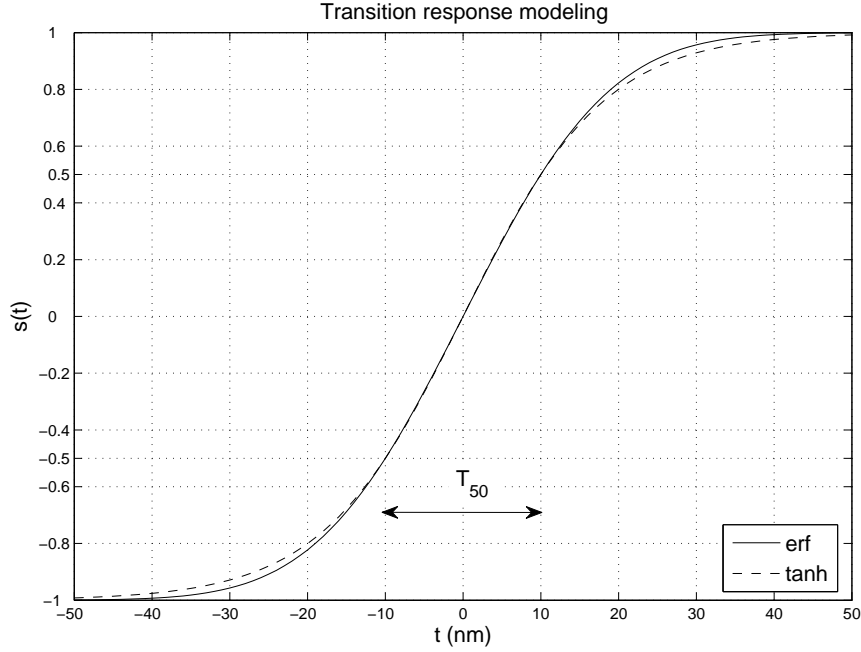


Figure 1.2 The approximations of the ideal transition response in perpendicular recording systems.

where $\text{erf}(\cdot)$ has the form

$$\text{erf}(t) = \frac{2}{\sqrt{\pi}} \int_0^t e^{-u^2} du. \quad (1.5)$$

The parameter T_{50} in this approximation has the same meaning as in the \tanh approximation. It satisfies equation (1.3) as well. A comparison of the two approximations is shown in Figure 1.2, using the parameters $T_{50} = 20\text{nm}$ and $V_{\max} = 1$.

The noiseless read-back signal can also be expressed as the superposition of the *dipulse response*, where the input data are considered to be the NRZ pattern, not the NRZI pattern for transitions. Let the binary data sequence be $\{x_i\}$, where $x_i \in \{+1, -1\}$. The transition d_i can be calculated as $d_i = \frac{x_i - x_{i-1}}{2}$. Denoting the dipulse response by $h(t)$, the read-back signal can also be represented by

$$z(t) = \sum_i x_i h(t - iB) \quad (1.6)$$

where $h(t) = (s(t) - s(t - B))/2$. An example of the dipulse response is shown in

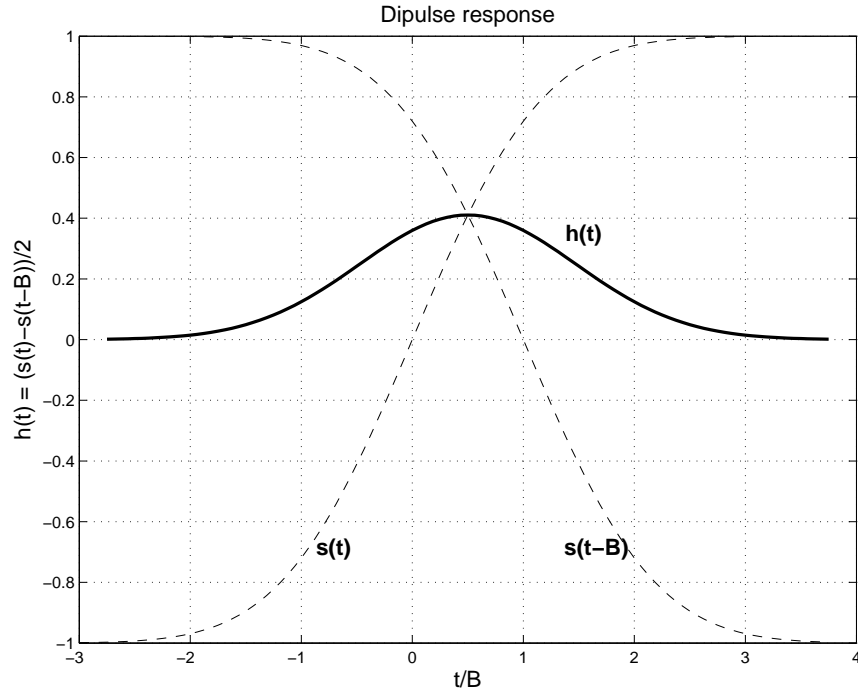


Figure 1.3 The dipulse response using the error function approximation of the transition response. $T_{50} = 20\text{nm}$; $B = 15\text{nm}$.

Figure 1.3, which uses the error function approximation of the transition response.

1.1.2 Noise and Distortions in Perpendicular Recording Systems

The read back signal will be affected by various noise and distortions, which cause errors during the recovery of the data. The fundamental noise sources are media noise, the noise from the head and the noise from the electronic circuits such as the preamplifier [10]. The noise from the head and the electronic circuits are usually modeled as the additive white Gaussian noise (AWGN).

The noise from the media is the dominant noise source in perpendicular recording systems [11]. Among the components of the media noise, the *transition noise* is dominant [4, 6, 12]. The transition noise originates from the randomness of the magnetization of grains in the transition region. It can be modeled as a random jitter of the

transition position, which can be included into the read-back signal as follows:

$$z(t) = \sum_i d_i s(t + a_i - iB) \quad (1.7)$$

where a_i is the transition jitter for transition d_i . If d_i is zero, there is no transition and thus a_i is also zero. When d_i is non-zero, the transition jitter a_i is a random variable. In much of the literature, for simplicity, the transition jitter a_i is assumed to have a Gaussian distribution, or a truncated Gaussian distribution. The random variables representing the transition jitters are assumed to be independent to each other.

Nonlinear distortions can also degrade the system performance, such as the nonlinear transition shift and the read head nonlinearities [13]. The nonlinear transition shift (NLTS) is due to the magnetic field from the neighboring recorded transitions, on the same track and from adjacent tracks. Also because of the neighboring magnetization, the read head could operate in a nonlinear regime, yielding asymmetry in the dipulse response [14, 15].

Inter-track interference (ITI) is another problem that arises with increasing density. It can be modeled as a linear signal added to the main track signal. Signal processing techniques for ITI in perpendicular channel were studied [16].

Thermal asperities, which also happens in longitudinal recording, arise from the defects on the medium surfaces. The friction between the asperity and the magnetoresistive read head causes the head to heat rapidly, which results in a change of the head resistance and therefore the read-back signal. Long bursts of errors usually happen because of thermal asperities. In practice, thermal asperities can be detected and canceled by signal processing techniques, as proposed in [17, 18].

1.2 Signal Processing and Coding Techniques in Read Channel

From the viewpoint of signal processing and coding techniques, the data storage system can be modeled as a communication system. The communication channel here

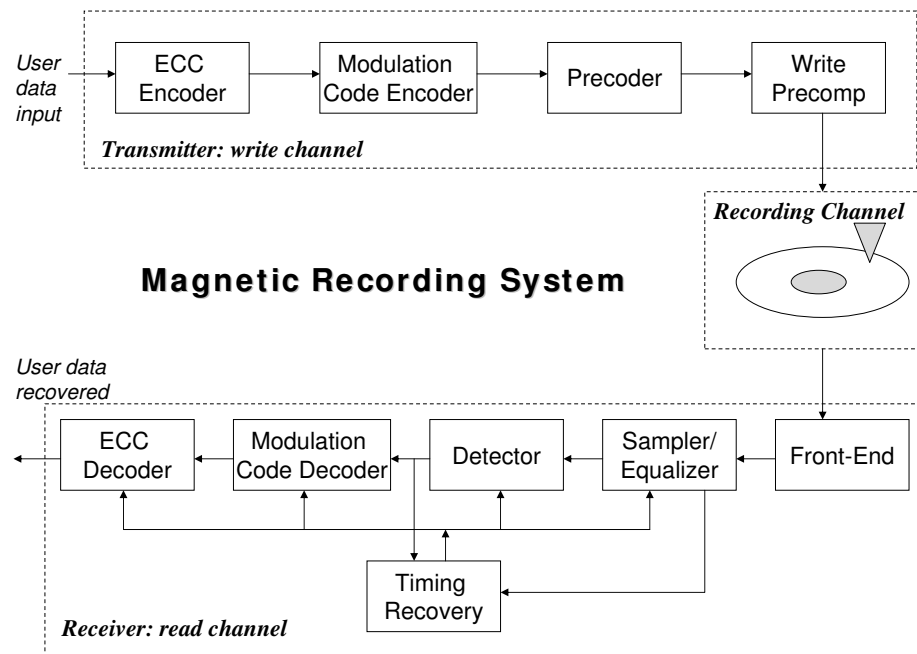


Figure 1.4 System diagram of the perpendicular recording system.

is the recording channel. A system diagram of the magnetic recording system is shown in Figure 1.4.

The writing process can be regarded as the transmitter. The binary user data are usually first encoded with an error correction code (ECC) and then with a modulation code. A precoder is sometimes used. The transition positions in the write current are adjusted by the write precompensation process before going through the coil of the write head, to reduce the effect of the nonlinear transition shift.

The read back and data recovery process can be regarded as a receiver in the communication system. The read-back signal from the read head is an analog waveform. It is processed by the front-end circuit, which usually contained a preamplifier and an analog low-pass filter. The timing recovery loop provides the clock for sampling and the digital signal processing afterwards. The sampled signal is usually equalized before

the detector. The modulation code and the error correcting code are decoded afterwards to finally recover the user data.

In the following paragraphs, we are going to introduce some signal processing and coding techniques used in the system.

Partial Response Equalization

The magnetic recording channel is a channel with intersymbol interference. In early longitudinal magnetic recording systems, peak detection was used for low density recording. The pulses produced by two neighboring transitions are enough apart so that the interference between each other can be ignored. However, with increasing recording density, the interference between pulses becomes more severe, and the peak detection gives very poor performance. A partial response maximum likelihood detector was proposed to deal with the channel ISI [19, 20]. In order to reduce the number of states in the detector, a FIR equalizer is used to equalize the channel response to a desired, shorter partial-response target. One class of equalization target, labeled “Class-4” or “extended Class-4”, is widely used in longitudinal magnetic recording channels with different channel bit densities. A general expression of these target polynomials takes the form

$$g(D) = (1 - D)(1 + D)^N \quad (1.8)$$

where $N \geq 1$. When $N = 1$, a simplified channel that uses $g(D)$ as the channel transfer function is usually called a PR4 (Partial Response Class-4) channel. When $N \geq 2$, the channels are called extended Class-4 channels, denoted by $E^{N-1}\text{PR4}$.

Generalized partial response (GPR) targets were proposed later. The coefficients of the GPR target can be arbitrary real numbers. The noise-predictive maximum likelihood (NPML) detector [21] is based on such a target, since the noise whitening filter response can be built into the generalized partial response polynomial. By using GPR targets, the equalizer reduces the noise enhancement and at the same time, whitens the noise. In present systems, a GPR target is usually used.

There are many studies on the design of the equalizer, such as in [22, 23, 24].

The design objective is usually to minimize the mean squared error (MMSE) of the targeted signal and the equalized channel output. Since the channel parameters might change slightly on one disk drive, for example, because of the temperature varying, the equalizer can be designed to be adaptive to the channel, so that a stable targeted signal is provided for the detector afterwards [25].

Detection

Before the use of partial responses, the detection used a peak detection. Given a partial response target, a Viterbi, or Viterbi-like detector is used to detect the data sequence.

The conventional Viterbi detector is a maximum likelihood sequence detector for an ISI channel with AWGN [26]. For any received sequence \bar{r} , the maximum likelihood sequence detector gives the most probable input data sequence \bar{x} , such that the probability $\Pr(\bar{r}|\bar{x})$ is maximized.

The Viterbi algorithm runs on a trellis. The number of trellis states is determined by the partial response target length. The detected sequence is selected to be the one that minimizes the accumulated branch metric. For the conventional Viterbi detector, the branch metric is the Euclidean distance between the received samples and the labeled targeted outputs.

As we stated previously, the noise in the perpendicular recording channel is dominated by the transition noise which can not be modeled as AWGN. Besides, after the equalizer, the noise is usually colored. Therefore, the conventional Viterbi detector is not the optimal one. Modifications to the conventional Viterbi detector were proposed. A noise-predictive maximum likelihood detector was proposed in [21], which uses a noise prediction filter in the Viterbi algorithm. The pattern-dependent noise-predictive detector, which uses a different noise-prediction filter for each branch to capture the data-dependent nature of the jitter noise, was studied later [27, 28]. The system performance is much improved by using these modified detectors.

Modulation Code

Modulation codes, also called the constrained codes, are used in digital recording systems to avoid some of the data recovery failure mechanisms, such as timing recovery failure and detection failure [29, 30].

For example, in peak detection systems, a runlength-limited (d, k) constrained codes was used. The code limits the runlength of zeros in a codeword. This constraint is very suitable to the transition sequence recorded on the disk. Since the read head can only sense transitions on the medium, if there is no transition on a long stretch of track, i.e., there is a long sequence of zeros in the transition sequence, the timing recovery loop could lose track of the clock. Therefore, we need to make sure that the runlength of zeros can not be longer than a certain number k . However, the run length of zeros can not be smaller than d bits in order to prevent interference between two neighboring transitions. If two transitions are too close, they will be affected by ISI and the detection might fail.

In PR4 channels, a $(0, G/I)$ modulation code is usually used. The parameter G limits the runlength of zeros in the transition sequence for the sake of the timing recovery, which is the same as the k constraint in the (d, k) codes. The constraint I is used to ensure the successful detection by the Viterbi detector. Since the trace-back length in the Viterbi detector is limited in real system, the maximum number of zeros in the even and odd substrings of the sequence needs to be limited by I .

A variety of other constraints are also used in the magnetic recording channels. For example, a DC-free constraint is extremely useful for a perpendicular recording channel where the DC is nonzero. Some constrained codes, such as the maximum-transition-run (MTR) constrained codes, are designed such that the distance between events in the PRML detector is increased, thus improving the system error performance.

The modulation codes adds overhead to the data sequence to ensure the constraint. Thus the rate of the code is very important to the system designers. The constrained codes used in real systems usually have very high code rates.

Error Correction Codes

Because of noise and distortions from the channel, there may be errors in the detected data. For hard disk drives, because of the huge volume of reads and writes of the data in a computer system, the ultimate error rate of the recovered user data needs to be very low, for example, 10^{-15} , which is much lower than in other communication system, such as in wireless communication. An error correction code is used after the detection and modulation code, to correct the errors which occurred during the recording process.

In magnetic recording systems, a Reed-Solomon (RS) code is the error correcting codes used for decades and is still being used today. RS codes are maximum distance separable (MDS) codes which have the maximum minimum Hamming distance given the number of parity check symbols. It can correct and detect burst errors. Hard decision decoding of RS codes is very well defined and simple to implement. Because of these advantages, it is used in various communication systems, including storage systems.

However, many other coding schemes were proposed to improve the system sector-error-rate (SER) and save the overhead further. For example, concatenating other codes with the RS codes, where sometimes the decoding of the inner code is regarded as the postprocessing of the detector, were proposed in [31, 32]. Multilevel error correcting codes were also proposed to handle both random and burst errors in the recording systems [33, 29]. The low-density parity-check (LDPC) codes are the most likely candidate today to substitute for the RS code in magnetic recording systems, because of its channel capacity approaching property [34]. The decoding of the LDPC codes uses a message passing algorithm, which is practical for implementation [35].

1.2.1 Advanced Techniques

Signal processing and coding techniques for magnetic recording systems are improving rapidly today. Here, we only introduce two advanced techniques among them.

Reverse Concatenation of Modulation Code and ECC Code

The order of the concatenation of the modulation code and the ECC code is not necessarily in the order shown in Figure 1.4. A reverse concatenation of the modulation code and ECC code modules was proposed [36, 37]. Since the decoding of the modulation code might result in error propagation, the reverse concatenation enables the ECC code to correct the errors directly from the detector, which is more efficient.

Turbo Equalization and Iterative Detection and Decoding

Turbo equalization is an iterative decoding and detection process for ISI channels. With the recent improvement of the iterative decoding techniques proposed for Turbo codes and LDPC codes, there is a trend toward using Turbo equalization techniques in the read channel.

In Turbo equalization, soft-decision detection algorithms such as the BCJR algorithm [38] and the soft-output Viterbi algorithm (SOVA) [39] are used to provide soft information on each channel bit. The ECC code decoder is a decoder that can handle soft input and output soft information as well. We call this type of decoder a *soft-input-soft-output (SISO)* decoder. The soft output of the detector is the input to the SISO decoder, while the soft output of the decoder can be fed back to the channel detector, to improve the detector performance. Thus, a Turbo-like structure is built between the detector and the decoder, and soft information is iteratively exchanged between them.

The ECC code which is required to have a practical SISO decoder, can be either a convolutional code [40], an LDPC code [41] or a Turbo code [42]. With the development of SISO decoder for the RS codes [43, 44], RS codes are also in consideration. The performance of the system with Turbo equalization is shown to be improved substantially compared to the conventional system.

1.3 Summary of Dissertation

In this dissertation, we will focus on the channel modeling of perpendicular recording systems with nonlinearities, and the study of the system performance, optimization, and related signal processing and coding techniques.

Chapter 2 will discuss the tradeoffs among channel parameters for perpendicular recording systems with specific signal processing and coding schemes. An information theoretic study will be discussed to show the system density limit and the tradeoff of the channel parameters.

Chapter 3 to 5 will focus on the study of the nonlinear transition shift (NLTS) phenomenon in perpendicular recording systems and related topics. In Chapter 3, write precompensation of different levels is studied. In Chapter 4, a bit-error-rate analysis is presented for the system with NLTS and write precompensation. Chapter 5 proposed a new modification of the Viterbi detector, which can reduce the effect of NLTS.

Chapter 6 presented a study on coding schemes in magnetic recording systems. A soft-decision decoding algorithm is proposed for tensor-product parity codes. A list decoding algorithm is also presented for the generalized error-locating (GEL) codes.

Bibliography

- [1] IBM 350 disk storage unit. [Online]. Available: http://www-03.ibm.com/ibm/history/exhibits/storage/storage_350.html
- [2] G. E. Moore, "Cramming more components onto integrated circuits," *Electronics Magazine*, Apr. 2006.
- [3] S. Khizroev and D. Litvinov, *Perpendicular Magnetic Recording*. Kluwer Academic Publishers, Boston, 2004.
- [4] B. F. Valcu, "Studies in perpendicular magnetic recording," Ph.D. dissertation, Department of Physics, University of California, San Diego, 2004.
- [5] J. Moon and J. Park, "Detection of prescribed error events: application to perpendicular recording," *Communications, 2005. ICC 2005. 2005 IEEE International Conference on*, vol. 3, pp. 2057–2062 Vol. 3, May 2005.

- [6] M. Madden, M. Oberg, Z. Wu, and R. He, "Read channel for perpendicular magnetic recording," *IEEE Trans. Magn.*, vol. 40, no. 1, pp. 241–246, Jan. 2004.
- [7] X. Yang and E. M. Kurtas, "Signal and noise generation for magnetic recording channel simulations," in *CRC Handbook for Coding and Signal Processing for Recording Systems*, B. Vasic and E. M. Kurtas, Eds. CRC Press, 2004.
- [8] M. Erden, I. Ozgunes, E. Kurtas, and W. Eppler, "General transform filters in perpendicular recording architectures," *IEEE Trans. Magn.*, vol. 38, no. 5, pp. 2334–2336, Sep 2002.
- [9] P. Chaichanavong, H. N. Bertram, and P. H. Siegel, "Design parameter optimization for perpendicular magnetic recording systems," *IEEE Trans. Magn.*, vol. 42, no. 10, pp. 2549–2554, Oct. 2006.
- [10] H. N. Bertram, *Theory of Magnetic Recording*. Cambridge University Press, 1994.
- [11] H. Sawaguchi, Y. Nishida, T. Nakagawa, H. Takano, and H. Aoi, "Signal processing and coding techniques for perpendicular recording," *Joint NAPMRC 2003. Digest of Technical Papers [Perpendicular Magnetic Recording Conference 2003]*, p. 8, Jan. 2003.
- [12] M. Hashimoto, K. Miura, H. Muraoka, H. Aoi, and Y. Nakamura, "Influence of magnetic cluster-size distribution on signal-to-noise ratio in perpendicular magnetic recording media," *IEEE Trans. Magn.*, vol. 40, no. 4, pp. 2458–2460, July 2004.
- [13] B. Wilson, "Signal processing for perpendicular recording," *Joint NAPMRC 2003. Digest of Technical Papers [Perpendicular Magnetic Recording Conference 2003]*, p. 9, Jan. 2003.
- [14] H.-S. Lee, J. Bain, S.-Y. Hong, and H. Lee, "Influence of polarity of media DC background on write/read performance in perpendicular media with an antiferromagnetically stabilized soft underlayer," *IEEE Trans. Magn.*, vol. 41, no. 10, pp. 3157–3159, Oct. 2005.
- [15] P. Dhagat, D. Palmer, and B. Xu, "Neighborhood induced reader bias shift in perpendicular recording," *Journal of Applied Physics*, vol. 93, no. 10, pp. 7729–7731, May 2003.
- [16] W. Tan and J. Cruz, "Signal processing for perpendicular recording channels with intertrack interference," *IEEE Trans. Magn.*, vol. 41, no. 2, pp. 730–735, Feb. 2005.

- [17] M. Erden and E. Kurtas, "Thermal asperity detection and cancellation in perpendicular magnetic recording systems," *IEEE Trans. Magn.*, vol. 40, no. 3, pp. 1732–1737, May 2004.
- [18] G. Mathew and I. Tjhia, "Thermal asperity suppression in perpendicular recording channels," *IEEE Trans. Magn.*, vol. 41, no. 10, pp. 2878–2880, Oct. 2005.
- [19] J. G. Proakis, "Partial response equalization with application to high density magnetic recording channels," in *CRC Handbook for Coding and Signal Processing for Recording Systems*, B. Vasic and E. M. Kurtas, Eds. CRC Press, 2004.
- [20] K. A. S. Immink, P. H. Siegel, and J. K. Wolf, "Codes for digital recorders," *IEEE Trans. Inf. Theory*, vol. 44, no. 6, pp. 2260 – 2299, Oct. 1998.
- [21] J. Coker, E. Eleftheriou, R. Galbraith, and W. Hirt, "Noise-predictive maximum likelihood (NPML) detection," *IEEE Trans. Magn.*, vol. 34, no. 1, pp. 110–117, Jan 1998.
- [22] P. Kovintavewat, I. Ozgunes, E. Kurtas, J. R. Barry, and S. W. McLaughlin, "Generalized partial-response targets for perpendicular recording with jitter noise," *IEEE Trans. Magn.*, vol. 38, no. 5, 2002.
- [23] J. Moon and W. Zeng, "Equalization for maximum likelihood detectors," *IEEE Trans. Magn.*, vol. 31, no. 2, pp. 1083–1088, Mar. 1995.
- [24] H. Yang and G. Mathew, "Joint design of optimum partial response target and equalizer for recording channels with jitter noise," *IEEE Trans. Magn.*, no. 1, pp. 70–77, Jan. 2006.
- [25] P. M. Aziz, "Adaptive equalization architectures for partial response channels," in *CRC Handbook for Coding and Signal Processing for Recording Systems*, B. Vasic and E. M. Kurtas, Eds. CRC Press, 2004.
- [26] C. Schlegel and L. Perez, *Trellis and Turbo Coding*. Wiley-IEEE Press, 2003, ch. 7.
- [27] A. Kavčić and J. M. Moura, "The Viterbi algorithm and Markov noise memory," *IEEE Trans. Inf. Theory*, vol. 46, no. 1, pp. 291–301, Jan. 2000.
- [28] J. Moon and J. Park, "Pattern-dependent noise prediction in signal-dependent noise," *IEEE J. Sel. Areas Commun.*, vol. 19, no. 4, pp. 730–743, Apr. 2001.
- [29] A. Fahrner, "On signal processing and coding for digital magnetic recording system," Ph.D. dissertation, Department of Telecommunications and Applied Information Theory, University of Ulm, Germany, 2004.

- [30] S. Aviran, “Constrained coding and signal processing for data storage system,” Ph.D. dissertation, Department of Electrical and Computer Engineering, University of California, San Diego, 2006.
- [31] P. Chaichanavong and P. H. Siegel, “Tensor-product parity code for magnetic recording,” *IEEE Trans. Magn.*, vol. 42, no. 2, pp. 350–352, Feb. 2006.
- [32] J. Park and J. Moon, “High-rate error-correction codes targeting dominant error patterns,” *IEEE Trans. Magn.*, vol. 42, no. 10, pp. 2573–2575, Oct. 2006.
- [33] K. Abdel-Ghaffar and M. Hassner, “Multilevel error-control codes for data storage channels,” *IEEE Trans. Inf. Theory*, vol. 37, no. 3, pp. 735–741, May 1991.
- [34] J. B. Soriaga, “On near-capacity code design for partial response channels,” Ph.D. dissertation, Department of Electrical and Computer Engineering, University of California, San Diego, 2005.
- [35] W. E. Ryan, “An introduction to LDPC codes,” in *CRC Handbook for Coding and Signal Processing for Recording Systems*, B. Vasic and E. M. Kurtas, Eds. CRC Press, 2004.
- [36] J. Fan and A. Calderbank, “A modified concatenated coding scheme, with applications to magnetic data storage,” *IEEE Trans. Inf. Theory*, vol. 44, no. 4, pp. 1565–1574, Jul 1998.
- [37] P. Chaichanavong and P. Siegel, “Tensor-product parity codes: combination with constrained codes and application to perpendicular recording,” *IEEE Trans. Magn.*, vol. 42, no. 2, pp. 214–219, Feb. 2006.
- [38] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, “Optimal decoding of linear codes for minimizing symbol error rate,” *IEEE Trans. Inf. Theory*, vol. 2, pp. 284–287, Mar. 1974.
- [39] J. Hagenauer and P. Hoeher, “A Viterbi algorithm with soft-decision outputs and its applications,” *Global Telecommunications Conference, 1989, and Exhibition. Communications Technology for the 1990s and Beyond. GLOBECOM '89*, IEEE, pp. 1680–1686 vol.3, Nov 1989.
- [40] T. Souvignier, M. Oberg, P. Siegel, R. Swanson, and J. Wolf, “Turbo decoding for partial response channels,” *Communications, IEEE Transactions on*, vol. 48, no. 8, pp. 1297–1308, Aug 2000.
- [41] J. L. Fan, A. Friedmann, E. Kurtas, and S. McLaughlin, “Low density parity check codes for magnetic recording,” in *Proceedings, Allerton Conference*, Allerton, IL, 1999.

- [42] W. Ryan, "Performance of high rate turbo codes on a PR4-equalized magnetic recording channel," *Communications, 1998. ICC 98. Conference Record.1998 IEEE International Conference on*, vol. 2, pp. 947–951 vol.2, Jun 1998.
- [43] R. Koetter and A. Vardy, "Algebraic soft-decision decoding of Reed-Solomon codes," *IEEE Trans. Inf. Theory*, vol. 49, pp. 2809–2825, Nov. 2003.
- [44] J. Jiang and K. Narayanan, "Iterative soft-input soft-output decoding of Reed-Solomon codes by adapting the parity-check matrix," *IEEE Trans. Inf. Theory*, vol. 52, no. 8, pp. 3746–3756, Aug. 2006.

Chapter 2

Design Curves and Information-Theoretic Limits for Perpendicular Recording Systems

The optimization of high density magnetic recording systems requires the analysis and evaluation of tradeoffs among the many system parameters. For example, code rate optimization for different channel models has been addressed in several recent studies [1, 2, 3]. In particular, Chaichanavong et al. [1] proposed a quasi-analytical methodology to optimize the system for a channel model with three parameters, the replay pulse width T_{50} , the transition jitter noise standard deviation σ_J , and the signal-to-electronic-noise ratio SNR_W . The signal processing scheme included an MMSE equalizer with a unit energy constraint, a Viterbi detector, and a Reed-Solomon (RS) error-correcting code. For a specified user bit spacing B_{user} and SNR_W , the results in [1] take the form of a design curve that defines the acceptable (T_{50}, σ_J) pairs for a specified sector error rate (SER). Each point on the curve has a corresponding optimal code rate, R , and an optimal channel bit spacing $B = RB_{\text{user}}$.

In this chapter, we will present the design curves for systems using coding and equalization schemes different from those used in the system in [1]. The same channel model will be considered. Then we will apply information-theoretic tools to derive de-

sign curves that will shed light on the acceptable region of (T_{50}, σ_J) pairs when optimal detection and linear coding techniques are used.

2.1 Design Curves for Perpendicular Recording Systems

2.1.1 Perpendicular Recording System with Jitter-Noise and AWGN

In this section, we will introduce the channel model we are going to use.

Assume the channel isolated transition response to be

$$s(t) = V_{\max} \operatorname{erf} \left(\frac{0.954t}{T_{50}} \right), \quad (2.1)$$

where $\operatorname{erf}(\cdot)$ is the error function defined by

$$\operatorname{erf}(t) = \frac{2}{\sqrt{\pi}} \int_0^t e^{-u^2} du. \quad (2.2)$$

The constant 0.954 comes from the definition of T_{50} , which is the width of the transition response at half of the maximum amplitude, i.e., $s(T_{50}/2) = V_{\max}/2$. The channel dipulse response is defined as

$$h(t) = [s(t) - s(t - B)]/2, \quad (2.3)$$

where B is the channel bit spacing.

The channel input is a binary sequence $\{x_i\}$, where $x_i \in \{+1, -1\}$. The noiseless channel output $y(t)$ can be written as the convolution of the channel input and the dipulse response

$$y(t) = \sum_i x_i h(t - iB). \quad (2.4)$$

The jitter noise is modeled as

$$n_J(t) = \sum_i a_i \left(\frac{x_i - x_{i-1}}{2} \right) s'(t - iB) \quad (2.5)$$

where the $\{a_i\}$ are independent Gaussian random variables with zero mean and variance σ_J^2 and $s'(t)$ is the first derivative of the transition response $s(t)$.

Electronic noise and a small amount of stationary medium noise, not included in the jitter noise, can be modeled as AWGN represented by $n_W(t)$. Therefore, the output of the channel can be written as

$$z(t) = y(t) + n_J(t) + n_W(t). \quad (2.6)$$

After low-pass filtering and sampling at the channel bit spacing B , the discrete-time channel output is thus

$$z_k = \sum_i h_i x_{k-i} + \sum_i a_{k-i} \left(\frac{x_{k-i} - x_{k-i-1}}{2} \right) s'_i + n_W(kB) \quad (2.7)$$

where h_i is the sample of $h(t)$ at iB , s'_i is the sample of $s'(t)$ at iB . The signal-to-electronic-noise ratio is defined by $SNR_W = 10 \log_{10}(V_{\max}^2/\sigma_W^2)$, where σ_W^2 is the variance of the Gaussian random variable $n_W(kB)$. When SNR_W is fixed, the channel depends only on the normalized parameters T_{50}/B and σ_J/B .

2.1.2 Parameter Optimization and Design curves

In a recording system, we would like to store as much user data as possible per unit area. However, during the recording process, various types of overhead will be added to the user data, such as the parity check bits for the error correcting code (ECC), the extra bits inserted for the modulation codes and so on. In this chapter, we assume a simplified recording system as shown in Figure 2.1, where the ECC code is considered as the only source of overhead. For such a system, the user bit spacing is defined as $B_{\text{user}} = B/R$, where R is the code rate for the error correcting code. The user density is thus $1/B_{\text{user}}$.

Raising the channel bit density does not necessarily increase the user density. Higher channel density will result in more ISI and higher normalized jitter noise. Therefore, a stronger ECC code, which usually has lower code rate, is needed to ensure the system performance, usually measured by the sector error rate (SER). The user density might not increase in this process because of the amount of overhead required.

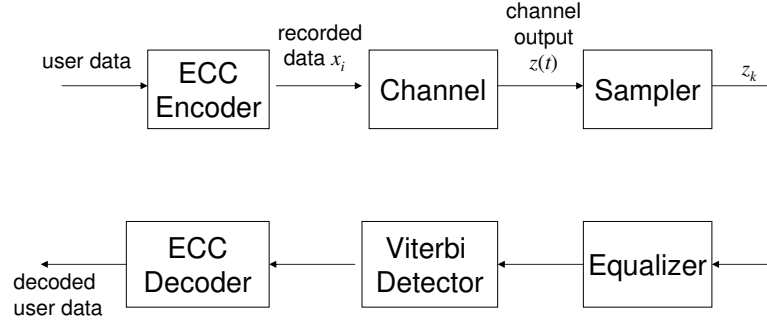


Figure 2.1 The system diagram for design curve computation.

For example, Figure 2.2 shows a curve of the user density $1/B_{\text{user}}$ verses the channel density $1/B$ for a given channel. The ECC code is a Reed-Solomon code with information length 4100 bits and 10-bit symbols. The system utilized an MMSE equalizer design with unit energy constraint [4] and a conventional Viterbi detector. T_{50} is set to be 20nm, σ_J is set to be 1nm and SNR_W is 20dB. For each channel bit spacing value, we used the block multinomial model [5] to estimate the SER for different ECC code rate. The maximum code rate that ensures a target SER is chosen to calculate B_{user} which is B/R . We can see that the higher the channel density $1/B$ is, the lower the code rate goes. However, the user density is not a monotonic function of the channel density. When $1/B = 0.064$, the maximum user density is achieved.

A very natural system optimization question thus arises. For a given channel with parameters $(T_{50}, \sigma_J, SNR_W)$, what is the proper channel bit spacing and the ECC code rate that maximize the user density, or minimize the user bit spacing B_{user} ? The answer for different channel models has been addressed in several recent studies [1, 2, 3]. Besides this question, the design curves in [1] also define the acceptable (T_{50}, σ_J) pairs for a specified sector error rate (SER), given the user bit spacing B_{user} and SNR_W . Here, we will briefly review the methodology to calculate the design curves proposed in [1].

As we have shown in Figure 2.2, for a given set of channel parameters $(T_{50},$

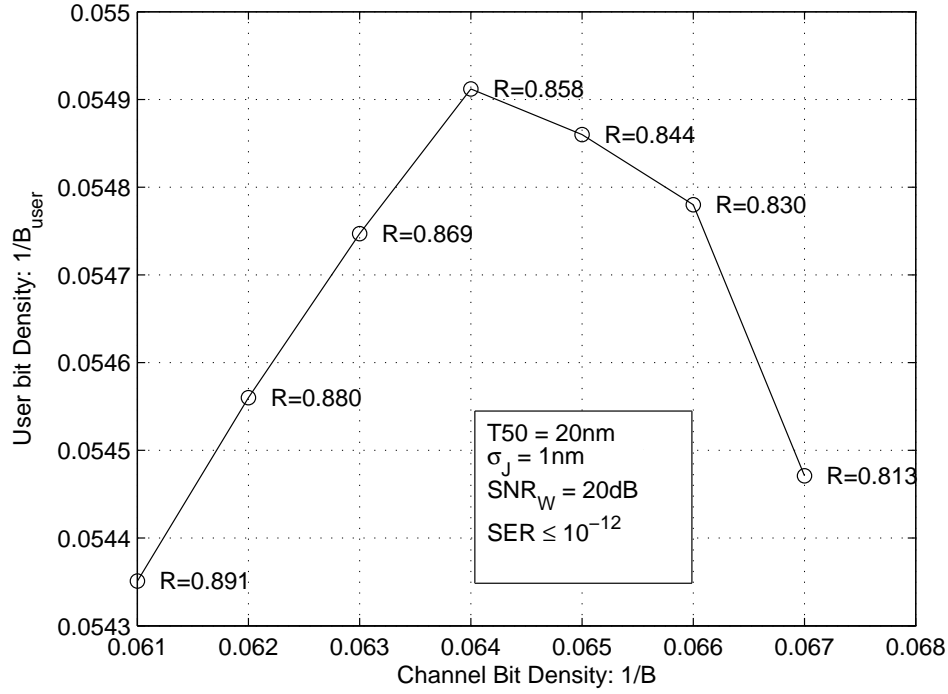


Figure 2.2 An example of the user density optimization. $T_{50}=20\text{nm}$; $\sigma_J=1\text{nm}$; $SNR_W=20\text{dB}$; SER for each point is no greater than 10^{-12} . A MMSE unit energy equalizer design is used. The ECC code is a Reed-Solomon code with information length 4100 bits and 10-bit symbols.

σ_J, SNT_W), there exists an optimal B and a corresponding code rate R that maximize the user density. Since the channel model we used depends only on the normalized parameters T_{50}/B and σ_J/B when the SNR_W is fixed, by changing both T_{50}/B and σ_J/B while keeping the ratio T_{50}/σ_J constant, we can obtain the code rates that meet the requirement of SER for all the $(T_{50}/B, \sigma_J/B)$ pairs. Among these code rates, the optimal code rate R is the code rate that maximizes the T_{50}/B_{user} . It also maximizes the σ_J/B_{user} since the ration of T_{50}/σ_J is fixed. Thus, a point on the normalized design curve can be drawn in the $(T_{50}/B_{\text{user}}, \sigma_J/B_{\text{user}})$ -plane for a given SNR_W . By connecting those points for the same SNR_W of different T_{50}/σ_J ratio, as shown in Figure 2.3(a), a normalized design curve is obtained. For any given B_{user} requirement, for example, 20nm in Figure 2.3(b), a design curve in (T_{50}, σ_J) -plane can be created from the corre-

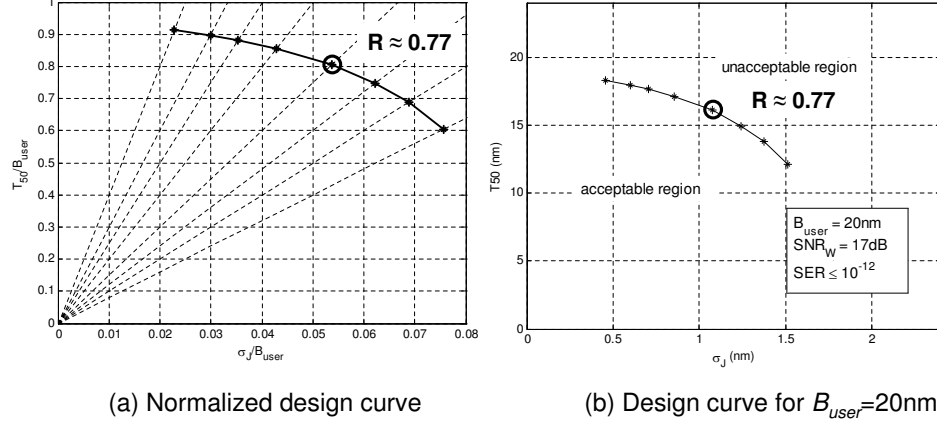


Figure 2.3 Illustration of the normalized design curve and the design curve in the (T_{50}, σ_J) -plane.

sponding normalized design curve of the same SNR_W . The code rates on the curve are the same as those on the normalized curve of the same T_{50}/σ_J ratio. From this design curve, the acceptable region of (T_{50}, σ_J) pair that can achieve the given user density and SER is identified by the region under the curve. For any given pair of (T_{50}, σ_J) and a fixed SNR_W , we are also able to determine the maximum user density we can achieve and the code rate we should choose according to the normalized design curves.

2.1.3 Design Curves of Different Coding and Equalization Schemes

The design curves are different for systems with different signal processing methods and ECC coding schemes. As we mentioned before, the system in [1] includes an MMSE equalizer of unit energy design, the conventional Viterbi detector and the RS code. In this section, we will show the simulation results for two systems that differ from the system used in [1].

In the first system, we modified the constraint used in the MMSE equalizer design [4]. The monic constraint, which sets the first tap of the equalization target to be

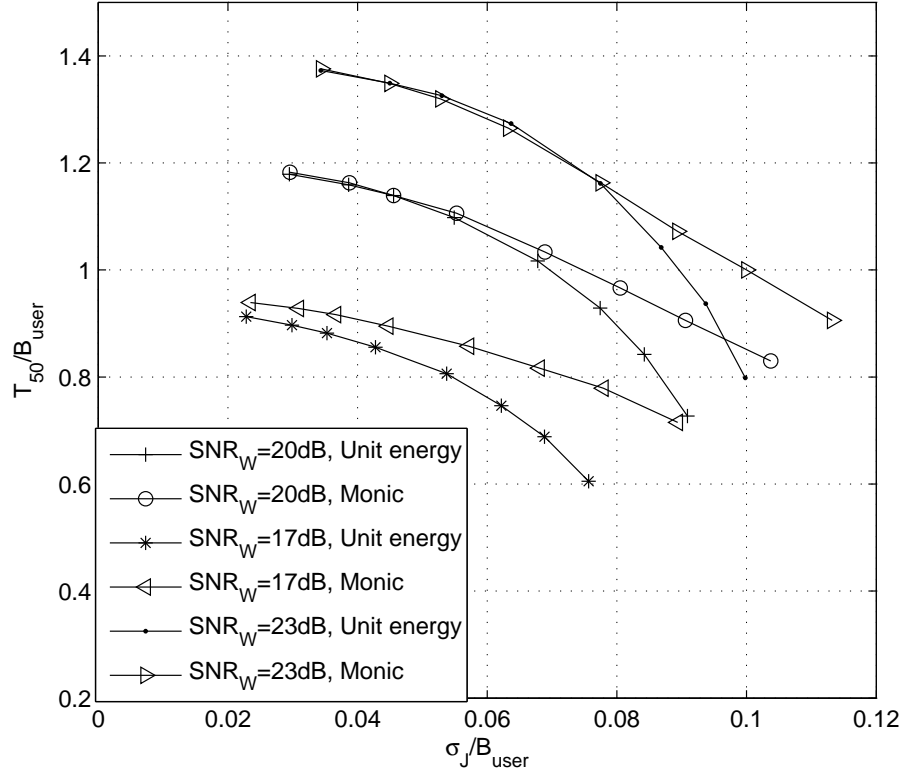


Figure 2.4 Comparison of the normalized design curves for MMSE equalizers with unit energy constraint and monic constraint. SER is required to be no greater than 10^{-12} . RS code with 4100 information symbols and 10 bits each symbol is used.

1, was shown to perform better than the unit energy constraint, which sets the energy, i.e., the squared sum of the target taps, to 1. The design curves for a system using an MMSE equalizer with the monic constraint are compared with those using the unit energy constraint in Figure 2.4. The coding scheme is the same RS code as in [1], which has 410 information symbols per codeword and 10 bits per symbol. We can see that the design curves for the monic constraint do not curve downward as much as those for the unit energy constraint. It means that the monic constraint has a better performance especially when the jitter noise is severe.

The second system used a different coding scheme proposed in [6]. The coding

scheme is a combination of the tensor product parity (TPP) code with the RS code. The TPP code will be further discussed in Chapter 6. In this system, a single parity check bit is calculated for every 10 bits in the sector. These parity check bits form a codeword of a BCH code. The RS code is still the same RS code with 10 bits per symbol and 410 information symbols per codeword. A brief diagram of the encoding and decoding process proposed in [6] is shown in Figure 2.5.

Now we need to find the optimal code rate for such a combined coding scheme that satisfies the SER requirement.

The SER for such a system is

$$\text{SER} = P_{\text{BCH}} + (1 - P_{\text{BCH}}) \times P_{\text{RS}} \approx P_{\text{BCH}} + P_{\text{RS}} \quad (2.8)$$

where P_{BCH} is the BCH code decoding failure rate and P_{RS} is the RS code decoding failure rate, as marked in Figure 2.5. The approximation is appropriate for a good channel where P_{BCH} is very small.

The total set of parity bits includes those from both the TPP code and the RS code. For a fixed number of parity bits, the most efficient way to distribute them among the BCH code and the RS code is such that $P_{\text{BCH}} \approx P_{\text{RS}}$, because if any one of the two codes fails to decode, the whole decoding process fails. Therefore, if the target SER is 10^{-12} , the requirement on P_{BCH} and P_{RS} is that both need to be no greater than 5×10^{-13} . To estimate P_{BCH} and P_{RS} , we used the block multinomial model for both the BCH code and the RS code. However, the block multinomial model can only give us the required error correcting ability of the code. For the RS code, the number of parity symbols required for correcting t errors is $2t$. But the relationship between the number of parity bits and the number of correctable errors for the BCH code is not obvious. Therefore, for the BCH code, we set up look-up tables for the number of errors and the corresponding parity bits. The length of the BCH code is determined by the total number of the parity bits. If the length of BCH code is smaller than 511, we will use the look-up table for shortened code of length-511 BCH code, and if the length is larger than 511, we will use the look-up table for shortened code of length-1023 BCH code. In this way, the number of parity bits required for each code can be calculated and the total

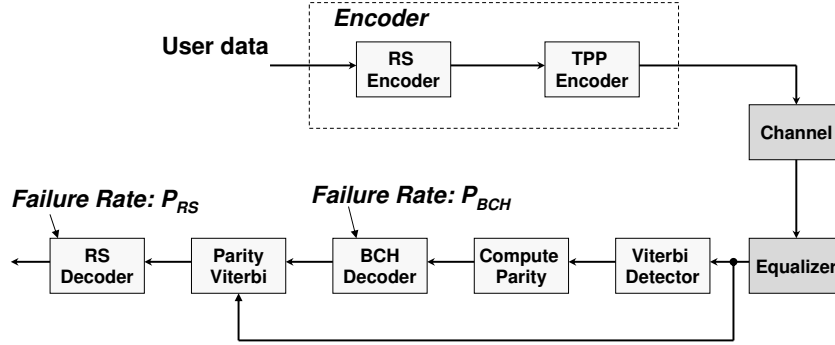


Figure 2.5 The system with tensor product parity codes.

code rate can be obtained.

The design curves for the TPP code plus the RS code are shown in Figure 2.6. We can see that this coding scheme actually does not always perform better than the RS code alone. The design curves of these two coding schemes cross each other for SNR_W larger than 17dB. The TPP plus RS coding scheme can handle more jitter noise while the RS code alone is better suited for channels with more ISI.

2.2 Information-Theoretic Limit of the Perpendicular Recording System

Information theory tells us that there exist coding schemes that provide an arbitrarily small error rate if the rate of the code is smaller than the capacity of the channel. Intuitively, we expect that if we use the maximum mutual information rate instead of the rate of a specific code, the acceptable region for the parameters will be extended to the limit. From the information-theoretical perspective, the previous optimization questions for a specific coding scheme becomes for any detection and linear coding scheme:

- Given T_{50} , σ_J , and SNR_W , what is the maximum achievable user areal density?

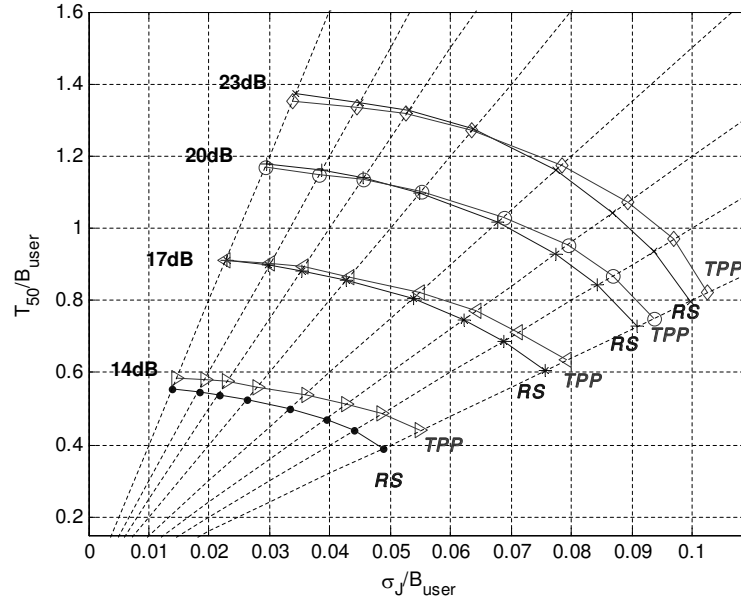


Figure 2.6 Comparison of the normalized design curves for TPP+RS scheme and the RS only scheme. The curves marked with *TPP* is for the TPP+RS scheme and the curves marked with *RS* is for the RS code only system.

- For a given areal density, what would be an acceptable set of values for T_{50} , σ_J , and SNR_W ?

There exists substantial literature addressing the computation of information rates and capacity for magnetic recording channels [7, 8, 9, 10]. As an application of these techniques, Ryan et al. [11] determined the symmetric information rate (SIR) for a Lorentzian channel corrupted by AWGN, and then used this result to maximize the linear and areal user density. In this chapter, for the perpendicular recording channel we modeled previously, we will calculate the upper bounds on the SIR and apply it to form the information-theoretic design curves that, for a given user linear density, provide insight into the tradeoff among the three parameters T_{50} , σ_J , and SNR_W . The results will be compared to those of the RS-coded system described in [1], and they show the possibility of obtaining considerable performance improvement with the help of advanced

detection and coding techniques.

2.2.1 Monte-Carlo Estimation of Information Rate

Before we go into the details of estimating the information rate, let us review the channel model based on equation (2.7). Since the sampled dipulse response and the first order derivative of the transition response vanish when i is far away from zero, we assume $h_i \approx 0$ when $i < -A_1$ and $i > A_2$ where $A_1, A_2 > 0$. Similarly, we assume that $s'_i \approx 0$ when $i < -D_1$ and $i > D_2$, where $D_1, D_2 > 0$. The channel is therefore modeled as a finite-memory intersymbol interference (ISI) channel with data-dependent, finite-memory noise and AWGN. We can write the discrete-time channel output as follows:

$$z_k = \sum_{i=-A_1}^{A_2} h_i x_{k-i} + \sum_{i=-D_1}^{D_2} a_{k-i} d_{k-i} s'_i + w_k \quad (2.9)$$

where $d_{k-i} = \frac{x_{k-i} - x_{k-i-1}}{2}$ and $w_k = n_W(kB)$.

The symmetric information rate (SIR) of a channel is the mutual information rate corresponding to independent, identically distributed, equiprobable binary inputs. The SIR is generally interpreted as the maximum achievable rate for which reliable recording is possible using a binary, linear code. For the channel described above, we use the SIR to examine the tradeoffs among system design parameters assuming optimal detection and linear coding. In this section, the method we used to evaluate the SIR is introduced.

In [7] and [8], independently, a simulation-based method to evaluate the information rate of a Markovian channel was proposed. This method was extended in [9, 10] to evaluate the information rate and the capacity for the magnetic recording channel, assuming linear ISI and data-dependent, colored Gaussian noise. We use a similar method in this section.

In the following analysis, we denote a column vector $(x_i, x_{i+1}, \dots, x_j)^T$ by \bar{x}_i^j . The information rate between the channel input X and the channel output Z can be written as

$$I(X; Z) = \lim_{n \rightarrow \infty} \frac{1}{n} I(\bar{x}_1^n; \bar{z}_1^n) = h(Z) - h(Z|X) \quad (2.10)$$

where $h(Z) = \lim_{n \rightarrow \infty} (1/n) h(\bar{z}_1^n)$ is the entropy rate of the channel output and $h(Z|X) = \lim_{n \rightarrow \infty} (1/n) h(\bar{z}_1^n | \bar{x}_1^n)$ is the entropy rate of the channel output conditioned on the channel input. We proceed to evaluate these two values separately.

According to [7], $-(1/n) \log \Pr(\bar{z}_1^n)$ converges to $h(Z)$ with probability one. Since we are considering a finite-memory Markovian channel, it can be computed by the forward recursion of the BCJR algorithm [12]. Let S_k be the trellis state at time k . Then the probability of the channel output sequence is

$$\Pr(\bar{z}_1^n) = \sum_{m \in \Omega} \Pr(S_n = m, \bar{z}_1^n) \quad (2.11)$$

where Ω is the set of all states.

Define the forward state metric as

$$\alpha_k(m) = \Pr(S_k = m, \bar{z}_1^k). \quad (2.12)$$

Using Bayes' rule

$$\begin{aligned} \alpha_{k+1}(m) &= \sum_{m' \in \Omega} \Pr(S_k = m', S_{k+1} = m, \bar{z}_1^k, z_{k+1}) \\ &= \sum_{m' \in \Omega} \Pr(S_k = m', \bar{z}_1^k) \Pr(z_{k+1}, S_{k+1} = m | S_k = m', \bar{z}_1^k). \end{aligned} \quad (2.13)$$

If we define the branch metric to be

$$\gamma_k(m', m) = \Pr(z_k, S_k = m | S_{k-1} = m', \bar{z}_1^{k-1}) \quad (2.14)$$

then we have the following recursion:

$$\alpha_{k+1}(m) = \sum_{m' \in \Omega} \alpha_k(m') \gamma_{k+1}(m', m). \quad (2.15)$$

Since $\alpha_k(m)$ is very close to zero when k is large, we normalized the sum of all forward-state metrics at each time k to be 1 in order to avoid losing precision in the computation. Therefore, denoting the normalized forward state metric at time k and state m by $\tilde{\alpha}_k(m)$, and the normalization factor by λ_k , the new recursion takes the form

$$\tilde{\alpha}_{k+1}(m) = \lambda_{k+1} \sum_{m' \in \Omega} \tilde{\alpha}_k(m') \gamma_{k+1}(m', m). \quad (2.16)$$

Setting the same initial values for recursions (2.15) and (2.16), i.e., let $\tilde{\alpha}_0(m) = \alpha_0(m)$, we can show that

$$\tilde{\alpha}_k(m) = \left(\prod_{i=1}^k \lambda_i \right) \alpha_k(m). \quad (2.17)$$

Thus

$$\begin{aligned} -\frac{1}{n} \log \Pr(\bar{z}_1^n) &= -\frac{1}{n} \log \left[\sum_{m \in \Omega} \alpha_n(m) \right] \\ &= -\frac{1}{n} \log \left[\frac{\sum_{m \in \Omega} \tilde{\alpha}_n(m)}{\prod_{i=1}^n \lambda_i} \right] = \frac{1}{n} \sum_{i=1}^n \log(\lambda_i). \end{aligned} \quad (2.18)$$

The computation of the branch metric depends on the channel model. According to the derivations in the Appendix, the branch metric is

$$\gamma_k(m', m) = c_1 \exp \left\{ c_2 \left[(\bar{z}_{k-L}^k - \bar{y}_{k-L}^k)^T \bar{\omega} \right]^2 \right\} \quad (2.19)$$

where c_1 and c_2 are edge-dependent constants and $\bar{\omega}$ is an edge-dependent column vector, as defined in (2.35)-(2.33). Here, L is the memory of the jitter noise, defined as $L = D_1 + D_2$.

Using the branch metrics, the initial values for the forward state metrics, and the recursion (2.16), we can compute the probability of a long channel output realization, and thus get an estimate of the output entropy rate.

Computing the conditional entropy rate $h(Z|X)$ is quite straightforward for this channel model [9]. The conditional entropy rate is

$$h(Z|X) = \lim_{n \rightarrow \infty} \frac{1}{n} E[h(\bar{z}_1^n | \bar{x}_1^n = \bar{x}^*)], \quad (2.20)$$

where \bar{x}^* is a realization of the input sequence \bar{x}_1^n and the expectation is over all possible input sequences.

Since we are considering the symmetric information rate, we assume the input sequences are equiprobable. For a random input sequence realization \bar{x}^* , $(1/n) h(\bar{z}_1^n | \bar{x}_1^n = \bar{x}^*)$ converges to $h(Z|X)$ when n goes to infinity. With the knowledge that

the channel output sequence obeys a joint Gaussian distribution when conditioned on a specific channel input sequence, the conditional entropy rate is

$$h(Z|X) \approx \frac{1}{2} \log(2\pi e |\mathbf{R}_{\bar{x}^*}|) \quad (2.21)$$

where $\mathbf{R}_{\bar{x}^*}$ is the covariance matrix of the output sequence, given the input sequence \bar{x}^* . The length of the input sequence should be large for this approximation to be valid.

2.2.2 Reducing computation complexity

The number of states in the trellis grows exponentially with the sum of the memory length of the channel ISI and the memory length of the jitter noise. Thus, if these quantities are large, the complexity in computing the SIR is dominated by the channel output entropy rate calculation.

One method to reduce the computational complexity is to reduce the number of states in the trellis by truncating the channel dipulse response, $\{h_i\}$, and the sampled first derivative of $s(t)$, $\{s'_i\}$. In [9], it is stated that if we use an approximation of the real channel in the BCJR forward recursion, the computed output entropy rate is an upper bound on the true value of $h(Z)$. However, although this method can reduce the computational complexity significantly, it generates a loose upper bound.

Another method is to try to reduce the number of branch metric computations in the recursion. Recall that, for each state, the new state metric is obtained by calculating the branch metric for each incoming edge, multiplying each of these by the state metric of the corresponding state from the previous stage, and then summing up these products. However, the normalized forward-state metric $\tilde{\alpha}_k(m')$, which equals the conditional probability $\Pr(S_k = m' | \bar{z}_1^k)$, may be very close to zero if the current state m is unlikely to be the actual state at time k , given the past channel output sequence. Since the branch metric is a multivariate Gaussian density function, which is bounded, we can ignore terms in the summation (2.16) corresponding to sufficiently small values of $\tilde{\alpha}_k(m')$. Therefore, we set a threshold value ε , and if the forward-state metric $\tilde{\alpha}_k(m')$ is smaller than ε , it is set to zero and the branch metrics on the edges starting from state

m' are not calculated. This approximation results in forward state metrics, and therefore estimated output sequence probabilities, that are smaller than their true values. It follows that the computed estimate of the output entropy rate is an upper bound for the true value.

Simulations confirmed that this method significantly reduces the required amount of computation, while producing a tight upper bound on the output entropy rate. For example, for a 512-state trellis and a threshold value $\varepsilon = 10^{-3}$, the computation time consumed using this approximation is about one-fifth of that required by the exact calculation. We observed that the number of non-zero forward state metrics was reduced to no more than 100 after a few steps, and sometimes dropped to less than 10. At the same time, the entropy rate upper bound differed only slightly from the exact value. Therefore, in the simulations used to generate the design curves, we chose a reasonably large number of states, namely 2^{13} , and applied this approximation method to compute our upper bounds on the SIR.

2.3 Information-Theoretic Limit of the Design Curves

For a specified user density and SNR_W , the information-theoretic design curves computed in this paper essentially determine the region of (T_{50}, σ_J) pairs that are acceptable in a system utilizing a code that achieves the SIR.

As in [1], we simplify the calculation of the design curves by normalizing with respect to the user bit spacing, defined by $B_{\text{user}} = B/R$, where the code rate R is set equal to the SIR. The method for deriving the normalized curve is modeled after the approach used in [1]. Specifically, for a given SNR_W and various values of the ratio T_{50}/σ_J , we first compute the SIR bound as a function of σ_J/B . We then determine the corresponding values of $T_{50}/B_{\text{user}} = RT_{50}/B$, where R is the calculated SIR bound.

Figures 2.7 and 2.8 illustrate the numerical results for these two steps for the case where $SNR_W=17\text{dB}$. Note that for each curve in Figure 2.8, there is a point representing a maximum value of T_{50}/B_{user} . The corresponding points in the $(T_{50}/B_{\text{user}}, \sigma_J/B_{\text{user}})$ -

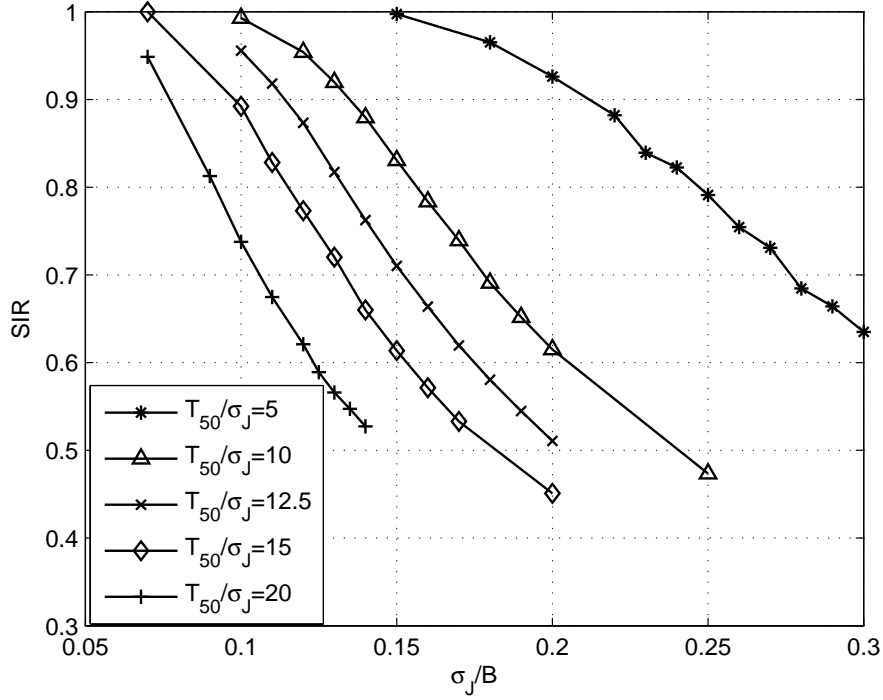


Figure 2.7 SIR curves v.s. jitter noise for $SNR_W=17\text{dB}$.

plane form the normalized information-theoretic design curve for $SNR_W=17\text{dB}$, shown in Figure 2.9 along with the normalized design curves for SNR_W values of 14dB and 20dB.

For purposes of comparison, Figure 2.9 also shows the normalized design curves derived in [1] for the RS-coded system at the same values of SNR_W . (We remark that, in order to facilitate the comparison to the curves taken from [1], the information-theoretic results plotted in Figure 2.9 were computed for different values of the ratio T_{50}/σ_J than those used to generate the curves in Figure 2.7 and Figure 2.8.)

It is clear that there is a significant gap between the curves representing the same SNR_W . Thus, we can infer that better detection methods and coding schemes might extend the acceptable region of system parameters considerably.

Also, it is interesting to note that the shapes of the two sets of curves are not the

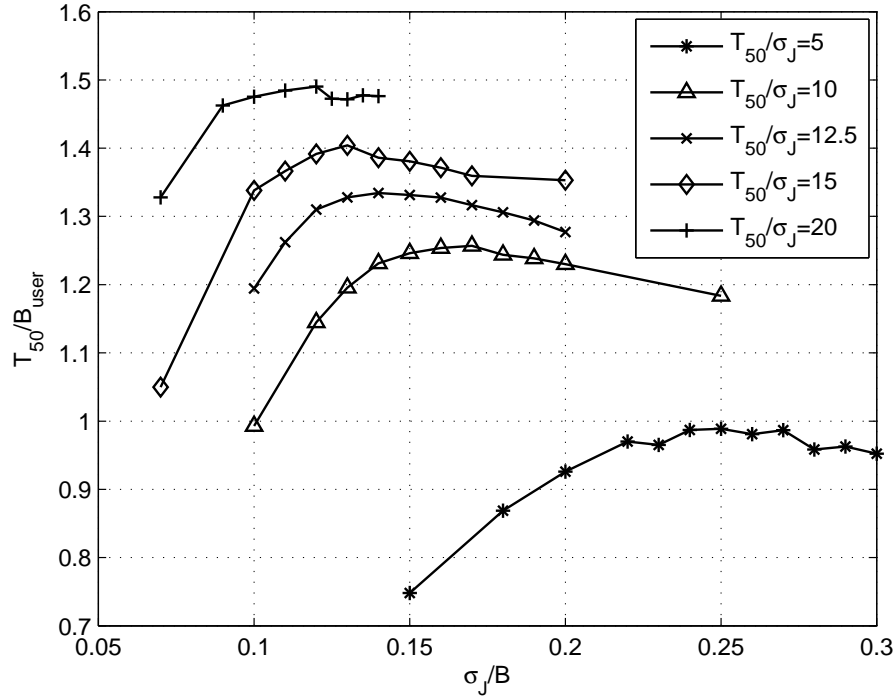


Figure 2.8 Normalized user density curves v.s. jitter noise for $SNR_W=17\text{dB}$.

same. This suggests that the tradeoff between the system parameters might be different for SIR-achieving codes than for the optimized RS codes examined in [1].

In Figure 2.10, we plot the design curves in the (T_{50}, σ_J) -plane for $SNR_W=17\text{dB}$ and for user bit spacing $B_{\text{user}}=20\text{nm}$. The region below the lower design curve is the acceptable region of (T_{50}, σ_J) pairs for the RS-coded system in [1]. The region above the upper design curve can be interpreted as (T_{50}, σ_J) pairs for which no detection method and linear coding scheme can guarantee reliable data retrieval. The region between these two curves gives insight into head and media parameters that are acceptable with sufficiently powerful detection and coding schemes.

The numerical results represented by the curves in Figure 2.9 are shown in Table 2.1 and Table 2.2. Table 2.1 gives the T_{50}/B_{user} values for points on the normalized design curves for both the RS-coded system and the information-theoretic limiting case.

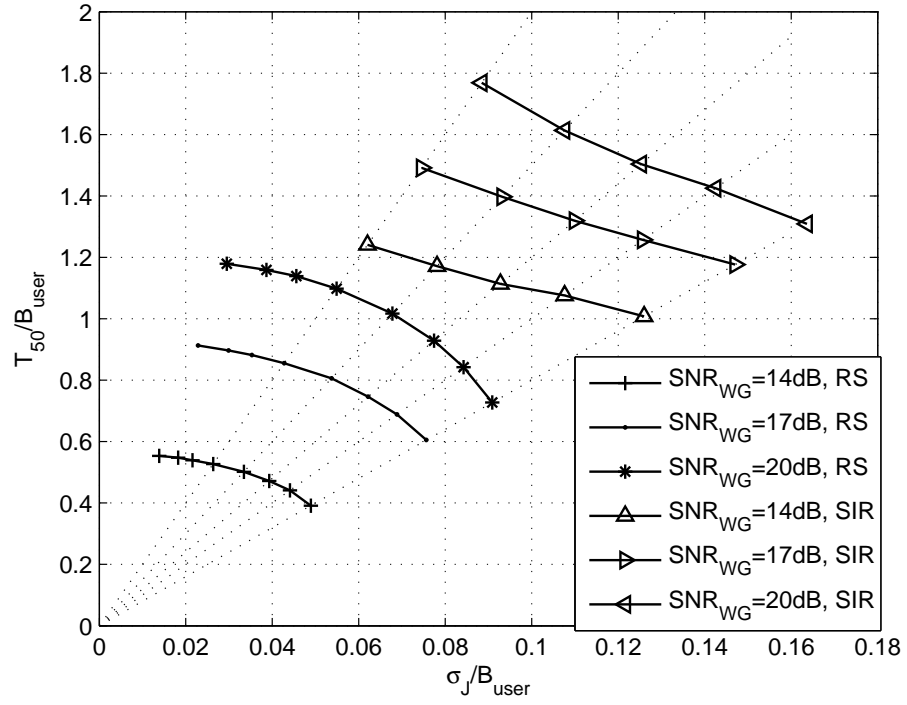


Figure 2.9 Normalized design curves for information-theoretic limit.

Table 2.1 Comparing T_{50}/B_{user} of the RS-coded system and SIR.

T_{50}/σ_J	RS-coded system			SIR		
	14dB	17dB	20dB	14dB	17dB	20dB
8	0.39	0.60	0.73	1.01	1.17	1.31
10	0.44	0.69	0.84	1.08	1.26	1.43
12	0.47	0.75	0.93	1.11	1.33	1.50
15	0.50	0.81	1.02	1.17	1.40	1.61
20	0.53	0.86	1.10	1.24	1.49	1.77

Table 2.2 shows the corresponding optimal RS code rates and the SIR upper bound values. Note that the SIR-achieving code almost doubles the user density T_{50}/B_{user} relative to the optimized RS code, whereas the optimal RS code rates and SIR upper bounds are relatively close.

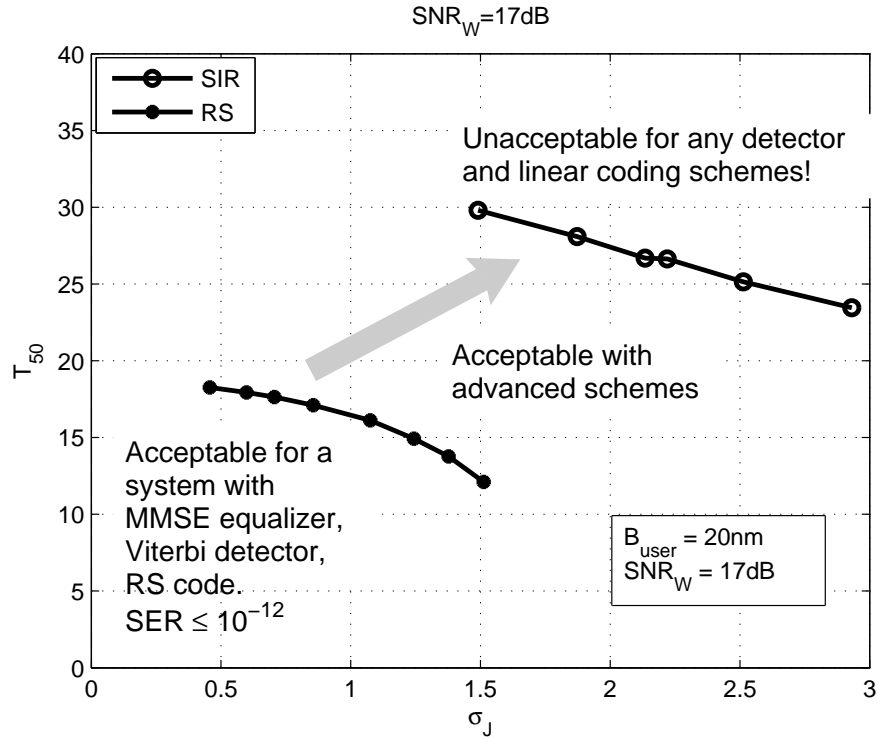


Figure 2.10 Comparison of the design curves for optimal RS code and SIR at $SNR_W = 17\text{dB}$, $B_{\text{user}} = 20\text{nm}$.

Table 2.2 Comparing code rate of the RS-coded system and SIR.

T_{50}/σ_J	RS-coded system			SIR		
	14dB	17dB	20dB	14dB	17dB	20dB
8	0.55	0.71	0.77	0.66	0.70	0.71
10	0.60	0.76	0.79	0.63	0.74	0.75
12	0.61	0.78	0.80	0.71	0.69	0.78
15	0.63	0.77	0.82	0.65	0.72	0.72
20	0.63	0.81	0.86	0.54	0.62	0.77

Acknowledgement

This chapter is partly a reprint of the paper Z. Wu, P. H. Siegel, H. N. Bertram, and J. K. Wolf, "Design curves and information-theoretic limits for perpendicular recording systems", *IEEE Trans. Magn.*, vol. 43, no. 2, pp. 721-726, Feb. 2007. The

dissertation author is the primary author of this paper.

Appendix: Derivation of the Branch Metric

Referring to the channel model in Section 2.2, we define $I = \max\{A_1, D_1\}$ and $T = \max\{A_2, D_2 + 1\}$. We also recall that $L = D_1 + D_2$, so the state of the channel is given by $S_k = \{x_{k-T-L+1}, \dots, x_{k+I}\}$. With these definitions, the Markovian channel satisfies the relations

$$\Pr(S_k | \bar{S}_1^{k-1}, \bar{z}_1^{k-1}) = \Pr(S_k | S_{k-1}) \quad (2.22)$$

and

$$\Pr(z_k | \bar{S}_1^m, \bar{z}_1^{k-1}) = \Pr(z_k | S_{k-1}, S_k, \bar{z}_{k-L}^{k-1}). \quad (2.23)$$

The branch metric defined in Section 2.2.1 can therefore be written as

$$\begin{aligned} \gamma_k(m', m) &= \Pr(z_k, S_k = m | S_{k-1} = m', \bar{z}_1^{k-1}) \\ &= \Pr(z_k | S_k = m, S_{k-1} = m', \bar{z}_1^{k-1}) \Pr(S_k = m | S_{k-1} = m', \bar{z}_1^{k-1}) \\ &= \Pr(z_k | S_k = m, S_{k-1} = m', \bar{z}_{k-L}^{k-1}) \Pr(S_k = m | S_{k-1} = m'). \end{aligned} \quad (2.24)$$

The transition probability from state S_{k-1} to state S_k is either 0, if there is no edge between them, or 0.5, assuming i.i.d. equiprobable binary channel inputs.

The channel output satisfies

$$\bar{z}_{k-L}^k = \bar{y}_{k-L}^k + \mathbf{Q} \bar{a}_{k-L-D_2}^{k+D_1} + \bar{w}_{k-L}^k, \quad (2.25)$$

where \mathbf{Q} is an $(L+1)$ by $(2L+1)$ matrix defined as

$$\mathbf{Q} = \begin{pmatrix} s'_{D_2} & s'_{D_2-1} & \cdots & s'_{-D_1} & 0 & \cdots & 0 \\ 0 & s'_{D_2} & s'_{D_2-1} & \cdots & s'_{-D_1} & 0 & \cdots \\ 0 & 0 & \cdots & \ddots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & s'_{D_2} & s'_{D_2-1} & \cdots & s'_{-D_1} \end{pmatrix} \begin{pmatrix} d_{k-L-D_2} & & \\ 0 & \ddots & 0 \\ & & d_{k+D_1} \end{pmatrix}, \quad (2.26)$$

where, $d_i = \frac{x_i - x_{i-1}}{2}$.

The conditional joint density function $\Pr(\bar{z}_{k-L}^k | S_{k-1}, S_k)$ is multivariate Gaussian with covariance matrix

$$\begin{aligned} \mathbf{R}^{(L)} &= E\{(\mathbf{Q}\bar{a}_{k-L-D_2}^{k+D_1} + \bar{w}_{k-L}^k)(\mathbf{Q}\bar{a}_{k-L-D_2}^{k+D_1} + \bar{w}_{k-L}^k)^T\} \\ &= \sigma_J^2 \mathbf{Q}\mathbf{Q}^T + \sigma_W^2 \mathbf{I}. \end{aligned} \quad (2.27)$$

Similarly, the conditional density of \bar{z}_{k-L}^{k-1} given S_{k-1} and S_k is multivariate Gaussian with covariance matrix $\mathbf{R}^{(L-1)}$ which is obtained from $\mathbf{R}^{(L)}$ by eliminating the last row and column; that is

$$\mathbf{R}^{(L)} = \begin{pmatrix} \mathbf{R}^{(L-1)} & \bar{r}^{(L)} \\ (\bar{r}^{(L)})^T & r_{LL} \end{pmatrix}, \quad (2.28)$$

Noting that

$$\Pr(z_k | S_k = m, S_{k-1} = m', \bar{z}_{k-L}^{k-1}) = \frac{\Pr(\bar{z}_{k-L}^k | S_k = m, S_{k-1} = m')}{\Pr(\bar{z}_{k-L}^{k-1} | S_k = m, S_{k-1} = m')}, \quad (2.29)$$

we can express the branch metric as

$$\begin{aligned} \gamma_k(m', m) &= \Pr(S_k = m | S_{k-1} = m') \\ &= \sqrt{\frac{|\mathbf{R}^{(L-1)}|}{2\pi|\mathbf{R}^{(L)}|}} \exp \left\{ -\frac{1}{2}(\bar{z}_{k-L}^k - \bar{y}_{k-L}^k)^T (\mathbf{R}^{(L)})^{-1} (\bar{z}_{k-L}^k - \bar{y}_{k-L}^k) \right. \\ &\quad \left. + \frac{1}{2}(\bar{z}_{k-L}^{k-1} - \bar{y}_{k-L}^{k-1})^T (\mathbf{R}^{(L-1)})^{-1} (\bar{z}_{k-L}^{k-1} - \bar{y}_{k-L}^{k-1}) \right\} \end{aligned} \quad (2.30)$$

According to the inversion relation ship of $\mathbf{R}^{(L)}$ and $\mathbf{R}^{(L-1)}$ [13],

$$(\mathbf{R}^{(L)})^{-1} = \begin{pmatrix} (\mathbf{R}^{(L-1)})^{-1} & 0 \\ 0 & 0 \end{pmatrix} - 2c_2 \bar{\omega} \bar{\omega}^T \quad (2.31)$$

where

$$c_2 = -\frac{1}{2[r_{LL} - (\bar{r}^{(L)})^T (\mathbf{R}^{(L-1)})^{-1} \bar{r}^{(L)}]} \quad (2.32)$$

$$\bar{\omega} = \begin{pmatrix} -(\mathbf{R}^{(L-1)})^{-1} \bar{r}^{(L)} \\ 1 \end{pmatrix} \quad (2.33)$$

it is not difficult to derive the branch metric to be

$$\gamma_k(m', m) = c_1 \exp \left\{ c_2 \left[(\bar{z}_{k-L}^k - \bar{y}_{k-L}^k)^T \bar{\omega} \right]^2 \right\} \quad (2.34)$$

where

$$c_1 = \begin{cases} \frac{1}{2} \sqrt{\frac{|\mathbf{R}^{(L-1)}|}{2\pi|\mathbf{R}^{(L)}|}}, & \text{if there is an edge from } S_{k-1} \text{ to } S_k \\ 0, & \text{otherwise} \end{cases} \quad (2.35)$$

All the three parameters c_1 , c_2 and $\bar{\omega}$ are edge-dependent.

Bibliography

- [1] P. Chaichanavong, H. N. Bertram, and P. H. Siegel, "Design parameter optimization for perpendicular magnetic recording systems," *IEEE Trans. Magn.*, vol. 42, no. 10, pp. 2549–2554, Oct. 2006.
- [2] W. E. Ryan, "Optimal code rates for concatenated codes on a PR4-equalized magnetic recording channel," *IEEE Trans. Magn.*, vol. 36, no. 6, pp. 4044–4049, Nov. 2000.
- [3] R. D. Cideciyan, E. Eleftheriou, and S. Tomasin, "Performance analysis of magnetic recording systems," in *IEEE International Conference on Communications, 2001*, vol. 9, Helsinki, Finland, Jun. 2001, pp. 2711–2715.
- [4] J. Moon and W. Zeng, "Equalization for maximum likelihood detectors," *IEEE Trans. Magn.*, vol. 31, no. 2, pp. 1083–1088, Mar. 1995.
- [5] 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.
- [6] P. Chaichanavong and P. H. Siegel, "Tensor-product parity code for magnetic recording," *IEEE Trans. Magn.*, vol. 42, no. 2, pp. 350–352, Feb. 2006.
- [7] D. Arnold and H.-A. Loeliger, "On the information rate of binary-input channel with memory," in *Proceedings IEEE International Conference on Communications 2001*, Helsinki, Finland, Jun. 2001.
- [8] H. D. Pfister, J. B. Soriaga, and P. H. Siegel, "On the achievable information rates of finite state ISI channels," in *Proceedings IEEE Global Communications Conference 2001*, San Antonio, Texas, USA, Nov. 2001.

- [9] Z. Zhang, T. M. Duman, and E. Kurtas, "Information rates of binary-input inter-symbol interference channels with signal-dependent media noise," *IEEE Trans. Magn.*, vol. 39, no. 1, pp. 599–607, Jan. 2003.
- [10] S. Yang, A. Kavčić, and W. E. Ryan, "Optimizing the bit aspect ratio of a recording system using an information-theoretic criterion," *IEEE Trans. Magn.*, vol. 39, no. 5, pp. 2228–2230, Sep. 2003.
- [11] W. E. Ryan, F. Wang, R. Wood, and Y. Li, "Optimal code rates for the Lorentzian channel: Shannon codes and LDPC codes," *IEEE Trans. Magn.*, vol. 40, no. 6, pp. 3559–3565, Nov. 2004.
- [12] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inf. Theory*, vol. 2, pp. 284–287, Mar. 1974.
- [13] L. L. Scharf, *Statistical Signal Processing: Detection, Estimation, and Time Series Analysis*. Addison-Wesley Publishing Company, Inc, 1991.

Chapter 3

Nonlinear Transition Shift and Write Precompensation in Perpendicular Recording Systems

In a high density perpendicular recording system, nonlinear effects can distort the read-back signal and degrade the system performance. Nonlinear transition shift (NLTS) induced by demagnetization from previously written transitions is one example. As in longitudinal recording, the NLTS in a perpendicular recording channel can be measured by time or frequency analysis of the read-back signal corresponding to a carefully chosen input data pattern [1, 2]. The distortion caused by NLTS can be reduced by the use of write precompensation, whereby, for specific data patterns, deterministic offsets are added to timing of written transitions. A simple and commonly used precompensation scheme is dibit precompensation, which affects the second transition of a dibit pair; see, e.g., [3, 4]. In practice, the timing offsets in write precompensation are optimized empirically in order to minimize the bit-error-rate (BER).

There are very few theoretical results on optimal precompensation of NLTS in recording channels. This is due, in part, to the complex nature of the nonlinear effects. It also stems from the fact that, even with a good model for NLTS, it is very difficult to derive an analytical solution for a precompensation scheme that minimizes the BER. Lim

and Kavčić [5] presented a dynamic programming method to optimize write precompensation for a longitudinal recording channel with partial erasure, NLTS and additive white Gaussian noise (AWGN). Their objective was to minimize the mean-squared error (MSE) between the output signal of the noisy, nonlinear channel model and that of the noiseless, linear channel model, rather than to minimize BER. They allowed the use of a different precompensation value for each transition. The optimization procedure and the resulting precompensation scheme would be too complex to implement in a real system, however.

In this chapter, we will first describe briefly the reason of the existence of NLTS. The NLTS calculation model proposed by Bertram and Nakamoto [6, 7] will be introduced. A perpendicular recording channel model including jitter noise, AWGN and NLTS with write precompensation will then be discussed. Various precompensation schemes will be discussed including dibit, two-level, 3-level and 7-level precompensation scheme. We will numerically find the precompensation levels that minimize the BER at the detector output and compare the performance with the case where there is no NLTS in the channel. This result will also be compared to the optimal precompensation levels based upon two other criteria: minimizing the MSE as in [5] and minimizing the variance of the net transition shift.

3.1 Perpendicular Recording Channel with Nonlinear Transition Shift

3.1.1 Nonlinear Transition Shift Model

The nonlinear transition shift (NLTS) is induced by the demagnetization from the previously written transitions. When the write head writes a transition, it generates a magnetic field opposite from the previous bit and magnetizes the media to the opposite direction. The transition positions are controlled by the write current. But with the interference of the magnetic field generated by the previously written transitions, the

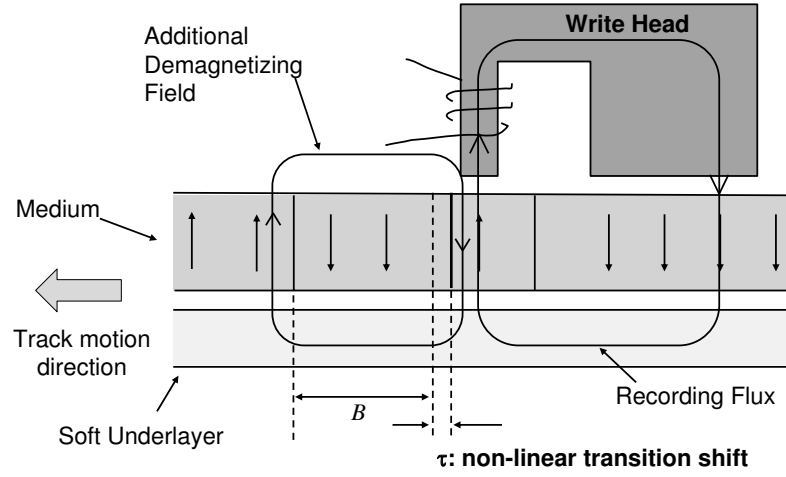


Figure 3.1 The generation of the nonlinear transition shift. B is the channel bit spacing.

writing field will be affected and thus the real transition position will be shifted from the proper position. This is illustrated in Figure 3.1.

The amount of the transition shift is determined by media and head parameters, as well as the pattern and the distances of the previous transitions to the current one. When the recording density is increased, the previous transitions are closer to the current one; thus, a larger transition shift is generated. Therefore, with increasing density, the NLTS becomes one of the nonlinear effects that can not be ignored.

Bertram and Nakamoto [6, 7] proposed a model to calculate the NLTS for a perpendicular recording channel. The net demagnetizing field was calculated by superposition of the field produced by previous transitions. Figure 3.2 illustrates the parameters used: T , the thickness of the medium; S , the medium to soft underlayer spacing; M_r , the remanent magnetization; and HFG , the head field gradient at the writing position. The distances of the previous transitions to the current transition are denoted by L_1, L_2, \dots , where L_1 corresponds to the most recent transition, L_2 corresponds to the next most recent and so on.

Therefore, according to Bertram and Nakamoto, the net demagnetizing field is

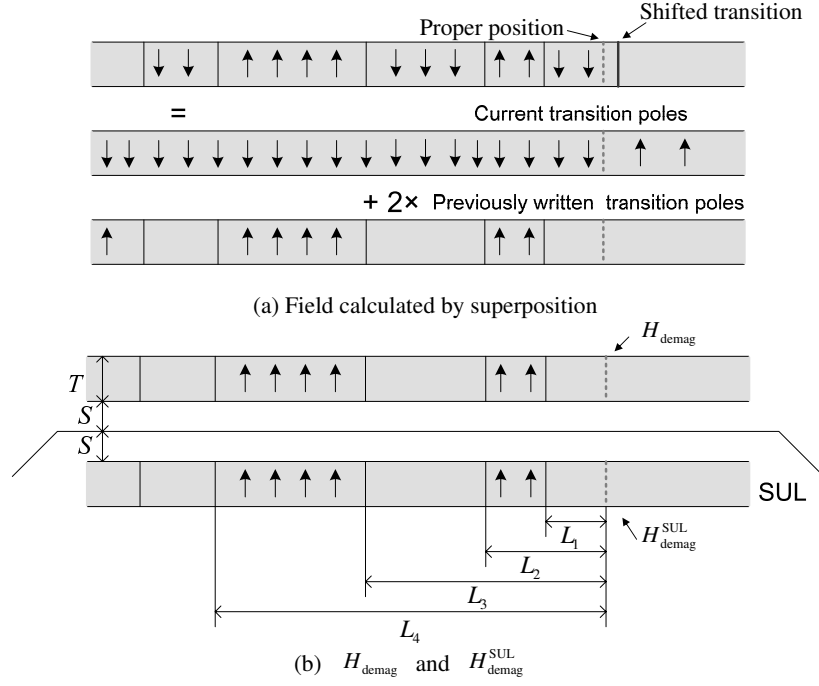


Figure 3.2 The calculation of NLTS.

the sum of the fields from the medium and the soft underlayer.

$$H_{\text{demag}}^{\text{net}} = H_{\text{demag}} + H_{\text{demag}}^{\text{SUL}} \quad (3.1)$$

where

$$H_{\text{demag}} = -\frac{8M_r}{T} \int_0^T \left(\arctan \frac{y}{L_1} - \arctan \frac{y}{L_2} + \arctan \frac{y}{L_3} - \dots \right) dy \quad (3.2)$$

$$H_{\text{demag}}^{\text{SUL}} = -\frac{4M_r}{T} \int_0^T \left(\begin{aligned} &\arctan \frac{y+2S}{L_1} - \arctan \frac{y+2S}{L_2} \\ &- \arctan \frac{y+2S+T}{L_1} + \arctan \frac{y+2S+T}{L_2} \\ &\arctan \frac{y+2S}{L_3} - \arctan \frac{y+2S}{L_4} \\ &- \arctan \frac{y+2S+T}{L_3} + \arctan \frac{y+2S+T}{L_4} \\ &+ \dots \end{aligned} \right) dy. \quad (3.3)$$

By using the following definite integral,

$$\int_0^t (\arctan y) dy = t \arctan t - \frac{1}{2} \ln(1 + t^2), \quad (3.4)$$

we can express the demagnetizing field equations without integrals:

$$H_{\text{demag}} = -\frac{8M_r}{T} \sum_{i=1}^{\infty} (-1)^{i-1} \left[T \arctan \frac{T}{L_i} - \frac{L_i}{2} \ln\left(1 + \frac{T^2}{L_i^2}\right) \right] \quad (3.5)$$

$$\begin{aligned} H_{\text{demag}}^{\text{SUL}} = & -\frac{4M_r}{T} \sum_{i=1}^{\infty} (-1)^{i-1} \left[(T + 2S) \arctan \frac{T + 2S}{L_i} - 2S \arctan \frac{2S}{L_i} \right. \\ & - (2T + 2S) \arctan \frac{2T + 2S}{L_i} + (T + 2S) \arctan \frac{T + 2S}{L_i} \\ & \left. - \frac{L_i}{2} \ln\left(1 + \frac{T^2 + 4ST}{L_i^2 + 4S^2}\right) + \frac{L_i}{2} \ln\left(1 + \frac{T^2 + 2T(T + 2S)}{L_i^2 + (T + 2S)^2}\right) \right]. \end{aligned} \quad (3.6)$$

The NLTS can then be written as

$$\tau = \frac{H_{\text{demag}}^{\text{net}}}{HFG}. \quad (3.7)$$

From equations (3.5) and (3.6) we can see that the current NLTS is determined by all the previous transition positions, for the given head and medium parameters. The contribution of a previously recorded transition to the shift of a transition currently being written decreases as the distance between the transitions increases.

In Figure 3.3, we show an example of the contributions of the past transitions. The contributions are normalized by the contribution of the neighboring transition that is B away. The x-axis represents the separation between the previous transition and the current one, as a multiple of the channel bit spacing B . The y-axis represents the absolute value of the normalized contribution of the corresponding transition. The medium thickness is 10nm. The medium to soft underlayer spacing is 20nm. The channel bit spacing, denoted by B , is 16nm. The remanent magnetization to head field gradient ratio M_r/HFG is 1.5077. With these parameters, the NLTS caused by a neighboring transition in the past, i.e., the dibit induced NLTS is about 20% of the channel bit spacing B . We can see that the contribution of the past transitions decreases very fast. For example, the absolute value of the contribution of a transition $4B$ away from the current

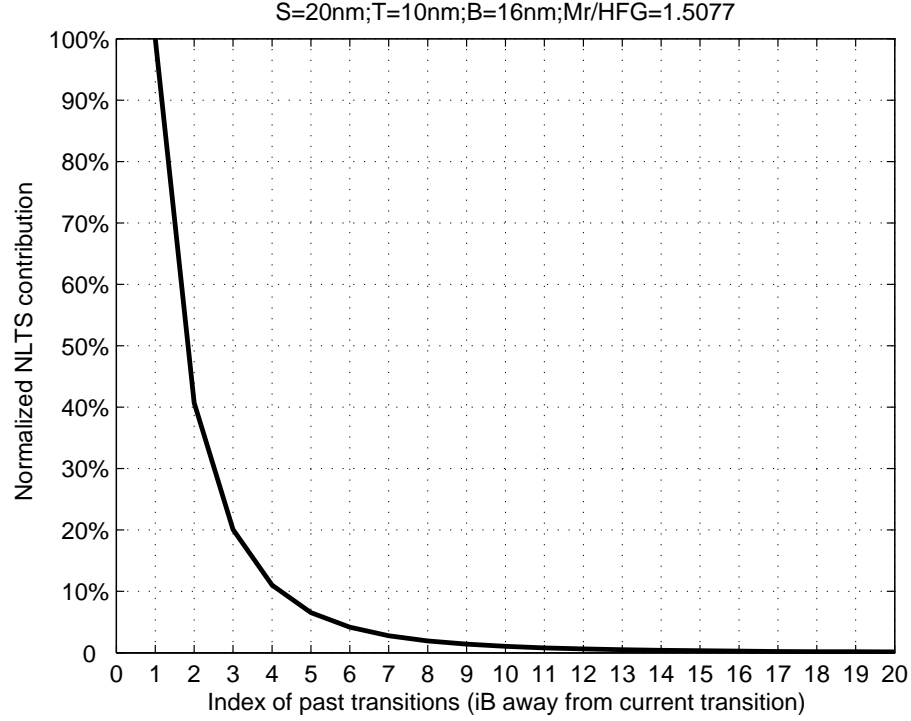


Figure 3.3 The normalized absolute contributions of the past transitions. $T = 10\text{nm}$; $S = 20\text{nm}$; $M_r/HFG = 1.5077$; $B = 16\text{nm}$.

transition decreases to 10% of that of a neighboring transition that is B away. The shift induced by a transition $10B$ away is only 1% of that of the neighboring transition, and the relative shift resulting from a transition $20B$ away drops to 0.1%. Therefore, when calculating the NLTS, we can ignore the influence of the transitions far away from the current transition. The equations (3.5) and (3.6) can be truncated. We denote by K the number of preceding bits used to calculate the NLTS. The larger K is, the more accurate the calculated NLTS will be. We call K the NLTS calculation window length.

In equations (3.5) and (3.6), the terms for the transitions in the past have different signs, which means some of the transitions in the past push the current transition away while others pull it back. We can see that the terms with L_i , where i is odd, have negative signs in the net demagnetization field while those with L_i , where i is even, have positive signs. Since the contribution of the previous transitions decreases as i increases, the

Table 3.1 NLTS for different data patterns. (consider 3 previous NRZI bits)

Bit pattern	τ/B (No NLTS on previous transitions)	τ/B (NLTS on previous transitions considered)
$\dots 0000(1)$	0	0
$\dots 0001(1)$	20%	20%
$\dots 0010(1)$	8.1%	8.1%
$\dots 0011(1)$	11.9%	17.0%
$\dots 0100(1)$	4.0%	4.0%
$\dots 0101(1)$	16.0%	17.9%
$\dots 0110(1)$	4.1%	5.5%
$\dots 0111(1)$	15.9%	18.8%

demagnetization field will always be non-positive, i.e., either zero or the opposite direction of the write field. Therefore, the direction of the NLTS in perpendicular recording systems is always away from the previous transitions, and τ is always non-positive.

In Table 3.1, an example of the NLTS values for different data patterns are listed. The ‘1’ in the data pattern represents a transition while ‘0’ represents no transition. The second column shows the NLTS values calculated for the data patterns where all the previous transitions are assumed to be at their nominal positions. In real systems where the data are written sequentially, the transition shifts of the previous transitions will affect the distance of these transitions to the current one. The third column takes this effect into account and gives the NLTS values where the previous transitions are shifted by NLTS. We only consider 3 bits of transitions before the current one and all the bits before that are assumed to be zero (no transitions). Since the most recent neighboring bit plays the most significant role in NLTS, we can see that if this bit is shifted by NLTS, the NLTS for current transition will be shifted more, as can be seen in the case of data pattern “011”.

Figure 3.4 shows the change in the distribution of NLTS values for the two scenarios. Both curves represent a histogram of the NLTS values. The histogram is obtained by counting the number of transitions in a long random sequence that are shifted by a value within each bin of length 0.001. The dashed curve is the histogram of NLTS values which are calculated under the assumption that the previous transitions are not

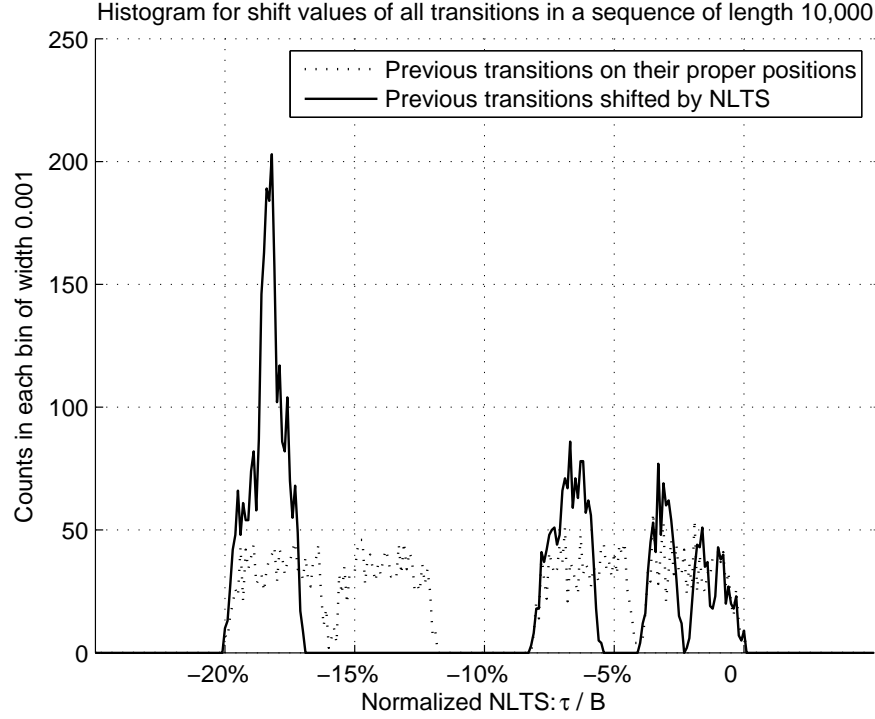


Figure 3.4 The distribution of NLTS for a random sequence. $T = 10\text{nm}$; $S = 20\text{nm}$; $M_r/HFG = 1.5077$; $B = 16\text{nm}$.

shifted. The solid curve considers the NLTS of the previous transitions. The absolute shifts larger than 0.1 correspond to the data patterns that have a neighboring transition. We can see that this part of the histogram changes most.

3.1.2 Write Precompensation

NLTS can be offset by writing the transition a little earlier so that the resulting transition position is close to its nominal position. We call this write precompensation. Let us denote Δ_i the precompensation for the i^{th} transition in a data sequence. The current net shift of the transition, denote by δ_i , is the sum of the precompensation and the NLTS.

$$\delta_i = \Delta_i + \tau_i. \quad (3.8)$$

Here, τ_i is a function of the distances of the previous transitions to the current transition position. Note that the current transition position is shifted by the precompensation value Δ_i when writing. The previous transitions are also shifted by the combination of their precompensation and the NLTS.

In theory, we can calculate the proper precompensation value Δ_i that reduces the net transition shift δ_i to zero. But this is not practical in real systems for three reasons. First, the precompensation precision cannot be arbitrarily small. Second, the precompensation values that make the net transition shifts zeros might require the knowledge of quite a few bits in the past, which could cost a lot of memory. And finally, these precompensation values are not easy to obtain for real systems.

3.1.3 Channel Output and Approximations

In this chapter, we will consider a channel model with NLTS, as well as jitter noise and electronic noise. The transition response is modeled as an error function, the same as in Chapter 2.

Let the input transitions be d_i , $d_i \in \{-1, 0, +1\}$. The channel output $z(t)$ can be written as

$$z(t) = \sum_i d_i s(t + \delta_i + a_i - iB) + n_W(t). \quad (3.9)$$

Here, δ_i is the net shift of the transition d_i with respect to its nominal location in the recording medium, a_i is the random position jitter for transition d_i , B is the channel bit spacing (as well as the sampling period), and $n_W(t)$ is the electronics noise. For $d_i = 0$, we set $a_i = 0$, whereas for $d_i \neq 0$, a_i is a zero mean Gaussian random variable with variance σ_J^2 . The jitter values for recorded transitions are mutually independent. The electronics noise $n_W(t)$ is modeled as a zero-mean, AWGN process. The variance of the sampled AWGN $n_W(kB)$ is denoted by σ_W^2 . We define the signal-to-AWGN ratio to be $SNR_W = 10 \log_{10}(V_{\max}^2/\sigma_W^2)$, which is the same as in Chapter 2.

If the net transition shift is small, we can approximate the channel output again

using a truncation of the Taylor series. Expanding the transition shift at $t - iB$,

$$z(t) = \sum_i d_i \left[s(t - iB) + (\delta_i + a_i) s'(t - iB) + \frac{(\delta_i + a_i)^2}{2} s''(t - iB) + \cdots \right] + n_W(t). \quad (3.10)$$

We call the result an *order-1* channel approximation when we consider only the first derivative of the transition response:

$$z_{o1}(t) \approx \sum_i d_i s(t - iB) + \sum_i d_i (\delta_i + a_i) s'(t - iB) + n_W(t) \quad (3.11)$$

and an *order-2* channel approximation when we take the first and second derivatives into account:

$$\begin{aligned} z_{o2}(t) \approx & \sum_i d_i s(t - iB) + \sum_i d_i (\delta_i + a_i) s'(t - iB) \\ & + \sum_i d_i \frac{(\delta_i + a_i)^2}{2} s''(t - iB) + n_W(t). \end{aligned} \quad (3.12)$$

As we did in Chapter 2, the data sequences can be represented by NRZ data x_i as well as in terms of the transitions. The order-1 and order-2 channel approximations can be written as

$$z_{o1}(t) \approx \sum_i x_i h(t - iB) + \sum_i d_i (\delta_i + a_i) s'(t - iB) + n_W(t) \quad (3.13)$$

$$\begin{aligned} z_{o2}(t) \approx & \sum_i x_i h(t - iB) + \sum_i d_i (\delta_i + a_i) s'(t - iB) \\ & + \sum_i d_i \frac{(\delta_i + a_i)^2}{2} s''(t - iB) + n_W(t) \end{aligned} \quad (3.14)$$

where $d_i = \frac{x_i - x_{i-1}}{2}$ and $h(t)$ is the dipulse response.

The accuracy of these approximations will be discussed later.

3.1.4 System Diagram

In this chapter, we consider a simple system shown in Figure 3.5. The channel output is first sampled and then pass through a digital equalizer. The detector is assumed to be a conventional Viterbi detector matched to the equalizer target. We will

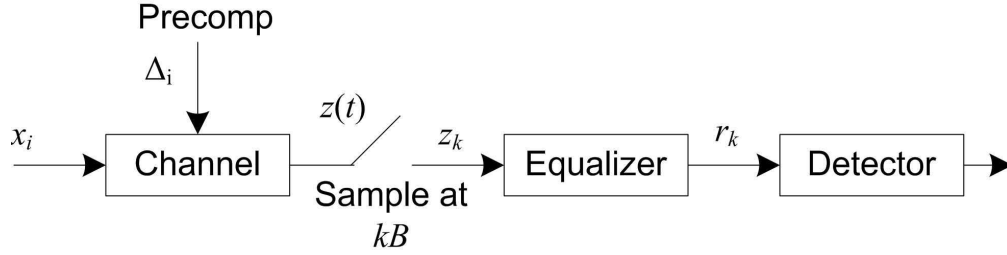


Figure 3.5 The system diagram for NLTS and precompensation study.

measure the BER at the detector output in the simulations. The performance of various precompensation schemes will also be compared and analyzed.

3.2 Precompensation Schemes

As we have mentioned, for a given transition sequence $\{d_i\}$ one could in principle determine a corresponding set of precompensation values $\{\Delta_i\}$ such that the net shift of all transitions would be zero. However, in practice, recording systems usually use only a small number of precompensation values corresponding to selected patterns of recently recorded transitions, which play the most important role in determining the NLTS value of the transition being written. Therefore, these precompensation schemes can not completely eliminate the effects of NLTS. However, as our later simulations will show, in some cases, the performance of a simple precompensation scheme can be even better than the performance of a channel with no NLTS.

3.2.1 Dibit Precompensation

The simplest precompensation scheme, usually called dibit precompensation, applies a shift only to those transitions with a neighboring transition in the preceding bit position. As we have mentioned before, the neighboring transition is the most significant contributor to the NLTS. Using the NRZI description, where ‘0’ represents no

transition and ‘1’ represents a transition, we can specify the precompensation rule for the i^{th} transition d_i as:

$$\Delta_i = \begin{cases} \Delta & \text{if } d_{i-1} \neq 0 \text{ and } d_i \neq 0 \\ 0 & \text{otherwise} \end{cases} \quad (3.15)$$

3.2.2 Two-level Precompensation

The NLTS values for different patterns of the 3 preceding bits are shown in Table 3.1. If we only look at the 2 preceding bits, we will find that the two cases where the transitions are shifted the most is when the preceding 2 bits are ‘01’ and ‘11’. Therefore, we now propose a two-level precompensation scheme that applies precompensation to these two cases. Specifically, we define the precompensation value as

$$\Delta_i = \begin{cases} \Delta_H & \text{if } d_i \neq 0, d_{i-1} \neq 0 \text{ and } d_{i-2} = 0 \\ \Delta_L & \text{if } d_i \neq 0, d_{i-1} \neq 0 \text{ and } d_{i-2} \neq 0 \\ 0 & \text{otherwise.} \end{cases} \quad (3.16)$$

3.2.3 Multilevel Precompensation

More levels of precompensation can be used, according to different previous data patterns. However, the number of levels of the precompensation increases exponentially to the number of bits involved in the pattern. If the number of bits involved is k , and we assign one level to each pattern except the all-zero pattern, the number of levels of precompensation will be $2^k - 1$.

For example, the 3-level precompensation scheme can be expressed as the follows:

$$\Delta_i = \begin{cases} \Delta_H & \text{if } d_i \neq 0, d_{i-1} \neq 0 \text{ and } d_{i-2} = 0 \\ \Delta_M & \text{if } d_i \neq 0, d_{i-1} \neq 0 \text{ and } d_{i-2} \neq 0 \\ \Delta_L & \text{if } d_i \neq 0, d_{i-1} = 0 \text{ and } d_{i-2} \neq 0 \\ 0 & \text{otherwise.} \end{cases} \quad (3.17)$$

The precompensation levels for a 7-level (3-bits look-back) scheme can be expressed as in Table 3.2

Table 3.2 The precompensation levels corresponding to the preceding 3-bit patterns.

NRZI pattern	000(1)	001(1)	010(1)	011(1)	100(1)	101(1)	110(1)	111(1)
Precomp Δ_i	0	$\Delta_{(1)}$	$\Delta_{(2)}$	$\Delta_{(3)}$	$\Delta_{(4)}$	$\Delta_{(5)}$	$\Delta_{(6)}$	$\Delta_{(7)}$

The more levels we have, the more complex it is to find the best precompensation levels through Monte-Carlo simulation searches. For the 3-level and 7-level precompensation scheme, instead of using brute-force search for the optimal values in the Monte-Carlo BER simulation, we used the performance analysis technique in Chapter 4 to help reduce the computational complexity. Thus, we can compare the performances of different precompensation schemes.

3.3 Simulation Results

The channel simulations use pseudorandom input data sequences of length 5120 bits. Using the NLTS model, the net transition shifts are calculated sequentially for each sequence. The noisy channel output signal is sampled at multiples of the bit spacing B . We use a minimum mean-squared error (MMSE) equalizer design with monic constraint [8]. The equalizer is a 21-tap FIR filter and the equalization target has 3 taps. The equalizer output is passed into a Viterbi detector matched to the target. The BER is measured at the Viterbi detector output.

For the NLTS calculation, we set the same parameters as in Table 3.1. The channel bit spacing B is 16nm in most of the simulations except in Figure 3.15, where B is 14nm.

3.3.1 Dibit Precompensation

Figure 3.6 shows the simulated system BER as a function of the normalized dibit precompensation level Δ/B for window lengths $K = 5, 10$, and 20. The simulation uses the order-2 channel approximation. The channel density T_{50}/B is 1, the jitter noise σ_J/B is set to 0.08, and the signal-to-AWGN ratio SNR_W is 29dB. As can be seen, the

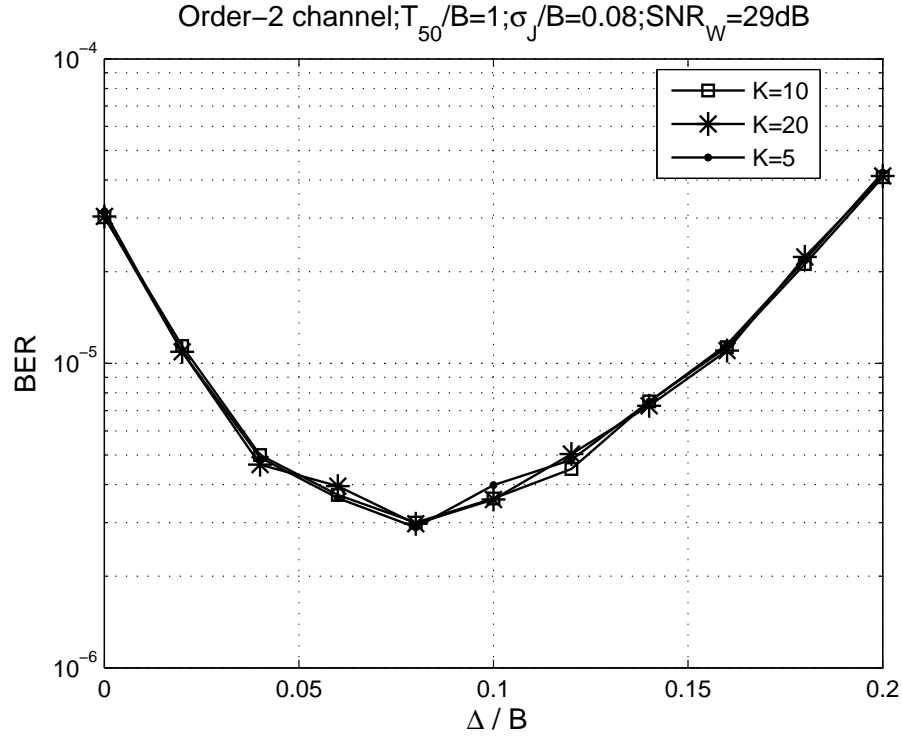


Figure 3.6 Influence on BER of NLTS window length.

curves corresponding to the different values of M are almost identical. Therefore, in all of the remaining simulations, we use the NLTS window length $K = 10$. Note that the precompensation level that minimizes BER is $\Delta = 0.08B$.

In Figure 3.7, we compare the BER results obtained using the exact channel model with those obtained with the order-1 and order-2 channel approximations. We see that the BER produced by the order-1 channel approximation is optimistic relative to that of the exact channel over the entire range of precompensation values, while, in contrast, the BER corresponding to the order-2 approximation is pessimistic over the same range. Since the order-2 estimate is fairly close to the exact channel BER, and is considerably less complex to compute, we will use the order-2 channel approximation in subsequent simulations in this section.

For the dibit precompensation, a question we would like to know the answer

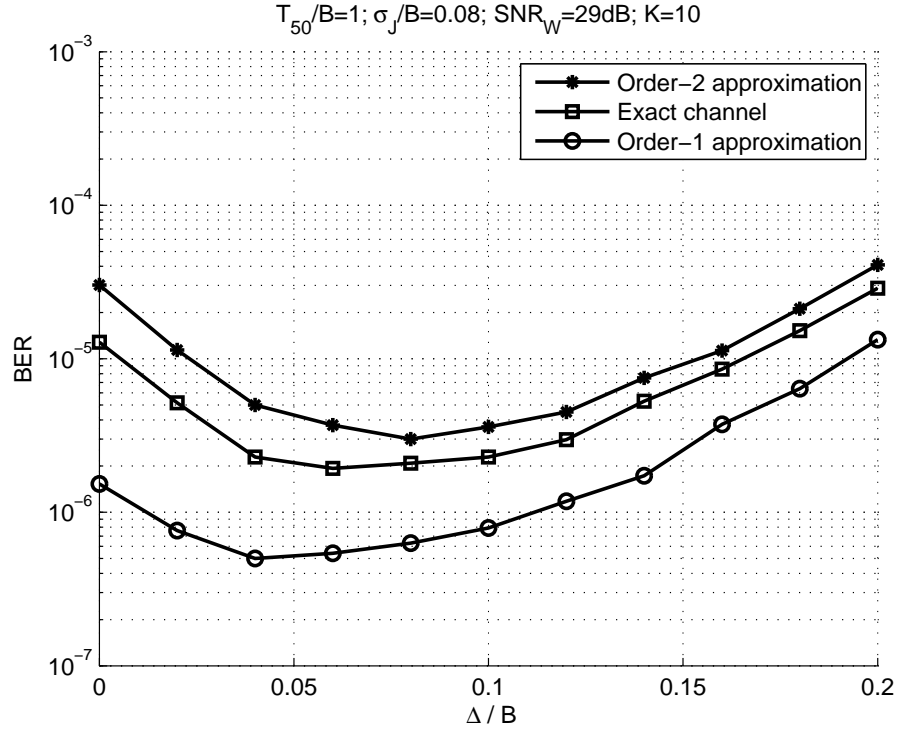


Figure 3.7 Comparison of simulated BER for exact channel, order-1 channel approximation, and order-2 channel approximation.

to is how the best precompensation level changes when some of the channel parameters change. Therefore, in Figure 3.8, Figure 3.9 and Figure 3.10 we fix the NLTS calculation parameters and change only the jitter noise, AWGN and ISI separately.

In Figure 3.8 we changed the jitter noise level σ_J . All the other parameters remain the same. We can see only a very slight change in the best precompensation level when σ_J/B changes from 0.08 to 0.14. The curves become shallower when the jitter noise increases. The performance improvement brought by write precompensation decreases because the NLTS is relatively small when jitter noise variance increases.

The BER performance of dibit precompensation for different AWGN levels is shown in Figure 3.9. The best precompensation levels for different SNR_W remain almost the same as the case for the jitter noise. However, in Figure 3.10, we find that the best precompensation level needs to be larger when T_{50} increases. When $T_{50} = 1.5B$,

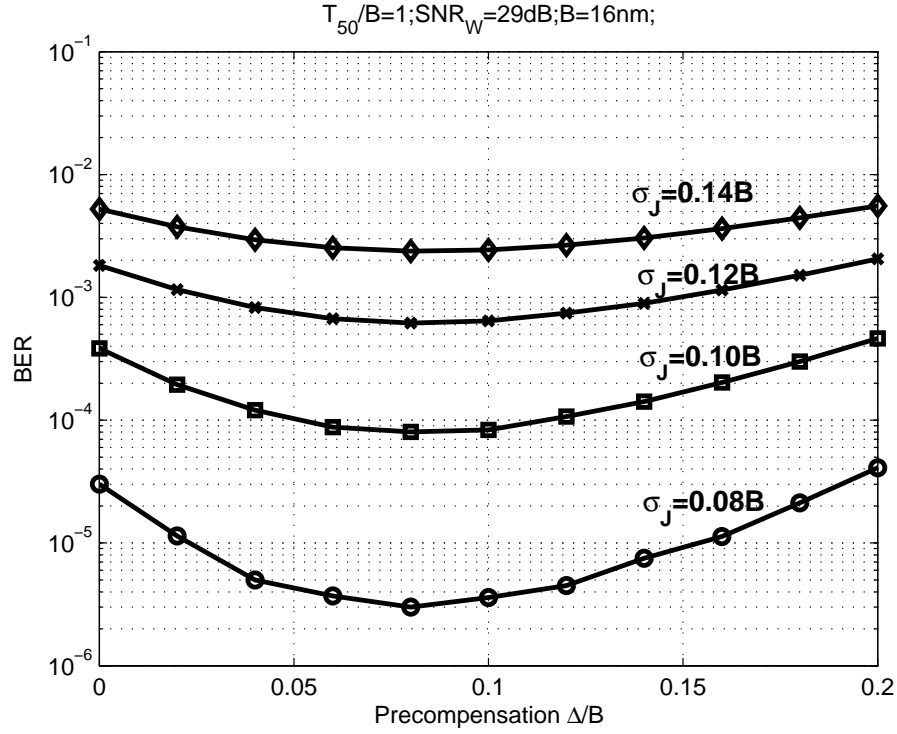


Figure 3.8 Best dibit precompensation levels for different jitter noise levels. $B = 16\text{nm}$; $T_{50}/B = 1$; $\text{SNR}_W = 29\text{dB}$; $\sigma_J/B = 0.08, 0.10, 0.12, 0.14$.

the best precompensation level is $0.12B$, compared to $0.08B$ when $T_{50} = 1B$.

3.3.2 Two-level Precompensation

The BER simulation results for the two-level precompensation scheme are shown in Figure 3.11. The simulations used the same system parameters as in previous simulations, with NLTS window length 10 and an order-2 channel approximations. The surface plot shows the BER as a function of the normalized precompensation values Δ_H/B and Δ_L/B . The optimal two-level precompensation values are seen to be $\Delta_H^* = 0.09B$ and $\Delta_L^* = 0.05B$.

Figure 3.12 shows a comparison between the performance of the two-level precompensation scheme and the dibit precompensation. The curve for the dibit precompensation scheme is the same as the order-2 channel approximation curve shown in

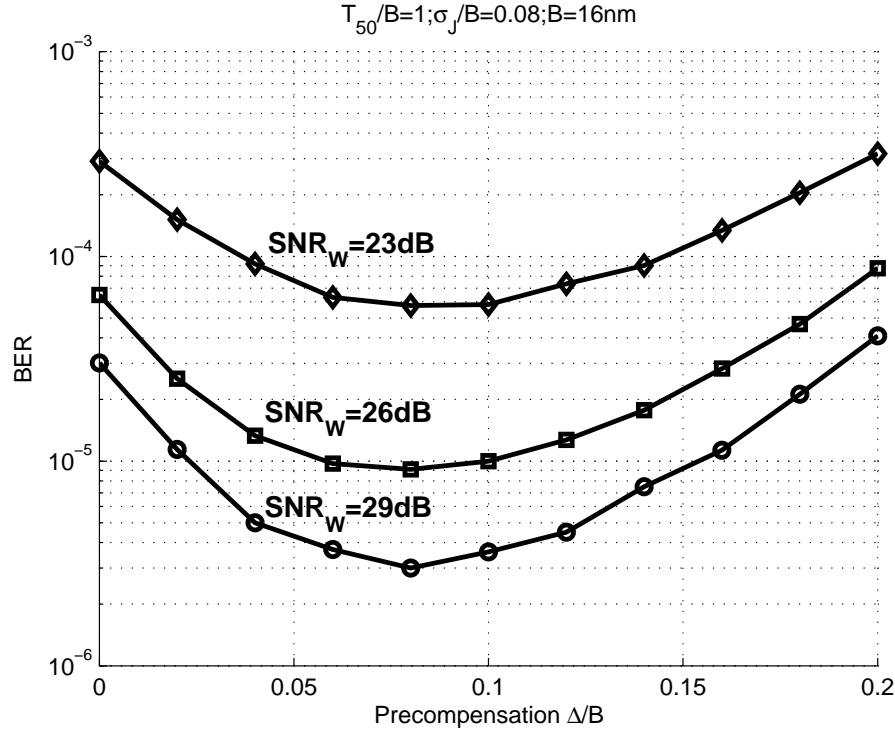


Figure 3.9 Best dibit precompensation levels for different AWGN levels. $B = 16\text{nm}$; $T_{50}/B = 1$; $\sigma_J/B = 0.08$; $SNR_W = 23, 26, 29\text{dB}$.

Figure 3.7. For the two-level precompensation scheme, we fixed Δ_L to be the optimum value $0.05B$ and plot the BER versus Δ_H/B . The point on this curve that achieves the minimum BER is thus optimal for the scheme. The horizontal dash-dot line represents the BER for a channel with no NLTS and no precompensation but with the same amount of jitter and electronics noise. We see that the dibit precompensation scheme reduces the BER by approximately one order of magnitude compared to a system without any precompensation. There is a small gap between the lowest BER achieved by the dibit precompensation scheme and the BER for the no-NLTS case. However, the system with optimized two-level precompensation actually performs better than the system with no NLTS. Specifically, the BERs for the optimal dibit precompensation scheme, the no-NLTS case, and the optimal two-level precompensation scheme are 3.0×10^{-6} , 1.7×10^{-6} , and 1.5×10^{-6} , respectively.

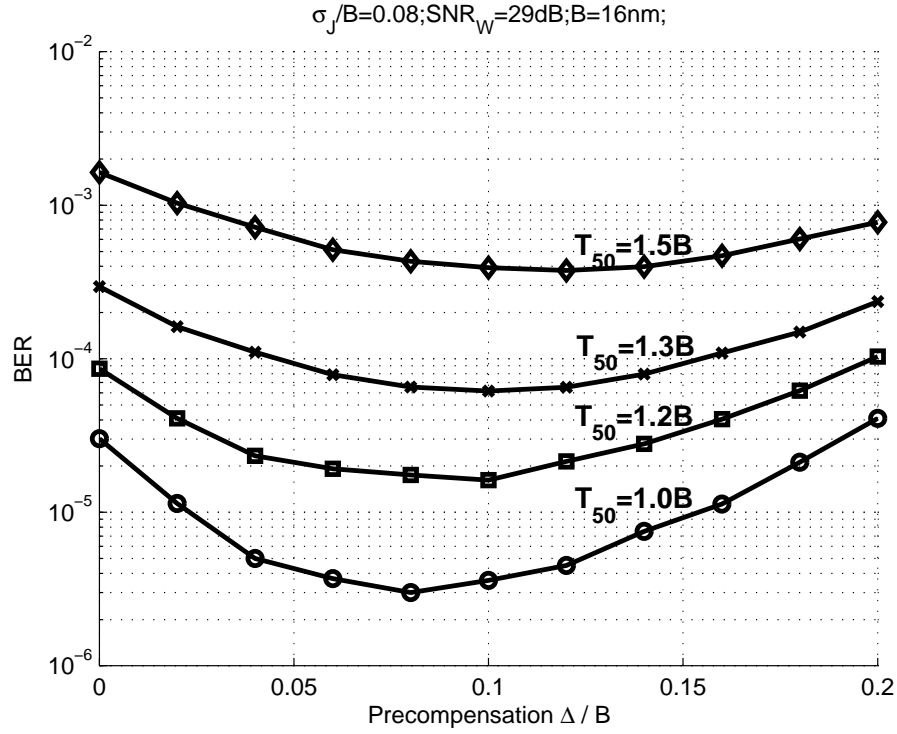


Figure 3.10 Best dibit precompensation levels for different T_{50} . $B = 16nm$; $\sigma_J/B = 0.08$; $SNR_W=29dB$; $T_{50}/B= 1.0,1.2,1.3,1.5$.

In order to verify and better understand these comparative results, particularly the overall superiority of the optimized two-level precompensation scheme, we ran longer channel BER simulations. Indeed, the longer simulations pointed to the same conclusion. In Table 3.3, we list the frequency of occurrence of the dominant error events for the system without NLTS and for the system with NLTS and optimal two-level precompensation. In the simulations, a total of 100,000 sectors of 5120 bits each were used as channel input. The same pseudorandom input sequence was used in both cases. We also used the same sequence of randomly generated jitter and electronics noise samples. We see from the table that the dominant error event is the single error event for both channels, but the system using two-level precompensation scheme experienced about 9% fewer of these.

However, this relative performance may not be observed if the channel param-

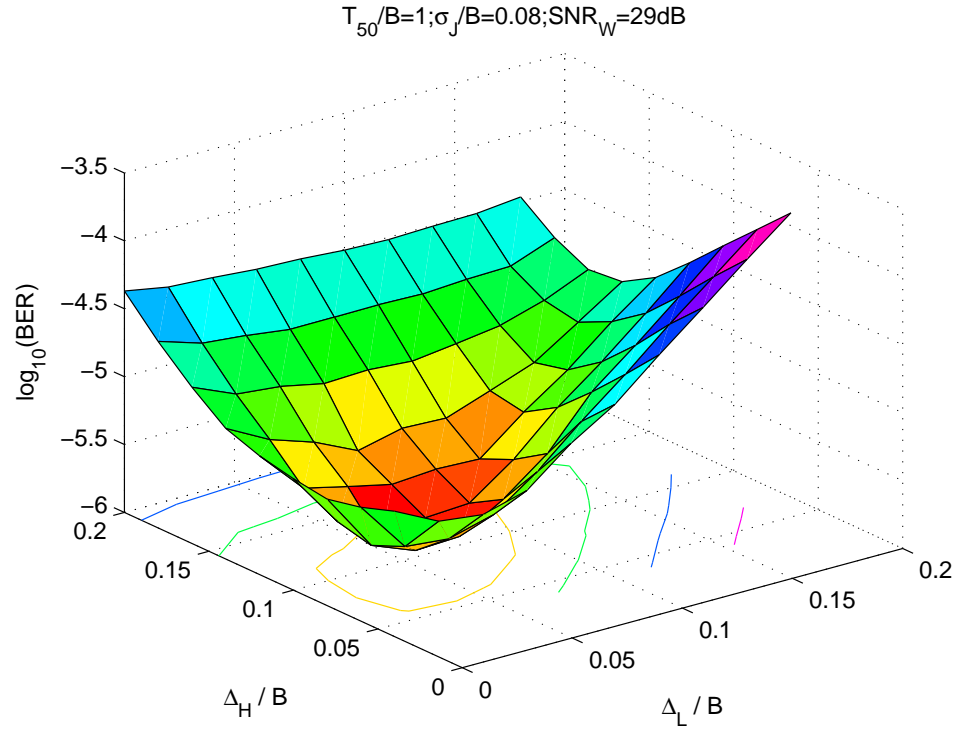


Figure 3.11 Surface plot of BER for the two-level precompensation scheme.

Table 3.3 Error-event counts for system with no NLTS and with optimal two-level precompensation

Error events	No NLTS	Optimal two-level precomp
1	414	368
-1	410	385
-1 1	12	4
1 -1	4	6
1 -1 1	4	1
-1 1 -1	4	2
Total erroneous bits	888	790
BER	1.73×10^{-6}	1.54×10^{-6}

eters change. Figure 3.13 shows a comparison of BER performance when the precompensation schemes are used on a channel with more severe jitter noise, specifically with $\sigma_J = 0.12B$. All the other system parameters remain unchanged. We see that while the

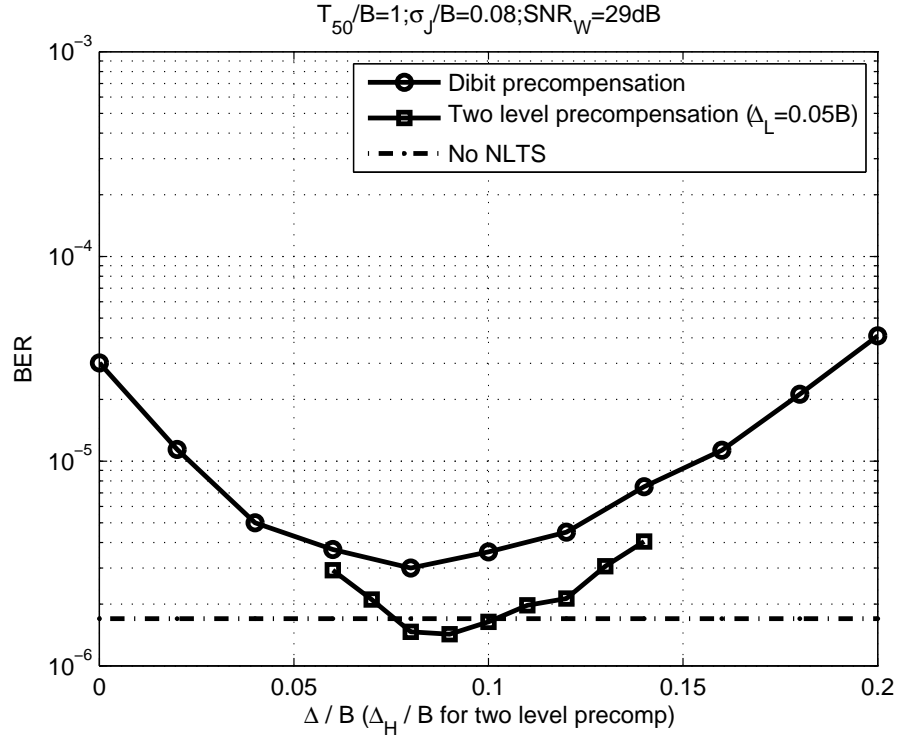


Figure 3.12 Comparison of simulated BER for dibit precompensation and two-level precompensation with $\sigma_J = 0.08B$.

optimized two-level precompensation scheme continues to have an advantage over the dibit scheme, it does not outperform the system with no NLTS, although the results are very close.

3.3.3 Higher Density and Multilevel Precompensation

According to the previous simulations, the optimal two-level precompensation can perform better than the no NLTS situation. How will other precompensation schemes with more levels perform? In this section, we will show some simulation results that addresses this question.

As mentioned earlier, the search complexity for the best precompensation values grows exponentially with the number of precompensation levels. It is not practical to use the brute force search of Monte-Carlo BER simulations to obtain the optimal pre-

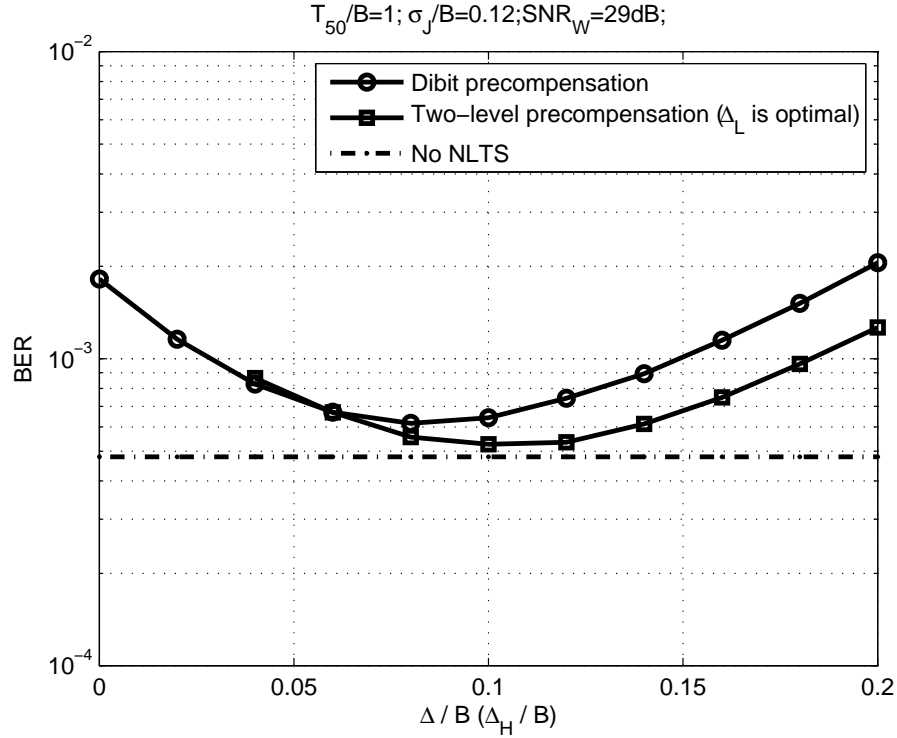


Figure 3.13 Comparison of simulated BER for dibit precompensation and two-level precompensation with $\sigma_J = 0.12B$.

omprecompensation values for 3-level and 7-level precompensation schemes. Instead, we will use the performance analysis in Chapter 4 to help reduce the computational time.

For example, for the 3-level precompensation scheme, we used the BER analysis in Chapter 4 to estimate the optimal precompensation values. For a system with the same parameters as in Figure 3.12, the resulted precompensation values are $(\Delta_H, \Delta_M, \Delta_L) = (0.114B, 0.056B, 0.093B)$. To verify the result using the Monte-Carlo simulation, we chose $(0.12B, 0.06B, 0.1B)$ as the center point, and simulated on points $((0.12 \pm 0.02)B, (0.06 \pm 0.02)B, (0.1 \pm 0.02)B)$. The step size is chosen to be consistent with the previous simulations on dibit precompensation. The Monte-Carlo simulation shows that the optimal point is $(0.12B, 0.06B, 0.1B)$, the center point. The curve where Δ_H is fixed to $0.12B$ and Δ_M is fixed to $0.1B$ is shown in Figure 3.14.

For the 7-level precompensation scheme, we didn't verify the estimation result

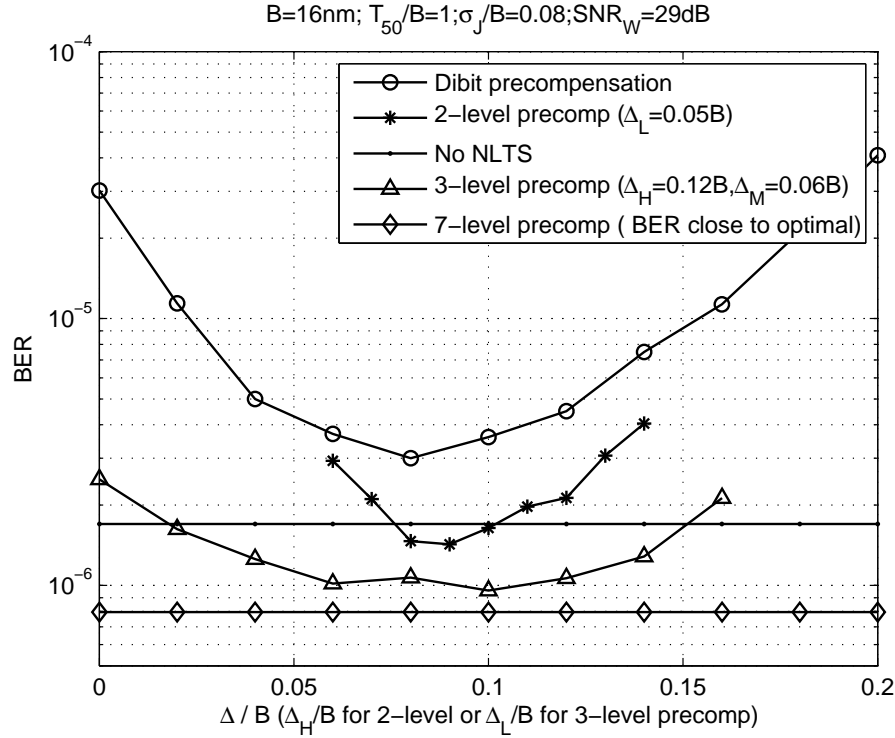


Figure 3.14 Comparison of the Dibit, Two-level and multilevel precompensation schemes for $B=16\text{nm}$. All the parameters are the same as in Figure 3.12.

with the Monte-Carlo simulation, since even the verification similar to the 3-level precompensation scheme is too complex. We used directly the estimated values in the BER Monte-Carlo simulation and draw the BER as a line in Figure 3.14. According to the results of other precompensation schemes, the estimated values are very close to the Monte-Carlo simulation result. Therefore, we think the performance given in Figure 3.14 for the 7-level precompensation scheme is very close to the best performance the 7-level scheme can achieve.

In Figure 3.15, we ran simulations on a system with higher channel density. The channel bit spacing B is set to 14nm and all the other parameters remain the same as in Figure 3.14. With the increase of the channel density, the NLTS of the second transition in an isolated dibit pattern is increased from 20% to approximately 26%. The normalized channel density T_{50}/B is about 1.14 and σ_J/B is about 0.09. The SNR_W is

assumed to be 29dB. As in Figure 3.14, the performance for dibit, 2-level, 3-level and the 7-level precompensation schemes are shown. The curve for 3-level precompensation and the line for 7-level precompensation are obtained by the same process as described previously. Generally, because of the increased NLTS, the precompensation values for all the levels are larger than those in Figure 3.14. We can also see that though the 2-level precompensation scheme didn't give a better performance than that of a system without NLTS, the 3-level and 7-level precompensation schemes all outperform the no-NLTS case. On the other hand, the performances for the 3-level precompensation and the 7-level precompensation are very close. The main reason is that the increased ISI and the normalized jitter noise variance in the higher density channel can not be reduced by more levels of precompensation. Therefore, the improvement of BER resulting from an increased number of precompensation levels is limited.

3.4 Other Criteria

In the comparisons above, we used Monte Carlo simulation to find the precompensation values for dibit and two-level precompensation schemes that minimize the BER after the detector. This optimization approach is very time-consuming, however, so it is desirable to consider criteria other than minimum BER in the selection of precompensation parameters. These criteria might not guarantee a minimal BER after detection, but they might produce a suboptimal solution for the precompensation values that nevertheless gives a relatively low BER. We now consider two such alternative optimization criteria.

One of these criteria is commonly known as the mean-squared error (MSE) criterion [5]. The objective is to minimize the MSE between the output signal of the channel with no noise or NLTS and the output signal of the channel impaired by both. The former is given by

$$y(t) = \sum_i d_i s(t - iB) \quad (3.18)$$

and the latter is given by $z(t)$ as defined in (3.9), or the approximations in (3.11) and

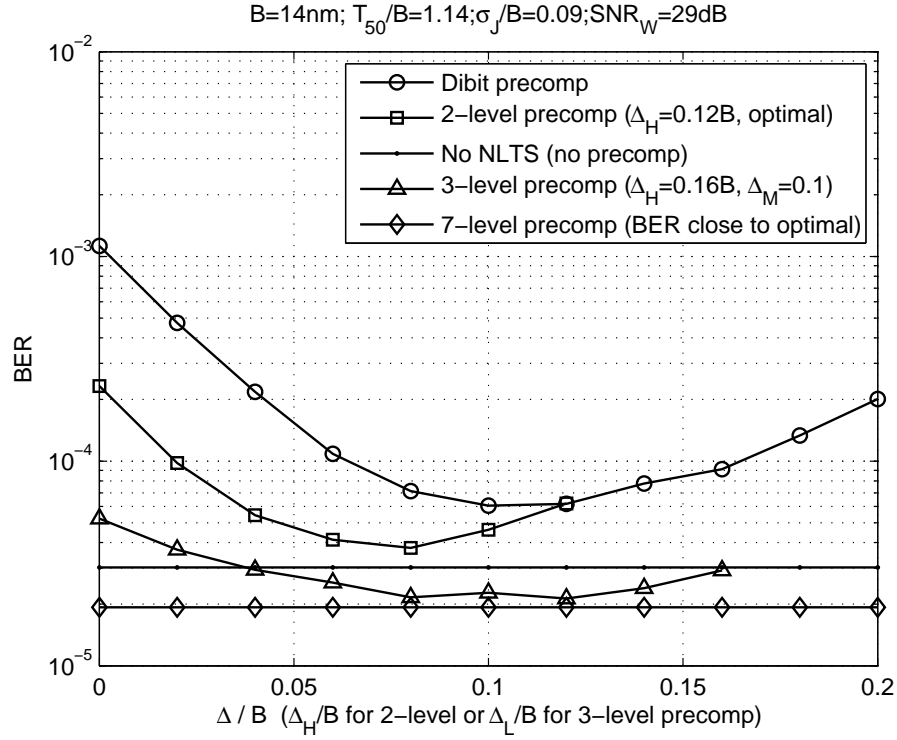


Figure 3.15 Comparison of the Dibit, Two-level and multilevel precompensation schemes for $B=14\text{nm}$. All other parameters are the same as in Figure 3.14. Therefore, $T_{50}/B \approx 1.14$; $\sigma_J/B \approx 0.09$; dibit NLTS $\approx 26\%$.

(3.12). The squared error for each sample is defined by

$$\epsilon_k^2 = [z(kB) - y(kB)]^2, \quad (3.19)$$

and the MSE is then $E\{\epsilon_k^2\}$, where the expectation is taken over all possible samples. It can be approximated by

$$E\{\epsilon_k^2\} \approx \frac{1}{N} \sum_{k=1}^N \epsilon_k^2. \quad (3.20)$$

With this formulation of the MSE, we can estimate it closely by simulation over a long enough random input sequence.

Another possible optimization criterion is the minimization of the variance of the net transition shifts, δ_i . Again, we can accurately estimate this by simulating the effects

Table 3.4 Optimal precompensation values for different criteria
 $T_{50}/B=1; \sigma_J/B=0.08; SNR_W=29\text{dB}$

	BER	Shift Variance	MSE
Dibit (Δ/B)	0.08	0.10	0.12
Two-level ($\Delta_H/B, \Delta_L/B$)	(0.09, 0.05)	(0.12, 0.08)	(0.14, 0.12)

of NLTS and precompensation over a long random input sequence and computing the sample variance of the observed net transition shifts.

We numerically determined the optimal precompensation values according to these two criteria for the same channel as used to generate the results in Figure 3.12. For the dibit precompensation scheme, we found that the value which minimizes the MSE is $0.12B$, while the value producing the minimum shift-variance is $0.10B$. Note that these precompensation values differ from that which we previously found to minimize the BER, namely $0.08B$. In this example, we see that the minimum shift-variance criterion gives an optimal precompensation value that is closer to the value that minimizes the BER. Referring to Figure 3.12, we see that, indeed, the BER obtained using this precompensation value is lower than that obtained using the minimum MSE value.

In Table 3.4, we give the corresponding results for the two-level precompensation scheme. Referring to Figure 3.11, we again find that the precompensation levels that minimize the net shift variance give a lower BER than the levels that minimize the MSE.

It is important to note that the results of such a comparison might not hold for different channel parameters. For example, when we use dibit precompensation, the precompensation value that achieves the minimum BER will be larger when T_{50} is larger, according to our simulations. On the other hand, the variance of net transition shifts does not depend on the value of T_{50} . So the precompensation value that minimizes this variance will remain at $0.10B$. For the minimum MSE criterion, we ran simulations for T_{50} equal to $1.2B$, $1.3B$, and $1.5B$. The result is shown in Figure 3.16. In the figure, we normalized the MSE by their minimum values for each case, so that it is easy to compare the optimal values for all the cases. We found that, for all three cases, the

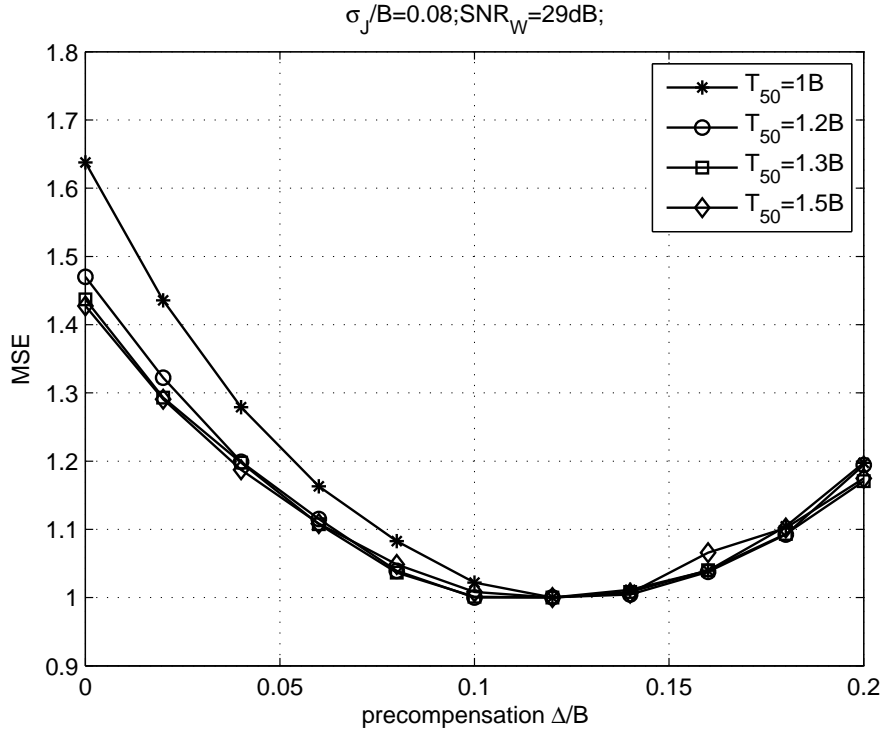


Figure 3.16 Normalized MSE using the dibit precompensation for different T_{50} values. The MSE is normalized by its minimum value for all precompensation levels in each curve. $B = 16\text{nm}$; $T_{50}/B=1, 1.2, 1.3, 1.5$.

optimal precompensation level was $0.12B$. Now, for the channel with $T_{50} = 1.5B$, we found through Monte Carlo simulation that the precompensation value that minimizes the BER is $0.12B$. Hence, in this situation, the minimum MSE criterion is superior to the minimum shift-variance criterion.

Acknowledgement

This chapter is partly a reprint of the materials in following papers Z. Wu, H. Bertram, P. Siegel, and J. Wolf, “Nonlinear transition shift and write precompensation in perpendicular magnetic recording”, Communications, 2008. ICC’08. IEEE International Conference on, pp. 1972-1976, May 2008; Z. Wu, P. H. Paul, J. K. Wolf, and H.

Bertram, “Analysis on nonlinear transition shift and write precompensation in perpendicular magnetic recording”, submitted to JSAC, 2009. The dissertation author is the primary author of the above papers.

Bibliography

- [1] X. Che, “Nonlinearity measurements and write precompensation studies for a PRML recording channel,” *IEEE Trans. Magn.*, vol. 31, no. 6, pp. 3021–3026, Nov. 1995.
- [2] A. Taratorin, J. Fitzpatrick, S. X. Wang, and B. Wilson, “Non-linear interactions in a series of transitions,” *IEEE Trans. Magn.*, vol. 33, no. 1, pp. 956–961, Jan. 1997.
- [3] W. Zhu, D. Kaiser, J. Judy, and D. Palmer, “Experimental study of reader nonlinearity in perpendicular recording using pseudorandom sequences,” vol. 39, no. 5, pp. 2636–2638, Sep. 2003.
- [4] K. Miura, K. Seki, M. Hashimoto, H. Muraoka, H. Aoi, and Y. Nakamura, “Optimization of precompensation for NLTS in perpendicular magnetic recording,” *IEEE Trans. Magn.*, vol. 42, no. 10, pp. 2297–2299, Oct. 2006.
- [5] F. Lim and A. Kavčić, “Optimal precompensation for partial erasure and nonlinear transition shift in magnetic recording using dynamic programming,” in *IEEE Global Telecommunications Conference, Globecom '05*, vol. 1, St. Louis, Missouri, USA, Nov. 28 - Dec. 2 2005.
- [6] H. N. Bertram, *Theory of Magnetic Recording*. Cambridge University Press, 1994.
- [7] K. Nakamoto and H. N. Bertram, “Analytic perpendicular-recording model for transition parameter and NLTS,” *Journal of The Magnetic Society of Japan*, vol. 26, no. 2, pp. 79–85, Feb. 2002.
- [8] J. Moon and W. Zeng, “Equalization for maximum likelihood detectors,” *IEEE Trans. Magn.*, vol. 31, no. 2, pp. 1083–1088, Mar. 1995.

Chapter 4

Analysis of the Bit-Error-Rate for Perpendicular Recording Channels with NLTS

In Chapter 3, we showed the performance of different precompensation schemes for perpendicular recording systems with NLTS. We used the Monte-Carlo method to simulate the bit error rate for each precompensation scheme. The best precompensation values were found by brute force search (for dibit and two-level precompensation) and the performances were compared.

In this chapter, we will develop a method to analyze the bit error rate for the system in Chapter 3. An estimate of the lower bound will be derived and compared to the Monte-Carlo simulation result. The optimal precompensation levels obtained using the lower bound are very close to those found with Monte-Carlo simulation. Since the Monte-Carlo method is time-consuming, it can not be easily used to optimize the precompensation values in a multilevel precompensation scheme. Using the analytical method, the computation time can be greatly reduced and the performance can be determined, as already shown in Chapter 3.

The Viterbi detector was briefly introduced in Chapter 1. The error probability analysis for the Viterbi detector involves the analysis of the error events. In this chapter,

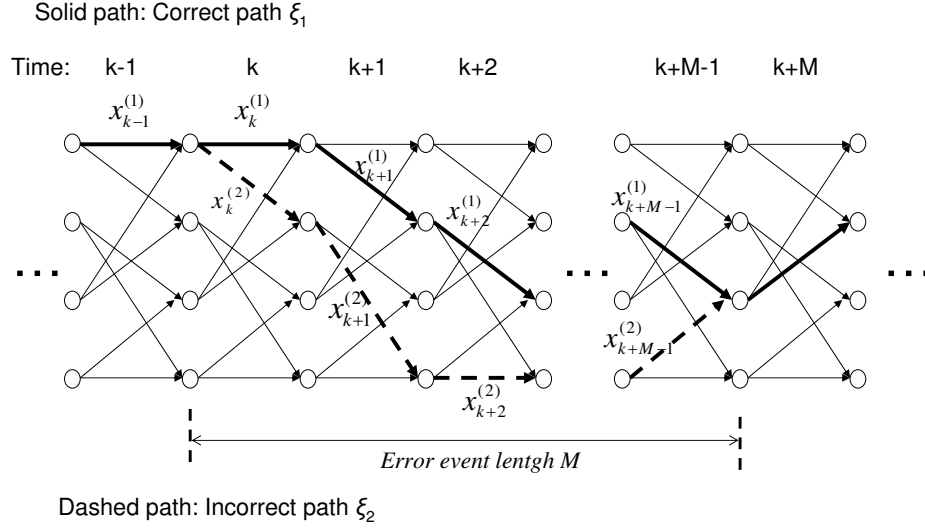


Figure 4.1 Illustration of error event $\epsilon = (\xi_1 \rightarrow \xi_2)$.

we will first analyze the error events in our system. The union bound and a lower bound for the bit error rate will be derived. An estimate of the lower bound will then be proposed and compared with the result of Monte-Carlo simulation.

4.1 Pairwise Error Probability

Suppose we transmit data sequence $\bar{x}^{(1)}$ and after detection, we get $\bar{x}^{(2)}$ as the detected sequence. Let ξ_1 and ξ_2 be the paths in the trellis of Viterbi detector that correspond to $\bar{x}^{(1)}$ and $\bar{x}^{(2)}$, as shown in Figure 4.1. Therefore, path ξ_1 is the correct path and path ξ_2 is the incorrect path. Let's assume that ξ_1 and ξ_2 diverge at time k and remerge at time $k + M$. Between time k and $k + M$, they do not cross at any state.

We usually call the probability that ξ_2 is chosen as the detected path instead of ξ_1 , given that \bar{x}_1 was transmitted, the *pairwise error probability* of the error event $\epsilon = (\xi_1 \rightarrow \xi_2)$. We should note that in our case, the pairwise error probability for event $(\xi_1 \rightarrow \xi_2)$ is different from that for event $(\xi_2 \rightarrow \xi_1)$ because of the data dependent noise and NLTS.

An event error happens when the accumulated metric on the incorrect path is smaller than that on the correct path. In a conventional Viterbi detector, the Euclidean distance is used as the branch metric. Therefore, the pairwise error probability can be expressed as follows:

$$\Pr(\xi_1 \rightarrow \xi_2 | \bar{x}^{(1)}) = \Pr\left(\sum_{i=k}^{k+M-1} (r_i - y_i^{(1)})^2 > \sum_{i=k}^{k+M-1} (r_i - y_i^{(2)})^2 | \bar{x}^{(1)}\right), \quad (4.1)$$

where r_i is the equalized channel output sample when $\bar{x}^{(1)}$ is transmitted, and $y_i^{(1)}$ and $y_i^{(2)}$ are branch output labels at time i in the trellis corresponding to input data sequence $\bar{x}^{(1)}$ and $\bar{x}^{(2)}$, respectively. Assume that the equalization target of the system is $g(D) = g_0 + g_1 D + \dots + g_J D^J$, where D is a unit delay operator. We can write the branch output labels as follows:

$$y_i^{(1)} = \sum_{l=0}^J g_l x_{i-l}^{(1)} \quad (4.2)$$

$$y_i^{(2)} = \sum_{l=0}^J g_l x_{i-l}^{(2)}. \quad (4.3)$$

Let the noise at the input of the detector be $n_i = r_i - y_i^{(1)}$. After some derivations, from equation (4.1) we get

$$\Pr(\xi_1 \rightarrow \xi_2 | \bar{x}^{(1)}) = \Pr\left(\sum_{i=k}^{k+M-1} 2n_i(y_i^{(1)} - y_i^{(2)}) < - \sum_{i=k}^{k+M-1} (y_i^{(1)} - y_i^{(2)})^2 | \bar{x}^{(1)}\right) \quad (4.4)$$

In the above equation, the branch output labels $y_i^{(1)}$ and $y_i^{(2)}$ are deterministic values, given the transmitted sequence and the specific error event. Therefore, the left side of the inequality in the probability calculation is a linear combination of the random noise $n_i, i = k, \dots, k + M - 1$, while the right side is a deterministic value. If we can derive the distribution of the linear combination of the noise, the probability can be calculated directly. We will discuss how to calculate this probability for order-1 and order-2 approximations of the channel with NLTS in the remainder of this section .

4.1.1 Order-1 channel Approximation

In the order-1 channel approximation, the noise at time i can be written as

$$n_i = \sum_j x_{i-j}^{(1)} \tilde{h}_j - \sum_j x_{i-j}^{(1)} g_j + \sum_j d_{i-j}^{(1)} (\delta_{i-j} + a_{i-j}) \tilde{s}'_j + w_i, \quad (4.5)$$

where \tilde{h}_j and \tilde{s}'_j denote the convolution of the FIR equalizer taps with the samples of the channel dibit response and the first derivative of the transition response, respectively. The equalized sample of the AWGN at time k is denoted w_k .

In perpendicular recording systems, \tilde{h}_j and \tilde{s}'_j will vanish when j goes to $+\infty$ and $-\infty$. Let's assume that $\tilde{h}_j = 0$ and $\tilde{s}'_j = 0$ when $i < -A$, or $i > B$. We can rewrite equation (4.5) as follows:

$$n_i = \sum_{j=-A}^B x_{i-j}^{(1)} \tilde{h}_j - \sum_{j=0}^J x_{i-j}^{(1)} g_j + \sum_{j=-A}^B d_{i-j}^{(1)} (\delta_{i-j} + a_{i-j}) \tilde{s}'_j + w_i \quad (4.6)$$

In the above equation, $\{\delta_i\}$ are the NLTS values when $\bar{x}^{(1)}$ is recorded; they are deterministic given $\bar{x}^{(1)}$. The jitter noise values $\{a_i\}$ are i. i. d. random Gaussian distribution variables, and $\{w_i\}$ are colored Gaussian distribution random variables. Therefore n_i , $i = k, \dots, k + M - 1$ are non-zero mean, jointly distributed Gaussian random variables when conditioned on $\bar{x}^{(1)}$.

Thus equation (4.4) can be calculated by the Q-function:

$$\Pr(\xi_1 \rightarrow \xi_2 | \bar{x}^{(1)}) = Q\left(\frac{\bar{\lambda}^T \bar{\lambda} + 2\bar{\lambda}^T \bar{\mu}}{2\sqrt{\bar{\lambda}^T \Sigma \bar{\lambda}}}\right) \quad (4.7)$$

where column vector $\bar{\lambda}$ is $(y_k^{(1)} - y_k^{(2)}, y_{k+1}^{(1)} - y_{k+1}^{(2)}, \dots, y_{k+M-1}^{(1)} - y_{k+M-1}^{(2)})^T$, $\bar{\mu}$ is the mean of random variable vector $\bar{n} = (n_k, n_{k+1}, \dots, n_{k+M-1})^T$ given $\bar{x}^{(1)}$, and Σ is the covariance matrix of \bar{n} given $\bar{x}^{(1)}$. In order to distinguish these from the noise mean and variance of the order-2 channel approximation, we add the subscript 1 to them in the following derivations. The mean and the variance can be written as:

$$\bar{\mu}_1 = \mathbf{H} \cdot \bar{x}^{(1)} - \mathbf{G} \cdot \bar{x}^{(1)} + \mathbf{S}' \cdot \mathbf{D}^{(1)} \cdot \bar{\delta}^{(1)} \quad (4.8)$$

$$\Sigma_1 = \sigma_J^2 \mathbf{S}' \cdot (\mathbf{D}^{(1)})^2 \cdot \mathbf{S}'^T + \sigma_W^2 \mathbf{F} \mathbf{F}^T \quad (4.9)$$

Here, \mathbf{H} , \mathbf{G} , \mathbf{S}' and \mathbf{F} are all Toeplitz matrices of similar form. For example, \mathbf{H} is a $M \times (M + A + B)$ matrix with each row equal to a shifted version of the sequence \tilde{h}_i , in reverse order:

$$\mathbf{H} = \begin{pmatrix} \tilde{h}_B & \tilde{h}_{B-1} & \cdots & \tilde{h}_{-A} & 0 & \cdots & 0 \\ 0 & \tilde{h}_B & \tilde{h}_{B-1} & \cdots & \tilde{h}_{-A} & 0 & \cdots \\ \vdots & \vdots & & \cdots & \cdots & & \vdots \\ 0 & \cdots & 0 & \tilde{h}_B & \tilde{h}_{B-1} & \cdots & \tilde{h}_{-A} \end{pmatrix} \quad (4.10)$$

Similarly, the rows of \mathbf{G} , \mathbf{S}' and \mathbf{F} correspond to the sequences of $\{g_i\}$, $\{\tilde{s}'_i\}$ and the equalizer coefficients. Since sequences \tilde{h}_i and g_i probably are of different lengths, the data vectors multiplied by \mathbf{H} and \mathbf{G} are of different lengths. For example, the data bits involved in multiplication with \mathbf{H} are from time $k - B$ to $k + M - 1 + A$, while the data bits involved in multiplication with \mathbf{G} are from time $k - J$ to $k + M - 1$. To simplify the notation, we omit the index of the data bits in equation (4.8) and (4.9).

$\mathbf{D}^{(1)}$ is a matrix whose diagonal elements are the transition values d_i and zeros elsewhere. We use the superscript (1) to emphasize that here the transitions are for the recorded sequence $\bar{x}^{(1)}$. Similarly, the size of $\mathbf{D}^{(1)}$ and the length of $\bar{\delta}^{(1)}$ are determined by the range of \tilde{s}' .

The derivation of the mean and the variance is very direct when we write out the matrix representation of the noise vector \bar{n} . As we can see, the mean of the noise vector is affected by NLTS while the covariance matrix is not affected. The distance between two paths are changed by NLTS.

4.1.2 Order-2 Channel Approximation

In the order-2 channel approximation, the noise at time i can be written as

$$\begin{aligned} n_i = & \sum_j x_{i-j}^{(1)} \tilde{h}_j - \sum_j x_{i-j}^{(1)} g_j + \sum_j d_{i-j}^{(1)} (\delta_{i-j} + a_{i-j}) \tilde{s}'_j \\ & + \sum_j d_{i-j}^{(1)} \frac{(\delta_{i-j} + a_{i-j})^2}{2} \tilde{s}''_j + w_i. \end{aligned} \quad (4.11)$$

Because of the second derivative term in the above equation, the noise is no longer Gaussian in nature. The joint distribution of the noise is complicated and the exact pairwise error probability cannot be calculated easily.

Therefore, we approximated the pairwise error probability by still using the Q-function in equation 4.7. Of course, the mean and the covariance matrix of the noise are different from those for the order-1 channel approximation. According to the derivations in the Appendix, the mean and the variance for the order-2 channel approximation are represented as follows:

$$\bar{\mu}_2 = \mathbf{H} \cdot \bar{x}^{(1)} - \mathbf{G} \cdot \bar{x}^{(1)} + \mathbf{S}'\mathbf{D}^{(1)}\bar{\delta}^{(1)} + \frac{\sigma_J^2}{2}\mathbf{S}''\mathbf{D}^{(1)} + \mathbf{S}''\mathbf{D}^{(1)}\mathbf{Q}^{(1)}\bar{\delta}/2 \quad (4.12)$$

$$\Sigma_2 = \sigma_J^2(\mathbf{S}' + \mathbf{S}''\mathbf{Q}^{(1)})(\mathbf{D}^{(1)})^2(\mathbf{S}' + \mathbf{S}''\mathbf{Q}^{(1)})^T + \sigma_J^4\mathbf{S}''(\mathbf{D}^{(1)})^2\mathbf{S}''^T + \sigma_W^2\mathbf{F}\mathbf{F}^T \quad (4.13)$$

Here, \mathbf{H} , \mathbf{G} , \mathbf{S}' , $\mathbf{D}^{(1)}$ and \mathbf{F} are the same as in equations (4.8) and (4.9). The matrix \mathbf{S}'' is the Toeplitz matrix in which each row is a shifted version of the sequence \tilde{s}''_i , in reverse order. The size of \mathbf{S}'' can be the same as \mathbf{H} and \mathbf{S}' , since we can always find A and B such that $\tilde{h}_j = 0$, $\tilde{s}'_j = 0$ and $\tilde{s}''_j = 0$ when $i < -A$, or $i > B$, as assumed in equation (4.6).

Matrix $\mathbf{Q}^{(1)}$ in equations (4.12) and (4.13) is the matrix whose diagonal elements are the net transition shifts δ_i for recorded sequence $\bar{x}^{(1)}$, and with zero entries elsewhere.

Comparing the noise mean and covariance matrix in order-1 and order-2 channel approximations, we see that the NLTS affects only the noise mean in the order-1 channel approximation, while in the order-2 channel approximation, NLTS affects both the noise mean and the covariance matrix. Similarly, the jitter noise variance appears only in the covariance matrix calculation in the order-1 channel approximation while in the order-2 channel approximation, it appears both in the mean and the covariance matrix calculation. The AWGN noise variance only appears in the covariance matrix calculation for both cases.

4.2 Upper and Lower Bounds on Bit-Error-Rate

For the Viterbi detector, the union bound on the probability that an error event occurs at time k can be expressed by the summation of the probabilities of all the possible error events that start at time k and end at time $k + M - 1$ for any given recorded data sequence $\bar{x}^{(1)}$:

$$P_E \leq \sum_{\bar{x}^{(1)}} \Pr(\bar{x}^{(1)}) \sum_{\epsilon} \Pr(\epsilon | \bar{x}^{(1)}) \quad (4.14)$$

If we group the error events according to the error event lengths, the union bound of the sequence error probability can also be written as

$$P_E < \sum_{M=M_{\min}}^{\infty} \sum_{\bar{x}^{(1)}} \Pr(\bar{x}^{(1)}) \sum_{\substack{\bar{x}^{(2)} \text{ such that} \\ \epsilon_M = (\xi_1 \rightarrow \xi_2) \in E_M}} \Pr(\epsilon_M | \bar{x}^{(1)}) \quad (4.15)$$

where ϵ_M denotes a length- M error event, M_{\min} is the minimum length of the error event, and E_M is the set of all error events of length M .

The union bound on the bit error probability can be derived from P_E . Denote by $N_b(\epsilon_M)$ the number of erroneous bits corresponding to the error event ϵ_M , i.e., the number of bits by which sequence $\bar{x}^{(1)}$ and $\bar{x}^{(2)}$ differ. The bit error probability is the probability that the bit belongs the set of incorrect bits of an error event. The union bound for the bit error probability is thus given by

$$P_b < \sum_{M=M_{\min}}^{\infty} \sum_{\bar{x}^{(1)}} \Pr(\bar{x}^{(1)}) \sum_{\substack{\bar{x}^{(2)} \text{ such that} \\ \epsilon_M = (\xi_1 \rightarrow \xi_2) \in E_M}} \Pr(\epsilon_M | \bar{x}^{(1)}) N_b(\epsilon_M) \quad (4.16)$$

The union bounds in the above equations are usually very loose. A lower bound on bit error rate can be obtained if the error events we pick are all disjoint. According to the simulation results in Table 3.3, the bit error rate is dominated by the single-bit error, which results from the minimum-length error events. Therefore, we will pick these error events and derive a lower bound.

The minimum-length error events are error events where the two paths ξ_1 and ξ_2 diverge at time k and remerge in the shortest time. This happens when $\bar{x}^{(2)}$ differs from

$\bar{x}^{(1)}$ only in the bit at time k . The minimum length of the error events is thus $M_{\min} = J + 1$. For a given recorded sequence $\bar{x}^{(1)}$, there is only one such error event because the input data is binary and there is only one sequence $\bar{x}^{(2)}$ that can differ from $\bar{x}^{(1)}$ at time k . Therefore, all the minimum-length error events are disjoint since they correspond to different recorded sequences. For each minimum-length error event, $N_b(\epsilon_{M_{\min}}) = 1$.

The lower bound on the bit error probability considering only the minimum length error events can thus be written as follows:

$$P_b > \sum_{\bar{x}^{(1)}} \Pr(\bar{x}^{(1)}) \Pr(\text{minimum-length error event } \xi_1 \rightarrow \xi_2 | \bar{x}^{(1)}) \quad (4.17)$$

Suppose the recorded data sequences are equiprobable. Then equation (4.17) can be written as

$$P_b > \frac{1}{2^{L_E}} \sum_{\bar{x}^{(1)}} \Pr(\text{minimum-length error event } \xi_1 \rightarrow \xi_2 | \bar{x}^{(1)}) \quad (4.18)$$

where L_E is the effective calculation length of the recorded sequence. The value of L_E is determined by the ISI channel memory, the memory of the noise correlation, the data-dependence noise memory, and the memory of the NLTS.

If the noise n_i is AWGN, L_E is just the length of the error events, which is determined by the ISI channel memory. In our case, the noise is data-dependent and has memory. According to equations (4.8), (4.9), (4.12) and (4.13), the data bits involved in calculating the pairwise error probability are determined by the span of $\{\tilde{h}_i\}$, $\{\tilde{s}'_i\}$, $\{\tilde{s}''_i\}$ and $\{g_i\}$. However, there is another factor that will implicitly affect the effective length L_E , namely, the net transition shifts. Theoretically, the data bits are written sequentially and the net transition shift of the current transition is affected by previous transition positions, which in turn have been affected by their preceding transitions. Thus, the memory of NLTS can be tracked back to the very beginning of a track. In this case, L_E needs to be a very large number, which is not practical for computation. Therefore, we will truncate the data sequence and shorten L_E to estimate the lower bound.

One more simplification of the calculation can be applied in the estimation of the bit error rate lower bound considering only the minimum-length error events. The

vector $\bar{\lambda}$ in this scenario is $\bar{\lambda} = 2x_k^{(1)}(g_0, g_1, \dots, g_J)^T$, where $x_k^{(1)}$ is either $+1$ or -1 . Therefore, when calculating $\bar{\lambda}^T \bar{\mu}$ and $\bar{\lambda}^T \Sigma \bar{\lambda}$, we can calculate in advance those vectors such as $|\bar{\lambda}|^T \cdot \mathbf{H}$, which are not related to the recorded data sequence $\bar{x}^{(1)}$. These vectors can then be truncated according to L_E before they are applied to the pairwise error probability calculation.

4.3 Simulation Results

The lower bound estimate can be calculated for both order-1 and order-2 channel approximations. In Figure 4.2, the lower bound is shown for the dibit precompensation scheme using the order-1 channel approximation. The bit error probability calculated using equation (4.18) is compared to the BER verses precompensation value curve simulated in Figure 3.7. All the channel parameters are the same as in Figure 3.7. The value of T_C in the legend of Figure 4.2 determines the effective calculation length L_E , which is $L_E = 2T_C + 1$. The data sequence $\bar{x}^{(1)}$ we considered in the calculation is from time $k - T_C$ to $k + T_C$. We can see that the curves for $L_E = 11$ and $L_E = 15$ are almost identical. For $L_E = 31$, because of the computational complexity, we simulated only the point where no precompensation was used. Still the result is very close to the results corresponding to $L_E = 11$ and $L_E = 15$.

The lower bound curve for the order-1 channel approximation is very close to the Monte-Carlo simulation result, as shown in Figure 4.2. The optimal precompensation value that minimizes the BER can also be deduced from the lower bound estimate. It is the same as the Monte-Carlo simulation, and it took much less time to compute.

Figure 4.3 shows the estimate of the lower bound for the order-2 channel approximation. The channel parameters are the same as in Figure 4.2. We can see that the lower bound is not as tight compared to that for the order-1 channel approximation because of the Q-function approximation in the pairwise error probability calculation. However, the optimal precompensation value obtained by using the estimate is the same as that which resulted from the Monte-Carlo simulation. Again, the difference between

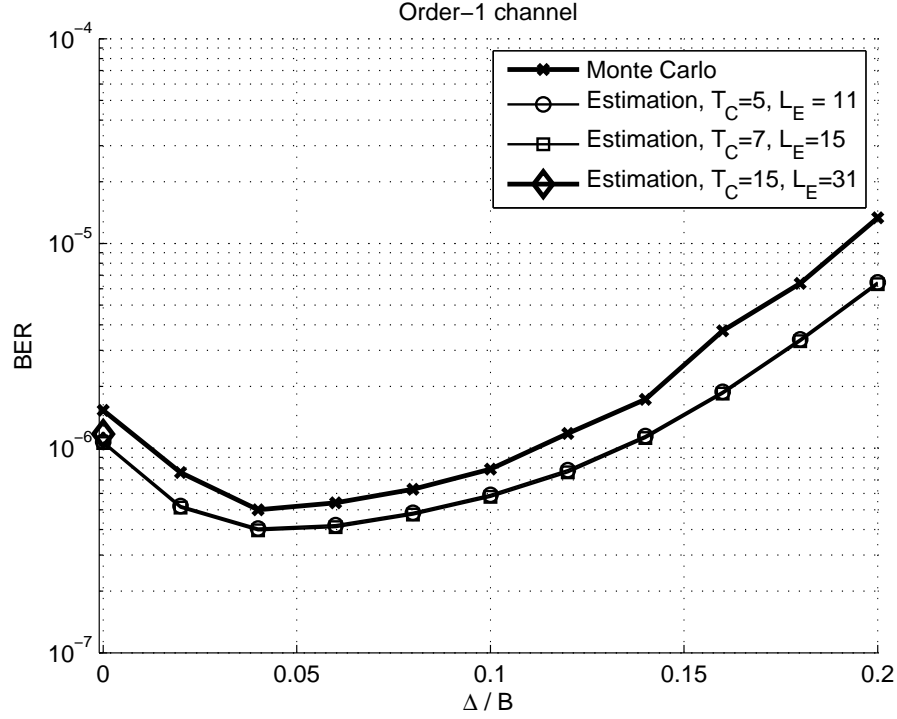


Figure 4.2 The lower bound estimation for order-1 channel approximation with dibit precompensation scheme. $T_{50}/B = 1$; $\sigma_J/B = 0.08$; $SNR_W = 29\text{dB}$; $B = 16\text{nm}$.

$L_E = 11$ and $L_E = 15$ is very small.

The computation time required for the lower bound estimate is much less than required for the Monte-Carlo simulation. For example, the curve obtained by Monte-Carlo simulation in Figure 4.3 took several hours to compute, while the lower bound estimate curve in the same figure took only several minutes. The computation time for the Monte-Carlo simulation is related to number of simulated sectors, which need to increase when the bit error rate decreases. However, the computation time of the estimate does not depend on the bit error rate.

As we have shown in Figure 4.3, the lower bound estimate can give us the same optimal precompensation value for the dibit precompensation scheme in much less time, though the lower bound is not very tight. Similar results were observed for the two-level precompensation and 3-level precompensation schemes. In Table 4.1, we

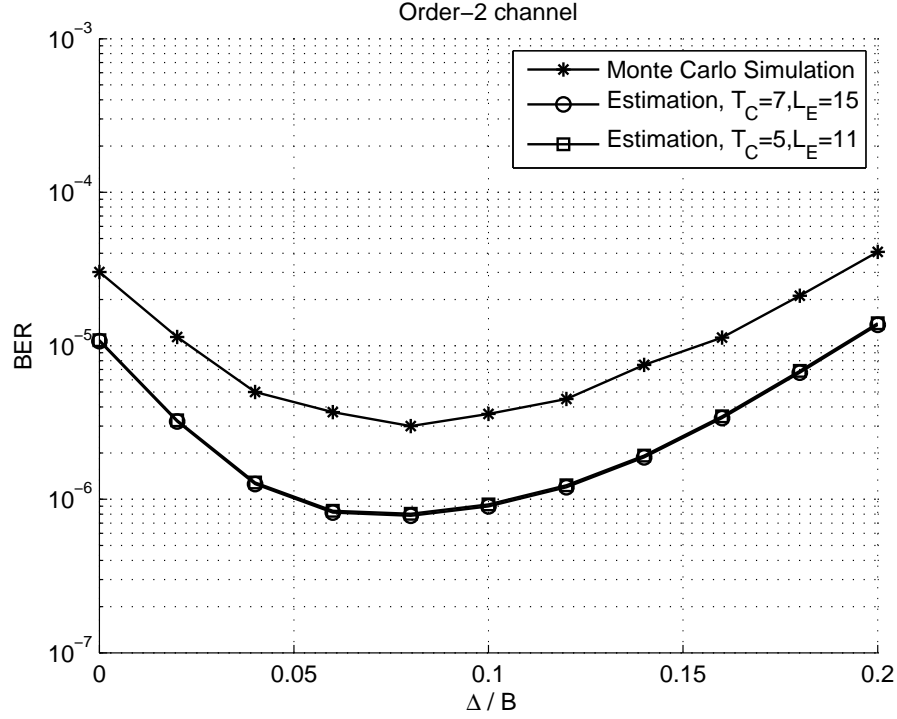


Figure 4.3 The lower bound estimation for order-2 channel approximation with dibit precompensation scheme. $T_{50}/B = 1$; $\sigma_J/B = 0.08$; $SNR_W = 29\text{dB}$; $B = 16\text{nm}$

compare the best precompensation values obtained through the Monte-Carlo simulation and the lower bound estimation. Except for the dibit precompensation scheme, the optimal precompensation values for other schemes are obtained by using the `fminunc` function in MATLAB. This function can give us the minimum value of a function and its corresponding function input when the function input has no constraint. The input of the function can be either a scalar, or a vector. The format of the precompensation values in the table is as follows: $(\Delta_H/B, \Delta_L/B)$ for two-level precompensation, $(\Delta_H/B, \Delta_M/B, \Delta_L/B)$ for the 3-level precompensation, $(\Delta_{(1)}/B, \Delta_{(2)}/B, \dots, \Delta_{(7)}/B)$ for the 7-level precompensation. In the Monte-Carlo simulation, we used a step size of 0.01 when tuning the precompensation values for the two-level precompensation scheme and 0.02 for other schemes.

Table 4.1 Comparison of the optimal precompensation values obtained from the Monte-Carlo simulation and the lower bound estimate.

$$T_{50}/B = 1; \sigma_J/B = 0.08; SNR_W = 29\text{dB}; B = 16\text{nm}$$

	Dibit precomp	Two-level Precomp	3-level Pre-comp	7-level precomp
Monte-Carlo Simulation	0.08	(0.09,0.05)	(0.12,0.06, 0.1)	N/A
Lower Bound Estimation	0.08	(0.088,0.043)	(0.114,0.056, 0.093)	(0.128, 0.113, 0.063, 0.079, 0.123, 0.089, 0.059)

4.4 Application of the Analysis

The BER analysis of the lower bound can be used to find the optimal precompensation values for a multi-level precompensation scheme, when a brute force search by Monte-Carlo simulation is not affordable. We used the lower bound estimate to help compare the performance of the multi-level precompensation schemes in Chapter 3, significantly reducing the required amount of computation time.

The analysis can also be used to explain some of the simulation results. In Chapter 3, we showed the simulation results for various precompensation schemes. One observation is that in some cases, the multi-level precompensation schemes can perform better than the system without NLTS and precompensation.

Table 3.3 compared the error events for the system with NLTS and the best two-level precompensation, with those for the system without NLTS. It shows that the single-bit error event is dominant and the system with two-level precompensation scheme apparently has fewer single-bit error events. Since the minimum-length error events correspond to the single-bit error events, we try to give the reason after observing the distribution of the pairwise error probabilities for the minimum-length error events. Figure 4.4 shows the histograms of the pairwise error probability for all the minimum-length error events for both cases. The x-axis is the logarithm of the pairwise error probability. We divided the x-axis into bins of size 0.1 and count the number of error events with probability falling in each bin. These numbers are shown as the y-axis. The BER is dominated by those error events that have large pairwise error probability, as

indicated roughly in the figure. We can see that those error events that have the largest pairwise error probability in the no-NLTS case, concentrated in the right most peak of the distribution. However, these error events span a wider range along the x-axis for the two-level precompensation case. There are a small amount of the error events whose probabilities increases very slightly, but a large amount of error events that has a decreased probability. The count of the highest error probability events is much less than that of the no-NLTS case. Although the highest error probability is slightly larger in the two-level precompensation case, the overall BER still decreases from 3.59×10^{-7} to 3.03×10^{-7} , according to the lower bound estimate.

The histogram for the error events shows that the effective distance between the correct path and the incorrect path is larger for many single-error events than those in the no-NLTS case. And therefore, the BER is improved.

Acknowledgement

This chapter is partly a reprint of the following paper: Z. Wu, P. H. Paul, J. K. Wolf, and H. Bertram, “Analysis on nonlinear transition shift and write precompensation in perpendicular magnetic recording”, submitted to JSAC, 2009. The dissertation author is the primary author of this paper.

Appendix: Derivation of Noise Mean and Covariance Matrix for Order-2 Channel Approximations

The equalizer output of the order-2 channel approximation is

$$r_k = \sum_{i=0}^J g_i x_{k-i} + \sum_i d_{k-i} [(\delta_{k-i} + a_{k-i}) \tilde{s}'_i + \frac{(\delta_{k-i} + a_{k-i})^2}{2} \tilde{s}''_i] + w_k + q_k \quad (4.19)$$

where $\{g_i\}$ are the target taps, $\{d_i\}$ are transitions, $\{x_i\}$ are binary data bits of +1 and -1, and δ_i is the net shift for transition i . If d_i equals zero, which means there is no transition,

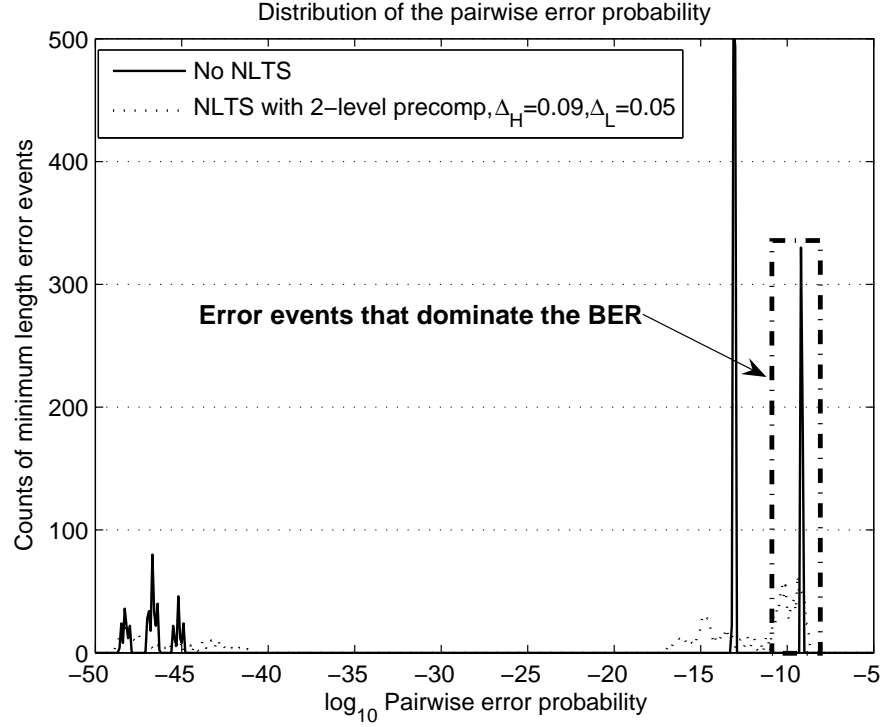


Figure 4.4 The histogram of the pairwise error probability for a system with no NLTS and a system with optimal two-level precompensation. $T_{50}/B = 1$; $\sigma_J/B = 0.08$; $SNR_W = 29\text{dB}$; $B = 16\text{nm}$.

the net shift δ_i equals to zero. The jitter noise is denoted by a_i . It is zero when there is no transition and a Gaussian random variable with zero mean and variance σ_J^2 , otherwise. The sequence $\{\tilde{s}'_i\}$ and $\{\tilde{s}''_i\}$ are the equalized samples of the first-order and second-order derivatives of the transition response. The term w_k is the equalized AWGN, and q_k is the equalizer residue, given by

$$q_k = \sum_i x_{k-i}^{(1)} \tilde{h}_i - \sum_i x_{k-i}^{(1)} g_i. \quad (4.20)$$

Therefore, the noise at the equalizer output can be written as

$$n_k = v_k + w_k + q_k \quad (4.21)$$

where

$$v_k = \sum_i d_{k-i} [(\delta_{k-i} + a_{k-i}) \tilde{s}'_i + \frac{(\delta_{k-i} + a_{k-i})^2}{2} \tilde{s}''_i] \quad (4.22)$$

Now we are going to derive the mean and covariance matrix for n_k conditioned on data sequence \bar{x} . The mean can be written as

$$E\{n_k\} = E\{v_k\} + E\{w_k\} + E\{q_k\} \quad (4.23)$$

Since w_k is colored AWGN with zero mean, and q_k is determined when the data is given, the mean is actually

$$E\{n_k\} = E\{v_k\} + q_k. \quad (4.24)$$

The first term, $E\{v_k\}$, can be derived as follows:

$$\begin{aligned} E\{v_k\} &= E\left\{\sum_i d_{k-i}[(\delta_{k-i} + a_{k-i})\tilde{s}'_i + \frac{(\delta_{k-i} + a_{k-i})^2}{2}\tilde{s}''_i]\right\} \\ &= \sum_i d_{k-i}[\delta_{k-i}\tilde{s}'_i + \frac{E\{(\delta_{k-i} + a_{k-i})^2\}}{2}\tilde{s}''_i] \\ &= \sum_i d_{k-i}[\delta_{k-i}\tilde{s}'_i + \frac{\sigma_J^2 + \delta_{k-i}^2}{2}\tilde{s}''_i] \end{aligned} \quad (4.25)$$

The covariance between n_k and n_{k-m} can be expressed as

$$Cov\{n_k n_{k-m}\} = Cov\{v_k v_{k-m}\} + Cov\{w_k w_{k-m}\} \quad (4.26)$$

since the AWGN is independent with respect to the jitter noise.

The covariance matrix for the colored Gaussian noise is not difficult to derive. With the assumption that the equalizer taps are $\bar{f} = (f_0, f_1, \dots, f_K)$, the covariance matrix for the vector \bar{w} is $\sigma_W^2 \mathbf{F} \mathbf{F}^T$, where \mathbf{F} is a Toeplitz matrix generated by \bar{f} .

Therefore, we now consider the covariance of the v_k :

$$Cov\{v_k v_{k-m}\} = E\{v_k v_{k-m}\} - E\{v_k\}E\{v_{k-m}\} \quad (4.27)$$

The expectation $E\{v_k v_{k-m}\}$ can be derived as follows:

$$\begin{aligned}
E\{v_k v_{k-m}\} &= E\left\{ \sum_i d_{k-i} [(\delta_{k-i} + a_{k-i}) \tilde{s}'_i + \frac{(\delta_{k-i} + a_{k-i})^2}{2} \tilde{s}''_i] \right. \\
&\quad \left. \sum_j d_{k-m-j} [(\delta_{k-m-j} + a_{k-m-j}) \tilde{s}'_j + \frac{(\delta_{k-m-j} + a_{k-m-j})^2}{2} \tilde{s}''_j] \right\} \\
&= \sum_i \sum_j d_{k-i} d_{k-m-j} \left[\tilde{s}'_i \tilde{s}'_j E\{(\delta_{k-i} + a_{k-i})(\delta_{k-m-j} + a_{k-m-j})\} \right. \\
&\quad + \frac{\tilde{s}''_i \tilde{s}''_j}{4} E\{(\delta_{k-i} + a_{k-i})^2 (\delta_{k-m-j} + a_{k-m-j})^2\} \\
&\quad + \frac{\tilde{s}'_i \tilde{s}''_j}{2} E\{(\delta_{k-i} + a_{k-i})(\delta_{k-m-j} + a_{k-m-j})^2\} \\
&\quad \left. + \frac{\tilde{s}''_i \tilde{s}'_j}{2} E\{(\delta_{k-i} + a_{k-i})^2 (\delta_{k-m-j} + a_{k-m-j})\} \right] \\
&= \sum_i \sum_{j: k-i \neq k-m-j} d_{k-i} d_{k-m-j} \left[\tilde{s}'_i \tilde{s}'_j \delta_{k-i} \delta_{k-m-j} \right. \\
&\quad + \frac{\tilde{s}'_i \tilde{s}''_j}{2} \delta_{k-i} (\sigma_J^2 + \delta_{k-m-j}^2) + \frac{\tilde{s}''_i \tilde{s}'_j}{2} \delta_{k-m-j} (\sigma_J^2 + \delta_{k-i}^2) \\
&\quad \left. + \frac{\tilde{s}''_i \tilde{s}''_j}{4} (\sigma_J^2 + \delta_{k-i}^2) (\sigma_J^2 + \delta_{k-m-j}^2) \right] \\
&\quad + \sum_i d_{k-i}^2 \left[\tilde{s}'_i \tilde{s}'_{i-m} (\sigma_J^2 + \delta_{k-i}^2) \right. \\
&\quad + \frac{\tilde{s}'_i \tilde{s}''_{i-m} + \tilde{s}''_i \tilde{s}'_{i-m}}{2} \delta_{k-i} (\delta_{k-i}^2 + 3\sigma_J^2) \\
&\quad \left. + \frac{\tilde{s}''_i \tilde{s}''_{i-m}}{4} (\delta_{k-i}^4 + 6\delta_{k-i}^2 \sigma_J^2 + 3\sigma_J^4) \right] \\
&= \sum_i \sum_j d_{k-i} d_{k-m-j} \left[\tilde{s}'_i \tilde{s}'_j \delta_{k-i} \delta_{k-m-j} + \frac{\tilde{s}'_i \tilde{s}''_j}{2} \delta_{k-i} (\sigma_J^2 + \delta_{k-m-j}^2) \right. \\
&\quad + \frac{\tilde{s}''_i \tilde{s}'_j}{2} \delta_{k-m-j} (\sigma_J^2 + \delta_{k-i}^2) + \frac{\tilde{s}''_i \tilde{s}''_j}{4} (\sigma_J^2 + \delta_{k-i}^2) (\sigma_J^2 + \delta_{k-m-j}^2) \Big] \\
&\quad + \sum_i d_{k-i}^2 \left[\tilde{s}'_i \tilde{s}'_{i-m} \sigma_J^2 + (\tilde{s}'_i \tilde{s}''_{i-m} + \tilde{s}''_i \tilde{s}'_{i-m}) \delta_{k-i} \sigma_J^2 \right. \\
&\quad \left. + \frac{\tilde{s}''_i \tilde{s}''_{i-m}}{2} (2\delta_{k-i}^2 \sigma_J^2 + \sigma_J^4) \right]. \tag{4.28}
\end{aligned}$$

Since

$$\begin{aligned}
E\{v_k\}E\{v_{k-m}\} &= \sum_i d_{k-i} [\delta_{k-i} \tilde{s}'_i + (\sigma_J^2 + \delta_{k-i}^2) \tilde{s}''_i / 2] \\
&\quad \cdot \sum_j d_{k-m-j} [\delta_{k-m-j} \tilde{s}'_j + (\sigma_J^2 + \delta_{k-m-j}^2) \tilde{s}''_j / 2] \\
&= \sum_i \sum_j d_{k-i} d_{k-m-j} \left[\tilde{s}'_i \tilde{s}'_j \delta_{k-i} \delta_{k-m-j} \right. \\
&\quad \left. + \frac{\tilde{s}''_i \tilde{s}''_j}{4} (\sigma_J^2 + \delta_{k-i}^2) (\sigma_J^2 + \delta_{k-m-j}^2) + \frac{\tilde{s}'_i \tilde{s}''_j}{2} \delta_{k-i} (\sigma_J^2 + \delta_{k-m-j}^2) \right. \\
&\quad \left. + \frac{\tilde{s}''_i \tilde{s}'_j}{2} \delta_{k-m-j} (\sigma_J^2 + \delta_{k-i}^2) \right]. \tag{4.29}
\end{aligned}$$

Therefore,

$$\begin{aligned}
Cov\{v_k v_{k-m}\} &= \sum_i d_{k-i}^2 \sigma_J^2 \left[\tilde{s}'_i \tilde{s}'_{i-m} + (\tilde{s}'_i \tilde{s}''_{i-m} + \tilde{s}''_i \tilde{s}'_{i-m}) \delta_{k-i} \right. \\
&\quad \left. + \tilde{s}''_i \tilde{s}''_{i-m} \delta_{k-i}^2 + \frac{\tilde{s}''_i \tilde{s}''_{i-m}}{2} \sigma_J^2 \right] \\
&= \sum_i d_{k-i}^2 \sigma_J^2 \left[(\tilde{s}'_i + \tilde{s}''_i \delta_{k-i}) (\tilde{s}'_{i-m} + \tilde{s}''_{i-m} \delta_{k-i}) + \frac{\tilde{s}''_i \tilde{s}''_{i-m}}{2} \sigma_J^2 \right]. \tag{4.30}
\end{aligned}$$

Therefore, we can express the mean $\bar{\mu}_2$ and the covariance matrix Σ_2 for the order-2 channel approximation as:

$$\bar{\mu}_2 = \mathbf{H} \cdot \bar{x}^{(1)} - \mathbf{G} \cdot \bar{x}^{(1)} + \mathbf{S}' \mathbf{D}^{(1)} \bar{\delta}^{(1)} + \frac{\sigma_J^2}{2} \mathbf{S}'' \mathbf{D}^{(1)} + \mathbf{S}'' \mathbf{D}^{(1)} \mathbf{Q}^{(1)} \bar{\delta} / 2 \tag{4.31}$$

$$\Sigma_2 = \sigma_J^2 (\mathbf{S}' + \mathbf{S}'' \mathbf{Q}^{(1)}) (\mathbf{D}^{(1)})^2 (\mathbf{S}' + \mathbf{S}'' \mathbf{Q}^{(1)})^T + \sigma_J^4 \mathbf{S}'' (\mathbf{D}^{(1)})^2 \mathbf{S}''^T + \sigma_W^2 \mathbf{F} \mathbf{F}^T \tag{4.32}$$

where \mathbf{H} , \mathbf{G} , \mathbf{S}' , $\mathbf{D}^{(1)}$ and \mathbf{F} are the same as in equation (4.8) and (4.9). The rows of \mathbf{H} , \mathbf{G} and \mathbf{S}' are shifted versions of sequences $\{\tilde{h}_i\}$, $\{g_i\}$ and $\{\tilde{s}'_i\}$, respectively, in reverse order. \mathbf{S}'' is a similar Toeplitz matrix where each row is a shifted version of the sequence $\{\tilde{s}''_i\}$, in reverse order. The matrices $\mathbf{D}^{(1)}$ and $\mathbf{Q}^{(1)}$ are diagonal matrices, whose diagonal elements are the transition data d_i and the net shifts δ_i respectively, with zeros elsewhere. The transitions and the net transition shifts are both for the recorded sequence $\bar{x}^{(1)}$.

Chapter 5

Mean-Adjusted Pattern-Dependent Noise Prediction Detector for Perpendicular Systems with Nonlinear Transition Shift

In Chapter 3 and 4, we introduced the modeling of NLTS in the perpendicular recording channel and evaluated the performance of the system with write precompensation. Write precompensation is the predominant method that is used today to counteract the distortion induced by NLTS.

In this chapter, we propose a new detection method designed to reduce the performance degradation resulting from NLTS. We show that the new detector performs significantly better than a conventional Viterbi detector and a pattern-dependent noise prediction (PDNP) detector, while its computational complexity is comparable to that of a PDNP detector. In contrast to write precompensation, which generally requires empirical optimization of the precompensation levels, the new detection algorithm incorporates parameters that reflect the channel nonlinearity and noise statistics. These parameters can be determined either adaptively or through the use of training sequences.

Several equalization and detection techniques have previously been proposed to

reduce the effects of nonlinear distortion and media noise in the magnetic recording readback channel. A Volterra equalizer design method was proposed in [1] to combat channel nonlinearities. Detector design has generally concentrated on techniques that mitigate the effects of nonlinear media noise and noise correlation. In particular, the conventional Viterbi detector has been modified in various ways. For example, in [2, 3], partial local feedback noise prediction was used to reduce the impact of correlated noise and media noise. Kavčić and Moura [4] derived a maximum-likelihood sequence detector (MLSD) for an intersymbol-interference (ISI) channel with data-dependent finite-memory Gauss-Markov noise and applied the detector to an autoregressive (AR) model for the magnetic recording channel. The detector incorporates pattern-dependent noise prediction filters. Moon and Park [5] examined various sub-optimal pattern-dependent noise prediction (PDNP) detectors that offer a trade-off between performance and implementation complexity in the presence of media noise. Zayed and Carley [6] and Sun et al. [7] both proposed a modified Viterbi detector with a data-dependent offset in the branch metric calculation, intended to deal with both nonlinearities and media noise.

In this chapter, we address the data-dependent nature of NLTS and derive a modified PDNP detector for perpendicular recording channels with additive Gaussian noise, transition jitter noise, and NLTS. Computer simulations show that the new detector, which we refer to as the mean-adjusted PDNP (MA-PDNP) detector, improves the bit-error-rate (BER) performance when compared to the conventional Viterbi detector and the PDNP detector. In our simulations, we calculate the NLTS according to the model proposed by Bertram and Nakamoto [8, 9] and illustrated in Chapter 3. We also show that the MA-PDNP detector can be combined with write precompensation schemes to achieve further performance improvement.

We will first give a brief introduction to the derivation of the PDNP detector and then describe the structure of the MA-PDNP detector. The simulation results and comparisons between the MA-PDNP detector, Viterbi detector, and PDNP detector will then be given. The performance of the MA-PDNP detector combined with a dibit precompensation scheme will also be presented.

5.1 Channel Model and Notations

The channel model we considered in this chapter is the same as in Chapter 3 and Chapter 4, which considers the NLTS, jitter noise and AWGN. More precisely, we will use the order-2 channel approximation in this chapter. The notation is also the same as in Chapter 3 and Chapter 4. The system diagram is the same as in Figure 3.5. In this section, we will briefly review the order-2 channel approximation.

The order-2 channel approximation incorporates both first- and second-derivative terms of the Taylor expansion of the transition response, given by

$$\begin{aligned} z(t) \approx & \sum_i d_i s(t - iB) + \sum_i d_i (\delta_i + a_i) s'(t - iB) \\ & + \sum_i d_i \frac{(\delta_i + a_i)^2}{2} s''(t - iB) + n_W(t). \end{aligned} \quad (5.1)$$

The discrete-time signal at the detector input can thus be written as

$$\begin{aligned} r_k = & \sum_i x_i \tilde{h}_{k-i} + \sum_i d_i (\delta_i + a_i) \tilde{s}'_{k-i} \\ & + \sum_i d_i \frac{(\delta_i + a_i)^2}{2} \tilde{s}''_{k-i} + w_k, \end{aligned} \quad (5.2)$$

where \tilde{h}_j , \tilde{s}'_j and \tilde{s}''_j denote the convolution of the FIR equalizer taps with the samples of the channel dibit response, the first derivative of the transition response, and the second derivative of the transition response, respectively. The binary input data sequence to the channel is represented by $\{x_i\}$. The corresponding transition sequence is given by $\{d_i\}$, where $d_i = \frac{x_i - x_{i-1}}{2}$. The jitter noise and the net transition shift for transition d_i is represented by a_i and δ_i , respectively. They are zero when d_i is zero, i.e., no transition at time iB . The jitter noise a_i is modeled as independent Gaussian random variables with variance σ_j^2 for transitions d_i not equal to zero. The equalized sample of the AWGN at time k is denoted w_k , where $w_k = n_W(kB)$.

With an equalization target $g(D) = g_0 + g_1 D + \dots + g_J D^J$, the equalizer output

can be written as

$$r_k = \sum_{i=0}^J g_i x_{k-i} + n_k, \quad (5.3)$$

where

$$n_k = \sum_i d_i (\delta_i + a_i) \tilde{s}'_{k-i} + \sum_i d_i \frac{(\delta_i + a_i)^2}{2} \tilde{s}''_{k-i} + w_k + q_k. \quad (5.4)$$

The noise n_k , which has non-zero mean and a non-Gaussian density, includes contributions from the NLTS, transition jitter noise, equalized AWGN, and misequalization error q_k , which is

$$q_k = \sum_i x_{k-i} \tilde{h}_i - \sum_{i=0}^J x_{k-i} g_i. \quad (5.5)$$

5.2 Pattern-Dependent Noise Predictive Detector

The conventional Viterbi algorithm with squared-Euclidean metric is an MLSD only if the noise term n_k in (5.3) is sampled AWGN. The number of trellis states in such a Viterbi detector is 2^J , where $J + 1$ is the length of the equalization target. For a channel with zero-mean, data-dependent, finite-memory Gauss-Markov noise, the MLSD was derived in [4]. We refer to this detector as the pattern-dependent noise-predictive (PDNP) detector. In this section, we will briefly discuss the derivation of the PDNP detector

Assume the channel output is

$$r_k = y_k + n_k \quad (5.6)$$

where the following Markovian properties hold

$$\Pr(S_k | \bar{S}_1^{k-1}) = \Pr(S_k | S_{k-1}) \quad (5.7)$$

$$\Pr(r_k | \bar{S}_1^n, \bar{r}_1^{k-1}) = \Pr(r_k | S_{k-1}, S_k, \bar{r}_{k-L}^{k-1}). \quad (5.8)$$

In the above equations, S_k is the state at time k defined by channel inputs and L is the Markov memory length of the noise.

The MLSD is a detector that gives an estimate of the input sequence that maximizes the likelihood probability

$$\Pr(\bar{r}_1^N | \bar{x}_1^N) = \prod_k \Pr(r_k | \bar{r}_1^{k-1}, \bar{S}_1^N), \quad (5.9)$$

or minimizes the following:

$$-\ln \Pr(\bar{r}_1^N | \bar{x}_1^N) = -\sum_k \ln \Pr(r_k | \bar{r}_1^{k-1}, \bar{S}_1^N). \quad (5.10)$$

The branch metric of the trellis is thus

$$\mathcal{M}(e_k) = -\ln \Pr(r_k | \bar{r}_1^{k-1}, \bar{S}_1^N), \quad (5.11)$$

where e_k is a trellis branch at time k , from state S_{k-1} to S_k .

Using the Markovian property,

$$\mathcal{M}(e_k) = -\ln \Pr(r_k | \bar{r}_{k-L}^{k-1}, S_{k-1}, S_k) \quad (5.12)$$

$$= -\ln \frac{\Pr(\bar{r}_{k-L}^k | S_k = m, S_{k-1} = m')}{\Pr(\bar{r}_{k-L}^{k-1} | S_k = m, S_{k-1} = m')}. \quad (5.13)$$

According to equation (2.30), the branch metric \mathcal{M} can be written as

$$\mathcal{M}(e_k) = -\ln \frac{\gamma(m', m)}{\Pr(S_k = m | S_{k-1} = m')}, \quad (5.14)$$

where $\gamma(m', m)$ is defined in equation (2.14).

Therefore, according to the derivations in Chapter 2, it is not difficult to write the branch metric for the MLSD as:

$$\begin{aligned} \mathcal{M}(e_k) = & \log(\sqrt{2\pi\sigma^2(e_k)}) \\ & + \frac{1}{2\sigma^2(e_k)} \left[\sum_{i=0}^L p_i(e_k)(r_{k-i} - y_{k-i}) \right]^2, \end{aligned} \quad (5.15)$$

where $\sigma^2(e_k)$ is the equivalent noise variance for branch e_k and $p_j(e_k), j = 1, \dots, L$ are the noise prediction coefficients for branch e_k . By comparing equations 2.34 and 5.15, the equivalent variance and the noise prediction coefficients can be represented as

$$\sigma^2(e_k) = r_{LL} - (\bar{r}^{(L)})^T (\mathbf{R}^{(L-1)})^{-1} \bar{r}^{(L)}, \quad (5.16)$$

$$\bar{p}(e_k) = -(\mathbf{R}^{(L-1)})^{-1} \bar{r}^{(L)}, \quad (5.17)$$

where $\bar{p}(e_k) = (p_L(e_k), \dots, p_1(e_k))^T$ and $p_0(e_k) = 1$. The matrix $\mathbf{R}^{(L)}$ is the covariance matrix for noise vector $\bar{n} = (n_{k-L}, \dots, n_k)$ and $\mathbf{R}^{(L-1)}$ is the covariance matrix for $\bar{n}' = (n_{k-L}, \dots, n_{k-1})$. The relationship between $\mathbf{R}^{(L)}$ and $\mathbf{R}^{(L-1)}$ is shown in equation (2.28). Since the noise samples $\{n_k\}$ are data dependent, the covariance matrices $\mathbf{R}^{(L)}$ and $\mathbf{R}^{(L-1)}$ vary from branch to branch.

Compared to the conventional Viterbi detector, the trellis complexity for the PDNP detector is increased because of the memory length of the Markovian noise as well as the span of its data dependence. For example, suppose that the signal-dependent noise n_k depends on the data values in positions $k - C$ to $k + D$. Then, the trellis state S_k at time k is defined by $S_k = (x_{k-L-\max\{C,J\}+1}, \dots, x_{k+D})$, and the number of states is therefore $2^{L+D+\max\{C,J\}}$.

The noise in a magnetic recording channel is not finite-memory Gauss-Markov, so the PDNP detector is, strictly speaking, not optimal, although it achieves near-optimal performance [5]. Several methods have been proposed to reduce the complexity of the PDNP detector, yielding a variety of simpler, yet still effective detectors [2, 3, 5]. Strategies include reducing the prediction filter length, limiting the required number of predictors by shortening the data-dependence length, and eliminating trellis states by using feedback of tentative decisions.

5.3 Mean-Adjusted Pattern-Dependent Noise Predictive Detector

As we discussed in Section 5.1, the noise n_k in (5.4) is non-Gaussian with non-zero mean. We have derived in Chapter 4 that the NLTS affects both the equivalent noise mean and variance, as well as the jitter noise. However, unlike the jitter noise, the NLTS is a deterministic, data-dependent nonlinear effect. To account for the NLTS, we propose a mean-adjusted PDNP (MA-PDNP) detector in which the branch metric for a

branch e_k is written as:

$$\begin{aligned} \mathcal{M}(e_k) = & \log(\sqrt{2\pi\sigma^2(e_k)}) \\ & + \frac{1}{2\sigma^2(e_k)} \left[\sum_{i=0}^L p_i(e_k)(r_{k-i} - y_{k-i} - m_i(e_k)) \right]^2, \end{aligned} \quad (5.18)$$

where $m_i(e_k)$ represents the data-dependent mean of the noise n_{k-i} .

The methods used to reduce the complexity of the PDNP detector can also be applied to the MA-PDNP detector. More precisely, we can rewrite the branch metric in (5.18) as

$$\begin{aligned} \mathcal{M}(\bar{x}_k) = & \log(\sqrt{2\pi\sigma^2(\bar{x}_k)}) \\ & + \frac{1}{2\sigma^2(\bar{x}_k)} \left[\sum_{i=0}^{L_y} p_i(\bar{x}_k)(r_{k-i} - y_{k-i} - m_i(\bar{x}_k)) \right]^2, \end{aligned} \quad (5.19)$$

where \bar{x}_k represents the data pattern from which the noise variance and the prediction filter coefficients corresponding to time k are determined. Since the noise is not a finite-memory Markovian process, we can not assign a value for the memory length L . Instead, we specify a value L_y to represent the length of the noise prediction filters, and we denote the span of the data-dependence by $L_x + 1$. The number of trellis states is given by 2^M . When $M < L_x$, the data pattern \bar{x}_k includes tentative decisions from the survivor path ending at the initial state of the branch e_k .

The computational complexity of the MA-PDNP detector depends upon the parameters M and L_y , while L_x determines the memory size required to store the pattern-dependent mean, variance and noise prediction filter coefficients. Thus, the overall implementation complexity of the MA-PDNP detector is comparable to that of the corresponding PDNP detector.

The branch metric calculation requires the values of the data-dependent mean, variance and filter coefficients. These parameters are derived from the channel noise statistics which can be determined by means of a training sequence, or adaptively by using an algorithm such as that proposed in [10, 7], or a combination of these two methods. Of course, an adaptive scheme requires extra computation to determine the

noise variances and filter coefficients using (5.16) and (5.17) at each time the noise statistics get updated.

5.4 Simulation Results

For the NLTS calculation, we set the medium to soft-underlayer spacing to 20nm, and the medium thickness is set to 10nm. The channel spacing is 16nm, corresponding to a linear density of about 1.59×10^6 bits/inch. The remanent magnetization to head field gradient ratio is set to 1.5. With these parameters, the NLTS of the isolated dibit pattern is about 20% of the channel bit spacing B .

The simulation uses pseudorandom input data divided into 5000-bit sectors. The equalizer utilizes the minimum mean-squared error (MMSE) monic constraint design [11]. The equalization target has length 3 and the number of FIR equalizer taps is set to 15. The noise statistics are obtained by means of a training sequence.

A comparison between the Viterbi detector, the PDNP detector, and the MA-PDNP detector is shown in Fig. 5.1. The normalized jitter noise variance is $\sigma_J/B = 0.1$. The bit density is set to $T_{50}/B = 1$. The BER for different detectors is plotted versus SNR_W . The PDNP detector and the MA-PDNP detector have the same number of states as the Viterbi detector, $M = J = 2$. The data-dependence parameter is set to $L_x = 6$ for both the PDNP detector and the MA-PDNP detector. The two detectors also have the same noise prediction filter length $L_y = 3$. From the figure, it is clear that the MA-PDNP detector is superior to the PDNP detector, which, in turn, outperforms the conventional Viterbi detector. The intuitive explanation for this relative behavior is that the pattern-dependent noise prediction reduces the effect of the correlated jitter noise, while the pattern-dependent mean compensates for the NLTS.

Fig. 5.2 compares MA-PDNP detector performance with various values of the noise prediction length L_y and the data-dependence length L_x , assuming $M = J = 2$. Note that the performance achieved with $L_x = 4$ is very close to that obtained with $L_x = 6$. We also note that the BER at $SNR_W = 26\text{dB}$ is nearly identical for $L_y =$

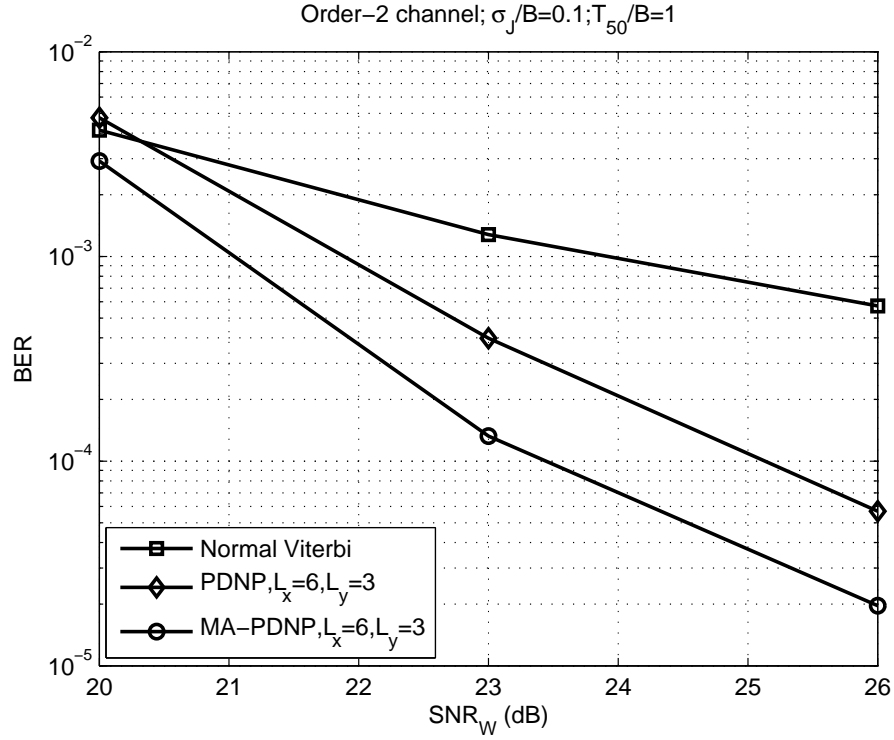


Figure 5.1 Comparison between Viterbi, PDNP, and MA-PDNP detectors.

1, 2, 3. Even when we set $L_y = 0$, corresponding to no noise prediction, the BER is only slightly higher.

The MA-PDNP detector can also be used in combination with write precompensation. To illustrate this, we simulated the BER for the MA-PDNP detector used in conjunction with a dibit precompensation scheme, with the results shown in Fig. 5.3. The channel parameters were set to $SNR_W = 26\text{dB}$, $\sigma_J/B = 0.1$ and $T_{50}/B = 1$. The parameter settings for the NLTS calculation were the same as in the previously discussed simulations. We set $L_x = 6$, and used the same number of trellis states in the MA-PDNP detector as in the conventional Viterbi detector, i.e., $M = J = 2$. The figure shows that for a channel with NLTS, the MA-PDNP detector can achieve a much lower BER than a Viterbi detector even with dibit precompensation. For prediction filter lengths $L_y = 1, 2$, we see that the MA-PDNP detector with dibit precompensation

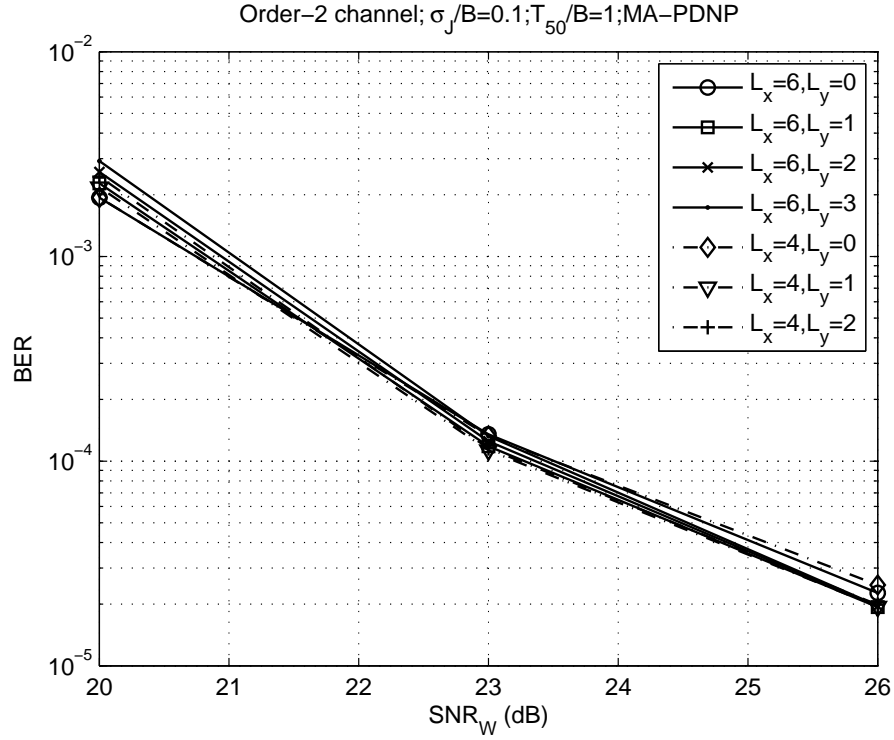


Figure 5.2 Comparison between different L_y and L_x .

achieves the same performance as that of a PDNP detector applied to a channel with no NLTS. This is true over a wide range of precompensation values.

5.5 Conclusions

In this chapter, we presented a mean-adjusted pattern-dependent noise-prediction (MA-PDNP) detector for perpendicular recording channels with nonlinear transition shift (NLTS), transition jitter, and additive Gaussian noise. The new detector reduces the performance degradation caused by data-dependent NLTS and media noise. According to simulation results for an order-2 channel approximation, the MA-PDNP detector improves the performance significantly as compared to both the conventional Viterbi detector and the PDNP detector. At the same time, the computational complexity of the

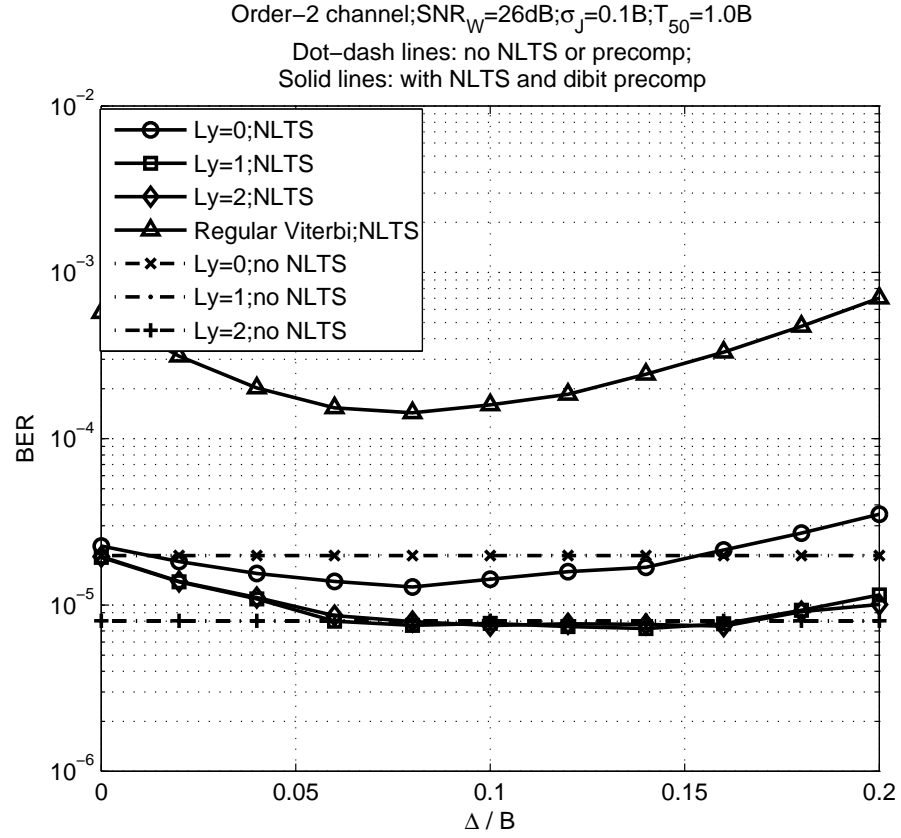


Figure 5.3 MA-PDNP detector with dibit precompensation.

MA-PDNP detector is comparable to that of the PDNP detector. The MA-PDNP detector can also be combined with write precompensation to achieve further performance improvements. Simulation results show that, when used with a simple dibit precompensation technique, the MA-PDNP detector provides the same BER performance as a PDNP detector applied to a channel with no NLTS, over a wide range of precompensation values.

Acknowledgement

This chapter is a reprint of the paper Z. Wu, P. H. Siegel, J. K. Wolf, and H. N. Bertram, “Mean-adjusted pattern dependent noise prediction for perpendicular recording channels with nonlinear transition shift”, *IEEE Trans. Magn.*, vol. 44, no. 11, pp. 3761-3764, Nov. 2008. The dissertation author is the primary author of this paper.

Bibliography

- [1] W. E. Ryan and A. Gutierrez, “Performance of adaptive Volterra equalizers on nonlinear magnetic recording channels,” *IEEE Trans. Magn.*, vol. 31, no. 6, pp. 3054–3056, Nov. 1995.
- [2] S. A. Altekari and J. K. Wolf, “Improvements in detectors based upon colored noise,” *IEEE Trans. Magn.*, vol. 34, no. 1, pp. 94–97, Jan. 1998.
- [3] J. Caroselli, S. A. Altekari, P. McEwen, and J. K. Wolf, “Improved detection for magnetic recording systems with media noise,” *IEEE Trans. Magn.*, vol. 33, no. 5, pp. 2779–2781, Sep. 1997.
- [4] A. Kavčić and J. M. Moura, “The Viterbi algorithm and Markov noise memory,” *IEEE Trans. Inf. Theory*, vol. 46, no. 1, pp. 291–301, Jan. 2000.
- [5] J. Moon and J. Park, “Pattern-dependent noise prediction in signal-dependent noise,” *IEEE J. Sel. Areas Commun.*, vol. 19, no. 4, pp. 730–743, Apr. 2001.
- [6] N. M. Zayed and L. R. Carley, “Equalization and detection for nonlinear recording channels with correlated noise,” *IEEE Trans. Magn.*, vol. 35, no. 5, pp. 2295–2297, Sep. 1999.
- [7] H. Sun, G. Mathew, and B. Farhang-Boroujeny, “Detection techniques for high-density magnetic recording,” *IEEE Trans. Magn.*, vol. 41, no. 3, pp. 1193–1199, Mar. 2005.
- [8] H. N. Bertram, *Theory of Magnetic Recording*. Cambridge University Press, 1994.
- [9] K. Nakamoto and H. N. Bertram, “Analytic perpendicular-recording model for transition parameter and NLTS,” *The Magnetic Society of Japan*, vol. 26, no. 2, pp. 79–85, 2002.

- [10] A. Kavčić and J. M. Moura, "Correlation-sensitive adaptive sequence detection," *IEEE Trans. Magn.*, vol. 34, no. 3, pp. 763–771, May 1998.
- [11] J. Moon and W. Zeng, "Equalization for maximum likelihood detectors," *IEEE Trans. Magn.*, vol. 31, no. 2, pp. 1083–1088, Mar. 1995.

Chapter 6

Error-Locating Codes and Soft-Decision Decoding

Error-locating (EL) codes were introduced first by Wolf and Elaspas in 1963 [1] and extended in [2, 3]. These codes can detect errors in sub-blocks, based on the idea of protecting the syndrome of each sub-block by another error correcting code. This idea was extended to a general case [4], where the syndromes corresponding to different codes of each sub-block are protected by different error correcting codes respectively. We call this type of codes generalized error-locating (GEL) codes. In general, EL and GEL codes have the ability to adjust their error detection/correction capability according to the channel, while decreasing the number of parity bits used.

Both EL and GEL codes have been proposed for application in magnetic recording systems [5, 6]. In [5], a tensor-product parity (TPP) code, which is a simple case of an EL code, is proposed to be concatenated with the Reed-Solomon (RS) code. Such a system outperforms a system with RS code only, and a system with a usual parity code concatenated with a RS code. In [6], a methodology to design a GEL code that adapts to the given magnetic recording channel is proposed. The GEL code was compared to a RS code of the same rate for several types of channels, including a PR4 channel with AWGN, ME2PR4 channel with AWGN and other channels. The performance of the GEL code was shown to be better than the RS code.

In the first part of this chapter, we will discuss soft-decision decoding of the TPP code in an ISI channel. Soft-decision decoding is now widely used for various types of codes, such as the Turbo codes, LDPC codes, Reed-Solomon codes, etc. Simulation results will be presented for an example for ISI channel with AWGN. However, the soft-decision decoding method we proposed can be applied to more sophisticated magnetic recording channels.

In the second part of this chapter, we will discuss the performance of a list decoding algorithm for GEL codes that use RS codes as the outer codes. Simulation results will be presented for a simple GEL code used in conjunction with an AWGN channel.

6.1 Tensor-Product Parity Code and Soft-Decision Decoding

In this section, we will first describe the structure of an error-locating code. After that, a TPP code will be introduced as a special case of an EL code. A hard decision decoding algorithm for TPP codes in ISI channels was proposed in [5]. We will propose two soft-decision decoding procedures for TPP codes. Simulation results will be presented for a TPP code using a convolutional code as the outer code. The performance advantage of soft-decision decoding as compared with hard decision decoding will be shown.

6.1.1 Introduction to Error-Locating Codes

Before we describe error-locating codes, we would like to first introduce some notation which we will use in this chapter. Assume a block code \mathcal{C} defined on a Galois field $\text{GF}(q)$. The code length of \mathcal{C} is N and the code dimension is K . The code rate is thus $R = K/N$. The minimum Hamming distance of code \mathcal{C} is d_{\min} . We denote such a code by $\mathcal{C}(q; N, K, d_{\min})$.

An error-locating code can be defined in terms of two component codes. Assume

the outer code is $\mathcal{A}(p^q; N_a, K_a, d_{\min,a})$ and the inner code is $\mathcal{B}(p; N_b, K_b, d_{\min,b})$, where $q = N_b - K_b$. Assume that a codeword of an EL code is represented by a matrix \mathbf{C} of size $N_b \times N_a$, the elements of which are from $\text{GF}(p)$. Denote the columns of \mathbf{C} by \mathbf{c}_i , where $i = 1, \dots, N_a$. The inner code \mathcal{B} is used to calculate the syndrome of each column of \mathbf{C} :

$$\mathbf{a}_i = \mathbf{H}_B \mathbf{c}_i, \quad i = 1, \dots, N_a \quad (6.1)$$

where \mathbf{H}_B is the parity check matrix for code \mathcal{B} . Therefore, \mathbf{a}_i is a column vector of length $N_b - K_b$ over $\text{GF}(p)$. Let \mathbf{a}_i represent a symbol from $\text{GF}(p^{N_b-K_b})$. The definition of an EL codes requires that the symbol vector $\mathbf{a} = (\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_{N_a})$ must form a codeword of code \mathcal{A} .

Consider the following example. A TPP code is defined as a binary code, with codeword size 3×7 . Therefore, $p = 2$. The inner code is a single parity check code of length 3, i.e., $\mathcal{B}(2; 3, 2, 2)$. The parity check matrix of \mathcal{B} is thus $\mathbf{H}_B = [1 \ 1 \ 1]$. Therefore, $q = N_b - K_b = 3 - 2 = 1$. The outer code is also a binary code. We define the outer code $\mathcal{A}(2; 7, 4, 3)$ to be a Hamming code of length 7. The following 3×7 matrix is a codeword of this EL code:

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}. \quad (6.2)$$

We now proceed to check the correctness. The syndrome for each code is a one bit symbol. Multiply \mathbf{H}_B with each column of \mathbf{C} , to obtain the vector $\mathbf{a} = (0, 0, 1, 1, 1, 0, 1)$. The parity check matrix for the outer code \mathcal{A} is defined as follows:

$$\mathbf{H}_A = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}. \quad (6.3)$$

It can be verified that the vector \mathbf{a} satisfies $\mathbf{H}_A \mathbf{a}^T = \mathbf{0}$.

We can further show the parity check matrix for this EL code. Write the codeword \mathbf{C} in the vector form $\bar{\mathbf{C}}$ by taking the elements in the matrix column-wise, i.e.,

let

$$\bar{\mathbf{C}} = (101 \ 011 \ 100 \ 111 \ 111 \ 000 \ 100)^T. \quad (6.4)$$

Then $\bar{\mathbf{c}}$ satisfy the equation

$$\mathbf{H}_{EL} \bar{\mathbf{C}} = \mathbf{0}, \quad (6.5)$$

where \mathbf{H}_{EL} is the tensor product of \mathbf{H}_A and \mathbf{H}_B :

$$\mathbf{H}_{EL} = \mathbf{H}_A \otimes \mathbf{H}_B = \begin{pmatrix} h_{11}^A \mathbf{H}_B & h_{12}^A \mathbf{H}_B & \cdots & h_{1N_a}^A \mathbf{H}_B \\ h_{21}^A \mathbf{H}_B & h_{22}^A \mathbf{H}_B & \cdots & h_{2N_a}^A \mathbf{H}_B \\ \vdots & \vdots & \vdots & \vdots \\ h_{r_a 1}^A \mathbf{H}_B & h_{r_a 2}^A \mathbf{H}_B & \cdots & h_{r_a N_a}^A \mathbf{H}_B \end{pmatrix} \quad (6.6)$$

$$= \begin{pmatrix} 111 & 111 & 111 & 000 & 111 & 000 & 000 \\ 000 & 111 & 111 & 111 & 000 & 111 & 000 \\ 111 & 111 & 000 & 111 & 000 & 000 & 111 \end{pmatrix}, \quad (6.7)$$

Let the element in the i^{th} row and j^{th} column of matrix \mathbf{H}_A be represented by h_{ij}^A . The variable r_a is the number of rows of the parity check matrix \mathbf{H}_A , which is equal to $N_a - K_a$.

In general, the parity check matrix for any EL code can be represented by the tensor product of the parity matrix \mathbf{H}_A and \mathbf{H}_B , as shown in equation (6.6). However, since \mathbf{H}_B is a matrix with elements from $\text{GF}(p)$ while h_{ij}^A is an element on $\text{GF}(p^q)$, when multiplying h_{ij}^A and \mathbf{H}_B , h_{ij}^A uses the matrix representation of elements in $\text{GF}(p^q)$ [7], where each component of such a matrix is from $\text{GF}(p)$. The corresponding codeword is the vector $\bar{\mathbf{C}}$, composed by taking the elements in \mathbf{C} column-wise, i.e., $\bar{\mathbf{C}} = (\mathbf{c}_1^T, \mathbf{c}_2^T, \dots, \mathbf{c}_{N_a}^T)$.

Since the number of parity symbols, i.e., the number of rows in matrix \mathbf{H}_{EL} , is $(N_b - K_b) \times (N_a - K_a)$, the code rate of the EL code is thus $R = 1 - \frac{(N_b - K_b)(N_a - K_a)}{N_a N_b} = \frac{N_a K_b + N_b K_a - K_a K_b}{N_a N_b}$. The code rate is greater than or equal to any of the code rates of the component codes \mathcal{A} and \mathcal{B} . However, the minimum distance of the EL code is no greater than the minimum distance of any component codes, which is $\min\{d_{\min,a}, d_{\min,b}\}$.

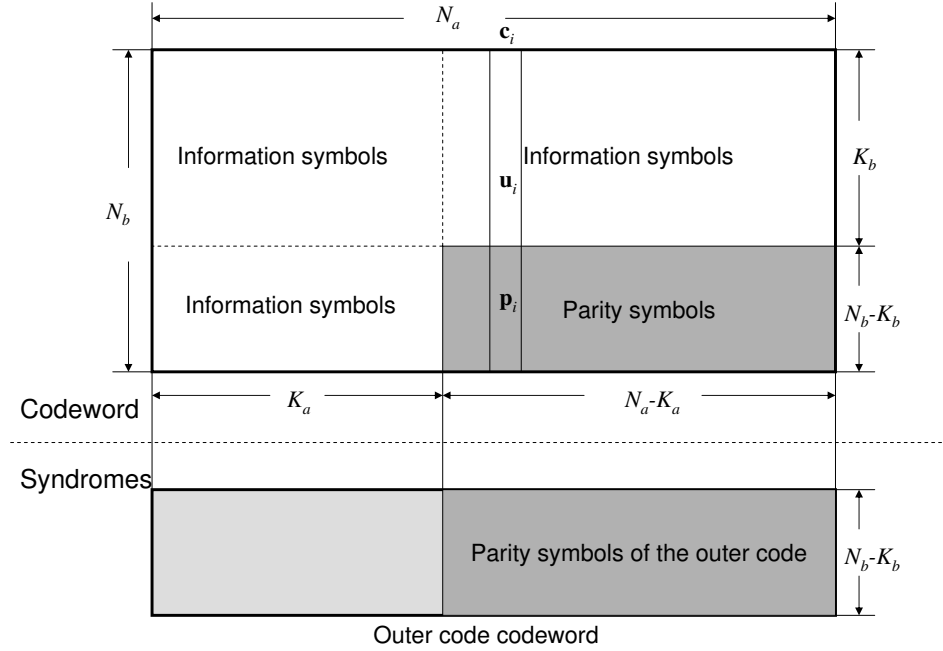


Figure 6.1 A systematic encoding procedure for error-locating codes.

Encoding of EL codes

A systematic encoding procedure for the EL codes can be described as follows:

1. Let the information symbols fill in all the rows of the codeword matrix \mathbf{C} from column 1 to column K_a and rows from 1 to K_b in column $K_a + 1$ to N_a , as shown in Figure 6.1.
2. Calculate the syndromes for columns from 1 to K_a . They are the information symbols for the outer code \mathcal{A} .
3. Encode the information symbols $\mathbf{a}_1, \dots, \mathbf{a}_{K_a}$ with a systematic encoder of code \mathcal{A} . The codeword of the outer code is thus formed as $(\mathbf{a}_1, \dots, \mathbf{a}_{K_a}, \mathbf{a}_{K_a+1}, \dots, \mathbf{a}_{N_a})$.
4. For columns from $K_a + 1$ to N_a of codeword \mathbf{C} , calculate the parity symbols,

shown as the shaded area of codeword \mathbf{C} in Figure 6.1. More precisely, if the parity check matrix for code \mathcal{B} is in the systematic form

$$\mathbf{H}_B = [\mathbf{P} \mathbf{I}], \quad (6.8)$$

where \mathbf{I} is an identity matrix of size $(N_b - K_b) \times (N_b - K_b)$, the parity symbols of each column can be calculated as

$$\mathbf{p}_i = \mathbf{a}_i - \mathbf{P}\mathbf{u}_i, \quad i = K_a + 1, \dots, N_a, \quad (6.9)$$

where \mathbf{u}_i is a vector of information symbols of column i and \mathbf{p}_i is the parity symbols of column i , as shown in Figure 6.1. Column \mathbf{c}_i is thus given as

$$\mathbf{c}_i = \begin{pmatrix} \mathbf{u}_i \\ \mathbf{p}_i \end{pmatrix}, \quad i = K_a + 1, \dots, N_a \quad (6.10)$$

Hard-Decision Decoding of EL Codes

During the decoding process of an EL code, a direct way of hard-decision decoding is to first decode the outer code and then the inner code. This algorithm is described as follows:

1. For the received matrix \mathbf{C}' of codeword \mathbf{C} , calculate the syndromes $(\mathbf{a}'_1, \dots, \mathbf{a}'_{N_a})$ of all the columns using \mathbf{H}_B .
2. Input the syndromes to the decoder of code \mathcal{A} . If the decoding fails, announce a decoding failure of the EL code. Otherwise, go to step 3.
3. Assume the decoded codeword of the outer code \mathcal{A} to be $(\hat{\mathbf{a}}_1, \dots, \hat{\mathbf{a}}_{N_a})$. For each column, we have

$$\mathbf{a}'_i = \mathbf{H}_B \mathbf{c}'_i = \mathbf{H}_B (\hat{\mathbf{c}}_i + \mathbf{e}_i) = \hat{\mathbf{a}}_i + \mathbf{H}_B \mathbf{e}_i \quad (6.11)$$

$$\Rightarrow \mathbf{H}_B \mathbf{e}_i = \mathbf{a}'_i - \hat{\mathbf{a}}_i \quad (6.12)$$

where $\hat{\mathbf{c}}_i$ is the estimate of column i after EL decoding, and \mathbf{e}_i is the error vector for column i . The decoder for the inner code \mathcal{B} is a syndrome decoder, which can

decode \mathbf{e}_i given the syndrome $\mathbf{a}'_i - \hat{\mathbf{a}}_i$. The estimate of column i is thus $\hat{\mathbf{c}}_i = \mathbf{c}'_i - \mathbf{e}_i$, where \mathbf{c}'_i is the received column i .

6.1.2 A Tensor-Product Parity Code with Soft-Decision Decoding for an ISI Channel

The tensor-product parity (TPP) code proposed in [5] is a special case of an EL code. The inner code \mathcal{B} is a binary parity check code $\mathcal{B}(2; N_b, N_b - 1, 2)$, while the outer code is a linear binary code $\mathcal{A}(2; N_a, K_a, d_{\min,a})$. The code definition in [5] differs from our definition of an EL code. In [5], the parity check bits of the inner code, not the syndromes, form a codeword of the outer code. However, for the TPP code defined in [5], the same code results using our definition. For example, the previous EL code example is exactly a TPP code. As defined in [5], the inner code \mathcal{B} would be a $(2;4,3,2)$ code, rather than a $(2;3,2,2)$ code which would result from our definition.

A TPP code in [5] was proposed to replace the conventional parity check code in the magnetic recording system, which is usually used in the post processing procedure. When using the conventional parity check code, a single parity check bit for each sub-block will be sent over the channel. This parity check bit can detect an odd number of bit errors, thus provide the information for the post processing. However, in high SNR scenarios, the parity check bits for the sub-blocks are mostly correct. By using the TPP code, we don't send the parity check bits of each sub-blocks. Instead, we put constraint on the transmitted bits, so that the parity check bits of sub-blocks form a codeword of the outer code. The errors in the parity check bits can be recovered mostly by the outer code. Therefore, less redundancy is used for the TPP code and the advantage is shown in [5].

Because the TPP code has a weak inner code which can only detect errors, a hard decision decoding process different from the hard decision decoding algorithm stated in 6.1.1 was proposed for the TPP code in [5]. The decoding process was described in Figure 2.5, as well as in Figure 6.3 (a). The bits in the TPP codeword matrix are recorded column-wise. After Viterbi detection of the equalized channel, the parity/syndrome bits

are calculated and decoded to an outer code codeword. An extended Viterbi trellis that combine the channel states and the parity bit state is then used to process the channel outputs r_i again. The output of such a detector is the estimate of the TPP codeword, which satisfies the TPP code constraint.

Soft-Decision Decoding of TPP Codes

In this part, we will propose two soft-decision decoding algorithms for TPP codes used for binary ISI channels.

In Figure 6.2, two system diagrams are shown. In (a), the bits in the TPP codewords are sent column-wise, which is the same as in [5]. However, we can also send the bits in the codewords row-wise, which is equivalent to inserting an interleaver at the transmitter side, as shown in (b). Of course, the hard decision decoding proposed in [5] can only be used for the system in (a), because the extended Viterbi trellis requires that the bits in the codewords are sent column-wise.

One method to achieve soft-decision decoding is to use the BCJR [8] algorithm instead of the Viterbi algorithm, and a soft input soft output (SISO) decoding algorithm instead of the hard decision decoder for the outer code of the TPP code, as shown in Figure 6.3(b). We call this decoder the *Ext-BCJR* decoder. Just as was the case for hard decision decoding, this decoder can only be applied to the system in Figure 6.2(a), because of the use of the extended trellis in the “*Extended BCJR*” block.

The “*Channel BCJR*” block in Figure 6.3 (b) applies the BCJR algorithm to the trellis matched to the ISI channel. It outputs soft information for each bit in the codeword, i.e., the probability of each bit being a 1 or 0 can be obtained. The next step is to calculate the probability that the syndrome bits is a 1 or a 0. We assume that the bits in each column of the TPP codeword are independent. Denote column i of the TPP codeword by \mathbf{c}_i as before, and denote each bit in this column by $c_{j,i}$, $j = 1, \dots, N_b$. Denote the syndrome, in this case only one bit, of column \mathbf{c}_i by a_i . The probability that

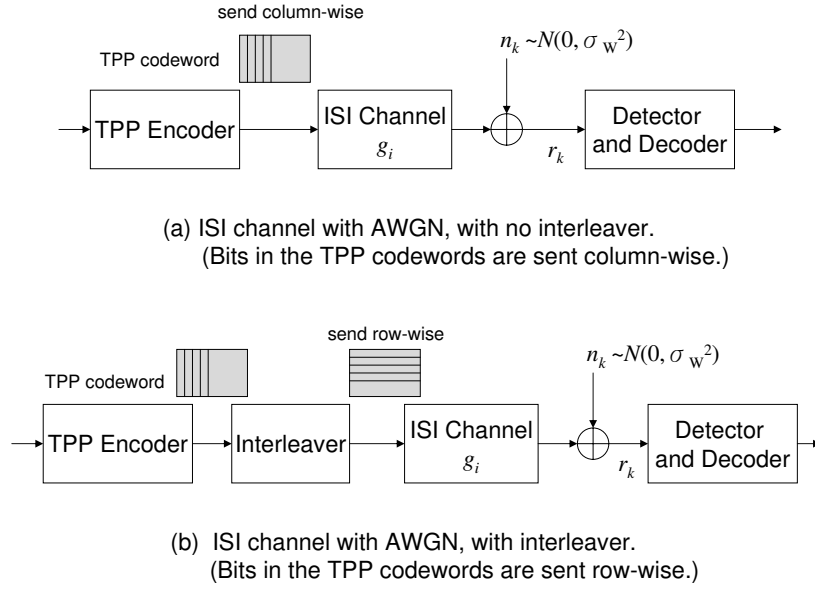


Figure 6.2 System diagrams using TPP code. (a) without interleaver; (b) with interleaver.

a_i is a 0 can be expressed as follows:

$$\Pr(a_i = 0) = \sum_{\substack{\text{even number of 1's in} \\ \{b_j, j=1, \dots, N_b\}}} \prod_j \Pr(c_{j,i} = b_j | \mathbf{r}) \quad (6.13)$$

where \mathbf{r} is the channel output sequence.

The probabilities of the syndrome bits are then used as the input to the SISO decoder of the outer code. The decoder output provides the soft information of the updated probability of the syndrome bits. These information are used as *a priori* information in an extended trellis of the BCJR algorithm to update the probability of each bit in the TPP codeword.

The extended trellis is a combination of the channel trellis and the parity bit trellis. Figure 6.4 shows an example where the channel memory is two. Each state in the extended trellis includes the two input bits as the channel memory, as well as a bit representing the accumulated parity bit p_k , where $p_{k+1} = p_k + x_k$, $k = 1, 2, \dots, N_a N_b$; $p_0 = 0$.

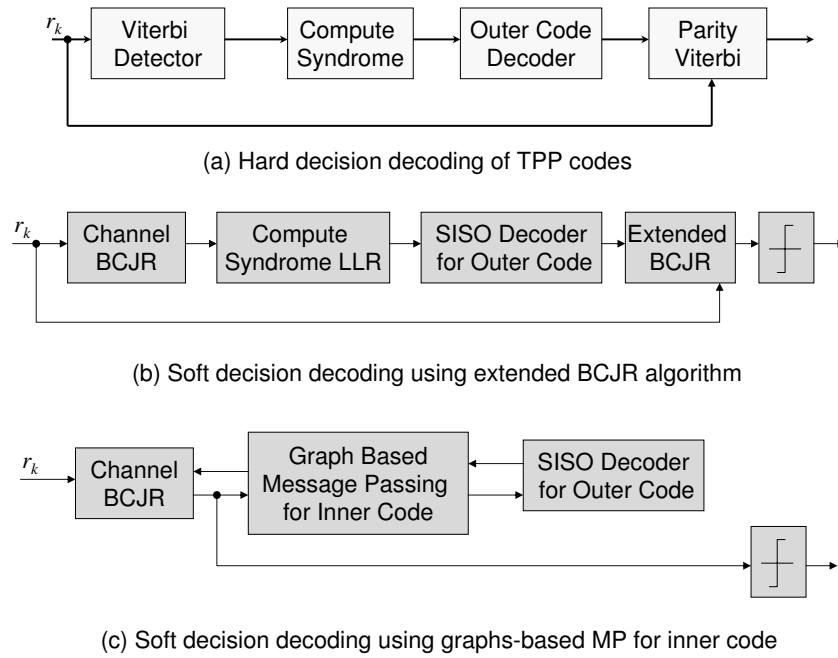


Figure 6.3 Hard and soft decision decoder structures for TPP code for an ISI channel.

Given the updated probabilities of a_i , $i = 1, \dots, N_a$, the probability of p_{iN_b} , $i = 1, 2, \dots, N_a$, can be calculated by the following relationship:

$$\Pr(p_{iN_b} = 0) = \Pr(p_{(i-1)N_b} = 1)\Pr(a_i = 1) + \Pr(p_{(i-1)N_b} = 0)\Pr(a_i = 0) \quad (6.14)$$

$$i = 2, \dots, N_a$$

$$\Pr(p_{N_b} = 0) = \Pr(a_1 = 0) \quad (6.15)$$

$$\Pr(p_{iN_b} = 1) = 1 - \Pr(p_{iN_b} = 0) \quad i = 1, \dots, N_a. \quad (6.16)$$

The probabilities of p_{iN_b} , $i = 1, 2, \dots, N_a$ are used as *a priori* information of the BCJR algorithm on the extended trellis. In the log-domain BCJR algorithm [9], the log-likelihood ratio (LLR) of p_{iN_b} is added to the branch metric, LLR of the conditional transition probability. However, this information is only available for branches at time iN_b . For branches at other times, the LLR of p_k is zero.

The output of the “*Extended BCJR*” block is thus the updated soft information for each bit of the TPP codeword. The codeword can be recovered from this soft information.

Another decoding structure for the TPP code, as shown in Figure 6.3(c), utilizes the Tanner graph [10] structure of the inner code. The graph structure of the decoder is shown in Figure 6.5. A message passing algorithm used to decode LDPC codes [11], such as the sum-product algorithm, can be used on the Tanner graph of the inner code to exchange information between the “*channel BCJR*” block and the “*SISO decoder for the outer code*” block. With this structure, an iterative decoding process is also possible. The soft information can be passed back and forth on the graph.

Unlike the Ext-BCJR algorithm, the second decoding algorithm, denoted by *MP-BCJR* algorithm, can be applied to both the system with and without the interleaver.

The type of SISO decoder for the outer code in both algorithms depends on the type of the outer code. For example, if the outer code is an LDPC code, a graph based message passing algorithm can be used. If the outer code is a convolutional code, the BCJR algorithm can be used. If the outer code is a Turbo code, an iterative Turbo decoder can be used.

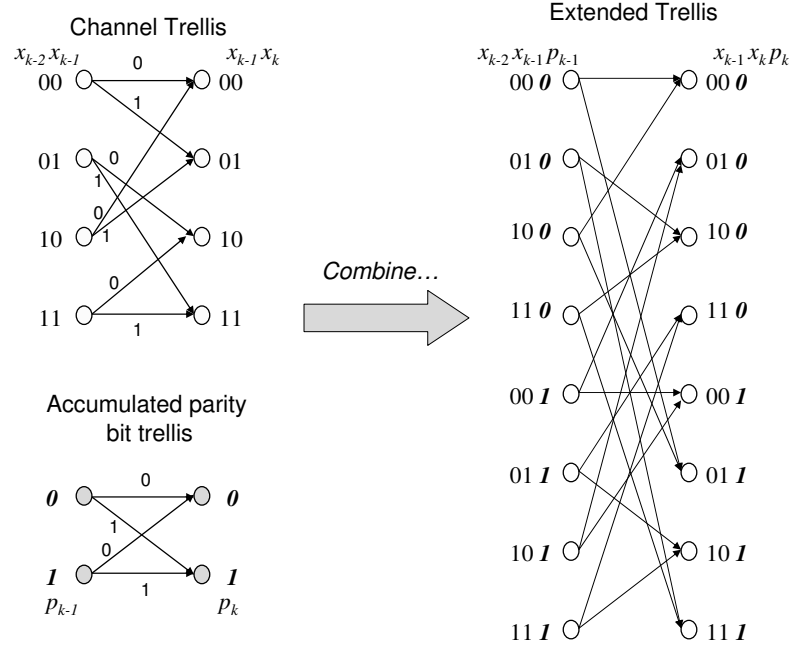


Figure 6.4 Combination of the channel trellis and the accumulated parity bit trellis.

6.1.3 Simulation Results

In this section, we will give an example of the soft-decision decoding of a TPP code over a PR4 channel with AWGN. The inner code of the TPP code is a single parity check code of length 10 bits. The outer code is a systematic convolutional code of rate $1/2$, with the generator matrix $G(D) = [1, 1 + D + D^3]$. The block length of the convolutional code is set to 468 bits. Thus the codeword size is 10×468 bits, which is about the size of a typical sector of hard disk drives. The rate of the code is 0.95.

We compared the performance of the following cases in Figure 6.6: (1) uncoded detection; (2) the data bits are coded with a TPP code, without an interleaver, and decoded using the hard decision decoding; (3) data bits are coded a TPP code, without an interleaver, and decoded using the Ext-BCJR algorithm; (4) data bits are coded with a TPP code, without an interleaver, and decoded using the MP-BCJR algorithm, with one iteration; (5) data bits are coded with a TPP code, with an interleaver, and decoded using

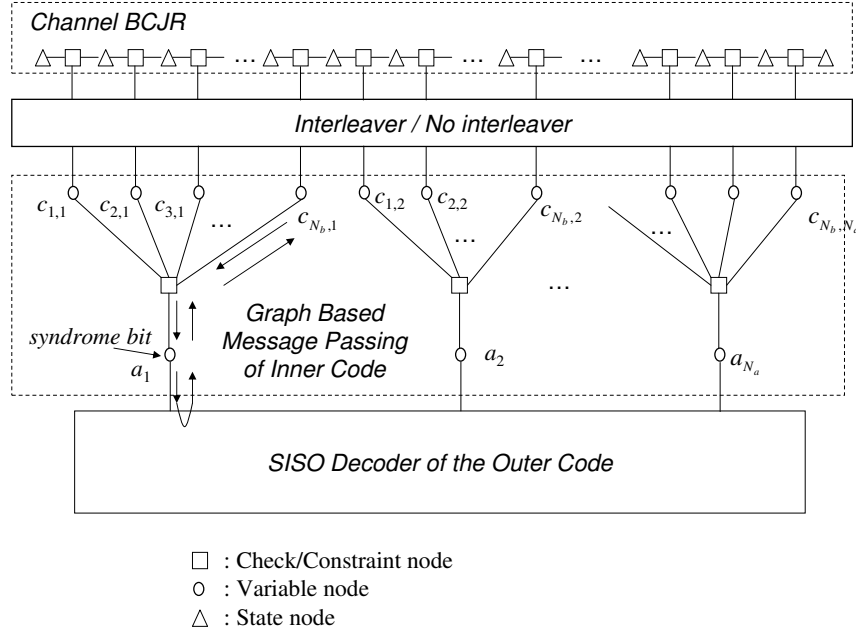


Figure 6.5 The graph structure of the MP-BCJR algorithm.

the MP-BCJR algorithm, with one, two and three iterations.

We can see that the performances of the soft-decision decoding for the cases without interleavers are very close to the performance of the hard decision decoding. The Ext-BCJR algorithm performs only about 0.12dB better than hard decision decoding and the MP-BCJR algorithm performs even worse than hard decision decoding.

However, if we use the interleaver, the performance is improved by about 1dB. With the interleaver, the burst errors can be spread over multiple columns rather than in one column, to be checked by multiple parity check bits. At the same time, the bits in one column of the TPP codeword are more independent compared to the case without the interleaver. The message passing algorithm works better for this scenario because it assumes the input bits to be independent.

Multiple iterations are also examined for the MP-BCJR algorithm for the system with an interleaver. Because of the weak inner code, we can see that the performance

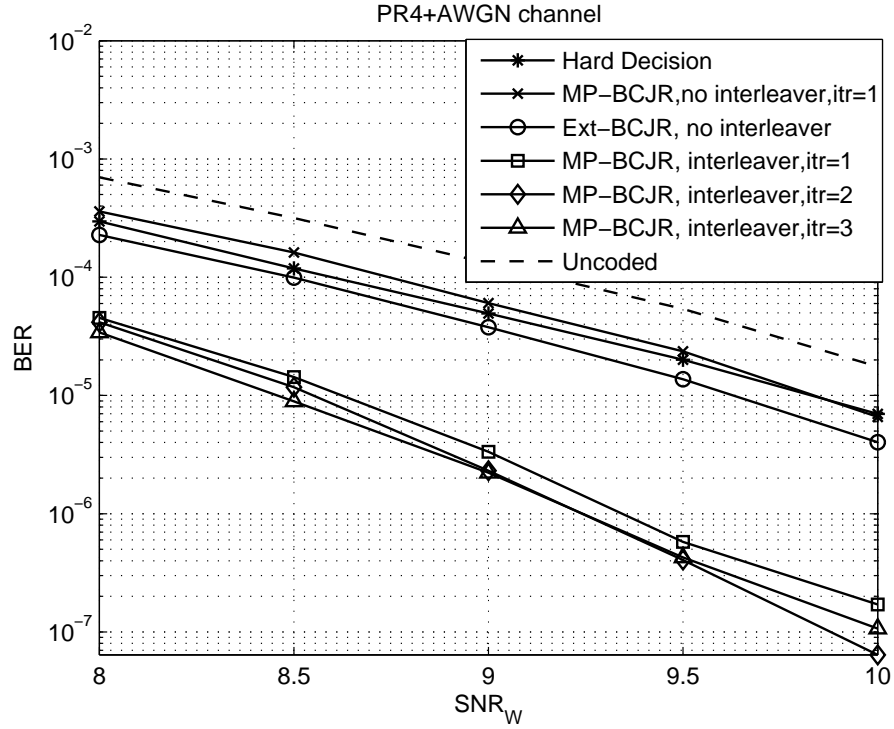


Figure 6.6 Performance comparison of several decoding algorithms for systems with and without interleaver.

does not vary much with one, two and three iterations.

6.2 Generalized Error-Locating Codes and List Decoding

In [6], a GEL code was proposed as a coding scheme that can substitute for a RS code in the magnetic recording channels. In this section, we will briefly introduce the advantage of the GEL code through examples and then propose a list decoding algorithm for the GEL code. The performance of the list decoding algorithm will be compared to one of the decoding algorithms in [6], using a simple GEL code example for a binary symmetric channel (BSC) and an AWGN channel.

6.2.1 Introduction to Generalized Error-Locating Codes

An EL code has two component codes. One is the inner code and the other is the outer code. In the definition of a GEL code, there are two sets of component codes, the set for the inner codes and the set for the outer codes.

A GEL code of order L is a combination of L outer codes $\mathcal{A}^{(l)}(p^{q_a^{(l)}}; N_a, K_a^{(l)}, d_{min,a}^{(l)})$ and L inner codes $\mathcal{B}^{(l)}(p; N_b, K_b^{(l)}, d_{min,b}^{(l)})$, where l is from 1 to L . All the outer codes have the same code length N_a and all the inner codes have the same code length N_b .

Let the GEL codeword also be represented in the form of an $N_b \times N_a$ matrix \mathbf{C} . We use the same notation for the columns of \mathbf{C} as for the EL codes. Therefore, the GEL code requires that the syndromes of each level

$$\mathbf{a}^{(l)} = (\mathbf{a}_1^{(l)}, \mathbf{a}_2^{(l)}, \dots, \mathbf{a}_{N_a}^{(l)}) = \mathbf{H}_B^{(l)} \mathbf{C}, \text{ for } i = 1, \dots, L \quad (6.17)$$

where $\mathbf{H}_B^{(l)}$ is the parity check matrix for the l^{th} inner code and syndromes $\mathbf{a}_i^{(l)}$, $i = 1, \dots, N_a$ are symbols in $\text{GF}(p^{q_a^{(l)}})$, represented by the vectors with elements in $\text{GF}(p)$, to be a codeword of the code $\mathcal{A}^{(l)}$.

The parity check matrices of the inner codes are restricted to have the following structure:

$$\mathbf{H}_B = \begin{pmatrix} \mathbf{H}_B^{(L)} \\ \mathbf{H}_B^{(L-1)} \\ \vdots \\ \mathbf{H}_B^{(1)} \end{pmatrix} = \begin{pmatrix} \mathbf{Q}_0^{(L)} & \mathbf{I}^{(L)} & 0 & \dots \\ \mathbf{Q}_0^{(L-1)} & \mathbf{Q}_1^{(L-1)} & \mathbf{I}^{(L-1)} & 0 & \dots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \mathbf{Q}_0^{(1)} & \mathbf{Q}_1^{(1)} & \dots & \mathbf{Q}_{L-1}^{(1)} & \mathbf{I}^{(1)} \end{pmatrix} \quad (6.18)$$

where $\mathbf{I}^{(l)}$ is the identity matrix of size $q_a^{(l)} \times q_a^{(l)}$, for $l = 1, \dots, L$.

Writing the codeword in vector form by taking the elements in \mathbf{C} column-wise, we can derive the parity check matrix for the GEL code using the tensor products of the inner codes and the outer codes, as for the EL codes. Any codeword of a GEL code is a codeword of the EL code with inner code $\mathcal{B}^{(l)}$ and outer code $\mathcal{A}^{(l)}$, for $l = 1, \dots, L$. Therefore, the codeword must satisfy all the parity check equations of all of these codes.

The parity check matrix for the GEL code is thus formed by combining the parity check matrices of all the individual EL codes:

$$\mathbf{H}_{\text{GEL}} = \begin{pmatrix} \mathbf{H}_A^{(L)} \otimes \mathbf{H}_B^{(L)} \\ \mathbf{H}_A^{(L-1)} \otimes \mathbf{H}_B^{(L-1)} \\ \vdots \\ \mathbf{H}_A^{(1)} \otimes \mathbf{H}_B^{(1)} \end{pmatrix}. \quad (6.19)$$

The dimension of \mathbf{H}_{GEL} is $\sum_{l=1}^L (N_a - K_a^{(l)})(N_b - K_b^{(l)})$ by $N_a N_b$. The rate of the GEL code is thus $R = 1 - \frac{\sum_{l=1}^L (N_a - K_a^{(l)})(N_b - K_b^{(l)})}{N_a N_b}$.

Encoding of GEL codes

GEL codes can be encoded using a procedure similar to that used for EL codes. Using the parity check matrix in equation (6.18), the encoding of the GEL code can be conducted level by level from L to 1.

We assume that the encoders for the outer codes $\mathcal{A}^{(l)}$ are systematic encoders. The encoding procedure can be described as follows:

1. Put the information symbols in the white spaces of the codeword matrix \mathbf{C} , as shown in Figure 6.7.
2. Use the encoding procedure of the EL codes in Section 6.1.1 for level L , i.e., using the inner code $\mathcal{B}^{(L)}$ to calculate the syndromes from column 1 to column $K_a^{(L)}$, then encode the syndromes using the outer code $\mathcal{A}^{(L)}$, and finally calculate the parity symbols in position P_L , as marked in Figure 6.7. Since the parity check matrix of the inner code is of the form shown in equation (6.18), the syndromes from column 1 to column $K_a^{(L)}$ do not depend on the parity symbols of other levels.
3. By repeating the same procedure as in level L , we fill in parity symbols in position P_{L-1} to P_1 sequentially, using inner and outer codes from level $L - 1$ to 1. The parity symbols filled into the codewords from higher levels will be used as the information symbols for the lower levels.

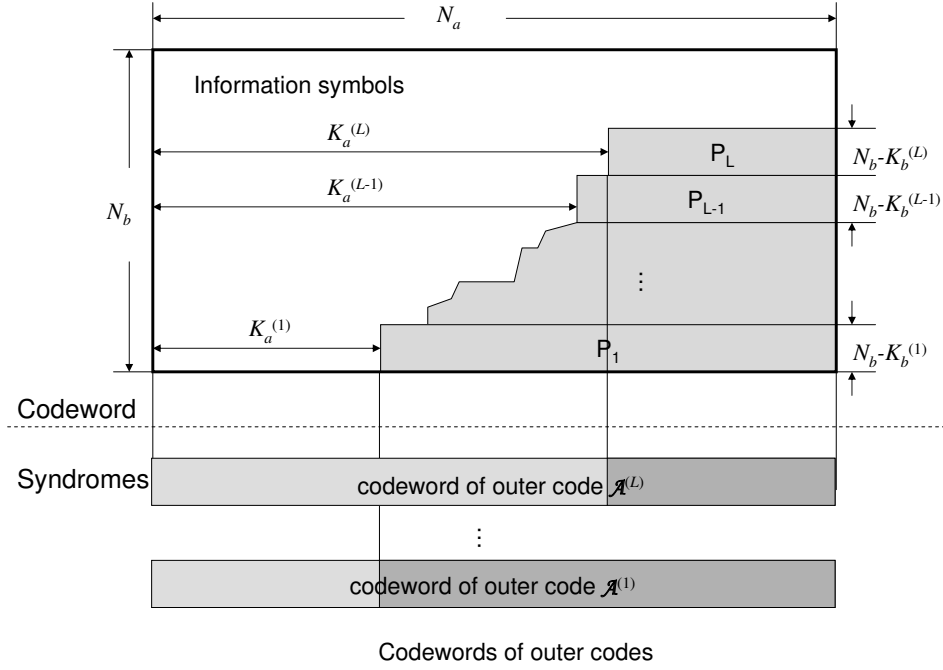


Figure 6.7 Encoding of the GEL codes.

Thus, the parity symbols all can be filled in the gray area and the encoding procedure is finished.

Decoding of GEL codes

The decoding procedure is more flexible. For example, we can calculate the syndromes for each inner code and decode the outer code of each level separately. The corrected syndromes are then used to correct the symbols of each column using the code corresponding to parity check matrix \mathbf{H}_B in equation (6.18). Fahrner proposed several decoding procedures in [6], considering the fact that some of the outer codes can correct more errors than others. A level-by-level hard-decision decoding procedure was proposed as follows. Here, we assume that the outer codes are ordered such that $K_a^{(i)} \leq K_a^{(j)}$ if $i < j$, and the error correcting capability of code $\mathcal{A}^{(i)}$ is greater than

$\mathcal{A}^{(j)}$. Therefore, we now start from level 1 of the inner and outer codes:

1. Calculate the syndromes $\mathbf{a}'_i^{(1)}$, $i = 1, \dots, N_a$ for the received word \mathbf{C}' , using matrix $\mathbf{H}_B^{(1)}$. Decode using outer code $\mathcal{A}^{(1)}$. If the number of erroneous columns in the word \mathbf{C}' is larger than $(d_{min,a}^{(1)} - 1)/2$, the outer code can not correct all the errors. In this case, declare a decoding failure. Otherwise, the syndromes $\mathbf{a}'_i^{(1)}$ can be corrected as $\hat{\mathbf{a}}_i^{(1)}$. Since the outer code $\mathcal{A}^{(1)}$ has the most powerful error correction capability among all the outer codes, it can correct the largest number of column errors.
2. Now, we erase the columns which are determined to have errors by code $\mathcal{A}^{(1)}$. Calculate $\mathbf{a}'_i^{(2)}$ for those unerased columns. If the number of erased columns is less than $d_{min,a}^{(2)} - 1$, the codeword can be recovered. Combine $\hat{\mathbf{a}}^{(1)}$ and $\hat{\mathbf{a}}^{(2)}$ together, to form the syndrome for the code with parity check matrix

$$\mathbf{H}_B^{1,2} = \begin{pmatrix} \mathbf{H}_B^{(2)} \\ \mathbf{H}_B^{(1)} \end{pmatrix}. \quad (6.20)$$

Thus, if $\mathbf{H}_B^{1,2}$ can correct t errors, the erroneous columns that contain no more than t errors can be corrected. The reason we combine the syndrome of the inner codes $\mathcal{B}^{(1)}$ and $\mathcal{B}^{(2)}$ together is that the error correcting capability is greater than any individual code. Now the remaining codeword matrix \mathbf{C} has only erroneous columns which have more than t errors. Therefore in the following level of decoding, the outer codes will see less errors.

3. Go to the next level of inner and outer codes decoding. At every level, we could choose either to decode the inner code of the erroneous columns, or to leave those columns as erasures for the next level of outer code decoding. Of course, at the last level, we will decode the inner code corresponding to parity check matrix \mathbf{H}_B .

In this process, any failure of the outer code decoding could result in decoding failure of the GEL code.

For a channel where the erroneous columns that contain more symbol errors are less likely to appear, the GEL code can be used to reduce the number of parity symbols.

Of course, the GEL code needs to be designed specifically for the given channel. For example, let the GEL codeword size be 31×31 , and further assume that the channel is a binary channel which has the following properties:

- The number of erroneous columns in one codeword is almost always smaller than 7.
- The number of erroneous columns in one codeword that have more than 1 error is almost always smaller than 3.
- The number of erroneous columns in one codeword that have more than 2 errors is almost always smaller than 1.
- There is almost never a column having more than 3 errors.

Then, we can design a GEL code with three levels. The inner codes are BCH codes with code length 31. The parity check matrix \mathbf{H}_B is the parity check matrix of a BCH code $(2; 31, 16, 7)$ whose rows are arranged such that $N_b - K_b^{(l)} = 5$, $l = 1, 2, 3$, code $\mathcal{B}^{(1)}$ has minimum distance 3, and the code corresponding to parity check matrix $\mathbf{H}_B^{1,2}$ has minimum distance 5. Therefore, the first, second and third level of decoding of the inner codes can correct 1, 2 and 3 errors, respectively, in each column.

The outer codes are RS codes over $\text{GF}(2^5)$, with length 31. Using the decoding procedure mentioned above, the outer code can be designed to adapt to the channel error statistics as follows:

- Code $\mathcal{A}^{(1)}$ needs to correct at most 7 errors. Set $d_{min,a}^{(1)} = 2 \times 7 + 1 = 15$.
- Code $\mathcal{A}^{(2)}$ will need to recover at most 7 erasures if we choose not to decode. Therefore, we could set the $d_{min,a}^{(2)} = 7 + 1 = 8$.
- Code $\mathcal{A}^{(3)}$ will need to recover 1 erasure if the level 2 column-wise decoding reports the failure on the 3-error columns. If it decoded to an incorrect codeword, the code needs to handle 1 error. Therefore, we could set the $d_{min,a}^{(3)} = 2 \times 1 + 1 = 3$.

The code rate for such a GEL code is about 0.88. However, if we use a shortened RS code over $\text{GF}(2^{31})$, which can correct any number of errors in each column, we will need to have 14 parity symbols to be able to correct 7 erroneous columns. The rate of such a RS code is only 0.55. Of course, this comparison is not entirely fair. However we can have a sense of how the GEL code could have an advantage over the RS code for some channels. In [6], more detailed simulations and comparisons of the GEL code over some magnetic recording channels were given.

6.2.2 List Decoding Algorithm for GEL Code

With the development of new decoding techniques for RS code, such as the Guruswami-Sudan list decoding [12] and the Koetter-Vardy soft-decision decoding [13], we are able to improve the performance of the GEL codes which use the RS codes as the outer codes. As we have mentioned in Section 6.2.1, the decoding failure of any outer codes will result in the decoding failure of the GEL code. Thus, by improving the decoding performance of the outer codes, we are able to improve the decoding performance of GEL codes.

In this section, we apply the Guruswami-Sudan (G-S) list decoding algorithm to the RS code decoding in the GEL code. A detailed description of the G-S algorithm can be found in [14]. Instead of obtaining one codeword for each outer code decoding, a list of codewords will be obtained at each level of outer code decoding. Each codeword in the list will be a candidate in the next level of decoding. At the last level, a list of TPP codewords will be obtained.

A simple GEL code of order 2 is simulated. The GEL code description is as follows:

Code size: 7×7 , $n_a = 7$ and $n_b = 7$

Code order: 2

Inner codes: $\mathcal{B}_1(7, 4; dmin = 3)$, Hamming code

$\mathcal{B}_{1,2}(7, 1; dmin = 7)$, Repetition code

$$\mathbf{H}_B = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

Outer codes: $\mathcal{A}_1(GF(2^3); 7, 3; dmin = 5)$, Reed-Solomon code;
 $\mathcal{A}_2(GF(2^3); 7, 4; dmin = 4)$, Reed-Solomon code

The performance of the list decoding algorithm is compared to the performance of the hard decision decoding algorithm given previously for the binary symmetric channel in Figure 6.8 and the AWGN channel Figure 6.9. We can see that the decoding performance is improved in both cases by the list decoding algorithm.

6.3 Conclusions

In this chapter, we first presented soft-decision decoding algorithms for the TPP codes and their performances in the ISI channel with AWGN. The performance can be improved by using an interleaver and there is not much need to decode with multiple iterations. Of course, the algorithm can be applied to the magnetic recording channels with other noise sources. The algorithms can also be applied to systems where soft-decision decoding is available for a code concatenated to a TPP code, such as a LDPC code or a RS code with soft-decision, to help improve the performance.

The outer code of the TPP code is not limited to convolutional codes. A stronger outer code can of course help improve the performance of the TPP code. However, since the inner code is very weak, the performance improvement might be limited. To verify this conjecture, further research and experiments will be needed.

In the second part of the chapter, we presented a list decoding algorithm for the GEL code. However, the soft-decision decoding algorithm can probably be further

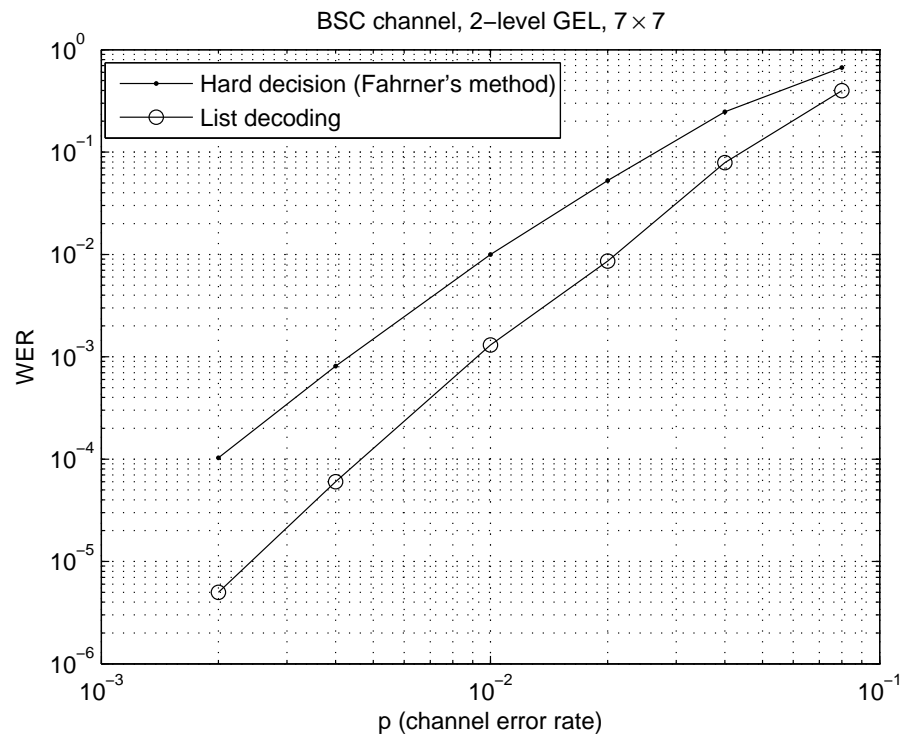


Figure 6.8 Comparison between Fahrner's hard decision decoding and the list decoding of the GEL code in BSC channel.

developed, using the soft decision decoding of the RS code and the information provided by each level of the inner code.

Bibliography

- [1] J. K. Wolf and B. Elspas, "Error-locating codes – a new concept in error control," *IEEE Trans. Inf. Theory*, vol. 9, no. 2, pp. 113–117, Apr. 1963.
- [2] J. K. Wolf, "On codes derivable from the tensor product of check matrices," *IEEE Trans. Inf. Theory*, vol. 11, no. 2, pp. 281–284, Apr. 1965.
- [3] —, "On an extended class of error-locating codes," *Information and Control*, vol. 8, no. 2, pp. 163–169, Apr. 1965.
- [4] J. Maucher, V. Zyablov, and M. Bossert, "On the equivalence of generalized con-

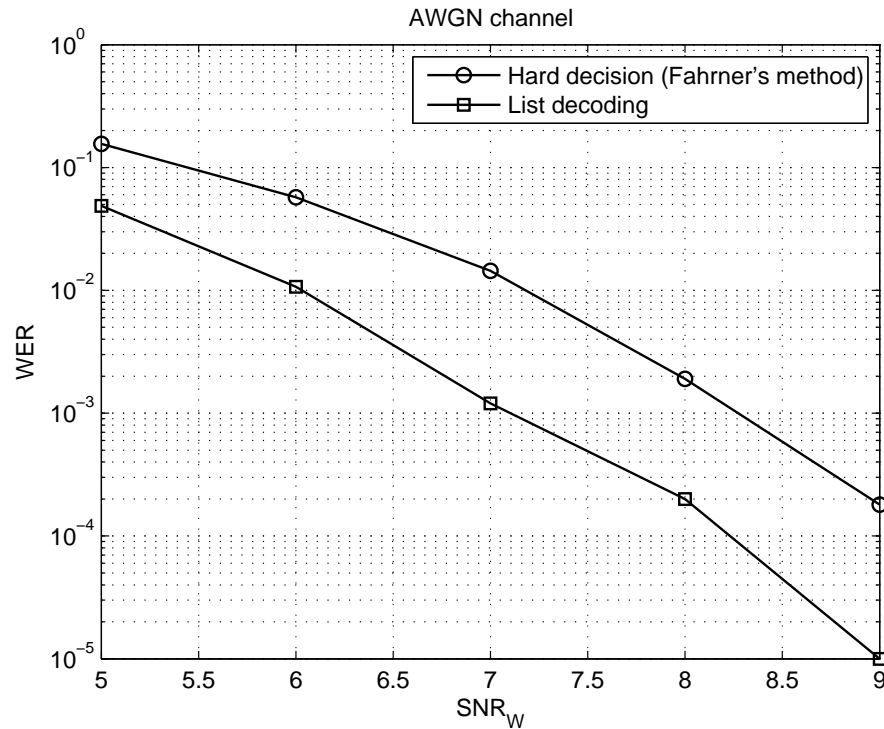


Figure 6.9 Comparison between Farhner's hard decision decoding and the list decoding of the GEL code in AWGN channel.

catenated codes and generalized error location codes," *IEEE Trans. Inf. Theory*, vol. 46, no. 2, pp. 642–649, Mar. 2000.

- [5] P. Chaichanavong and P. H. Siegel, "Tensor-product parity code for magnetic recording," *IEEE Trans. Magn.*, vol. 42, no. 2, pp. 350–352, Feb. 2006.
- [6] A. Fahrner, "On signal processing and coding for digital magnetic recording system," Ph.D. dissertation, Department of Telecommunications and Applied Information Theory, University of Ulm, Germany, 2004.
- [7] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*. Amsterdam, New York : North-Holland Pub. Co., 1996.
- [8] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inf. Theory*, vol. 2, pp. 284–287, Mar. 1974.
- [9] C. Schlegel and L. Perez, *Trellis and Turbo Coding*. Wiley-IEEE Press, 2003, ch. 7.

- [10] N. Wiberg, “Codes and decoding on general graphs,” Ph.D. dissertation, Department of Electrical Engineering, Linköping University, Sweden, 1996.
- [11] W. E. Ryan, “An introduction to LDPC codes,” in *CRC Handbook for Coding and Signal Processing for Recording Systems*, B. Vasic and E. M. Kurtas, Eds. CRC Press, 2004.
- [12] V. Guruswami and M. Sudan, “Improved decoding of Reed-Solomon and algebraic geometry codes,” *IEEE Trans. Inf. Theory*, vol. 45, pp. 1757–1767, 1999.
- [13] R. Koetter and A. Vardy, “Algebraic soft-decision decoding of Reed-Solomon codes,” *IEEE Trans. Inf. Theory*, vol. 49, pp. 2809–2825, Nov. 2003.
- [14] R. J. McEliece, “The Guruswami-Sudan decoding algorithm for Reed-Solomon codes,” California Institute of Technology, JPL Interplanetary Network Progress Report 42-153, May 2003.