UNIVERSITY OF OKLAHOMA
GRADUATE COLLEGE

CHANNEL CODING AND REDUCED-COMPLEXITY DETECTION FOR MAGNETIC RECORDING

A DISSERTATION
SUBMITTED TO THE GRADUATE FACULTY
in partial fulfillment of the requirements for
the degree of
DOCTOR OF PHILOSOPHY

By
Runsheng He
Norman, Oklahoma
1997

UMI Number: 9817717

UMI
300 North Zeeb Road
Ann Arbor, MI 48103

CHANNEL CODING AND REDUCED-COMPLEXITY
DETECTION FOR MAGNETIC RECORDING

A DISSERTATION APPROVED FOR THE
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

BY

Arthur M. Breipohl

Kevin Grasse

Acknowledgments

First, I would like to express my sincere gratitude to my advisor Dr. J. R. Cruz, for his guidance and support throughout my study and research in the past four years. My progress and accomplishment would not have been possible without his countless attention and valuable advice. I will always be indebted to Dr. Cruz for the help he provided.

Secondly, I would like to thank my committee members, Dr. Arthur Breipohl, Dr. John Cheung, Dr. Fred Lee, and Dr. Kevin Grasse. Some special thank goes to the late Dr. Richard Resco for two wonderful courses on algebra.

This work was supported by a graduate fellowship from Hitachi, Ltd., and I would like to thank them for all the financial and technical support provided.

Also, I would like to thank all my friends for their friendship and for their help.

Finally, I would like to express my gratitude to my parents, for their constant encouragement throughout my life.

This dissertation is dedicated to my wonderful wife Fang Wu

who has been so supportive over the past four years

Abstract

This work presents a high performance coding system and a novel reduced-complexity implementation technique to improve the performance of read channel chips in data storage systems. In particular, the coding system combines the list Viterbi algorithm (LVA) and a high rate error detection code. The novel reduced-state (RS) technique called complement states grouping technique (CSGT) reduces the implementation complexity of both the VA and the LVA with a negligible performance loss.

First, we develop a reduced-state technique that generates an RS trellis by grouping all pairs of complement states whose state-distance satisfies a predetermined criterion. While the traditional RS techniques often cause unacceptable performance loss for partial response channels with binary inputs, this technique reduces the number of states by a factor of about two with a negligible loss for such channels. Second, a coding system combining an LVA with an error detection code is proposed for magnetic recording channels. The error detection code can provide a substantial coding gain with a high code rate. The LVA is used to provide a candidate list to achieve the coding gain in a suboptimal way. Third, we develop a method to estimate the performance of the LVA; the method takes into account factors such as noise correlation and multiplicity of error events. Fourth, it is shown that the LVA can be realized using a pipelined implementation and the computation complexity increases only linearly with the number of LVA candidates.

As an application, a modified EEPR4 (MEEPR4) channel with a 3-candidate LVA is shown to exhibit a 2 dB LVA gain. In addition, an error detection code is designed to increase the minimum distance by more than 2 dB with a high code rate. The CSGT generated RS trellis for the MMEPR4 channel has half the number of states. The performances of both the VA and the LVA on the RS MEEPR4 trellis have negligible loss. Lastly, the Fettweis branch metric shifting method is used with the CSGT to further reduce the complexity of the MEEPR4ML system.

Table of Contents

List of Tables

List of Figures

Chapter 1
Introduction

1.1 Disk Storage Overview

1.1.1 Signal Processing in Data Storage Systems

Over the past decades, the areal storage density on hard disks has grown at an average rate of 29% [1], and is currently growing at a rate of 60% annually. From a historical point of view, the storage density growth is largely due to improvements in heads and media or improvements in disk drive mechanics. Signal processing techniques have made fewer contributions to this increase, in fact, the simple analog peak detector and its related signal processing technology remained unchanged until the early 1990's. But as improvements in heads and media slowed down and the mechanical refinements reached their limit, the application of advanced signal processing techniques is becoming more and more important to the disk drive industry. The first major change happened in 1990 when IBM reported the results of an experimental system [2], which used the so called Class-IV partial-response (PR4) signaling with maximum-likelihood (ML) sequence detection, or PR4ML. A complete description of the design of the PR4ML system can be found in [3]. Advanced signal processing techniques such as Viterbi decoding, digital equalization, digital timing and gain control were used for the first time in the PR4ML system. In both simulations and experiments, the density was found to increase by 20 to 30 percent by replacing peak detection with the PR4ML detector.

PR4ML is the simplest and therefore the fastest PRML system, however, higher order PR signaling formats are preferred at higher densities. For example, the extended

PR4 (EPR4) system exhibits a 2 dB gain over PR4 signaling at the density of 2.5. This allows the storage density to be increased by 20 to 30 percent without other changes. The disadvantage of using higher order PRML systems is that the implementation complexity is much higher than using the simpler PR4 signaling. However, as high performance and low-cost VLSI fabrication becomes widely available, sophisticated signal processing techniques are no longer impractical. In fact, commercial read channel chips using EPR4ML are starting to replace the PR4 chips.

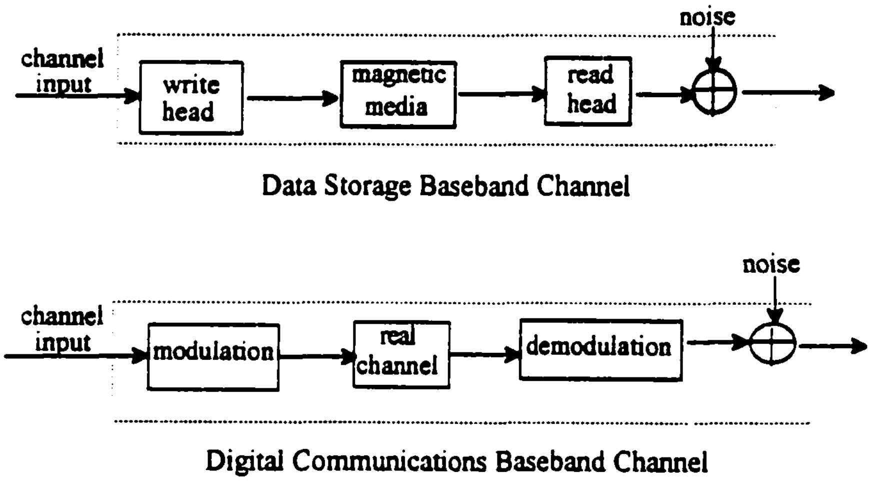
In addition to the EPR4ML or more sophisticated PRML systems, trellis codes are being considered as an alternative to improve the performance of the detector [4]. Another possible competitive candidate is decision feedback equalization (DFE) [5],[6], but only one company has come close to delivering a commercial read channel chip using the DFE. An experimental "dual-DFE" read channel chip was recently announced [7],[8]. Although the DFE detector has a high signal-to-noise ratio (SNR) with low complexity, it has the infamous error propagation problem which becomes more severe as the density increases.

Today, the magnetic storage industry is seeking more sophisticated signal processing techniques to increase the storage density. It is expected that the drive capacity will reach 10 GB to 20 GB by the year 2000, compared to 1 GB to 2 GB in 1996. In addition to higher density, high speed is another goal for chip designers: 200 Mbit/s chips are not unusual today and there is no limit to the demand for higher speed.

The objective of this dissertation is to provide an advanced coding system for magnetic recording with improved performance. One of the key elements of this system is a novel reduced-state technique which is well-suited for the magnetic recording channel, and opens a path for future low-power consumption and high-speed implementations.

1.1.2 Magnetic Data Storage Channel Model

A magnetic data storage baseband channel is similar to a digital communication baseband channel as illustrated in Fig. 1.1. The major difference between them is their channel inputs. In a digital communications baseband channel, the channel is not limited to binary inputs; multilevel inputs can be used to avoid intersymbol interference (ISI) without increasing the channel bandwidth as data rate increases. In magnetic data storage channels, because of the hysteresis effects in magnetic media, saturation recording is used to maximize the SNR of the channel. Binary data +1 or -1 is represented by saturating the medium to one orientation or the opposite orientation. Because of this constraint on magnetic data storage channels, increase in ISI is inevitable as the channel



Data Storage Baseband Channel

Digital Communications Baseband Channel

Fig. 1.1. Data storage baseband channel and digital communications baseband channel.

density increases. Although binary channel inputs are widely used in today's storage systems, a ternary channel can be created by adding a demagnetized state to represent the additional '0' symbol; the demagnetized state is created by using a high frequency alternating current. The ternary channel could provide an attractive alternative for applying trellis codes especially when the storage density is high [9] .

The noise in Fig. 1.1 is assumed to be additive white Gaussian noise (AWGN). We ignore the intertrack interference [10],[11] in our model, and the channel is assumed to be a linear Lorentzian channel whose step response is described by the Lorentzian pulse

$$s(t) = \frac{1}{1+(2t/pw_{50})^2} \qquad (1.1)$$

where pw_{50} is called the 'half-amplitude width' and is equivalent to the time span that $s(t)$ is greater than or equal to half of its peak value. The channel density is defined as

$$S = \frac{pw_{50}}{T} \qquad (1.2)$$

where T is the duration of an input bit or symbol period. Let $\tau = t/T$, the normalized step response is given by

$$s(\tau) = \frac{1}{1+(2\tau/S)^2} \qquad (1.3)$$

and express the pulse response of the storage channel as

$$p(\tau) = s(\tau) - s(\tau - 1). \qquad (1.4)$$

Although the linear channel model is a good model up to the density of current interest, the nonlinearity in the storage channel is one of the major obstacles preventing the use of higher channel densities in data storage systems. Various models have been developed for the nonlinear mechanism, including the partial erasure model [12], the peak shifting model [13], and the Voterra model [14]. Another nonlinear mechanism is closely related to the use of magnetoresistive (MR) heads. MR heads possess a resistive element with a resistance that is a function of the magnetic field strength. The output signal from an MR head is larger than the traditional inductive read head, but MR heads require an electrical bias to overcome their inherent asymmetry, and the inaccuracy of this bias causes the asymmetric output of MR heads [15].

1.1.3 Data Storage Systems

A block diagram of a typical data storage system is shown in Fig. 1.2. The outer code is a Reed-Solomon (RS) code [16], which is a byte-oriented nonbinary error-correction-code (ECC). RS codes are very efficient in applications where burst errors other than random bit errors are more often encountered. The output of the RS encoder is interleaved so that a burst of several error bytes can be corrected by the RS code. More details about RS codes and the interleaver will be given in Chapter 3. The inner code could be a pure baseband modulation code such as the rate 8/9 RLL code or the rate 16/17 RLL code, or it could be a code that is used to improve the noise immunity

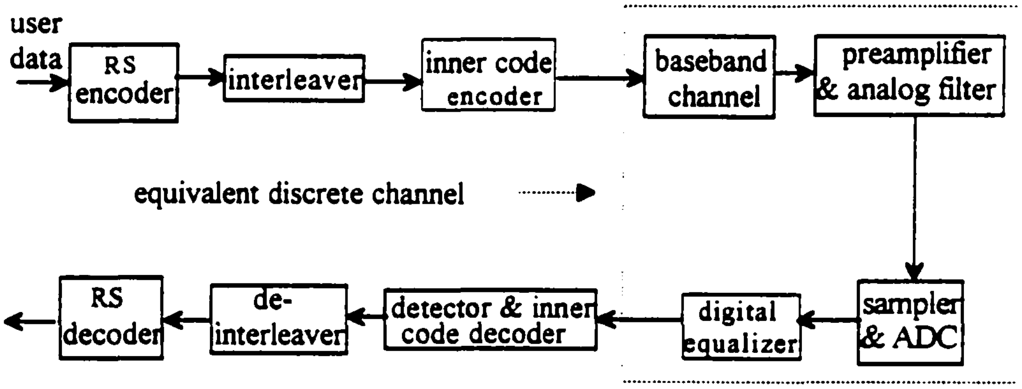


Fig. 1.2. Diagram of a data storage system.

while enforcing the RLL constraint such as a matched-spectral-null (MSN) code. The inner encoder outputs a coded binary sequence to the baseband channel which has been described in the previous subsection. At the receiver end, the channel output is first preamplified and filtered. The preamplifier is responsible for automatic gain control and nonlinearity compensation while the analog filter attenuates the out-of- band noise. Channel equalization is shared by the analog filter and the digital equalizer in a way determined by circuit designers. At the digital equalizer output, we have an equivalent discrete time channel that is set to a predefined target response such as PR4 or EPR4. The detector gives an estimation of the channel input based on the knowledge of the coded (or uncoded) predefined channel, and the Viterbi algorithm (VA) is usually used in the detector to realize maximum likelihood sequence estimation (MLSE). The bit error rate at the inner decoder output is expected to be 10^{-8}, the RS code is then responsible to reduce the bit error rate below 10^{-13} at the RS decoder output.

1.1.4 PR4ML and EPR4ML

PRML detection techniques were originally developed for high speed communications systems. Instead of equalizing the channel to an ISI free full response channel, a predefined amount of ISI is allowed in a PR system to match the channel characteristics and avoid severe noise enhancement. For example, in a PR4ML system, the channel is equalized to an equivalent discrete channel with a transfer function

$$h(D) = 1 - D^2 \tag{1.5}$$

or

$$x_k = a_k - a_{k-2} \tag{1.6}$$

where x_k is the noiseless channel output and a_k is the channel input taking values from the signal set {-1,+1}. The PR4 sequence can be viewed as two independent, interleaved dicode PR sequences with a transfer function

$$h(D') = 1 - D' \tag{1.7}$$

where D' refers to a delay $2T$. Therefore the PR4ML can be implemented with two 2-state Viterbi decoders operating on odd and even sequences, respectively. Because of its implementation simplicity and good match to the magnetic recording channel at densities around S=2, the PR4ML is one of the most widely used systems in data storage. However, higher order PR signaling is preferred for higher channel densities. For example, the EPR4 channel , with a transfer function

$$h(D) = 1 + D - D^2 - D^3 \ , \tag{1.8}$$

is a much better match to the magnetic recording channel than PR4 at a density of 2.5. The simulated performances of the EPR4ML and the PR4ML systems are compared in Fig. 1.3; the curves are plotted as the required channel SNR to achieve an error rate of 10^{-5}, versus user densities, and the channel SNR is defined as the ratio of the energy of the Lorentzian pulse to the double-sided power spectral density of the AWGN, *i.e.*,

$$SNR = \frac{\int s^2(t)dt}{N_0/2} \ . \tag{1.9}$$

In the simulation, a three-pole analog filter and a digital filter with seven taps are used. The filter design process began with a chosen sampling phase and pole locations for the analog filter. The optimum digital filter parameters are calculated and the mean-square error relative to the target response is measure. The pole locations of the analog filter are then modified recursively (and the digital filter redesigned at each step) using the

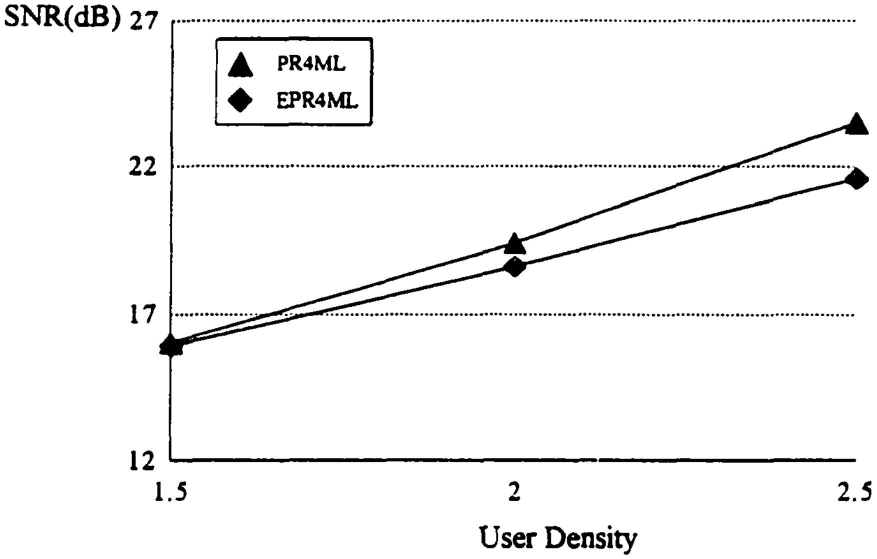


Fig. 1.3. Simulated performance of PR4ML and EPR4ML.

gradient of the mean-square error until a minimum is reached. The process is then repeated for a range of sampling phases, and the one found to yield the overall minimum mean-square error is used for system simulation. Since a rate 8/9 RLL code is used in the simulation, the relation between the channel density S_c and the user density S_u is given by

$$S_c = S_u \times \tfrac{9}{8} \ . \tag{1.10}$$

Although the complexity of EPR4ML is much higher than PR4ML, it offers a significant performance gain as shown in Fig. 1.3. The major concern of chip designers is how to increase the speed and bring down the power consumption of the EPR4ML chip.

1.2 Post EPR4ML Technology

1.2.1 MEEPR4 Signaling

As EPR4ML has started to replace PR4ML in read channel chips, searching for the post-EPR4ML technology becomes more and more pressing. The new detection scheme needs to deliver at least 2 dB gain over the EPR4ML. A straightforward solution is to look for a higher order general partial response (GPR) system, a so-called modified extended EPR4 (MEEPR4) system which has the same channel length as EEPR4 with the transfer function

$$h(D) = (1 - D^2)(a_0 + a_1 D + a_2 D^2) \tag{1.11}$$

where $\{a_0, a_1, a_2\}$ is left for optimization. As an example, consider the transfer function

$$h(D) = 5 + 4D - 3D^2 - 4D^3 - 2D^4 \tag{1.12}$$

which may provide as much as 1 dB gain over EPR4ML at a channel density around 2.5.

In the following, we illustrate how to calculate the error probability of an VA detector in a PR channel while establishing some notation to be used throughout this dissertation. Fig. 1.4 shows an equivalent discrete channel and related notation: $a_n \in \{-1, +1\}$ is the channel input , $\mathbf{a}$ denotes the correct input sequence, $\mathbf{a}'$ denotes an input sequence other than $\mathbf{a}$ and $\Delta\mathbf{a}'$ is the corresponding *input error sequence;* $\mathbf{h}$ is the impulse response of the partial response channel, and '*' denotes convolution; $\mathbf{x}$ is the correct output sequence, $\Delta\mathbf{x}'$ is the *output error sequence* corresponding to $\Delta\mathbf{a}'$; $\mathbf{y}$ is the noisy channel output and the noise $\mathbf{n}$ could be a correlated noise sequence. Let $p_e(\Delta\mathbf{x}')$ denote the error probability that the detector mistakes $\mathbf{a}'$ for $\mathbf{a}$. For a detector implemented with the VA, such an error event occurs when the path metric of $\mathbf{a}'$

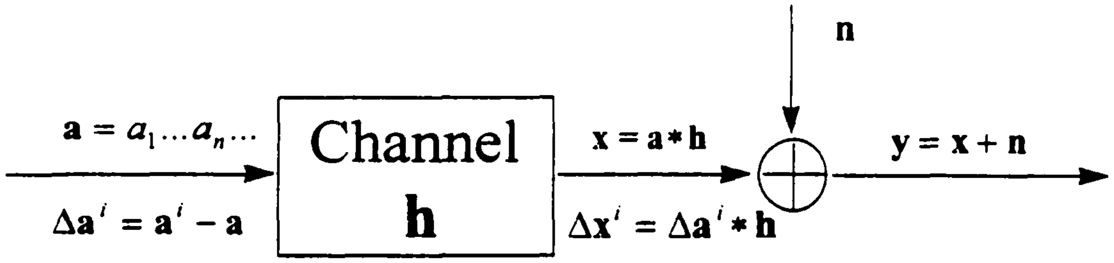

n
$\mathbf{a} = a_1 \ldots a_n \ldots$
$\Delta\mathbf{a}^i = \mathbf{a}^i - \mathbf{a}$
Channel
h
$\mathbf{x} = \mathbf{a} * \mathbf{h}$
$\Delta\mathbf{x}^i = \Delta\mathbf{a}^i * \mathbf{h}$
$\mathbf{y} = \mathbf{x} + \mathbf{n}$

Fig. 1.4. Equivalent channel and related notation.

is smaller than the path metric of $\mathbf{a}$, *i.e.*,

$$||\mathbf{x}^i - \mathbf{y}||^2 < ||\mathbf{x} - \mathbf{y}||^2 \tag{1.13}$$

or

$$||\Delta\mathbf{x}^i - \mathbf{n}||^2 < ||\mathbf{n}||^2 \tag{1.14}$$

$$2\Delta\mathbf{x}^i \bullet \mathbf{n}^T > ||\Delta\mathbf{x}^i||^2 \tag{1.15}$$

where the superscript "T"denotes vector transpose. The probability corresponding to the error event defined by (1.15) is then given by

$$p_e(\Delta\mathbf{x}^i) = Q(\sqrt{\tfrac{d^2(\Delta\mathbf{x}^i)}{4\sigma_n^2}}) \tag{1.16}$$

where σ_n^2 is the noise variance at the equivalent channel output, the $Q(\bullet)$ function is defined by

$$Q(z) = \int_z^{+\infty} \tfrac{1}{\sqrt{2\pi}} e^{-z^2/2} dz \ , \tag{1.17}$$

and the effective Euclidean distance $d(\Delta\mathbf{x}^i)$ is given by

$$d^2(\Delta\mathbf{x}^i) = \tfrac{(||\Delta\mathbf{x}^i||^2)^2}{\Delta\mathbf{x}^i \bullet \mathbf{r}_n(\Delta\mathbf{x}^i)^T} \tag{1.18}$$

where $\mathbf{r}$ is the noise correlation coefficient matrix. For AWGN, $\mathbf{r} = \mathbf{I}$ and the corresponding effective Euclidean distance coincides with the Euclidean distance

$$d^2(\Delta\mathbf{x}^i) = ||\Delta\mathbf{x}^i||^2 \quad . \tag{1.19}$$

Since Euclidean distance is a special case of the effective Euclidean distance, from now on, we will use (1.18) instead of (1.19) for performance evaluation but will refer to it as the Euclidean distance.

The noise correlation matrix is determined in the process of equalizing the Lorentizan channel to the target equivalent discrete channel. For example, using the aforementioned filter design approach, we design an analog filter and a digital equalizer for the MEEPR4 system, and the corresponding noise correlation coefficient function taking into account the equalization error is found to be

$$r_{nn} = [1, 0.054, -0.331, -0.057, -0.005, 0.062, -0.046, 0.031, -0.013] \tag{1.20}$$

and the corresponding correlation coefficient matrix is given by

$$\mathbf{r} = Toeplitz(r_{nn}) \quad . \tag{1.21}$$

The error rate of an MLSE detector is determined by the minimum Euclidean distance

$$d_{\min} = \min_{\Delta\mathbf{x}^i} \{d(\Delta\mathbf{x}^i)\} \; . \tag{1.22}$$

Using (1.18) , (1.20) and (1.22), the minimum distance for an uncoded MEEPR4 channel is found to be

$$d_{\min}^2 = 4 \times 42 \tag{1.23}$$

with the corresponding minimum distance error event defined by

$$\Delta\mathbf{a} = [\ldots, 0, 0, +2, -2, +2, 0, 0, \ldots] \tag{1.24}$$

or

$$\Delta\mathbf{x} = 2 \times [\ldots, 0, 0, 5, -1, -2, 3, -1, -2, -2, 0, 0, \ldots]. \qquad (1.25)$$

A high rate maximum transition run (MTR) code [17] can be used to eliminate this minimum distance error event by preventing three consecutive transitions in the input sequence, and the minimum distance is then increased to

$$d^2_{\min} = 4 \times 53 \ , \qquad (1.26)$$

with the corresponding minimum distance error event defined by

$$\Delta\mathbf{a} = [\ldots, 0, 0, 2, 0, 0, \ldots]$$

or

$$\Delta\mathbf{x} = 2 \times [\ldots, 0, 0, 5, 4, -3, -4, -2, 0, 0, \ldots]. \qquad (1.27)$$

The MTR code improves the minimum distance by 1.1 dB. In the following, the MEEPR4 channel will be assumed to be an MTR coded channel unless specified otherwise.

The overall error rate of an MLSE detector can be estimated from the minimum distance as

$$P_e \approx Q(\tfrac{d_{\min}}{2\sigma_n}) = Q(\sqrt{SNR_{eff}}) \qquad (1.28)$$

where the effective SNR is defined as

$$SNR_{eff} = (\tfrac{d_{\min}}{2\sigma_n})^2 \qquad (1.29)$$

as long as the SNR is high and the multiplicity only plays a minor role in the error rate evaluation.

1.2.2 Trellis Codes and PR Equalization

There is a large body of work on the use of trellis codes on PR channels [18]-[21].

The most successful efforts so far are the MSN codes [22]-[24] for the 1-*D* channel. The fundamental idea behind MSN codes is to design a code in such a way that the average power spectrum of the code has spectral nulls at the frequencies where the PR channel has zero frequency response. MSN codes can be implemented as a time-varying trellis code [25], and can deliver a 3 dB coding gain with a code rate of 8/10. However, the success of trellis codes in PR channels has been limited to the simple 1-*D* channel, and application to higher order PR channels has proven illusive. One obvious obstacle is the increase in the number of trellis states for longer PR channels. Trellis code design for storage systems using PR equalization will not be covered in this dissertation. It is, however, a promising topic for future study as discussed in Chapter 5.

1.2.3 Combining the LVA with High Rate Codes

The difficulty in applying trellis codes to storage channels is twofold. On one hand, the code needs to have a small trellis structure so that the decoder can estimate the most likely coded sequence with reasonable computation complexity; on the other hand, the storage channel requires a high rate code since the severe ISI in storage channels will cause a large code rate penalty that may negate all the coding gain. Keeping a high code rate and maintaining a substantial coding gain will inevitably lead to a large number of states for the trellis.

If the requirement of having a trellis structure is lifted, it is not difficult to design a high rate code such that the minimum distance of the coded output sequence is increased by a substantial amount (for example 2 dB). Although it will become difficult to optimally estimate the most likely coded sequence for such a code, it is not difficult to check whether a candidate sequence is a valid coded sequence. If there is an algorithm that can provide a list of best candidates that includes the correct sequence, then the coding gain can be fully achieved. A list of best candidates can be obtained by using the list Viterbi algorithm (LVA) [26] operating according to the trellis of the PR channel without considering the high rate code. Although the LVA cannot guarantee to provide a list of candidates that always includes the correct sequence, the probability that the correct sequence is not in the list should be smaller than the error rate of the VA, and the difference between these two probabilities is called the LVA gain. The LVA gain obviously improves with the number of candidates in the list, but the improvement will taper off as the number of candidates increases.

A coding scheme based on the above argument is proposed in Chapter 3. The LVA gain, or the probability that the correct sequence is not in the LVA candidate list, is investigated in Chapter 2, where a method for LVA performance evaluation is presented using the MEEPR4 channel as an example.

1.3 Reduced Complexity Implementation Techniques

While striving for more sophisticated signal processing techniques to improve the system performance, chip designers are interested in reducing the implementation complexity of these sophisticated algorithms without incurring a performance loss. The most important area of reduced complexity techniques are reduced-state techniques, where a number of ML trellis states are grouped together to form a reduced-state (RS) trellis. The traditional reduced-state sequence estimation (RSSE) [27] exhibits good performance for the magnetic channel modeled as a partial erasure channel [28], however, traditional RSSE techniques will cause significant performance loss for linear PR response channels encountered in data storage systems.

A novel reduced-state method called complement states grouping technique (CSGT) is developed in this work. The CSGT reduces the number of states by a factor of about two with only a negligible performance loss. It provides excellent solutions for reduced-complexity implementation of both the EPR4ML and the MEEPR4ML as shown in Chapter 4. In addition, Chapter 4 also investigates the performance of the LVA for the RS trellis generated by the CSGT, in particular, it is found that the LVA performance for the RS trellis is little different from its performance for the ML trellis of the MEEPR4 channel.

Chapter 2

The List Viterbi Algorithm: Implementation and Performance

2.1 Introduction

It is well known that the Viterbi algorithm can find the best (most likely) path through a trellis. The VA needs to be modified if the decoder requires knowledge of the $L>1$ best paths through the trellis. This modified VA is called the list Viterbi algorithm [26]. In this chapter, the implementation and performance of the LVA will be studied. First let us establish some notation needed for subsequent use. Fig. 2.1 shows a radix-2 subtrellis which is the basic unit of a trellis for an ISI channel with binary inputs.

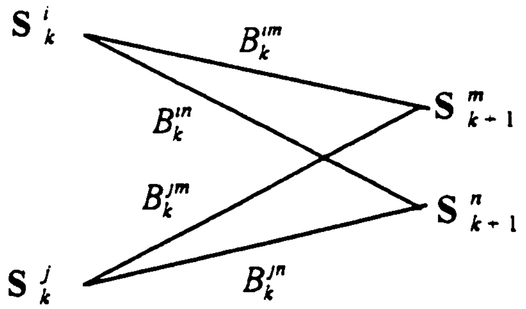


Fig. 2.1. Radix-2 subtrellis for the LVA.

Let $\mathbf{S}_k^m = [S_k^m(1), \ldots, S_k^m(l), \ldots, S_k^m(L)]$ where $S_k^m(l)$ denotes the state metric (or path metric) for the l^{th} best candidate that reaches state m at the k^{th} trellis stage, and B_k^{im} denotes the branch metric from state i to state m at the k^{th} stage.

The L globally best candidates can be found in a recursive way similar to the classical VA, using the following recursion for updating state m at the k^{th} stage

$$S^m_{k+1}(l) = \min^{(l)}_{1\le p\le L, 1\le q\le L} [S^i_k(p) + B^{im}_k, S^j_k(q) + B^{jm}_k] \tag{2.1}$$

where $\min^{(l)}$ denotes the l^{th} smallest value. Other state metrics can be updated in a similar way. Equation (2.1) can be implemented in a parallel way that simultaneously updates all the L best state metrics for state m , however, the implementation complexity will increase with L^2. A better solution is to implement (2.1) in a serial manner, which updates $S^m_{k+1}(l)$ only after the updating of $S^m_{k+1}(l-1)$. This will lead to a computation complexity which will only increase linearly with L. The resultant recursion time may need to increase linearly with L due to the sequential nature of the computation. Fortunately, as shown in next section, a pipelined architecture for the LVA leads to a recursion time which is independent of L.

2.2 Pipelined Implementation of the LVA

Fig. 2.2 shows a serial implementation of (2.1), where the step for updating $S^m_{k+1}(l)$ is shown. More specifically, let $I^m_k(l)$ and $J^m_k(l)$ be two pointers to the two contention paths to $S^m_{k+1}(l)$, one from state i, and one from state j. Suppose the contention path from state i is selected, then the two contention paths to $S^m_{k+1}(l+1)$ are shown as in the lower part of Fig. 2.2 . The procedure for updating S^m_{k+1} can be formally stated as follows:

<u>*Step 1*</u> Updating $S^m_{k+1}(1)$

$$I^m_k(1) = 1, J^m_k(1) = 1$$

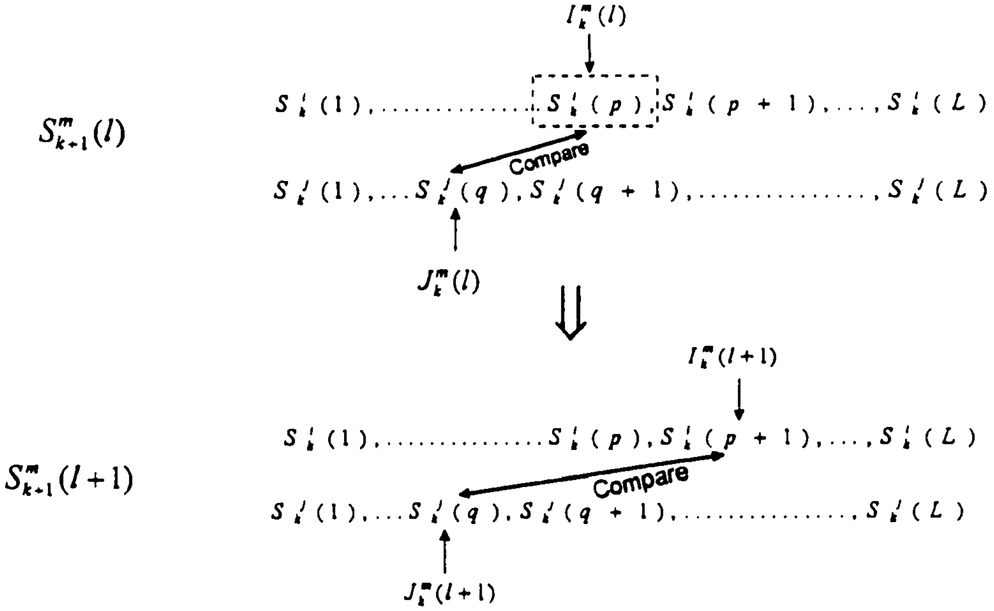

$S^m_{k+1}(l)$
$I^m_k(l)$
$S^i_k(1), \ldots, S^i_k(p), S^i_k(p+1), \ldots, S^i_k(L)$
Compare
$S^j_k(1), \ldots S^j_k(q), S^j_k(q+1), \ldots, S^j_k(L)$
$J^m_k(l)$
$S^m_{k+1}(l+1)$
$I^m_k(l+1)$
$S^i_k(1), \ldots, S^i_k(p), S^i_k(p+1), \ldots, S^i_k(L)$
Compare
$S^j_k(1), \ldots S^j_k(q), S^j_k(q+1), \ldots, S^j_k(L)$
$J^m_k(l+1)$

Fig. 2.2. Updating the l^{th} candidate.

$$[C^m_{k+1}(1), S^m_{k+1}(1)] = f_{ACS}[S^i_k(I^m_k(1)), S^j_k(J^m_k(1)), y_k]$$

$$\vdots$$

Step l Updating $S^m_{k+1}(l)$

$$I^m_k(l) = I^m_k(l-1) + C^m_{k+1}(l-1), J^m_k(l) = J^m_k(l-1) + \overline{C^m_{k+1}(l-1)}$$

$$[C^m_{k+1}(l), S^m_{k+1}(l)] = f_{ACS}[S^i_k(I^m_k(l)), S^j_k(J^m_k(l)), y_k]$$

$$\vdots$$

Step L Updating $S^m_{k+1}(L)$

$$I^m_k(L) = I^m_k(L-1) + C^m_{k+1}(L-1), J^m_k(L) = J^m_k(L-1) + \overline{C^m_{k+1}(L-1)}$$

$$[C^m_{k+1}(L), S^m_{k+1}(L)] = f_{ACS}[S^i_k(I^m_k(L)), S^j_k(J^m_k(L)), y_k]$$

where $f_{ACS}[\bullet]$ denotes the add-compare-select (ACS) operations specified as follows

$$[C^m_{k+1}, S^m_{k+1}] = f_{ACS}[S^i_k, S^j_k, y_k]$$

where

$$C^m_{k+1} = (S^i_k + B^{im}_k) < (S^j_k + B^{jm}_k)$$

$$S^m_{k+1} = mux[S^i_k + B^{im}_k, S^j_k + B^{jm}_k, C^m_{k+1}]$$

with

$$mux[x1, x2, c] = \begin{cases} x1, & \text{if } c = 1 \\ x2, & \text{if } c = 0 \end{cases}$$

and y_k is the noisy channel output used to compute the branch metrics.

Let $P_k(l)$ represent all the computations required for updating the l^{th} best state metrics for all states at the k^{th} trellis stage, and note that $P_k(l)$ will only depend on $P_k(1), ..., P_k(l-1)$ and $P_{k-1}(1), ..., P_{k-1}(l)$. Therefore, the following computations can be carried out simultaneously

$$..., P_{k-1}(l+1), P_k(l), P_{k+1}(l-1), .. \quad .$$

This will allow a pipelined implementation of the LVA. As an example, the computation schedule for a 3-candidate LVA is shown in Fig. 2.3.

In summary, with the pipelined implementation, the LVA recursion time is approximately the same as the ACS recursion time plus some overhead, while the computation complexity of the LVA is L times the complexity of the VA.

2.3 Performance Evaluation of the LVA

2.3.1 Introduction

Computing the LVA error rate is a multi-dimension integration problem. Let $p_f(\Delta\mathbf{x}^1, ..., \Delta\mathbf{x}^L)$ denote the *error event probability* that there are L output sequences

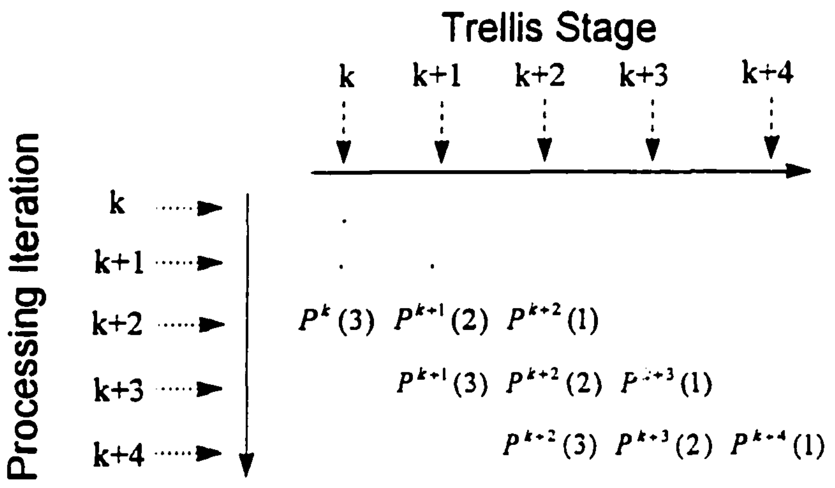

Trellis Stage
k
k+1
k+2
k+3
k+4
Processing Iteration
k
k+1
k+2
k+3
k+4
$P^k(3)$ $P^{k+1}(2)$ $P^{k+2}(1)$
$P^{k+1}(3)$ $P^{k+2}(2)$ $P^{k+3}(1)$
$P^{k+2}(3)$ $P^{k+3}(2)$ $P^{k+4}(1)$

Fig. 2.3. Pipelining schedule for a 3-candidate LVA.

that are more likely than the correct sequence, where $(\Delta\mathbf{x}^1,\ldots,\Delta\mathbf{x}^L)$ denotes the L corresponding *output error sequences* of the error event. The subscript f denotes the probability density function (pdf) of the noise, which is assumed to be Gaussian distributed but could be correlated. It follows that

$$p_f(\Delta\mathbf{x}^1,\ldots,\Delta\mathbf{x}^L) = \int_D f(\mathbf{n})d\mathbf{n} = \int_D f(n_1 \ldots n_m)dn_1 \ldots dn_m \tag{2.2}$$

where m is the dimension given by the length of the error event and D is the integration region determined by:

$$D:\left\{\begin{array}{c} ||\Delta\mathbf{x}^1-\mathbf{n}||^2 < ||\mathbf{n}||^2 \\ \vdots \\ ||\Delta\mathbf{x}^L-\mathbf{n}||^2 < ||\mathbf{n}||^2 \end{array}\right. \quad \text{or} \quad \left\{\begin{array}{c} \Delta\mathbf{x}^1 \bullet \mathbf{n}^T > ||\Delta\mathbf{x}^1||^2/2 \\ \vdots \\ \Delta\mathbf{x}^L \bullet \mathbf{n}^T > ||\Delta\mathbf{x}^L||^2/2 \end{array}\right. . \tag{2.3}$$

This integration can only be computed numerically, and deterministic numerical methods are not the best for this multi-dimension integration problem, because they may take a long time to give an acceptable result. A better way is to use the statistical Monte Carlo method [29]. Monte Carlo integration requires less computation time while providing an acceptable variance for the integration result. However, to compute the integration may still require an inordinate amount of time if we have to compute the *error event probabilities* for all types of error events. To further reduce the computation burden, it is desirable to derive an upper bound for the *error event probability* (as tight as possible) and use this upper bound to exclude all the unimportant error events so that the Monte Carlo method only needs to be applied to a small number of cases. In the following subsections, we derive a bound for the *error event probability*, the bound is used to exclude all the unimportant error events and the remaining cases form a list of dominant error events. Monte Carlo integration is then used to compute the *error event probabilities* of these dominant error events, and the error rate of the LVA can be evaluated from these dominant *error event probabilities.* Finally, it is pointed out that a lower bound of the LVA gain over VA is provided in [26], and is given by

$$G \geq \frac{2L}{L+1}. \tag{2.4}$$

This result was derived using simplex geometry [30], and is independent of the channel. It can provide a quick estimation of the LVA gain, however, there are several disadvantages in using this bound. First, it is a bound rather than a precise estimation. Second, the noise is required to be AWGN, which is not true in most cases, and this could make a significant difference in many cases. Third, the multiplicity of error events is not taken into account in (2.4), which may cause significant inaccuracy in some cases. All these problems can be overcome by the following method.

2.3.2 An Upperbound on the Error Event Probability

Let $\alpha_1, \ldots, \alpha_L$ be L constants. Multiplying both sides of each inequality in (2.3) by these constants gives

$$\left\{\begin{array}{l} \alpha_1 \Delta\mathbf{x}^1 \bullet \mathbf{n}^T > \alpha_1 ||\Delta\mathbf{x}^1||^2/2 \\ \quad\quad\quad\quad \cdot \\ \quad\quad\quad\quad \cdot \\ \quad\quad\quad\quad \cdot \\ \alpha_L \Delta\mathbf{x}^L \bullet \mathbf{n}^T > \alpha_L ||\Delta\mathbf{x}^L||^2/2 \end{array}\right. \quad \Rightarrow \sum_{i=1}^{L} \alpha_i \Delta\mathbf{x}^i \bullet \mathbf{n}^T > \sum_{i=1}^{L} \alpha_i ||\Delta\mathbf{x}^i||^2/2 \ . \tag{2.5}$$

For any choice of $\alpha_1, \ldots, \alpha_L$, the integration region defined by (2.3) is obviously included in the integration region defined by (2.5). Therefore the *error event probability* is upperbounded by the integration over the region defined by (2.5) which is equivalent to the probability that

$$\sigma_n \sqrt{\sum_{i=1}^{L} \sum_{j=1}^{L} \alpha_i \alpha_j (\Delta\mathbf{x}^i) \mathbf{r} (\Delta\mathbf{x}^j)^T}\, n_0 > \sum_{i=1}^{L} \alpha_i ||\Delta\mathbf{x}^i||^2/2 \tag{2.6}$$

where n_0 is $N(0,1)$ distributed, σ_n^2 is the noise variance and $\mathbf{r}$ is the noise correlation coefficient matrix given bv (1.21). The *error event probability* is then upperbounded by

$$p_f(\Delta\mathbf{x}^1, \ldots, \Delta\mathbf{x}^L) < Q(\sqrt{\frac{d_{bound}^2(\Delta\mathbf{x}^1, \ldots, \Delta\mathbf{x}^L)}{4\sigma_n^2}}) \tag{2.7}$$

where

$$d_{bound}^2(\Delta\mathbf{x}^1, \ldots, \Delta\mathbf{x}^L) = \underset{\{\alpha_1, \ldots, \alpha_L\}}{Max} \{[\sum_{i=1}^{L} \alpha_i ||\Delta\mathbf{x}^i||^2]^2 / [\sum_{i=1}^{L} \sum_{j=1}^{L} \alpha_i \alpha_j (\Delta\mathbf{x}^i) \mathbf{r} (\Delta\mathbf{x}^j)^T]\} \ . \tag{2.8}$$

A computer exhaustive search can be used to find the optimal $\{\alpha_1, \ldots, \alpha_L\}$ that achieves d_{bound}. It is not difficult to see that the solution is not unique, more specifically, if $\{\alpha_1, \ldots, \alpha_L\}$ is a solution then any constant k times $\{\alpha_1, \ldots, \alpha_L\}$ will also be a solution. For this reason, we choose $\alpha_L = 1$.

Example 2.1 : Consider a 3-candidate LVA for the MEEPR4 channel, and let us compute the *error event probability* of an error event defined by the *input error sequences*

$$\Delta\mathbf{a}^1 = [0,0,-2,0,0]$$
$$\Delta\mathbf{a}^2 = [0,0,0,0,2]$$
$$\Delta\mathbf{a}^3 = [2,-2,0,0,0]$$

with the corresponding *output error sequences*

$$\Delta\mathbf{x}^1 = 2\times[0,0,-5,-4,3,4,2,0,0]$$
$$\Delta\mathbf{x}^2 = 2\times[0,0,0,0,5,4,-3,-4,-2]$$
$$\Delta\mathbf{x}^3 = 2\times[5,-1,-7,-1,2,2,0,0,0].$$

To compute $d_{bound}(\Delta\mathbf{x}^1,\Delta\mathbf{x}^2,\Delta\mathbf{x}^3)$, a computer exhaustive search is carried out using (2.8). With the correlation coefficient function given by (1.20), the optimal $(\alpha_1,\alpha_2,\alpha_3)$ is found to be

$$\alpha_1 = 0.3,\ \alpha_2 = 1.2,\ \alpha_3 = 1$$

with the corresponding $d^2_{bound} = 80\times 4$, which is 1.81 dB more than $d^2_{\min} = 53\times 4$.

Define a lower bound of an *error event gain* as

$$G_{bound}(\Delta\mathbf{x}^1,\ldots,\Delta\mathbf{x}^L) = d_{bound}(\Delta\mathbf{x}^1,\ldots,\Delta\mathbf{x}^L)/d_{\min} \quad . \tag{2.9}$$

Table 2.1 lists all the error events with $G_{bound} < 2dB$. It is noted that we can get the *error input sequences* of case 8 by flipping the *error input sequences* of case 10, so case 10 and case 8 should have exactly the same performance. Similarly, pairs of (case 11, case 7), (case 12, case 2), (case 13, case 1) should have exactly the same performance. Error events with opposite sign of the cases shown above are not included in the table, since there is no difference between them as far as the performance is concerned.

The Monte Carlo integration only needs to be applied to the error events in Table 2.1 since the overall error rate is dominated by these important error events.

TABLE 2.1
IMPORTANT ERROR EVENTS FOR A 3-CANDIDATE LVA FOR THE MEEPR4 CHANNEL

	$\Delta\mathbf{a}^1$	$\Delta\mathbf{a}^2$	$\Delta\mathbf{a}^3$	$G_{bound}(dB)$
Case 1	0 0 -2 0 0	0 0 0 0 2	2 -2 0 0 0	1.81
Case 2	0 0 -2 0 0	0 0 0 0 2	2 0 0 0 2	1.96
Case 3	0 0 -2 2 0	0 0 0 0 2	2 -2 0 0 0	1.79
Case 4	0 0 -2 0 0	0 0 -2 2 0	2 -2 0 0 0	1.37
Case 5	0 2 -2 0 0	0 0 0 -2 0	2 0 0 0 0	1.32
Case 6	0 2 -2 0 0	0 0 0 -2 2	2 0 0 0 0	1.79
Case 7	0 2 -2 0 0	0 0 -2 0 0	2 0 0 0 0	1.44
Case 8	0 -2 0 0 0	0 0 -2 0 0	2 -2 0 0 0	1.74
Case 9	0 -2 0 0 0	0 0 -2 2 0	2 -2 0 0 0	1.36
Case 10	0 -2 0 0 0	0 -2 2 0 0	-2 0 0 0 0	1.74
Case 11	-2 0 0 0 0	0 0 2 0 0	-2 2 0 0 0	1.44
Case 12	-2 0 0 0 0	0 0 2 0 0	-2 0 0 0 -2	1.96
Case 13	0 0 -2 0 0	2 0 0 0 0	0 0 0 -2 2	1.81

2.3.3 Monte Carlo Integration

The Monte Carlo method is often used in simulations, but it can also be applied to integration problems especially multi-dimension integration. Suppose we want to do the integration

$$I = \int_D f(\mathbf{x})d\mathbf{x} \qquad (2.10)$$

where D is the integration region, defined by

$$A_j < x_j < B_j,\ j = 1, \ldots, m \qquad (2.11)$$

where A_j, B_j are the lower and upper limits of the j^{th} integration. We can rewrite the integration as

$$I = \int_D \frac{f(\mathbf{x})}{p(\mathbf{x})} p(\mathbf{x}) d\mathbf{x} = E[\frac{f(\mathbf{x})}{p(\mathbf{x})}] \tag{2.12}$$

where $p(\mathbf{x})$ is any pdf, and $\mathbf{x}$ is distributed according to $p(\mathbf{x})$. In most of cases, $p(\mathbf{x})$ is assumed to be a uniform, *i.e.*,

$$p(\mathbf{x}) = \begin{cases} \frac{1}{V}, \text{ if } \mathbf{x} \in D \\ 0, \text{ otherwise} \end{cases} \tag{2.13}$$

where V is the volume of D and

$$V = \prod_{j=1}^{m} (B_j - A_j) \ , \tag{2.14}$$

and

$$I = VE[f(\mathbf{x})] \ . \tag{2.15}$$

An unbiased estimator of I is its sample mean

$$\theta_1 = \frac{V}{N} \sum_{i=1}^{N} f(\mathbf{x}_i) \tag{2.16}$$

where N is the number of samples, and $\mathbf{x}_i$ is a random variable uniformly distributed over D. The variance of this estimate can be found as

$$\begin{aligned} Var(\theta_1) &= V^2 \frac{1}{N^2} \sum_{i=1}^{N} Var(f(\mathbf{x}_i)) = V^2 \frac{1}{N^2} \sum_{i=1}^{N} (E[f^2(\mathbf{x}_i)] - (I/V)^2) \\ &= V^2 \frac{1}{N} E[f^2(\mathbf{x})] - \frac{1}{N} I^2 = \frac{1}{N} \times (V \int_D f^2(\mathbf{x}) d\mathbf{x} - I^2) \\ &\approx \frac{V}{N} \int_D f^2(\mathbf{x}) d\mathbf{x} \end{aligned} \tag{2.17}$$

where the approximation can be used if the dimension is large. We define a roughness factor as

$$F = \frac{V \int_D f^2(\mathbf{x}) d\mathbf{x}}{I^2} \ . \tag{2.18}$$

F is a measurement of the roughness of the function $f(\mathbf{x})$ in the region D, and $F \geq 1$. Table 2.2 shows F/V for $f(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2}$ for different integration regions. We observe that for the same lower limit A, F is approximately proportional to V.

TABLE 2.2

F/V FOR $f(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2}$

D=[A, B]	F/V
[2, 3]	1.43
[2, 4]	1.27
[2, 5]	1.27
[2, 6]	1.27
[3, 4]	1.79
[3, 5]	1.71
[3, 6]	1.71
[3, 7]	1.71
[4, 5]	2.2
[4, 6]	2.17
[4, 7]	2.17
[4, 8]	2.17

To reduce the variance of the Monte Carlo integration, the most effective way is to use stratified sampling. In stratified sampling, we break the region D into N disjoint subregions $D_i, i = 1, 2, \ldots, N$. Define

$$I_i = \int_{D_i} f(\mathbf{x}) d\mathbf{x}$$

as the subregion of integration which can be estimated separately by the Monte Carlo method. For simplicity, only one sample is used for each subregion of integration although more samples (correlated sampling) may improve the efficiency. Then, the new estimator is

$$\theta_2 = \hat{I} = \sum_{i=1}^{N} \hat{I}_i = \sum_{i=1}^{N} V_i f(\mathbf{x}_i) \tag{2.19}$$

where V_i is the volume of D_i, and $\mathbf{x}_i$ is distributed uniformly in D_i. The variance of (2.19) is generally much smaller than (2.17), however, a precise comparison between $Var(\theta_1)$ and $Var(\theta_2)$ is almost impossible . A rough comparison can be made assuming f is a Gaussian distributed function.

$$\begin{aligned} Var(\theta_2) &= \sum_{i=1}^{N} Var(\hat{I}_i) = \sum_{i=1}^{N} F_i I_i^2 \approx \sum_{i=1}^{N} k^m V_i I_i^2 \\ &\approx \sum_{i=1}^{N} k^m \tfrac{V}{N} I_i^2 \approx \tfrac{V}{N} k^m I^2 \tfrac{1}{N} \approx Var(\theta_1)/N \,. \end{aligned} \tag{2.20}$$

To justify the above approximation, we need the following assumptions :

1. $F_i \approx k^m V_i$ (m is the dimension, k is a constant), which is roughly true according to Table 2.2.

2. $V_i = \frac{V}{N}$ because we assume the total number of samples to be the same for the two estimators and each subregion has the same size.

3. $I_i = \frac{I}{N}$ may not be a good assumption and therefore the actual variance may turn out to be larger than (2.20).

We can use (2.20) to give a rough estimate of how small D_i should be. The choice of the size of D_i is critical, too large will lead to an unacceptable variance; too small will lead to an unnecessary increase in computation time. For the evaluation of the LVA *error event probability*, an empirical choice of the size of the lattice D_i is $(1\sim2)\sigma_n$ in each dimension and this choice will be used for the integration in Subsection 2.3.4. It is noted that for deterministic numerical methods, this lattice size is too large to guarantee an acceptable solution and reducing the size will increase the computation time dramatically for multi-dimension integration.

2.3.4 Performance of a 3-Candidate LVA for the MEEPR4 Channel

To evaluate the LVA error rate for the MEEPR4 channel, the bound (2.8) is first used to select the important error events as shown in Table 2.1; the Monte Carlo method is then used to compute the ***error event probabilities*** of all these cases, and the overall error rate can be computed as

$$p_e^{LVA} \approx \Sigma M(\Delta\mathbf{a}^1, \ldots, \Delta\mathbf{a}^L) p_f(\Delta\mathbf{x}^1, \ldots, \Delta\mathbf{x}^L) \tag{2.21}$$

where $M(\Delta\mathbf{a}^1, \ldots \Delta\mathbf{a}^L)$ is the multiplicity term associated with the given error event and is given by

$$M(\Delta\mathbf{a}^1, \ldots, \Delta\mathbf{a}^L) = \frac{1}{2^{V(\Delta\mathbf{a}^1 \ldots \Delta\mathbf{a}^L)}} \tag{2.22}$$

where $V(\Delta\mathbf{a}^1, \ldots, \Delta\mathbf{a}^L)$ is the total number of error positions in $(\Delta\mathbf{a}^1, \ldots, \Delta\mathbf{a}^L)$. For example,

$$V(\Delta\mathbf{a}^1, \ldots, \Delta\mathbf{a}^3) = 4 \tag{2.23}$$

for$\{\Delta\mathbf{a}^1 = [0, 0, -2, 0, 0], \Delta\mathbf{a}^2 = [0, 0, 0, -2, 2], \Delta\mathbf{a}^3 = [2, 0, -2, 0, 0]\}$.

Before using Monte Carlo integration, (2.2) needs to be modified

$$p_f(\Delta\mathbf{x}^1, \ldots, \Delta\mathbf{x}^L) = \int_D f(\mathbf{n}) d\mathbf{n} \approx \int_{D'} f'(\mathbf{n}) d\mathbf{n} \tag{2.24}$$

where D is defined by the error event as in (2.3), and

$$f'(n) = \begin{cases} f(n) & \text{if } n \in D \\ 0 & \text{otherwise} \end{cases} ; \tag{2.25}$$

D' is defined by

$$D' : \quad \begin{cases} A_1 < n_1 < B_1 \\ \quad\quad . \\ \quad\quad . \\ \quad\quad . \\ A_m < n_m < B_m , \end{cases} \tag{2.26}$$

and the choice of the range of D' is important, too small will lead to an unacceptable biased result; too large will lead to an unnecessary increase in computation time. One practical suggestion is to choose a relative small size of D' , then increase the size (in all dimensions) by a certain amount , if the probability only increases by a negligible amount, then we do not need to increase the size of D' further. The following example shows how the integration result changes with the size of D'.

Example 2.2: The error event is given by

$$\Delta\mathbf{a}^1 = [2, 0, 0], \Delta\mathbf{a}^2 = [0, 0, -2], \Delta\mathbf{a}^3 = [2, -2, 0]$$

with the corresponding *output error sequences*

$$\Delta\mathbf{x}^1 = [5, 4, \ -3, \ -4, -2, 0, 0]$$
$$\Delta\mathbf{x}^2 = [0, 0, \ -5, \ -4, \ 3, \ 4, 2]$$
$$\Delta\mathbf{x}^3 = [5, -1, -7, -1, \ 2, 2, 0].$$

For the first D', each dimension is stratified in the following way

Dimension 1: $[-2, -0.5, 0.5, 1.5, 2.5, 3.5, 5.5]\sigma_n$

Dimension 2: $[-3.5, -1.5, -0.5, 0.5, 1.5, 2.5, 4.5]\sigma_n$

Dimension 3: $[-5.5, -3.5, -2.5, -1.5, -0.5, 0.5, 2]\sigma_n$

Dimension 4: $[-5, -3, -2, -1, 0, 1, 2.5]\sigma_n$

Dimension 5: $[-4, -2.5, -1.5, -0.5, 0.5, 1.5, 3.5]\sigma_n$

Dimension 6: $[-2.5, -0.5, 0.5, 1.5, 2.5, 3.5, 5]\sigma_n$

Dimension 7: $[-2.5, -1, 0, 1, 2, 3, 5]\sigma_n$.

For the second D', each dimension is stratified in the following way

Dimension 1: $[-3, -1, 0, 1, 2, 3, 4.5, 6.5]\sigma_n$

Dimension 2: $[-4, -2, -1, 0, 1, 2, 4, 6]\sigma_n$

Dimension 3: $[-6, -4, -3, -2, -1, 0, 1.5, 3.5]\sigma_n$

Dimension 4: $[-6, -4, -3, -2, -1, 0, 1.5, 3.5]\sigma_n$

Dimension 5: $[-4.5, -2.5, -1.5, -0.5, 0.5, 1.5, 3, 5]\sigma_n$

Dimension 6: $[-3.5, -2, -0.5, 0.5, 1.5, 2.5, 4, 6]\sigma_n$

Dimension 7: $[-3.5, -2, -0.5, 0.5, 1.5, 2.5, 4, 6]\sigma_n$

where the noise variance $\sigma_n^2 = 3.75$.

The results for the first and second D' are shown in Table 2.3.

TABLE 2.3
ERROR EVENT PROBABILITY VERSUS SIZE OF INTEGRATION REGION

	$p_f(\Delta\mathbf{x}^1, \ldots, \Delta\mathbf{x}^L) \times 10^7$									Average Error Event Probability
First D'	6.52	5.87	6.01	6.10	6.09	6.32	6.95	6.59	6.33	6.36E-7
Second D'	6.23	6.47	6.53	5.89	6.16	6.94	6.65	6.65	6.3	6.42E-7

It is obvious that the increase in the second integration is negligible, so the first D' is large enough. In addition , the results also confirm that the size of the subregion D'_i is small enough to give an acceptable integration variance.

At last, the Monte Carlo integration results for the cases in Table 2.1 are given in Table 2.4. The *error event gain* is defined as

$$G(\Delta\mathbf{x}^1, \ldots, \Delta\mathbf{x}^L) = \frac{SNR_{eff}^{VA}}{SNR_{eff}^{LVA}} = (\frac{\sigma_n^{LVA}}{\sigma_n^{VA}})^2 \quad , \tag{2.27}$$

TABLE 2.4

ERROR EVENT PROBABILITY OF A 3-CANDIDATE LVA FOR THE MEEPR4 CHANNEL

	$(\sigma_n^{LVA})^2$	$(\sigma_n^{VA})^2$	$p_f(\Delta\mathbf{x}^1, \ldots, \Delta\mathbf{x}^L)$	$G(\Delta\mathbf{x}^1, \ldots, \Delta\mathbf{x}^L)$	G_{bound}	$G\text{-}G_{bound}$
Case 1	4.25	2.33	9.33E-7	2.61	1.81	0.8
Case 2	4.25	2.33	9.25E-7	2.61	1.96	0.65
Case 3	3.75	2.10	2.60E-7	2.52	1.79	0.73
Case 4	3.75	2.31	8.40E-7	2.10	1.37	0.73
Case 5	3.75	2.36	10.8E-7	2.01	1.32	0.69
Case 6	4.25	2.4	13.4E-7	2.48	1.79	0.69
Case 7	3.75	2.26	6.40E-7	2.20	1.44	0.76
Case 8	4.25	2.36	10.5E-7	2.56	1.74	0.82
Case 9	3.75	2.30	8.10E-7	2.12	1.36	0.76
Case 14	5.00	2.32	8.80E-7	3.34	2.58	0.76

under the condition that $p_f(\Delta\mathbf{x}^1, \ldots, \Delta\mathbf{x}^L) = Q(\sqrt{SNR_{eff}^{VA}})$, where the effective *SNRs* for the VA and the LVA (SNR_{eff}^{VA} and SNR_{eff}^{LVA}) are both defined by (1.29). Using (2.27), the error event probability can be expressed as

$$p_f(\Delta\mathbf{x}^1, \ldots, \Delta\mathbf{x}^L) = Q(\sqrt{SNR_{eff}^{LVA} G(\Delta\mathbf{x}^1, \ldots \Delta\mathbf{x}^L)}\ ) \,. \tag{2.28}$$

Strictly speaking, the *error event gain* depends on the *SNR*, but the dependence is not strong, so it can be regarded as independent of the *SNR* at an error rate around the target error rate of 10^{-8}.

It is noted that there is no need to compute the *error event probabilities* for cases 10, 11, 12, 13, since these probabilities are exactly the same as for cases 8, 7, 2, 1, respectively. The *error event probability* for case 14 was computed in order to confirm that a large $G_{bound}(\Delta\mathbf{x}^1, \ldots, \Delta\mathbf{x}^2)$ corresponds to a smaller *error event probability*. This case is defined by the following input error sequences

$$\Delta\mathbf{a}^1 = [0,0,-2,2,0], \Delta\mathbf{a}^2 = [0,0,0,0,2], \Delta\mathbf{a}^3 = [2,0,0,2,0] .$$

Finally, the *LVA gain* $G_{\min}$ is defined as

$$G_{\min} = \min_{\{\Delta\mathbf{x}^1,\dots,\Delta\mathbf{x}^L\}} \{G(\Delta\mathbf{x}^1,\dots,\Delta\mathbf{x}^L)\} \cdot \tag{2.29}$$

The LVA performance will be governed by the ***LVA gain*** if the multiplicity of error events only plays a minor role. From Table 2.4, we conclude that the *LVA gain* of a 3-candidate LVA is about 2 dB for the MMEPR4 channel.

It is interesting to note that the difference between the *error event gain*, $G(\Delta\mathbf{x}^1,\dots,\Delta\mathbf{x}^L)$, and the lower bound of the gain, G_{bound} , is approximately fixed around 0.6~0.8 dB. If this were to hold, there would be no need to carry out the integration, unfortunately, this conclusion cannot be proved.

In summary, to evaluate the LVA performance, the first step is to use (2.8) to select a list of important error events, then Monte Carlo integration can be used to compute the *error event probabilities* for these important error events, finally (2.21) is used to obtain the overall LVA error rate.

Chapter 3

Combining a High Rate Error Detection Code with an LVA

3.1 Coding Strategy

Coding is a powerful tool for improving noise immunity in many communications channels. However, its application to magnetic recording systems has been very limited so far. Although there has been a significant amount of work devoted to the study of using trellis codes to improve noise immunity for magnetic recording channels, we have not yet seen an exciting solution. The performance of trellis codes in magnetic recording systems is much less impressive than in other communications channels [31]. As discussed in Chapter 1, the requirement for a high rate code and the limitation on decoder complexity are two major factors discouraging the use of trellis codes for magnetic recording systems, especially for those systems using high order PR signaling.

Compared to the design of a high rate trellis code, it is much easier to find a high rate block code (HR code) with substantial coding gain by adding some check bits at the end of a large data block. The disadvantage of such a coding scheme is obvious, *i.e.*, MLSE is impractical since the code does not possess a trellis structure. However, it should not be difficult to design the HR code such that the decoder can do an easy checking on whether a given candidate sequence is a valid codeword sequence. Therefore, if the channel detector can provide a list of candidates to the decoder such that the probability that the correct code sequence is not in the list is much smaller than the error probability of an MLSE channel detector, then a certain coding gain can be achieved. Using an LVA operating on the trellis defined by the PR channel, we can

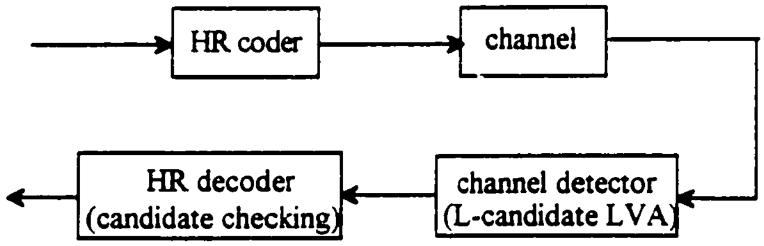


Fig. 3.1. Block diagram of the new coding system.

obtain a list of best candidates with the same complexity of an LVA channel detector. If the HR code is powerful enough, then the overall performance will be governed by the probability that the correct sequence is not in the LVA list. For example, in Chapter 2, we show that a 3-candidate LVA for the MEEPR4 channel exhibits a 2 dB LVA gain.

Based on the above argument, Fig. 3.1 shows a block diagram of the coding system. At the transmitter end, a HR code is inserted before the data is outputted onto the channel. The HR code is a systematic block code, *i.e.*, check bits are added to each data block to increase the minimum Euclidean $d_{\min}$ between codeword sequences. The size of the data block should be large enough to ensure a high coding rate. At the receiver end, first the channel detector uses an L-candidate LVA to obtain a list of L best candidates for each codeword block. Then the decoder performs the error detection checking on each of these candidates and selects the best candidate which passed the checking. If none of these candidates is a valid codeword, then the decoder can simply output the 1^{st} candidate or it can mark the codeword as an erasure depending on whether erasure information is utilized in the outer RS decoder.

The HR code design for ISI channels is given in Section 3.2 and Section 3.3

discusses boundary effects and guard bits. Performance of the system is analyzed in Section 3.4, where it is shown that the performance gain of the coding system can be approximated by the *LVA gain*. Finally, in Section 3.5, we address the issue of providing erasure information to the outer RS decoder, which provides an additional 1.3 dB improvement. Therefore, the overall performance gain of the concatenated coding system can be as high as 3.3 dB.

3.2 HR Code Design for ISI Channels

The strategy to increase the Euclidean distance between any two codeword sequences is to add check bits at the end of a data block such that the following condition is satisfied:

Any two input data sequences should have different check bit patterns if the Euclidean distance between the corresponding channel output sequences of these two data sequences is less than a predetermined value.

For example, let the predetermined value be $d_{\min}+2dB$, where $d_{\min}$ is the minimum distance of the ISI channel, then the HR code will increase the minimum distance by 2 dB.

To design a HR code, first we need to map the channel input from the set $\{-1,+1\}$ to the set $\{0,1\}$ by the mapping $\{-1\rightarrow 0, +1\rightarrow 1\}$. Then let d_0 be the predetermined value, and let $\Psi(d_0)$ denote a set of input error sequences with corresponding output error sequences having a norm less than d_0, *i.e.*,

$$\Psi(d_0)=\{\Delta\mathbf{a}' \mid \|\Delta\mathbf{x}'\| < d_0\}\ . \tag{3.1}$$

Let $F_g(c)$ denote a function that computes the remainder (module q) of the division of a polynomial $c(x)$ by a given generator polynomial $g(x) = g_0 + ... + g_s x^s$ where $g_i \in GF(q)$ and q is a prime. The code design procedure can be formally stated as follows:

Step 1. For a predetermined value d_0, find the set $\Psi(d_0)$ by a computer exhaustive search.

Step 2. Fix s, find the smallest q (or fix q, find the smallest s) such that there exists a $g(x)$ such that

$$F_g(\Delta \mathbf{a}') \neq 0, \quad \forall \Delta \mathbf{a}' \in \Psi(d_0) . \qquad (3.2)$$

Step 3 . Let H be the number of check bits such that

$$2^H \geq |R(F_g)|$$

where $R(F_g)$ denotes the range of F_g, and $|R_g(F_g)|$ denotes the size of $R(F_g)$.

Step 4. Defined a mapping from $R(F_g)$ to check bit patterns such that different elements of $R(F_g)$ correspond to different check bit patterns (one-to-one mapping).

Once F_g is determined, check bits of a data sequence $\mathbf{a}$ are determined by computing $F_g(\mathbf{a})$ and map it to a check bit pattern according to the mapping defined in Step 4. It is noted that the above code design approach satisfies the condition stated at the beginning of this section since F_g is a linear operation, *i.e.*,

$$F_g(\Delta \mathbf{a}') \neq 0 \Leftrightarrow F_g(\mathbf{a}) \neq F_g(\mathbf{a}') .$$

Example 3.1: Design a HR code for the MMEPR4 channel with the MTR code given that $d_0^2 = 92$ ($d_0/d_{\min} = \sqrt{92}/\sqrt{53} = 2.4\ dB$).

Step 1. Table 3.1 lists the set $\Psi(\sqrt{92})$ for the MMEPR4 channel with the MTR code.

Step 2. *Approach 1*: Fix q=2, the minimum order s is found to be 3, and the corresponding generator polynomials are

$$g(x) = 1 + x + x^3 \text{ or } g(x) = 1 + x^2 + x^3 .$$

Approach 2: Fix s=1, the minimum q is found to be 7, and the corresponding generator polynomials are

$$g(x) = x + 3 \text{ or } g(x) = x + 5 .$$

Step 3. Both approaches in Step 2 require that $H \geq 3$.

Step 4. There is no reason to prefer a particular mapping, so we can use any mapping scheme that maps $R(F_g)$ to the check bit patterns.

Although both approaches at Step 2 require at least 3 check bits, Approach 2 is more efficient since only 7 check bit patterns are required; this advantage can be utilized to enforce the MTR constraint. The disadvantage of Approach 2 is that its implementation complexity could be higher than Approach 1 except when $g(x) = (x - 1)$, which leads to $F_g(\mathbf{a}) = (\Sigma\, a_i)_{mod\,q}$.

TABLE 3.1

$\Psi(\sqrt{92})$ FOR THE MMEPR4 CHANNEL WITH THE MTR CODE

$\Delta\mathbf{a}'$	$\|\|\Delta\mathbf{a}' * \mathbf{h}\|\|^2$
[1 0 0 0 0 0 0]	53
[1 -1 0 0 0 0 0]	58
[1 0 1 0 0 0 0]	90
[1 0 0 1 0 0 0]	72
[1 0 0 0 1 0 0]	80
[1 0 1 0 0 1 0]	89
[1 0 0 1 0 1 0]	89
[1 0 0 1 0 0 1]	85

So far, the HR code is used after the precoder. If the HR code is used before the precoder, at Step 1, we need to transform each $\Delta\mathbf{a}'$ in the set $\Psi(d_0)$ to the $\Delta\mathbf{b}'$ defined by

$$\Delta\mathbf{b}' = \mathbf{b}' \oplus \mathbf{b} \tag{3.3}$$

where $\mathbf{b}'$ and $\mathbf{b}$ are the corresponding sequences of $\mathbf{a}'$and $\mathbf{a}$ before the precoder. The set $\Psi(d_0)$ is then formed by these $\Delta\mathbf{b}'$s accordingly. Note that the mapping $\Delta\mathbf{a}' \rightarrow \Delta\mathbf{b}'$ is well-defined, *i.e.*, there is only one $\Delta\mathbf{b}'$ that corresponds to each $\Delta\mathbf{a}'$. Table 3.2 shows the set $\Psi(\sqrt{92})$ comprising $\Delta\mathbf{b}'$s for the MEEPR4 channel with the MTR code and a $1/(1 \oplus D)$ precoder. The rest of the code design steps are the same except that only binary generator polynomials can be used at Step 2 (q=2). For the MEEPR4 channel given in Example 3.1, the minimum order of g(x) is found to be s=3 and the corresponding generator polynomials are

$$g(x) = 1 + x + x^3 \text{ or } g(x) = 1 + x^2 + x^3$$

TABLE 3.2

$\Psi(\sqrt{92})$ OF $\Delta\mathbf{b}'$S FOR THE MEEPR4 CHANNEL WITH THE MTR CODE

$\Delta\mathbf{a}'$	$\Delta\mathbf{b}'$	$d^2 = \|\|\Delta\mathbf{a} * \mathbf{h}\|\|^2$
[1 0 0 0 0 0 0]	[1 1 0 0 0 0 0 0]	53
[1 -1 0 0 0 0 0]	[1 0 1 0 0 0 0 0]	58
[1 0 1 0 0 0 0]	[1 1 1 1 0 0 0 0]	90
[1 0 0 1 0 0 0]	[1 1 0 1 1 0 0 0]	72
[1 0 0 0 1 0 0]	[1 1 0 0 1 1 0 0]	80
[1 0 1 0 0 1 0]	[1 1 1 1 0 1 1 0]	89
[1 0 0 1 0 1 0]	[1 1 0 1 1 1 1 0]	89
[1 0 0 1 0 0 1]	[1 1 0 1 1 0 1 1]	85

Finally, it is noted that if we choose a d_0 such that $d_0/d_{\min} < 3dB$, then each of the error sequences in the set $\Psi(d_0)$ corresponds to a path that diverges and merges with the correct path only once. Therefore the size of the set $\Psi(d_0)$ will not be too large. Whereas, if we choose a d_0 such that $d_0/d_{\min} \geq 3dB$, then $\Psi(d_0)$ will includes a large number of error sequences since there are a large number (provided that the codeword length is large) of paths that diverge and merge with the correct sequence twice with distances of only 3 dB more than $d_{\min}$. This kind of $\Psi(d_0)$ will need a large number of check bits to check each of the error sequences in the set $\Psi(d_0)$. Therefore, it is not easy to design a HR code with more than 3 dB increase in the minimum distance.

3.3 Boundary Effects and Guard Bits

The boundary between adjacent codewords and the boundary between the data bits and check bits need careful examination since the channel is an ISI channel. Guard bits are used to prevent the performance from being degraded by boundary effects. In the following, we assume the channel is a MEEPR4 channel with the MTR code.

3.3.1. Boundary Between Data Bits and Check Bits

If there is an incorrect path crossing the boundary of data bits and check bits with a smaller path metric (than the correct path), this incorrect path will have errors in the data bits as well as in the check bits. Therefore, even if the corresponding error event is one of the sequences in the set $\Psi(d_0)$, it is possible that the HR code may fail to detect such an error event since the check bits are contaminated by errors. Guard bits need to be inserted between data bits and check bits to reduce the probability of a cross-boundary error event. For example, we can form a codeword with the following format

$$a_1 \ldots a_{N-1}\, a_N\, g_1\, c_1\, c_2 \ldots c_H$$

where $a_i(i = 1,...,N)$ are data bits, $c_i(i = 1,...,H)$ are check bits and the guard bit $g_1 = a_N$. By inserting g_1, the first three error events in Table 3.1 will be prevented if they cross the boundary of data bits and check bits. The other advantage of inserting the guard bit g_1 is that it enforces the MTR constraint at the boundary of data bits and check bits.

3.3.2 Boundary Between Adjacent Codewords

Boundary effects between data bits and check bits usually are less harmful since we can add more check bits to design a powerful HR code such that the probability of failing to detect an error is very small. The boundary effects between adjacent codewords could be more harmful.

The difficulty here is how to select a termination channel state for a codeword block so that the L best candidates ending at the termination channel state can be outputted as the L best candidates for the codeword block. Strictly speaking, the ideal L best candidates for a codeword block can only be obtained if we terminate each codeword block to a known channel state such that the L best candidates ending at the known channel state can be outputted as the L best candidates for the codeword block. To terminate a codeword block to a known channel state, it needs 4 termination bits for an MEEPR4 channel, this overhead is obviously too large. On the other extreme, if no guard bit is used between adjacent codewords, the termination channel state is selected by looking ahead on the path and then select a termination channel state; we will choose an incorrect termination channel state if there is a cross-boundary incorrect path with smaller path metric; the probability that the correct sequence is not in the LVA list will be increased compared to the case when we select a correct termination channel state. A solution is to insert some guard bits (not necessary to terminate the path to a known state) such that the probability that a cross-boundary incorrect path has a smaller path metric is greatly reduced. For example, for the MEEPR4 channel, we form a codeword with the following format

$$a_1 \ldots a_{N-1}\, a_N\, g_1\, c_1\, c_2 \ldots c_H\, g_2$$

where $g_2 = c_H$. The guard bit g_2 will reduced the probability of a cross-boundary error event while enforcing the MTR constraints.

3.4 Performance of the HR Code

For simplicity, we will ignore the boundary effects. Therefore, we only need to consider the following two cases:

Case 1. An incorrect codeword appears before the correct codeword in the LVA candidate list. This will cause the decoder to make an error.

Case 2. None of the candidates in the LVA list is a valid codeword sequence. Although it is impossible for the decoder to output a correct decision, the decoder can mark the sequence as an erasure since the decoder knows there are some errors in the sequence. If the erasure information cannot be utilized, the decoder can simply output the most likely sequence since there is no better choice.

By adding additional check bits, we can design a HR code powerful enough to make Case 1 an unimportant error event. Therefore, we will focus on Case 2 in the following. As an example, we consider a 3-candidate LVA for the MEEPR4 channel

(The best tradeoff between complexity and performance is achieved by using a candidate list with a size of three to four). Fig. 3.2 shows all types of admissible path topologies if Case 2 happens, *i.e.*, there are at least three candidate sequences that are more likely than the correct sequence.

In Fig. 3.2, the Type 1 error event refers to the case when each candidate diverges and merges with the correct path only once. For Type 2-5 error events, there is at least one candidate who diverges and merges with the correct path twice.

Let p_e^{LVA} be the probability that a Case 2 error event starts from a given bit. If Type 1 is the dominant error event, then referring to Table 2.4, the probability can be approximated by

$$p_e^{LVA} \approx Q(\sqrt{SNR_{eff}^{LVA} G_{\min}}) \tag{3.4}$$

where $G_{\min}$ is given in (2.29).

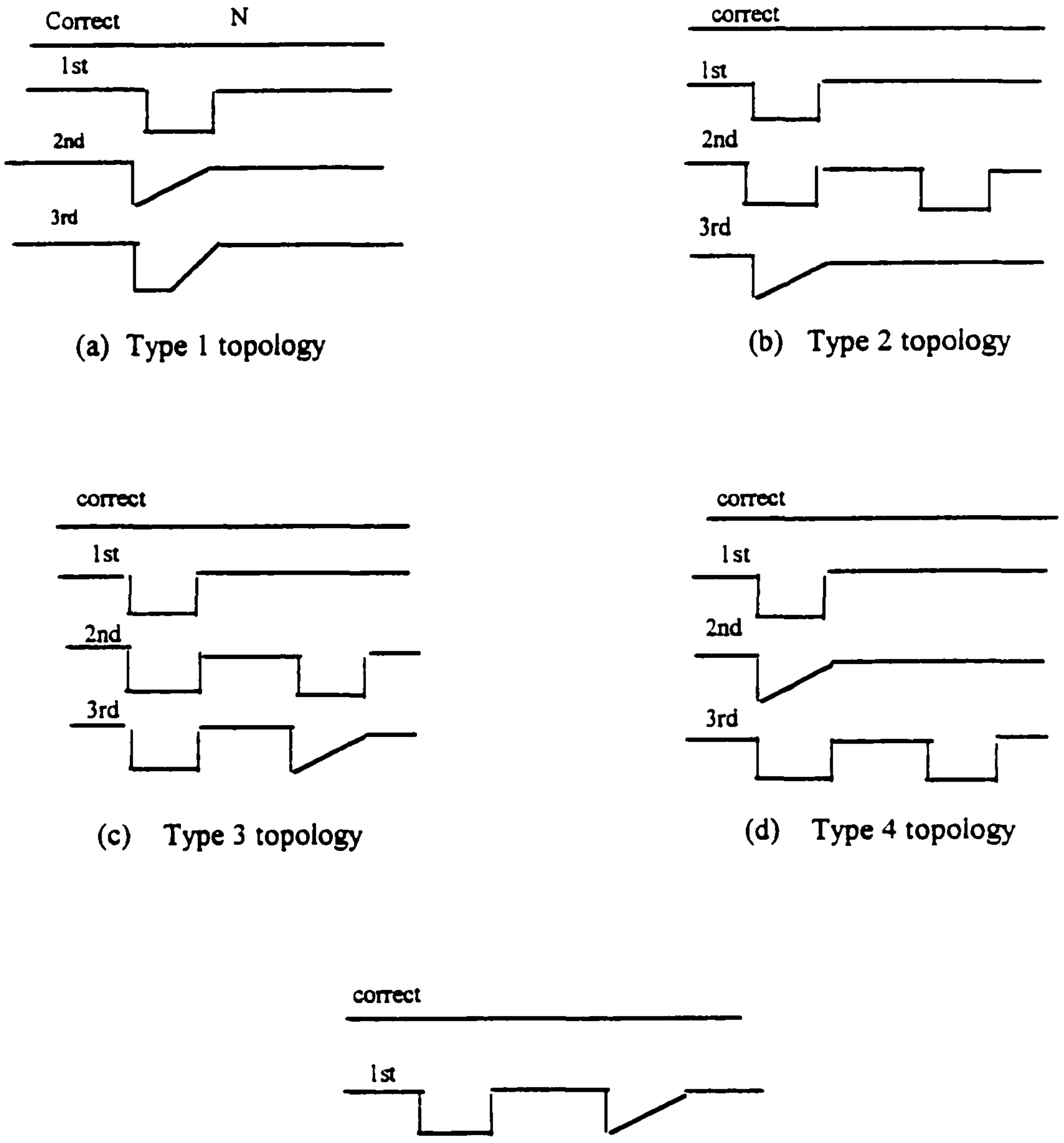
If the dominant error event is Type 2, then the error event probability can be bound by

$$p_e^{LVA} \leq NQ(\sqrt{SNR_{eff}^{LVA} G_{bound}}) \tag{3.5}$$

where N is the length of the HR codeword, and G_{boun} is the lower bound on the error event gain as defined in (2.9). Following the same procedure as in Subsection 2.3.2, it can be found that for each of the Type 2 error events

$$G_{bound}(Type\ 2) \geq 3.2\ dB.$$

Comparing (3.5) to (3.4), we note that (3.4) is larger than (3.5) for small values of N, while the opposite holds for large values of N. Therefore, Type 2 error events will eventually dominate as N increases. However, in order to increase the code rate without



(a) Type 1 topology

(b) Type 2 topology

(c) Type 3 topology

(d) Type 4 topology

(e) Type 5 topology

Fig. 3.2. Admissible path topologies when the correct sequence is not in the LVA list.

performance degradation, we would like to choose the largest value of N and still have Type 1 as the dominant type. For $p_e^{LVA} = 10^{-8}$, $G_{bound} = 3.2\ dB$ and $G_{\min} = 2\ dB$, it can be found with the use of Fig. 3.3, that the largest value of N is about 200.

Similarly, for Type 3 and Type 4 error events, the corresponding lower bounds for the error event gains are 3.4 dB and 3.2 dB, respectively. Therefore they can be ignored as long as N<200.

For a Type 5 error event, we have

$$p_e^{LVA} \le NQ^2(\sqrt{SNR_{eff}^{LVA}}) \ . \tag{3.6}$$

It is not difficult to verify that this type of error events can also be ignored when compared to Type 1 error events.

In summary, for a target error rate of 10^{-8}, if the codeword length N <200, Type 1 error events will be the dominant error events for the coding system and the error rate is approximated by (3.4). Define the erasure probability (P_{era}) as the probability that the correct sequence is not in the LVA candidate list for a codeword length of N, then this probability can be approximated as

$$P_{era} = Np_e^{LVA} . \tag{3.7}$$

3.5 Concatenated System

As shown in Fig. 1.2, RS codes are used as outer codes in data storage systems. It is well known that the RS code can correct twice as many erasures as errors [16]. An erasure is referred to as an errata for which location information is known, but its value is unknown. In the case when the HR decoder finds that all the candidates provided by the LVA channel detector are invalid codewords, it can mark this section of

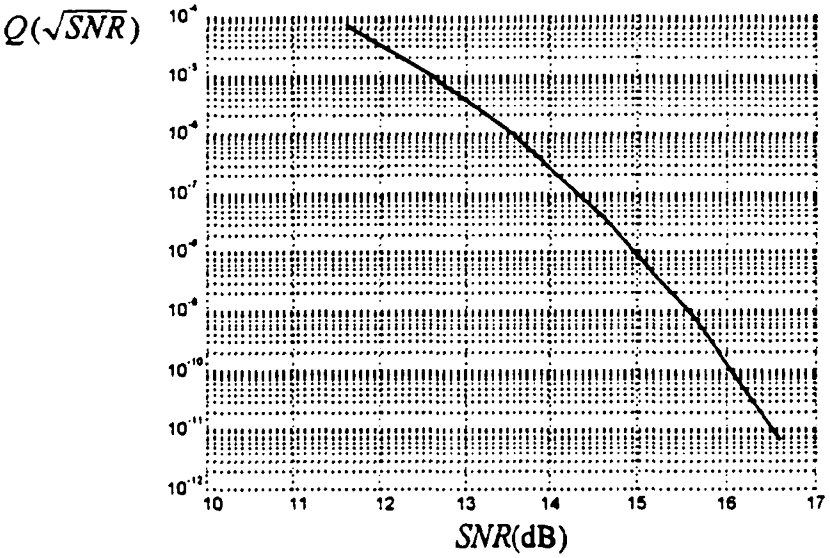


Fig. 3.3. Q function versus SNR.

data as an erasure and submit this information to the following RS decoder. Before analyzing the performance of this concatenated system, let us give a brief introduction to RS codes.

RS codes are a class of nonbinary BCH codes. Its symbols are from the Galois field GF(q), where q is a power of a prime number ($q = 2^8$ for most of the RS codes used in storage systems). A standard t-error-correcting RS code with symbols from GF(2^8) has the following parameters:

Codeword length $n=q-1=255$

Number of data symbols (bytes) $k=255-2t$

Number of parity check bytes $n-k=2t$

Minimum distance $d_{\min} = n - k+1$.

Storage systems use a shortened RS code to fit the data format used in these systems. A shortened RS code is derived simply by shortening the number of data symbols. The resultant RS code has at least the same error-correcting capability as the code from which it is derived.

Let α be a primitive element in GF(2^8), the generator polynomial of the RS code is

$$g(x) = (x+\alpha)(x+\alpha^2)\ldots(x+\alpha^{2t}) .$$

In a systematic form, the $2t$ parity check bytes are the coefficients of the remainder of the division of the data polynomial $a(x) = a_0 + a_1x\ldots + a_{k-1}x^{k-1}$ by the generator polynomial $g(x)$. The t-error- correcting RS code can correct $2t$ erasures. The following procedure describes a well-known algorithm [32] for erasure and error correction:

<u>Step 1</u>. Compute $2t$ raw syndromes S_i(i=1,2,...,$2t$). S_i is the remainder of the division of the received polynomial by $(x+\alpha^i)$.

<u>Step 2</u>. Generate an erasure-locator polynomial from the known erasure locations

$$\Gamma(X) = \prod_{i=1}^{l}(x+\alpha^{P_i})$$

where l is the number of available erasure pointers and P_i is the location specified by the i^{th} erasure pointer.

<u>Step 3</u>. Generate $(2t\text{-}l)$ modified syndromes T_i from the $2t$ raw syndromes S_i and the coefficients of the erasure-locator polynomial Γ_j $(j = 0, 1, \ldots, l)$

$$T_{i-l} = \sum_{j=0}^{l} \Gamma_j S_{i-j} \qquad\qquad i=l+1,\ldots,2t.$$

Step 4. Generate the coefficient of the error locator $\sigma(x)$ from the modified syndromes T_i.

Step 5. Find the error location using the error locator polynomial.

Step 6. Compute error values using the raw syndromes, the erasure pointers and error locations.

It is noted that a false erasure pointer will not cause a miscorrection (the error value for a false erasure will be zero), but each false erasure pointer decreases the remaining correction capability since it decreases by one the number of modified syndromes at Step 3.

The implementations of one-byte error correcting decoder and multiple-byte error correcting decoder are described in [33],[34]. A multiple-byte error correcting decoder is much more complicated than an one-byte error correcting decoder. Therefore, in the following, we will only consider RS codes that correct only one error (but may correct several erasures).

Let us consider an example to see how much performance improvement we can expect from utilizing the erasure information provided by the HR code. Fig. 3.4 shows the data format of a data storage system. The interleaver receives input user data row wise (byte by byte); the RS encoder adds three RS check bytes to each row to form an RS codeword; then the interleaver outputs the data column wise (byte by byte); after the rate 8/9 RLL encoder, the HR encoder adds H check bits to each column to form a HR codeword. At the receiver end the inverse operations are carried out. The interleaver size and the HR codeword length should satisfy

$$9I \geq N - H \tag{3.8}$$

so that each erasure in a HR codeword only causes an erasure for each of the RS codewords.

The RS code has three check bytes, thus it can correct one error or three erasures. Since a powerful HR code is assumed, the error probability is dominated by the case when there are four erasures in a RS code. Thus, the probability that the interleaved data block is in error will be

$$P_{e1} \approx \binom{n}{4} P_{era}^{4} = \binom{n}{4} N^{4} (p_{e1}^{LVA}) \, . \tag{3.9}$$

To evaluate the gain obtained by utilizing the erasure information, we need to consider a system without erasure information and with the same data format shown in Fig. 3.4. In case the erasure information is not utilized, the HR decoder simply outputs the 1st candidate, if none of the candidates is a valid codeword. Since the RS decoder can correct only one error, the error probability will be

$$P_{e2} \approx I \binom{n}{2} (C \times p_{e2}^{LVA})^{2} \, , \tag{3.10}$$

where Cp_{e2}^{LVA} is the probability that a byte is in error and C is a constant slightly larger than eight since the average length of error events is larger than one.

Equating (3.10) and (3.9), we have

$$p_{e1}^{LVA} \approx \{ 12IC^{2} (p_{e2}^{LVA})^{2} / (n^{2}) \}^{1/4} / N \cdot \tag{3.11}$$

As an example, let $p_{e2}^{LVA} = 10^{-8}$, I=12, C=12, N=120 (H=12), n=200, we have

$$p_{e1}^{LVA} = 7 \times 10^{-7} .$$

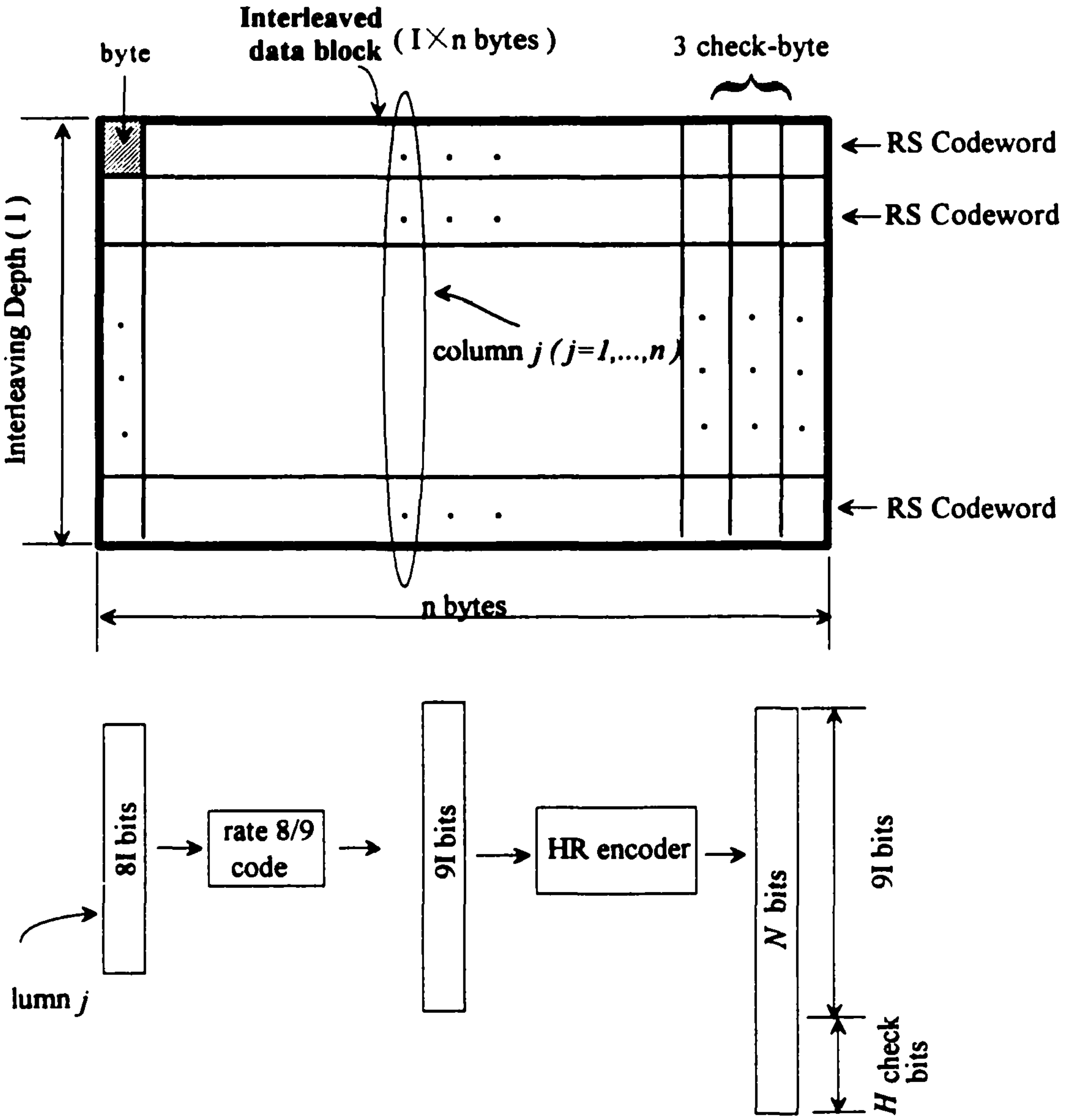


Fig. 3.4. Formats of the interleaved data block and HR codewords.

Comparing p_{e1}^{LVA} with $p_{e2}^{LVA} = 10^{-8}$, the corresponding SNR improvement is about 1.3 dB.

In summary, by providing the erasure information, the concatenated system can provide another 1.3 dB performance gain. Since the 3-candidate LVA provides a 2 dB gain, the total gain can be as high as 3.3 dB. Although, the code rate penalty may reduce some of the coding gain, the total net performance gain is still significant since the code rate of the HR code is very high. One of the problems associated with this coding system is that the interleaving size should be large enough so that (3.8) is satisfied. In order to satisfy this condition, the size of an interleaved data block may exceed the size of a disk data sector, this may require a change in current data formats.

Chapter 4
Complement States Grouping Technique

4.1 Reduced-State Sequence Estimation Overview

In decoding an ISI channel, maximum likelihood sequence estimation, implemented with the well-known Viterbi algorithm, has a significant performance gain compared to other detection techniques. However, the implementation complexity of the VA is generally much larger than other detection techniques. Increasing the complexity of the Viterbi decoder not only increases the circuit power consumption, but also complicates the circuit routing and makes it difficult to build a high speed circuit. Therefore, it is desirable to reduce the implementation complexity of the VA at the expense of a small performance loss. It is known that the complexity of the VA is directly related to the number of states which is given by

$$M^K \tag{4.1}$$

where M is the size of the channel input signal set and K is the length of the overall channel impulse response or the channel memory. Early work in this area includes [35], [36], where a feedback mechanism is used to truncate the channel impulse response. A more general treatment is given in [27], where the primary idea of the reduced-state sequence estimation (RSSE) technique is to form a reduced-state (RS) trellis by combining a certain number of states in the maximum likelihood (ML) trellis. The combination is based on the Ungerboeck set partitioning of the channel input signal set [37]. More specifically, a state vector at time k in the RS trellis is described by:

$$[C_{k-K}, \ldots, C_{k-2}, C_{k-1}] \tag{4.2}$$

where C_{k-j} denotes the index of the subsets in the $(K-j+1)^{th}$ level set partitioning $\Omega(K-j+1)$. The signal set partitioning must satisfy the condition that the $(i+1)^{th}$ level partitioning $\Omega(i+1)$ is either a further partition of $\Omega(i)$ or $\Omega(i+1)=\Omega(i)$. This condition guarantees a well-defined RS trellis [27]. The criterion used in the set partitioning is to maximize the minimum intra-subset Euclidean distance. The total number of states in the RS trellis is determined to be

$$J_1 J_2 \ldots J_K \tag{4.3}$$

where J_i is the number of subsets in $\Omega(i)$ and $1 \le J_1 \le J_2 \le \ldots J_k \le M$.

This traditional RSSE technique provides a good tradeoff between complexity and performance for many ISI channels especially channels with multi-level channel inputs. However, there are also a lot of channels where this conventional RSSE technique fails to provide a satisfactory solution with reduced complexity and a reasonable performance loss. For example, for channels with binary inputs, the RSSE technique becomes a state truncation technique and the state vector is defined by :

$$[a_{k-K'}, \ldots, a_{k-2}, a_{k-1}] \tag{4.4}$$

where $K' < K$, and a_{k-i} is the channel input taking values from {-1, +1}. For the EPR4 channel, it can be verified that the truncation of state vectors will result in at least 1.2 dB of performance loss, which is obviously unacceptable.

A new reduced-state technique, called *complement states grouping technique* (CSGT), is developed in this dissertation. The major advantage of the CSGT is that it ensures a negligible performance loss while reducing the number of states by a factor of about two. It is particularly useful for severe ISI channels with binary inputs where the traditional RSSE technique usually fails to provide a good solution.

In Subsection 4.2.1, we describe the CSGT for general ISI channels. In Subsections 4.2.2 and 4.2.3, the CSGT is applied to the EPR4 and MEEPR4 channels. In addition, in Section 4.3, the CSGT is combined with the Fettweis branch metric shifting method [38] to further reduce the detector implementation complexity. Finally, in Section 4.4, the LVA performance in a CSGT RS trellis is studied.

4.2 Complement States Grouping Technique

4.2.1 Introduction

Reduced-state techniques are basically about grouping ML states into superstates to form an RS trellis in such a way that it satisfies the following two conditions:

I. Performance loss is acceptable.

II. The resultant RS trellis should be (almost) well-defined.

Condition *II* will be discussed in detail in Subsection 4.2.4. For Condition *I*, if only **negligible loss** can be accepted, then an equivalent and more specific way to address condition *I* is as follows:

The "difference" between ML states grouped together to form a superstate in the RS trellis should be large enough such that mistaking one of the ML states for any other ML state in the same superstate should be at least as unlikely as making an error in the ML trellis.

To define the "difference" between state i and state j in a quantitative way, a new concept called *state distance* denoted as $D_{i,j}$ is introduced:

$D_{i,j}$ is the minimum Euclidean distance in the ML trellis between any two paths that diverge from a certain state and terminate at state i and state j at the same time , respectively.

Since the error probability in the *ML* trellis is determined by the minimum Euclidean distance of the ML trellis, Condition *I* is equivalent to the following criterion:

The state-distance between any pair of the ML states in the same superstate must be no less than the minimum Euclidean distance ($d_{\min}$) in the ML trellis.

Although the CSGT can be applied to channels with multilevel channel inputs, the channel input will be assumed to be binary from now on, unless specified otherwise. In the case of an ISI channel with binary inputs from the set {-1, +1}, the heuristic way of state grouping is to group complement states $[\bar{a}_{k-K}, \ldots, \bar{a}_{k-2,}\bar{a}_{k-1}]$ and $[a_{k-K}, \ldots, a_{k-1}, a_{k-1}]$ together where $\bar{a}_i = -a_i$. One advantage of this complement states grouping is that complement states are more likely to have a large state distance. Another advantage is that complement states grouping generates a well-defined RS trellis, if all pairs of complement states can be grouped together.

The complement states grouping technique is now formally stated as follows:

Step 1. Determine state distances for all pairs of complement states by a computer exhaustive search;

Step 2. Group pairs of complement states whose state distance is no less than $d_{\min}$; keep pairs of complement states unpaired if their state distance is less than $d_{\min}$;

Step 3. Form the RS trellis accordingly.

The RS trellis will not be a well-defined trellis if there exists at least one pair of complement states with a state distance less than $d_{\min}$. Fortunately, most of the complement states will have a state distances no less than $d_{\min}$. In most cases, the RS trellis will be at least an almost well-defined trellis as we will see in Subsections 4.2.2 and 4.2.3.

4.2.2 EPR4 Channel

The EPR4 channel has a channel response polynomial of $1+D-D^2-D^3$, therefore the ML trellis has eight states as shown in Fig. 4.1, where each branch is labeled in a (channel output/channel input) manner.

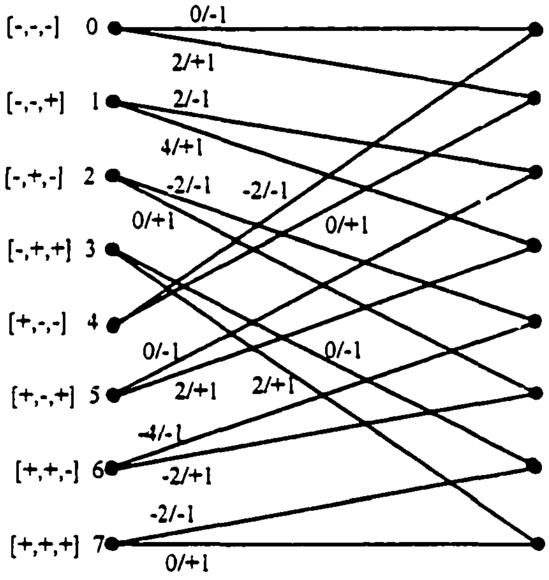


Fig. 4.1. ML trellis for the EPR4 channel.

The eight ML states for the EPR4 channel are listed as follows:

[-1,-1,-1] [-1, -1,+1] [-1,+1,-1] [-1,+1,+1]

[+1,-1,-1] [+1,-1,+1] [+1,+1,-1] [+1,+1,+1] .

They are numbered respectively as 0,1,2,...,7. The first step for the CSGT is to find the state distance for each pair of complement states. A computer exhaustive search is carried out to find complement state distances. Assuming the channel is AWGN, the results are as follows:

$$D_{0,7} = 4\ ,\ D_{1,6} = 4\ ,\ D_{3,4} = \sqrt{24}\ ,\ D_{2,5} = \sqrt{8}\ .$$

Since $d_{\min} = \sqrt{16}$ in the ML trellis, we can combine the following complement state pairs

(state 0, state 7), (state 1, state 6), (state 3, state 4).

But no grouping action should be taken for (state 2, state 5). The resultant RS trellis is shown in Fig. 4.2. It is noted that the branch from superstate 'c' to superstate 'd' is an impossible transition (denoted by '#') if the corresponding ML state is [+ + -] at superstate 'c'. Similarly the branch from superstate 'c' to superstate 'e' is impossible if the

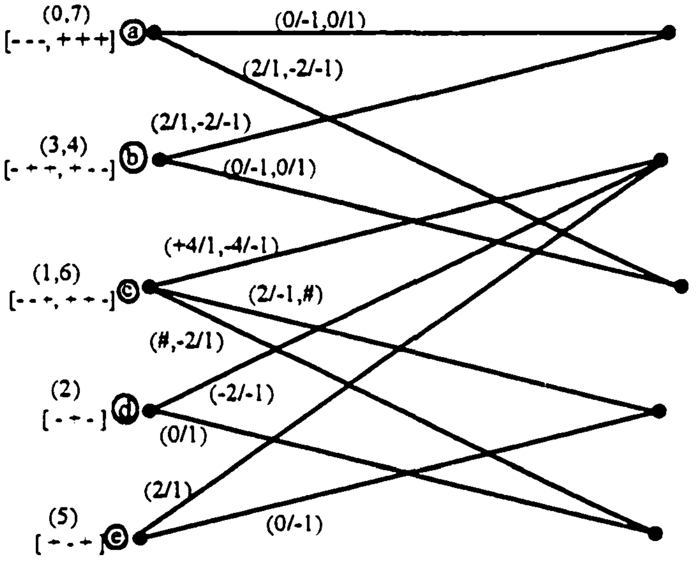


Fig. 4.2. RS trellis for the EPR4 channel.

corresponding ML state is [- - +] at superstate 'c'. Although the RS trellis in Fig. 4.2 cannot be regarded as a well-defined trellis, this trellis can be regarded as an almost well-defined trellis.

In the RS trellis, some of the branches have a pair of values, for example, the branch from superstate 'a' to superstate 'c' is labeled (2/1,-2/-1). To resolve this branch value ambiguity, a feedback from the survivor path of each superstate is used. For example, when a_{k-3} is -1 in the survivor path of superstate 'a', the corresponding ML state for 'a' is determined to be [-1,-1,-1], accordingly, 2/1 is selected for the above branch and the next stage ML state for superstate 'c' is updated as [-1,-1,+1]. Similarly, when $a_{k-3} = +1$ in the survivor path of superstate 'a', the corresponding ML state is determined to be [+1,+1,+1], accordingly, -2/-1 is selected for the branch and the next stage ML state for superstate 'c' is updated as [+1,+1,-1].

The complexity of the trellis can be further reduced by using an additional threshold detector. The resultant trellis is shown in Fig. 4.3. More specifically, referring to Fig 4.2, when the channel output y_k is greater than zero, the branch 'c' to 'e' and the branch 'd' to 'b' are eliminated. Similarly, when y_k is less than zero, then branch 'c' to 'd' and branch 'e' to 'b' are eliminated. While reducing the complexity of the implementation, this technique causes a negligible performance loss for the EPR4 channel as shown in [39] and confirmed by simulations shown below. The heuristic explanation is that when the branch value has an opposite sign from the noisy channel output, this branch will be an unlikely branch and may be eliminated without performance loss.

It is noted that channel encoders may have an impact on both the ML trellis and the

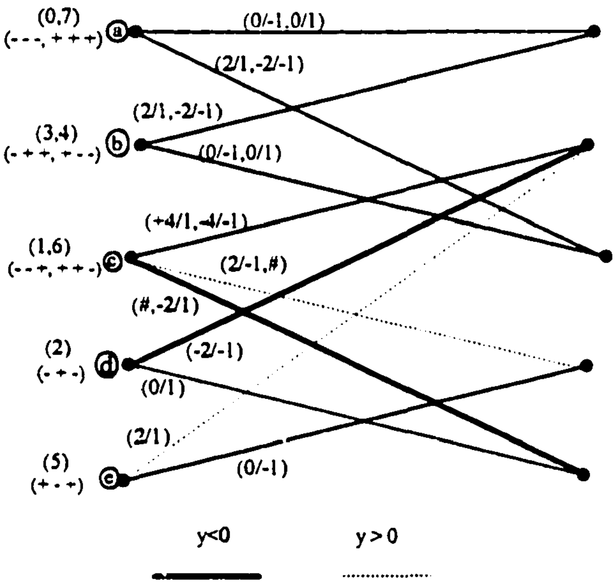


Fig. 4.3. Simplified RS trellis.

RS trellis , for example, the (1,7) RLL code will eliminate the channel state [-1,+1,-1] as well as [+1,-1,+1], whereas, the rate 8/9 RLL code impacts neither the ML trellis nor the RS trellis. Fig. 4.4 shows the RS trellis for EPR4 channels coded with the (1,7) code.

Fig. 4.5 is the implementation block diagram for the trellis shown in Fig. 4.3. The detector includes two radix-2 ACS units and a selection unit. Each radix-2 unit is a combination of two 2-way ACS units. The radix-2 ACS unit is a classical implementation

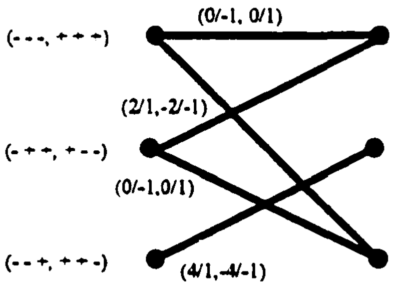


Fig. 4.4. Reduced-state trellis for the EPR4 channel with the (1,7) code.

unit for the Viterbi algorithm as described in [40]. Radix-2 units in Fig. 4.5 are similar to the conventional radix-2 unit, except that the survivor path of each superstate is used as a feedback to select one of the branch values in the aforementioned manner. The selection unit in Fig. 4.5 simplifies the implementation complexity using the sign of y_k.

Finally, computer simulations are carried out to evaluate the performance of the detector in Fig. 4.5. The simulated channel is a Lorentzian channel with user density of 2.5, and an 8/9 RLL code is also incorporated. The discrete channel is obtained by equalizing the Lorentzian channel to the EPR4 channel using a three-pole analog filter

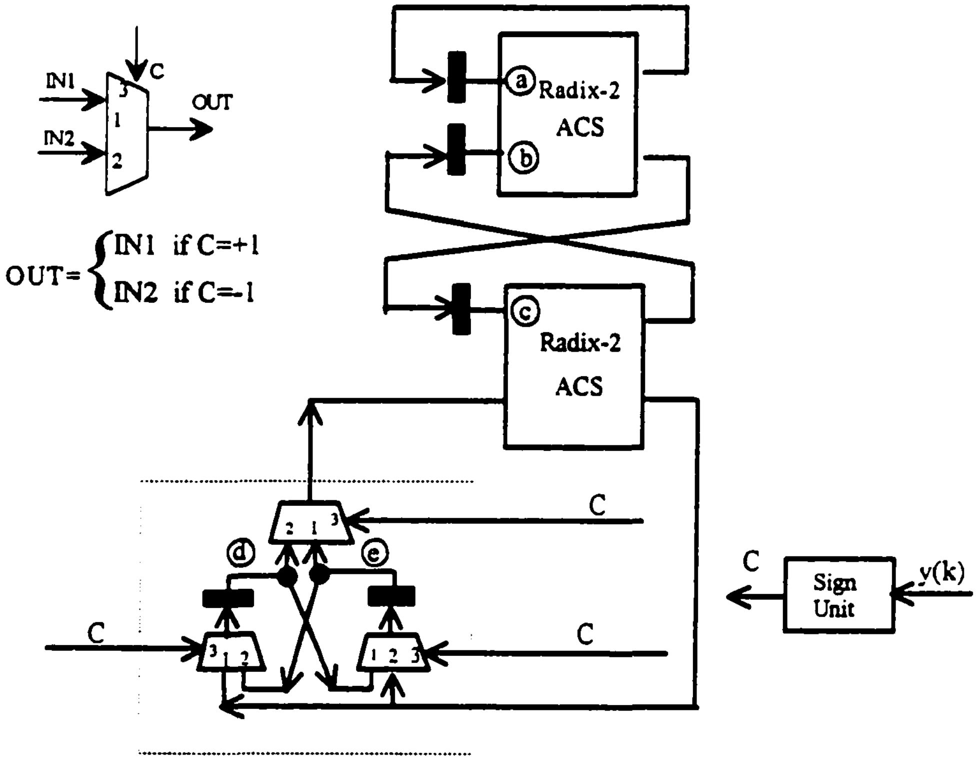


Fig. 4.5. Implementation of the RS trellis shown in Fig. 4.3.

and a 7-tap digital equalizer, AWGN is added to the Lorentzian channel. The performance is evaluated as the required channel SNR for achieving an error rate of 10^{-5} and the results are shown in Fig. 4.6. The required channel SNR for the reduced-state VA (RSVA) is 21.7 dB while the required channel SNR for a standard VA is 21.6 dB, the performance loss being indeed negligible.

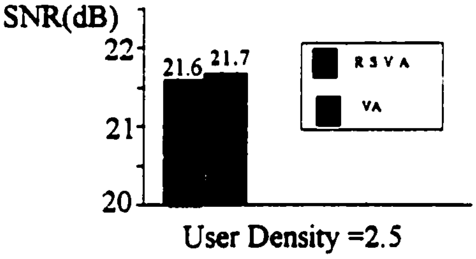


Fig. 4.6. Required channel SNR for RSVA and VA.

Table 4.1 compares the histograms of error events for the RSVA and the standard VA, at error rates of 10^{-5} and 10^{-4}, respectively. The comparison shows that the CSGT causes negligible extra error propagation, in spite of the fact that a feedback mechanism is used.

TABLE 4.1
HISTOGRAM OF ERROR EVENTS FOR VA AND RSVA

Error Rate	Length of Error Events	1	2	3	4	5	6	7	8	9	10	11	12	13
10^{-5}	VA	12	2	99	1	0	0	0	0	0	0	0	0	0
	RSVA	13	1	70	3	2	1	0	0	0	0	0	0	0
10^{-4}	VA	235	15	719	33	15	10	0	4	0	0	0	0	1
	RSVA	185	10	582	35	14	11	2	0	0	0	0	0	0

4.2.3 MEEPR4 Channel

The MEEPR4 channel considered has a channel response

$$h(D) = 5 + 4D - 3D^2 - 4D^3 - 2D^4 .$$

An MTR code can be used in the MEEPR4 channel to provide coding gain with a high code rate by preventing consecutive triple transitions in the input sequence. The corresponding ML trellis is shown in Fig. 4.7, where each branch is labeled with the ideal channel output associated with the transition.

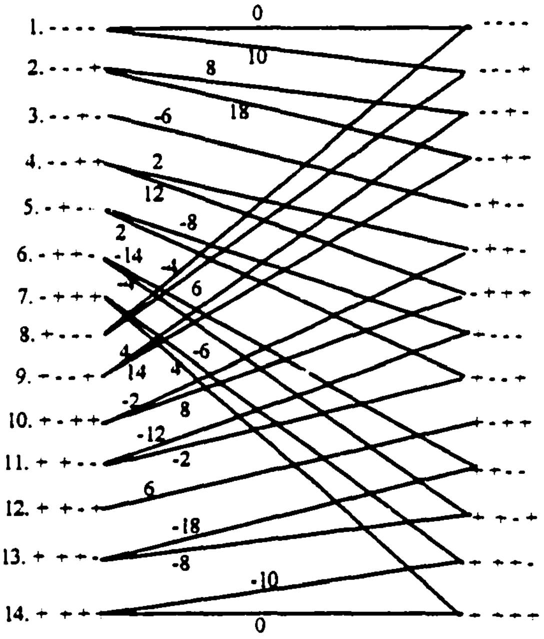


Fig. 4.7. ML trellis for the MEEPR4 channel.

To apply the CSGT to the ML trellis, first complement state distances are computed. In this case, instead of assuming an AWGN channel, we use a correlated noise with correlation coefficient function given by (1.20). It is noted that the correlation function depends on the channel density, however, a slight change of the channel density should not affect the correlation function significantly, therefore, (1.20) can still be used even if the RLL code is a rate 16/17 code.

Complement state distances which take into account the noise correlation are found by a computer exhaustive search to be

$$D^2_{1,14} = 382,\ D^2_{2,13} = 516,\ D^2_{3,12} = 390$$

$$D^2_{4,11} = 961,\ D^2_{5,10} = 356,\ D^2_{6,9} = 389,\ D^2_{7,8} = 551\ .$$

Obviously, all the complement state distances are larger than $d^2_{\min} = 4 \times 53$, therefore, all the complement state pairs can be grouped together and the resultant RS trellis is shown in Fig. 4.8. Compared to the ML trellis , the complexity of the RS trellis is reduced by a factor of two.

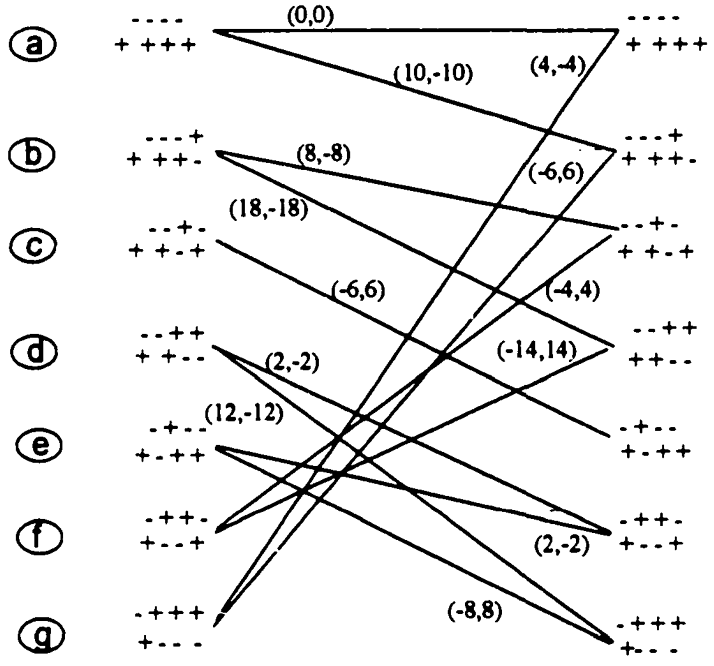


Fig. 4.8. RS trellis for the MEEPR4 channel.

4.2.4 RS Trellis Structure

In a well-defined RS trellis, each branch has a valid branch value associated with each ML state of the superstate where the branch originates. In other words, each transition in the RS trellis is a possible transition for any ML state of the superstate where the transition originates. The results in Subsections 4.2.3 and 4.2.2 show that the CSGT RS trellis for the MEEPR4 channel is a well-defined trellis, while the RS trellis for the EPR4 channel is an almost well-defined trellis.

If all complement state pairs can be grouped, it is not difficult to verify that the CSGT always leads to a well-defined RS trellis. It is also noted that the classical RSSE technique also leads to a well-defined RS trellis. The question is whether there is any other state grouping technique that will lead to a well-defined trellis.

<u>Lemma 4.1</u> : For a pairwise state grouping scheme, let $s_0' = [a_K, a_{K-1}, \ldots, a_1]$ denote the state grouped together with state $s_0 = [-1 - 1, \ldots, -1]$. If the RS trellis is well-defined, then either $s_0' = [+1, -1, -1, \ldots, -1]$ or $s_0' = [+1, +1, \ldots, +1]$.

<u>Proof</u> : In the ML trellis, the two destinations of the transitions from state $s_0 = [-1, -1, \ldots, -1]$ are state $s_0 = [-1, -1, \ldots, -1]$ and state $s_1 = [-1, \ldots, -1, +1]$; the two destinations of the transitions from state s_0' are state $s_p = [a_{K-1}, \ldots, a_1, +1]$ and state $s_q = [a_{K-1}, \ldots, a_1, -1]$. Since the RS trellis is well-defined then either state s_p or state s_q belongs to the same superstate as state s_0.

Case 1. s_p belongs to the same superstate as s_0.

$$s_p \neq s_0 \Rightarrow s_p = s_0' \Rightarrow$$

$$a_K = a_{K-1}, a_{K-1} = a_{K-2}, \ldots, a_2 = a_1, a_1 = +1 \Rightarrow s_0' = s_p = [+1, +1, \ldots, +1].$$

Case 2. s_q belongs to the same superstate as s_0.

Case 2.1 . $s_q = s_0$

$$s_0' \neq s_0 \Rightarrow a_K = +1 \text{ and } a_{K-1} = a_{K-2}, \ldots, a_1 = -1$$

$$\Rightarrow s_0' = [+1, -1, \ldots, -1].$$

Case 2.2 $s_q = s_0'$

$$a_K = a_{K-1}, a_{K-1} = a_{K-2}, \ldots, a_2 = a_1, a_1 = -1$$

$$\Rightarrow s_0' = s_0 = [-1, -1.., -1].$$

This is a contradiction.

QED

The situation for state $[+1, +1, \ldots, +1]$ is similar to state [-1,-1,...,-1].

Theorem 4.1 : For channels with memory $K \leq 3$, the complement states grouping and the classical RSSE are the only two ways that lead to well-defined RS trellises.

Proof : Using Lemma 4.1 and listing all the possibilities, it is easy to prove Theorem 4.1.

Theorem 4.1 no longer holds for channels with $K \geq 4$, for example, Fig. 4.9 shows a state grouping that leads to a well-defined trellis, but the grouping is neither the classical RSSE nor the complement states grouping.

Finally it is pointed out that a well-defined trellis is not necessary a radix-M trellis, where a radix-M trellis is referred to as a trellis with a structure like the ML trellis for an ISI channel with M-level channel inputs. More precisely, in a radix-M trellis, the set of states can be partitioned into a plurality of disjoint subsets with each subset consisting of M states which share the same M predecessor states. Fig. 4.9 is an RS

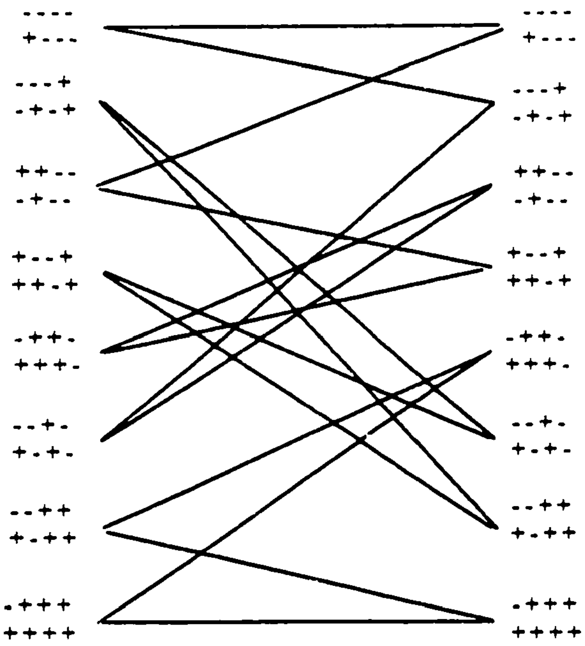


Fig. 4.9. A well-defined RS trellis obtained by a grouping other than RSSE or CSGT.

trellis which is a well-defined trellis but not a radix-2 trellis. It is obvious that both the classical kSSE and the CSGT (assuming that all pairs of complement states can be grouped together) will lead to a radix-2 RS trellis. Therefore, according to Theorem 4.1, for an ISI channel with $K \leq 3$, a well-defined RS trellis has to be a radix-2 RS trellis.

4.3 Further Reduction in Implementation Complexity

4.3.1 Branch Metric Shifting Method

A technique called branch metric shifting [38] is used to achieve a substantial reduction in complexity of the Viterbi detector for PR channels. Taking the EPR4 channel as an example, and referring to Fig. 4.1, the ACS recursion of the standard VA can be written as

$$
\begin{aligned}
S^0_{k+1} &= \min(S^0_k, S^4_k + (1+y_k)) \\
S^1_{k+1} &= \min(S^0_k + (1-y_k), S^4_k) \\[4pt]
S^2_{k+1} &= \min(S^1_k + (1-y_k), S^5_k) \\
S^3_{k+1} &= \min(S^1_k + (4-2y_k), S^5_k + (1-y_k)) \\[4pt]
S^4_{k+1} &= \min(S^2_k + (1+y_k), S^6_k + (4+2y_k)) \\
S^5_{k+1} &= \min(S^2_k, S^6_k + (1+y_k)) \\[4pt]
S^6_{k+1} &= \min(S^3_k, S^7_k + (1+y_k)) \\
S^7_{k+1} &= \min(S^3_k + (1-y_k), S^7_k)\ .
\end{aligned}
$$

A substitution can be made as follows:

$$
\begin{aligned}
&Q^0_k = S^0_k,\ Q^4_k = S^4_k + (1+y_k) \\
&Q^1_k = S^1_k + (1-y_k),\ Q^5_k = S^5_k \\
&Q^2_k = S^2_k,\ Q^6_k = S^6_k + 1 + y_k \\
&Q^3_k = S^3_k + (1-y_k),\ Q^7_k = S^7_k.
\end{aligned}
$$

Replacing path metric S^i_k with the modified path metric Q^i_k, the recursion becomes

$$
\begin{aligned}
Q^0_{k+1} &= \min(Q^0_k, Q^4_k) \\
Q^1_{k+1} &= \min(Q^0_k, Q^4_k - 2) + 2 - y_k - y_{k+1} \\[4pt]
Q^2_{k+1} &= \min(Q^1_k, Q^5_k) \\
Q^3_{k+1} &= \min(Q^1_k + 2, Q^5_k) + 2 - y_k - y_{k+1} \\[4pt]
Q^4_{k+1} &= \min(Q^2_k, Q^6_k + 2) + 2 + y_k + y_{k+1} \\
Q^5_{k+1} &= \min(Q^2_k, Q^6_k) \\[4pt]
Q^6_{k+1} &= \min(Q^3_k - 2, Q^7_k) + 2 + y_k + y_{k+1} \\
Q^7_{k+1} &= \min(Q^3_k, Q^7_k)\ .
\end{aligned}
$$

The corresponding trellis for the modified path metric Q_k is shown in Fig. 4.10 where each branch is labeled with the branch metric instead of the channel output. As shown in

Fig. 4.10, the computation in each recursion becomes a compare-select-add (CSA) instead of an ACS and the computation complexity is reduced significantly.

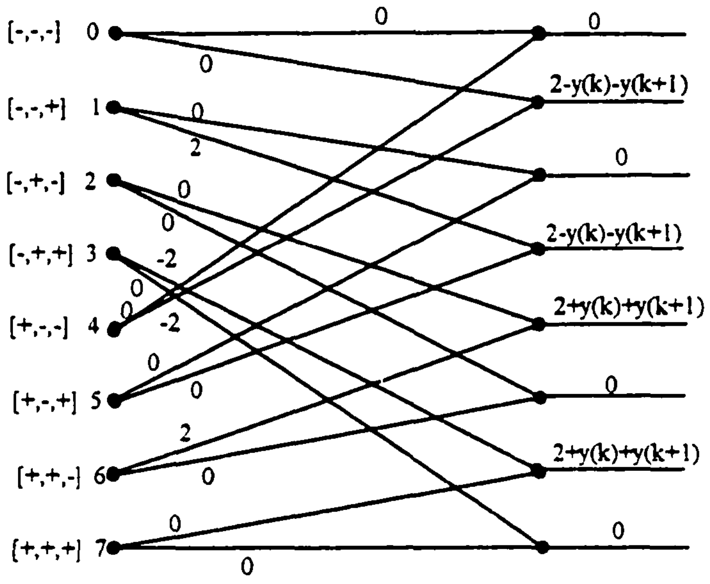


Fig. 4.10. Modified trellis for the EPR4 channel.

For a general PR channel with bipolar channel inputs, Fig. 4.11 shows a radix-2 subtrellis of the ML trellis, where 'a', 'b', 'c', 'd' denote branch metrics. The VA recursion

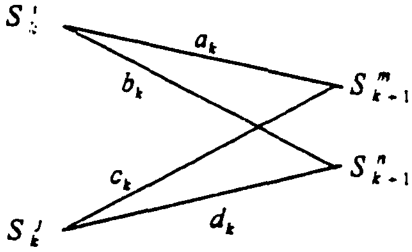


Fig. 4.11. Radix-2 subtrellis of a general ML trellis.

for this subtrellis is

$$S^m_{k+1} = \min(S^l_k + a_k, S^j_k \cdot - c_k)$$
$$S^n_{k+1} = \min(S^l_k + b_k, S^j_k + d_k).$$

Let

$$Q^l_k = S^l_k + a_k,\ \ Q^j_k = S^j_k + c_k,$$
$$Q^m_k = S^m_k + \gamma_k,\ \ Q^n_k = S^n_k + \beta_k,$$

then the recursion for Q_k is

$$Q^m_{k+1} = \min(Q^l_k, Q^j_k) + \gamma_{k+1}$$
$$Q^n_{k+1} = \min(Q^l_k, Q^j_k + W) + b_k - a_k + \beta_{k+1} \tag{4.5}$$

with

$$W = a_k + d_k - b_k - c_k\ , \tag{4.6}$$

where the parameter W in (4.6) is a constant for any channel with bipolar inputs. Although γ_k, β_k can be any value, they are usually optimized according to the subtrellis which includes either state m or state n as an input state. The CSA unit implementing (4.5) is shown in Fig. 4.12. Compared to the standard VA , savings in complexity are about 50% since the parameter W is always a constant.

4.3.2 Combining the CSGT with Fettweis's Method

Applying the branch metric shifting method for the RS trellis does not seem as appealing as for the ML trellis. This is because branches in the RS trellis may have two possible values, therefore the parameter W in Fig. 4.12 is not guaranteed to be a constant in the RS trellis, and this will increase the complexity by a significant amount. Fortunately, for some channels, such as MEEPR4, this problem can be solved by a

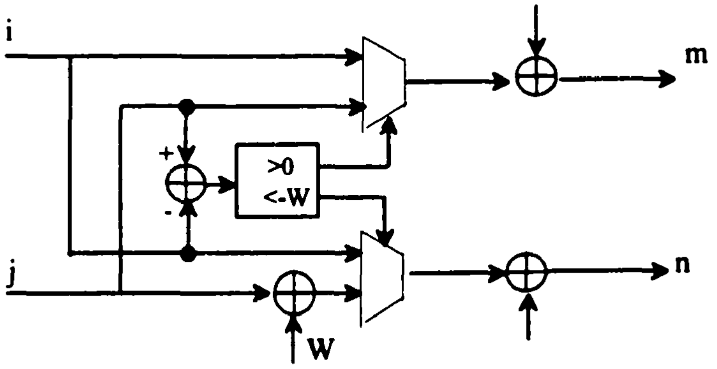


Fig. 4.12. The CSA block digram.

simple threshold detector. Referring to the RS trellis for the MEI:PR4 channel in Fig. 4.8, for superstate 'b', if the ideal channel outputs associated with the two branches entering sup?rstate 'b' have the same sign, then the parameter W for the corresponding radix-2 subtrellis will be a constant, and the Fettwe:s method can be applied . If the two branch values have opposite sign, then the parameter W is no longer a constant, fortunately if this is the case, the difference between the two branches is large enough to enable us to use a simple threshold (set at the midpoint between the two branch values) to select one of them. This only causes a negligible performance loss since the distance between the branches is $d = 16$ which is larger than the $d_{\min} = 2 \times \sqrt{53}$. Similar arguments can be made for superstate 'd' and 'g' .

Detailed derivation of the modified recursion is given below. Referring to Fig. 4.8, the VA recursion for the RS trellis is given by

$$S^a_{k+1} = \min(S^a_k, S^g_k + 2a^g_{k-4}y_k + 4)$$

$$S^b_{k+1} = \min(S^a_k + 5a^a_{k-4}y_k + 25, S^g_k - 3a^g_{k-4}y_k + 9)$$

$$S^c_{k+1} = \min(S^b_k + 4a^b_{k-4}y_k + 16, S^f_k - 2a^f_{k-4}y_k + 4)$$

$$S_{k+1}^{d} = \min(S_k^b + 9a_{k-4}^b y_k + 81, S_k^f - 7a_{k-4}^f y_k + 49)$$

$$S_{k+1}^{e} = S_k^c - 3a_{k-4}^c y_k + 9$$

$$S_{k+1}^{f} = \min(S_k^d + a_{k-4}^d y_k + 1, S_k^e + a_{k-4}^e y_k + 1)$$

$$S_{k+1}^{g} = \min(S_k^d + 6a_{k-4}^d y_k + 36, S_k^e - 4a_{k-4}^e y_k + 16)$$

where a_{k-4}^c denotes the $(k-4)^{th}$ bit in the survivor path of superstate '*c*', *etc.*

Let the substitution be

$$Q_k^a = S_k^a$$

$$Q_k^b = S_k^b + 4a_{k-4}^b y_k + 16$$

$$Q_k^c = S_k^c$$

$$Q_k^d = S_k^d + a_{k-4}^d y_k + 1$$

$$Q_k^e = S_k^e + a_{k-4}^e y_k + 1$$

$$Q_k^f = S_k^f - 2a_{k-4}^f y_k + 4$$

$$Q_k^g = S_k^g + 2a_{k-4}^g y_k + 4.$$

Then, the modified recursion will be

$$Q_{k+1}^a = \min(Q_k^a, Q_k^g)$$

$$Q_{k+1}^b = \min(Q_k^a, Q_k^g - 5y_k(a_{k-4}^a + a_{k-4}^g) - 20) + A_k$$

$$Q_{k+1}^c = \min(Q_k^b, Q_k^f)$$

$$Q_{k+1}^d = \min(Q_k^b, Q_k^f - 5y_k(a_{k-4}^b + a_{k-4}^f) - 20) + B_k$$

$$Q_{k+1}^e = Q_k^c + C_k$$

$$Q_{k+1}^f = \min(Q_k^d, Q_k^e) + D_k$$

$$Q_{k+1}^g = \min(Q_k^d, Q_k^e - 5y_k(a_{k-4}^d + a_{k-4}^e) - 20) + E_k$$

where

$$A_k = 4a_{k-3}^{b}y_{k+1} + 5a_{k-4}^{a}y_k + 41$$

$$B_k = a_{k-3}^{d}y_{k+1} + 5a_{k-4}^{b}y_k + 66$$

$$C_k = a_{k-3}^{e}y_{k+1} - 3a_{k-4}^{c}y_k + 10$$

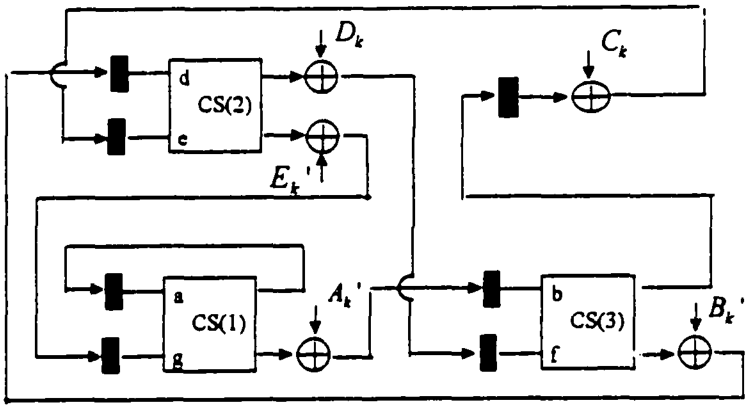
$$D_k = -2a_{k-3}^{f}y_{k+1} + 4$$

$$E_k = 2a_{k-3}^{g}y_{k+1} + 5a_{k-4}^{d}y_k + 39.$$

The implementation of the above modified recursion is shown in Fig. 4.13 where each CS unit implements a radix-2 subtrellis as shown in Fig. 4.14. In Fig. 4.14, the threshold detector is used to simplify the implementation. More specifically, *CTRL*(*i*) indicates when we should used the threshold detector *L(i)* to make a selection. For example, if $a_{k-4}^{a} \neq a_{k-4}^{g}$, referring to Fig. 4.8, the value of the branch from 'a' to 'b' has the same sign as the value of the branch from 'g' to 'b', then instead of using the threshold detector, the comparison result '*S*' should be used to make a selection for superstate 'b' ; whereas, if $a_{k-4}^{a} = a_{k-4}^{g}$, then the threshold detector *L(i)* should be used. The same argument can be made for updating superstates 'd' and 'g'. It is noted that the threshold setting will depend on the $(k-4)^{th}$ bit in the survivor path of the corresponding superstate.

4.4 The LVA on the CSGT RS Trellis

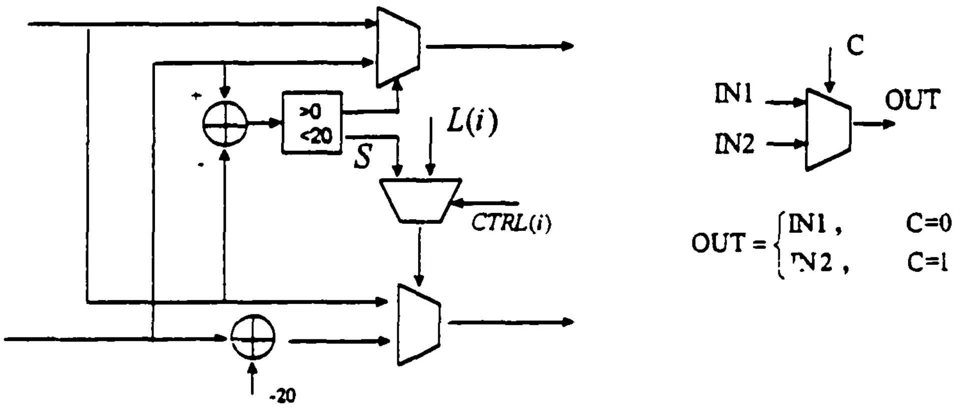
The performance of an LVA for a RS trellis obtained by the CSGT can be evaluated in a similar way as for the ML trellis in Section 2.3. The only difference is the correspondence between *input error sequence* $\Delta\mathbf{a}'$ and *output error sequence* $\Delta\mathbf{x}'$. For the ML trellis, $\Delta\mathbf{x}' = \Delta\mathbf{a}' * \mathbf{h}$, but this no longer holds for the CSGT RS trellis if there are K (channel length) consecutive error positions in $\Delta\mathbf{a}'$. For example, for an MEEPR4



$$A_k' = \begin{cases} 4a^b_{k-3}y_{k+1} - 5a^g_{k-4}y_k + 41, & \textbf{if} \quad L(1)=1 \\ 4a^b_{k-3}y_{k+1} + 5a^a_{k-4}y_k + 41 & \textbf{otherwise} \end{cases} \qquad B_k' = \begin{cases} a^d_{k-3}y_{k+1} - 5a^f_{k-4}y_k + 66, & \textbf{if} \quad L(3)=1 \\ a^d_{k-3}y_{k+1} + 5a^b_{k-4}y_k + 66 & \textbf{otherwise} \end{cases}$$

$$E_k' = \begin{cases} 2a^g_{k-3}y_{k+1} - 5a^e_{k-4}y_k + 39, & \textbf{if} \quad L(2)=1 \\ 2a^g_{k-3}y_{k+1} + 5a^d_{k-4}y_k + 39 & \textbf{otherwise} \end{cases}$$

Fig. 4.13. Implementation of the Fettweis method for the RS trellis of the MEEPR4 channel.



$$CTRL(1) = (a^e_{k-4} \neq a^g_{k-4}) \qquad CTRL(2) = (a^d_{k-4} \neq a^e_{k-4}) \qquad CTRL(3) = (a^b_{k-4} \neq a^f_{k-4})$$

$$L(1) = (y_k < -2a^g_{k-4}) \oplus (a^g_{k-4} = 1) \qquad L(2) = (y_k < -2d^f_{k-4}) \oplus (d^f_{k-4} = 1) \qquad L(3) = (y_k < -2a^f_{k-4}) \oplus (a^f_{k-4} = 1)$$

Fig. 4.14. Implementation of the radix-2 subtrellis (CS units of Fig. 4.13).

channel, if

$$\Delta \mathbf{a}' = [\ldots, 0, 2, 2, -2, 2, 0, \ldots],$$

the corresponding *output error sequence* in the ML trellis is

$$\Delta \mathbf{x}' = \Delta \mathbf{a}' * \mathbf{h} = 2 \times [\ldots 0, 5, 9, -4, -6, 1, -1, -2, -2, 0 \ldots],$$

while the corresponding *output error sequence* for the CSGT RS trellis shown in Fig. 4.8 is

$$\Delta \mathbf{x}' = 2 \times [\ldots, 0, 5, 9, -4, -6, 0, 0, 0, 0, 0, \ldots].$$

Note that thc error output sequence $\Delta \mathbf{x}'$ is truncated since the error path merges with the correct path earlier in the RS trellis. T':erefore, in the RS trellis, we need to truncate the *output error sequence* whenever necessary.

Once the *output error sequence* is found, we can follow the same procedure as in Section 2.3 to evaluate the performance of the LVA for an RS trellis. For example, a computer search has found that all the important error events of the LVA for the CSGT MEEPR4 RS trellis is exactly the same as for the ML trellis shown in Table 2.1. And those error events which only exist in the RS trellis have a large *error event gain* and therefore can be ignored in the performance evaluation.

In summary, using the CSGT, not only the RSVA has negligible performance loss compared to the VA, the RS-LVA also exhibits negligible performance loss compared to the LVA.

Chapter 5

Contributions and Future Work

5.1 Contributions of this Work

5.1.1 A Novel Reduced-State Technique

In Chapter 4, the complement states grouping technique was developed. Like traditional RSSE techniques, the CSGT leads to a well-defined RS trellis. The advantage of CSGT compared to traditional RSSE techniques is that the CSGT ensures a negligible performance loss while traditional RSSE techniques often cause unacceptable performance loss for storage systems using PR signaling. While the VA has a negligible performance loss for the RS trellis generated by the CSGT, the performance of the LVA for the RS trellis is no less impressive as shown in Section 4.4. Since the LVA is used in the coding system proposed in this dissertation, the CSGT can also provide a reduced-complexity implementation for this coding system with negligible performance loss.

In addition, the Fettweis reduced-complexity method [38] can be modified to be used in the CSGT RS trellis for the MEEPR4 channel. This leads to a further reduction of the implementation complexity for the MEEPR4ML system. The resultant reduced complexity MEEPR4ML system may prove to be an appealing post-EPR4 system.

5.1.2 LVA Performance Analysis

Although a lower bound on the LVA gain has been given in [26], Chapter 2 presents a method to evaluate the performance of the LVA with much more precision. It begins with the development of an upper bound on the error event probability, the bound is then used to exclude all the unimportant error events and obtain a complete list of important error events. Monte Carlo integration is then applied to evaluate the error event probabilities for these important error events and the overall error rate can be obtained accordingly. Using this method, a 3-candidate LVA is shown to exhibit a 2 dB gain over the VA for the MEEPR4 channel.

The method provided in this work has several advantages, first, it can handle situations where the noise is correlated which is crucial for PR channels used in storage systems. Second, it takes into account the multiplicity of error events, which is important in cases where the multiplicity plays a role that cannot be ignored. In addition, Chapter 4 illustrates how to evaluate the LVA performance on a CSGT RS trellis.

5.1.3 LVA Pipelined Implementation

The implementation of the LVA is also investigated in Chapter 2. Pipelined architectures turn out to be possible for the LVA. This leads to an LVA implementation that has the same recursion time as a standard VA (ignoring the overhead) while the computation complexity only increases linearly with the number of candidates.

5.1.4 An Efficient Coding System

Although the application of trellis codes to PR channels has been studied for a long time, the application of trellis codes to storage systems is limited by the requirement for high rate codes and the restriction on computation complexity.

Chapter 3 presents a coding system that combines the LVA with a high rate error detection code designed for ISI channels. The advantage of the coding system is that it can provide a substantial coding gain with a high code rate and with the computation complexity of an LVA decoder. In addition, it may also provide erasure information for the outer Reed-Solomon code which leads to the correction of twice as many erasures. The code design method and performance evaluation are also described in Chapter 3.

5.2 Future Work

5.2.1 Trellis Code Design Based on the LVA

Since traditional detectors use the Viterbi algorithm to achieve MLSE, the detector performance is governed by the minimum distance. Therefore, traditional trellis code design approaches aim at increasing the minimum distance in the coded trellis. This strategy needs to be changed if the LVA is used in the detector. Suppose the error detection code for the LVA is powerful enough to distinguish all codeword sequences so that the error rate is dominated by the event that the correct sequence is not in the LVA candidate list. By making this assumption, we do not need to worry about the error detection code, the only concern is how to design a trellis code to increase the *LVA gain*, *i.e.*, reduce the probability that the correct sequence is not in the LVA candidate list.

The following example illustrates how the code design approach for the LVA is

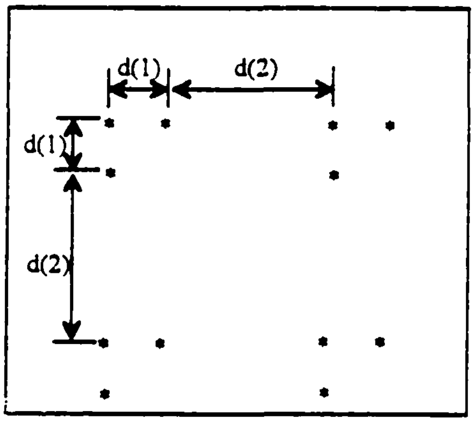


Fig. 5.1. A signal constellation.

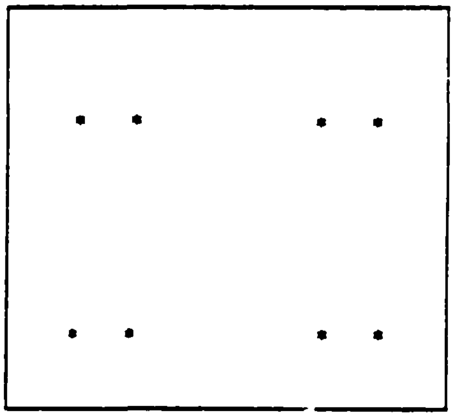


Fig. 5.2. A subset of the constellation.

different from the approach for the VA. Suppose we have a signal constellation as shown in Fig. 5.1 where each point could represent a channel output sequence. Fig. 5.2 is a subset of the constellation. If a code is designed by using the subset in Fig. 5.2 as the set of codewords, there will obviously be little coding gain if the VA is used, however, a large LVA gain is obtained if the LVA (with $L>1$) is used provided that $d(2)>>d(1)$.

5.2.2 Further Development of the CSGT

In this dissertation, the CSGT has only been applied to ISI channels with binary channel inputs where the definition of complement states is straightforward. In an ISI channel with multiple level inputs, there are more options for state grouping. For example, to generate an RS trellis, we may begin by using the traditional RSSE techniques to generate a first-step RS trellis, then apply the CSGT to this first-step RS trellis to further reduce the number of states. In addition to this scheme, there may be various forms of combinations of traditional RSSE techniques with the CSGT for generating RS trellises. The state distance metric developed in this dissertation can be used as a measurement of the performance of an RS trellis and therefore provides a guideline on how to group ML states into a superstate.

References

[1] P. H. Siegel and J. K. Wolf, "Modulation and coding for information storage," *IEEE Commun. Mag.*, vol. 29, pp. 68-86, Dec. 1991.

[2] T. Howell, *et al.*, "Error rate performance of experimental gigabit per square inch recording components," *IEEE Trans. Magn.*, vol. 26, pp. 2298-2302, Sept. 1990.

[3] R. Cideciyan, F. Dolivo, R. Hermann, W. Hirt, and W. Schott, "A PRML system for digital magnetic recording," *IEEE J. Select. Areas. Commun.*, vol. 10, pp. 38-56, Jan. 1992.

[4] J. Fife, "New coding scheme boosts read channel performance," *Data Storage*, pp. 45-52, Sept. 1997.

[5] P. Bednarz, S. Lin, C. Modlin, and J. Cioffi, "Design, performance , and extension of the RAM-DFE architecture," *IEEE Trans. Magn.*, vol. 31, pp. 1196-1201, Mar. 1995.

[6] P. Bednarz, N. Sands, C. Modlin, S. Lin, I. Lee, and J. Cioffi, "Performance evaluation of an adaptive RAM-DFE read channel," *IEEE Trans. Magn.*, vol. 31, pp. 1121-1127, Mar. 1995.

[7] E. Dahlman and B. Gudmundson, "Performance improvement in decision feedback equalizers by using 'soft decision'," *Electron. Lett.*, vol. 24, pp. 1084-1085, Aug. 1988.

[8] J. Bergmans, J.Voorman, and H. Wong-Lam, "Dual decision feedback equalizer," *IEEE Trans. Commun.*, vol. 45, pp. 514-518, May 1997.

[9] D. Krueger, J. R. Cruz, and R. He, "A ternary coded magnetic recording system," in *Proc. Internat. Conf. Commun.*, 1995, pp. 638-641.

[10] M. P. Vea, "Intertrack interference in high density recording modeling and equalization," *PhD Dissertation*, Carnegie Mellon University, Nov. 1993.

[11] P. Voois and J. Cioffi, "Multichannel signal processing for multiple-head digital magnetic recording," *IEEE Trans. Magn.*, vol. 30, pp. 5100-5113, Nov. 1994 .

[12] T. Yamauchi and J. Cioffi, "A nonlinear model for thin film disk recording systems," *IEEE Trans. Magn.*, vol. 29, pp. 3993-3995, Nov. 1993.

[13] A. Barany and H. Bertram, "Transition noise model for longitudinal thin-film media," *IEEE Trans. Magn.*, vol. 23, pp. 1776-1788, Mar. 1987.

[14] R. Hermann,"Volterra modeling of digital magnetic saturation recording channels," *IEEE Trans. Magn.*, vol. 26, pp. 2125-2127, Sept. 1990

[15] S. Yuan and H. Bertram, "Magnetoresistive heads for ultra high density recording," *IEEE Trans. Magn.*, vol. 29, pp. 3811-3816, Nov. 1993.

[16] S. B. Wicker and V. K. Bhargava, *Reed-Solomon Codes and Their Applications*, IEEE Press, NY, 1994

[17] J. Moon and B. Brickner, "Maximum transition run codes for data storage system," *IEEE Trans. Magn.*, vol. 32, pp. 3992-3994, Sept. 1996.

[18] A. Calderbank, C. Heegard, and T. Lee, "Binanry convolutional codes with application to magnetic recording," *IEEE Trans. Inform. Th.*, vol. 32, pp. 797-815, Nov. 1986.

[19] J. Wolf and G. Ungerboeck, "Trellis coding for partial response channels," *IEEE Trans Commun.*, vol. 34, pp. 765-773, Aug. 1986.

[20] D. Krueger, J. R. Cruz, and R. He, "Equalization of the convolutional-coded Lorentzian channel," *IEEE Trans. Magn.*, vol. 31, pp. 3030-3032, Nov. 1995.

[21] D. Krueger, "Trellis code design for intersymbol interference channels," *PhD Dissertation*, University of Oklahoma, 1995.

[22] R. Karabed and P. Siegel, "Match spectral null codes for partial response channels," *IEEE Trans. Inform. Th.*, vol. 37, pp. 818-855, May 1991.

[23] H. Thapar, J. Rae, C. Shung, R. Karabed, and P. Siegel, "On the performance of a rate 8/10 match spectral null code for class-4 partial response," *IEEE Trans. Magn.*, vol. 28, pp. 2883-2888, Sept. 1992.

[24] L. Fredrickson, R. Karabed, J. Rae, P. Siegel, H. Thapar, R. Wood, "Improve trellis-coding for partial response channels," *IEEE Trans. Magn.*, vol. 31, pp. 1141-1148, Mar. 1995.

[25] J. Rae, G. Christiansen, P. Siegel, K. Karabed, H. Thapar, and S. Shih, "Design and performance of a VLSI 120 Mb/s trellis-coded partial response channels," *IEEE Trans. Magn.*, vol. 31, pp. 1208-1214, Mar. 1995.

[26] N. Seshadri and C. W. Sundberg, "List Viterbi decoding algorithm with applications," *IEEE Trans. Commun.*, Vol. 42, pp. 313-322, Feb./Mar./Apr. 1994.

[27] M. V. Eyuboglu and S. U. H. Qureshi, "Reduced-state sequence estimation with set partitioning and decision feedback," *IEEE Trans. Commun.*, vol. 36, pp. 13-20, Jan. 1988.

[28] R. He, V. Guranovic, and J. R. Cruz, "Reduced state Viterbi decoder for simple partial erasure models," *Electron. Lett.*, vol. 32, pp. 1349-1350, Jul. 1996.

[29] R. Y. Rubinstein, *Simulation and the Monte Carlo Method*, John Wiley & Sons, NY, 1981, ch. 4, pp. 114-157.

[30] J. M. Wozencraft and I. M. Jacobs, *Principles of Communication Engineering*, John Wiley & Sons, NY, 1965.

[31] G. Ungerboeck, "Trellis-coded modulation with redundant signal sets," Pts. I and II, *IEEE Commun. Mag.*, vol. 25, pp. 5-21, Feb. 1987.

[32] N. Glover and T. Dudley, *Practical Error Correction Design for Engineers*, Data Systems Technology, Corp., CO, 1988.

[33] A. M. Patel, "Two-level coding for error control in magnetic disk storage products," *IBM J. Res. Develop.* vol. 33, pp. 470-484, Jul. 1989.

[34] A. M. Patel, "On-the-fly decoder for multiple byte errors," *IBM J. Res. Develop.* vol. 30, pp. 259-269, May. 1986.

[35] W. U. Lee and F. S. Hill, "A maximum-likelihood sequence estimator with decision-feedback equalization, " *IEEE Trans. Commun.*, vol. COM-25, pp. 971-979, Sept. 1977.

[36] A. Duel and C. Heegard, "Delayed decision-feedback sequence estimation," in *Proc. Allerton Conf. Commun., Cont., Comput.*, 1985, pp. 878-887.

[37] G. Ungerboeck, "Channel coding with multilevel/phase signals, " *IEEE Trans. Injorm. Th.*, voi. IT-28, pp. 55-67, Jan. 1982.

[38] G. Fettweis, R. Karabed, P. H. Siegel and H. K. Thapar, "Reduced-complexity Viterbi detector architectures for partial response signaling," in *Proc. Global Telecommun. Conf.*, 1995, pp. 559-563.

[39] N.Kobayashi, W.Sakurai, and S.Mita, "Simplified EPRML methods based on selective ACS for high-density magnetic recording channels," in *Proc. Global Telecommun. Conf.*, 1995, pp. 564-570.

[40] P.J.Black and T.H.Meng, "A 140-Mb/s, 32-state, radix-4 Viterbi decoder," *IEEE J. Solid-State Circuits*, vol. 27, pp. 1877-1885, Dec. 1992.