ADVANCED CHANNEL CODING TECHNIQUES USING BIT-LEVEL SOFT INFORMATION

A Dissertation

by

JING JIANG

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

August 2007

Major Subject: Electrical Engineering

ADVANCED CHANNEL CODING TECHNIQUES USING BIT-LEVEL SOFT INFORMATION

A Dissertation

by

JING JIANG

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Approved by:

Chair of Committee,	Krishna R. Narayanan
Committee Members,	Costas N. Georghiades
	Andrew K. Chan
	Jianxin Zhou
Head of Department,	Costas N. Georghiades

August 2007

Major Subject: Electrical Engineering

ABSTRACT

Advanced Channel Coding Techniques
Using Bit-level Soft Information. (August 2007)

Jing Jiang, B.S., Shanghai Jiao Tong University
Chair of Advisory Committee: Dr. Krishna R. Narayanan

In this dissertation, advanced channel decoding techniques based on bit-level soft information are studied. Two main approaches are proposed: bit-level probabilistic iterative decoding and bit-level algebraic soft-decision (list) decoding (ASD).

In the first part of the dissertation, we first study iterative decoding for high density parity check (HDPC) codes. An iterative decoding algorithm, which uses the sum product algorithm (SPA) in conjunction with a binary parity check matrix adapted in each decoding iteration according to the bit-level reliabilities is proposed. In contrast to the common belief that iterative decoding is not suitable for HDPC codes, this bit-level reliability based adaptation procedure is critical to the convergence behavior of iterative decoding for HDPC codes and it significantly improves the iterative decoding performance of Reed-Solomon (RS) codes, whose parity check matrices are in general not sparse. We also present another iterative decoding scheme for cyclic codes by randomly shifting the bit-level reliability values in each iteration. The random shift based adaptation can also prevent iterative decoding from getting stuck with a significant complexity reduction compared with the reliability based parity check matrix adaptation and still provides reasonable good performance for short-length cyclic codes.

In the second part of the dissertation, we investigate ASD for RS codes using bit-level soft information. In particular, we show that by carefully incorporating bit-level soft information in the multiplicity assignment and the interpolation step, ASD can significantly outperform conventional hard decision decoding (HDD) for RS codes with a very small amount of complexity, even though the kernel of ASD is operating at the symbol-level. More importantly, the performance of the proposed bit-level ASD can be tightly upper bounded for practical high rate RS codes, which is in general not possible for other popular ASD schemes.

Bit-level soft-decision decoding (SDD) serves as an efficient way to exploit the potential gain of many classical codes, and also facilitates the corresponding performance analysis. The proposed bit-level SDD schemes are potential and feasible alternatives to conventional symbol-level HDD schemes in many communication systems.

To my family

ACKNOWLEDGMENTS

Looking back to my graduate years, I would like to thank many people who have contributed to this dissertation. First of all, I wish to take this opportunity to express my deepest gratitude to my advisor Professor Krishna R. Narayanan. It was his fascinating advanced coding course that inspired me to work in the area of channel coding for communications. His brilliant insight, patient guidance and countless encouragement throughout the last five years helped me to grow quickly from an inexperienced student to a sophisticated researcher. Working with him was truly a pleasant, stimulating and rewarding experience. Without his continuous support, the completion of this dissertation would have never become possible.

My heartfelt thanks go to Professor Marc Fossorier and Professor Ralf Koetter. Two main ingredients of this dissertation, reliability based iterative decoding and algebraic list decoding of Reed-Solomon (RS) codes were largely inspired by their seminal works. Their invaluable comments and constructive criticisms lead to significant improvements in many aspects of this dissertation. It is my real fortune to have their opinions on my work. Besides, the visit to Professor Koetter's group at the University of Illinois was a fruitful and enjoyable experience to me.

I would also like to thank Professor Costas Georghiades, Professor Andrew Chan and Professor Jianxin Zhou, for serving as my dissertation committee and being supportive in different stages of my doctoral studies.

Moreover, I want to thank Professor Narayanan for generously offering me many academic opportunities: attending top conferences, visiting world-renowned research institutes, working as a summer intern in industry and lecturing a senior undergraduate DSP course, which motivated and inspired many interesting research works. The main results in Chapter V were initiated by some interesting discussions with Ratnakar Niranjan, Ivana Djurdjevic and Jorn Justesen during my visit in Illinois and during my internship at Seagate Research. I would like to thank many other friends during my study and visit the last five years for making my graduate life more enjoyable. I am very grateful to my group mates, Jing Li, Dung Doan, Yongzhe Xie, Guosen Yue, Jun Zheng, Wenyan He, Vivek Gulati, Janath Peiris, Hari Sankar, Nitin Nangare, Angelos Liveris, Vladimir Stankovic, Abhiram Prabhkar, Kapil Bhattad, Qiang Li, Lingjia Liu, Salim El Rouayheb, Makesh Pravin, Anantharaman Balasaubaramanian, Weiyu Chen, Sabaresan Mothi and Karthik Nagasubramanian in the WCL group. I would also like to thank my friends and colleagues, Dake He, Ashish Jagmohan, Jian Lou, Jun Chen, Vadim Sheinin, Ivana Djurdjevic, Ching He, Guanjun Zhang, Xueshi Yang, Erozan Kurtas, Alexander Kuznetsov, Fatih Erden, Haiyun Yang, Lei Chen, Lin Yang, Jinghu Chen, Xiangyu Tang, Fang Bao, Tianyu Zhan, Ratnakar Niranjan, Wai Fong, Jun Xu, Wei Zeng, Suquan Ding and Lifeng Su, during my stay at IBM, Seagate and Legend Silicon, UIUC, Institute Eurecom, etc..

Finally, this acknowledgement would be incomplete without mentioning my family. I sincerely thank my parents, my grandmother and my wife, Yiqing, for their unconditional love. Without their support, my dream of this dissertation would never have become true. This dissertation is dedicated to them.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE Page

FIGURE Page

FIGURE Page

LIST OF ALGORITHMS

CHAPTER I

INTRODUCTION

Reed-Solomon (RS) codes are one of the most popular error correction codes in many state-of-the-art communication and data storage systems. In most applications, RS codes are decoded via algebraic hard decision decoding (HDD), which does not fully exploit the error correcting capability of RS codes. When soft information about the channel output is available (for example, in an additive white Gaussian noise (AWGN) channel), HDD can incur a significant performance loss compared with optimal soft decision decoding (SDD). For the AWGN channel, the loss is about 2-4 dB for practical high rate RS codes at the target error rate where they are usually operating. More importantly, in many practical systems, RS codewords are mapped into their binary image expansions and then transmitted through channels using binary modulation formats. Therefore, it is of both theoretical and practical value to study decoding schemes for RS codes using bit-level soft information.

In this dissertation, we propose two advanced channel decoding techniques, namely, probabilistic iterative decoding [1][2] and algebraic soft-decision (list) decoding (ASD) [3] [4] for RS codes, using bit-level soft information. We have shown that, in spite of the fact that RS codes are non-binary codes, efficiently exploiting bit-level soft information plays a crucial role in achieving the potential gain of the codes with a moderate complexity and facilitates the corresponding performance analysis as well.

The rest of this dissertation is organized as follows: we first review some background on RS codes, which will be used throughout this dissertation in Chapter II.

The journal model is *IEEE Transactions on Automatic Control*.

We study iterative decoding schemes in Chapter III and Chapter IV. In Chapter III, we propose an iterative sum product algorithm (SPA) based decoding by adapting the parity check matrix in each iteration according to the bit reliabilities. In Chapter IV, another iterative decoding method based on stochastic shifting the bit-level reliability values of the coded bits in each iteration is presented. In Chapter V, we study the performance of ASD for RS codes using bit-level soft information. Finally, we summarize the main contributions of this dissertation and discuss potential future works of the bit-level advanced channel coding techniques in Chapter VI.

CHAPTER II

BACKGROUND ON REED-SOLOMON CODE

In this chapter, we briefly review background materials on RS codes. We first give a historic survey of RS codes and the symbol-level code structure in Section A. Commonly used algebraic HDD algorithms for RS codes are introduced in Section B. The binary image expansion of RS codes over GF(2^m) is presented in Section C. Two advanced channel coding techniques, i.e. iterative decoding and algebraic list decoding are discussed in Section D and Section E respectively. Other popular SDD algorithms of RS codes are introduced in Section F. Finally, the ensemble average performance of the binary image expansion of RS codes over GF(2^m) under maximum likelihood decoding (MLD) is analyzed in Section G.

A. Symbol-level Structure of Reed-Solomon Codes

In this section, we give a brief review of the symbol-level code structure of RS codes.

1. Evaluation Form of Reed-Solomon Codes

RS codes were invented in 1960 by Reed and Solomon [5] as the name stands for. The code was initially defined as a set of message polynomials evaluated at N distinct points, which is usually referred to as the evaluation form of RS codes.

Consider the evaluation form of an (N, K) RS code over $GF(q)$. Define the message vector $\underline{f}$ as:

$$\underline{f} = (f_0, f_1, \cdots f_{K-1}), f_i \in GF(q). \tag{2.1}$$

The polynomial form of the message is:

$$f(x) = f_0 + f_1 x + \cdots + f_{K-1} x^{K-1} \tag{2.2}$$

Let $D = \{x_1, x_2, \cdots, x_N\}$ be a set of distinct elements in $GF(q)$. An RS code is defined by evaluating the message polynomial at N points:

$$\mathcal{C}(N, K) = \{(f(x_1), \cdots, f(x_N))\} \tag{2.3}$$

for all message polynomials $f(x)$. If the evaluation points take the form $x_i = \beta^i(i = 0, \cdots, N-1)$, where β is a primitive element in GF(q), the RS code is evaluated using a set of fixed order points.

Reed and Solomon showed many nice properties of RS codes. For instance, RS codes are shown to be maximum distance separable (MDS) codes at the symbol-level. That is, for an (N, K) RS code, its minimum distance $d_{min} = N - K + 1$, which is the maximum possible at the symbol-level. The initial paper also proposed encoding and decoding schemes for RS codes, however, they are not very efficient for practical implementations.

2. Non-binary BCH Code Form of Reed-Solomon Codes

RS codes are closely related to another class of popular error correcting codes, Bose Chaudhuri and Hocquenghem (BCH) codes [6]. Gorenstein and Zierler showed that RS codes are equivalent to q-ary BCH codes over GF(q) [7].

Consider a BCH code over $GF(q)$. The generator polynomial of the code is:

$$g(x) = (x - \beta^b)(x - \beta^{b+1}) \cdots (x - \beta^{b+N-K-1}) \tag{2.4}$$

where β is a primitive element in $GF(q)$ and b is an integer number. When $b = 1$, the code becomes a narrow sense BCH code.

Since the generator polynomial has $N - K$ consecutive roots, the corresponding parity check matrix can be represented as follows [8]:

$$\mathbf{H}_s = \begin{pmatrix} 1 & \beta^{b} & \cdots & \beta^{(N-1)b} \\ 1 & \beta^{b+1} & \cdots & \beta^{(N-1)(b+1)} \\ & & \cdots & \\ 1 & \beta^{(b+\delta-2)} & \cdots & \beta^{(N-1)(b+\delta-2)} \end{pmatrix} \tag{2.5}$$

where $\delta = d_{min} = N - K + 1$. In addition, due to the cyclic property of BCH codes, it can be shown that the dual code of an RS code is also an RS code. Therefore the generator matrix of an RS code can also be represented in the form of (2.5). Comparing (2.5) when $b = 0$ with (2.2) evaluated by a set of fixed order points $x_i = \beta^i$ $(i = 0, \cdots, N-1)$, we can see that RS codes and q-ary BCH codes over GF(q) are equivalent.

B. Hard Decision Decoding of Reed-Solomon Codes

Many efficient HDD schemes have been proposed to decode RS codes up to $t = \lfloor \frac{d_{min-1}}{2} \rfloor$ symbol-level errors. Due to the equivalence of RS codes and non-binary BCH codes, the bounded distance HDD for non-binary BCH codes proposed by Peterson [9] can be applied to RS codes. However, the computational complexity of Peterson's algorithm is still quite large for long codes, mainly at the error location step. It was not until Berlekamp's seminal work [10], HDD of RS codes became truly efficient. Several years later, Massey showed that Berlekamp's decoding scheme is equivalent to the problem of synthesizing the shortest linear feedback shift register (LFSR) to generate a given sequence and HDD of RS can be realized efficiently using a set of shift registers in hardware [11]. The algorithm is therefore called Berlekamp and Massey (BM) decoding. Besides BM decoding, Sugiyama *et al.* showed that Euclid's algorithm can also be used for error location in RS HDD [12]. Frequency domain encoding and decoding methods of RS codes have been studied in depth in [13]. In

1986, Berlekamp and Welch proposed an interpolation based decoding algorithm for RS codes [14], which paved way for the later algebraic list decoding algorithms [3].

When error locations are known to the decoder, the error magnitudes can be obtained by solving a set of linear equations. Forney proposed a fast algorithm to calculate the error magnitudes [15], which is referred to as Forney's algorithm. BM algorithm can also be easily modified to accommodate erasures. Since the error location of erased symbols are known, BM algorithm can decode more erasures than errors. In general, HDD of RS codes will succeed if the number of errors e and the number of erasures g satisfies $2e + g < d_{min}$. In other words, if twice the number of erroneous symbols plus the number of erased symbols does not exceed the designed minimum distance of the code, we are guaranteed to be able to recover the original codeword. Besides, erasure decoding can also be used as a systematic way to encode RS codes. For more decoding techniques of RS codes, we refer interested readers to [8, 13, 16].

C. Binary Image Expansions of Reed-Solomon Codes

In this section, we consider RS codes over $GF(2^m)$, which are most commonly used. It is known that all the 2^m elements in $GF(2^m)$, 0, 1, β, β^2, $\cdots$, β^{2^m-2}, can be represented by an m-dimensional binary vector over $GF(2)$ using a basis which spans $GF(2^m)$. Addition operation in $GF(2^m)$ is nothing but component wise addition of the vector over $GF(2)$. Similarly, multiplication can be carried out by multiplying a binary vector with a binary multiplication matrix. Therefore, each entry in the parity check matrix $\mathbf{H}_s$ can be replaced by an $m \times m$ matrix over GF(2) for the purpose of multiplication. For instance, consider RS codes over GF(4) and let β be a root of the primitive polynomial $p(x) = x^2 + x + 1$. β has the binary vector representation

$[0, 1]$ and the multiplication operation $\times\beta$ corresponds the binary multiplication of the vector expansion with a multiplication matrix: $\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}$.

Hence, $\mathbf{H}_s$ has an equivalent binary image expansion $\mathbf{H}_b$ and the RS code has a binary linear block code representation. Let $n = N \times m$ and $k = K \times m$ be the length of the codeword and the message at the bit level, respectively. Hence, $\mathbf{H}_b$ is an $(n - k) \times n$ binary parity check matrix. In other words, an RS code over GF(2^m)can also be viewed as a binary linear block code.

D. Iterative Decoding

As discussed in Section C, RS codes can be represented using their binary image expansions. Therefore, RS decoding problem is turned into a general decoding problem for binary linear block codes.

Ever since the invention of turbo codes [1] and the rediscovery of low-density-parity-check (LDPC) codes [2], belief propagation (BP) based iterative decoding has been a panacea for many coding and communication problems. Iterative decoding of linear block codes and the sum product algorithm (SPA) was studied in [17] [18] [19]. However, BP algorithm is usually not considered to be suitable for high density parity check (HDPC) codes [18], since iterative decoding can easily get stuck at pseudo-equilibrium points due to the large number of short cycles in the corresponding bipartite graph of the code. Some research works have been focused on the construction of proper parity check sets for iterative decoding [18] [19], since the performance of iterative decoding will be different with the choice of parity check matrix even if the code is the same. Lucas *et al.* [18] suggested using minimum weight parity check sets for iterative decoding. Some algorithms for the small weight parity check sets search are also discussed. However, in general, finding a minimum parity check vector is NP-complete. Moreover, for MDS codes, such as RS codes, we cannot expect to get a very small weight parity check vector. Since the parity check matrix is nothing but the generator matrix of the dual code, which is also an MDS code, the minimum weight of every parity check must be at least $(K + 1)$ (typically much larger). For high rate RS codes, where K is large, the parity check matrix is necessarily non-sparse, which is unsuitable for iterative decoding.

Yedidia *et al*. [20] established the inherent connection between BP and statistical physics. In [19], Yedidia *et al*. proposed the “generalized belief propagation (GBP)” algorithm, which introduces hidden states in the bipartite graph of the code to help to improve the performance of iterative decoding. However, GBP still has problems in decoding HDPC codes over AWGN channels. Nevertheless, GBP inspired our study of BP algorithms for HDPC codes such as RS codes.

Recently, iterative decoding of RS codes have been studied by several research groups. Ungerboeck proposed a sub-trellis based iterative decoding scheme [21]. Yedidia proposed a factor graph based iterative decoding of RS codes using Galois-field fast Fourier transform (GFFT) as an instance of GBP [22]. Unfortunately, the proposed iterative decoding schemes only work well for short length RS codes. It is suggested that directly applying iterative decoding to HDPC codes does not give good results since the parity check matrix of an RS code is not sparse so that iterative decoding can quickly get stuck.

E. Algebraic Soft-decision List Decoding

In Section B, we have discussed many HDD schemes for RS codes. However, all the HDD algorithms can only decode up to $t = \lfloor \frac{N-K}{2} \rfloor$. In 1995, based on the previous work by Berlekamp [23] and Berlekamp and Welch [14], Sudan [3] built the first algebraic HDD algorithm which can correct beyond half d_{min} errors for low rate RS codes with polynomial time complexity. Two years later, Guruswami and Sudan [24] improved Sudan's decoder [3] and enlarged the decoding radius up to $t_{GS} = \lfloor N - \sqrt{N(K-1)} \rfloor$ for all rates with polynomial time complexity. Koetter and Vardy [4] generalized Guruswami and Sudan (GS) decoding algorithm and presented a multiplicity assignment strategy (MAS) for the GS list decoder, which can take into account the soft information available at the decoder input. Algebraic soft-decision decoding (ASD) algorithms have since then gained great research interest. In contrast to the iterative decoding algorithms in Section D, ASD is algebraic in nature. However, soft information can still be incorporated in the algebraic decoding procedure by appropriate multiplicity assignment.

From a theoretical perspective, optimal MAS for ASD and corresponding performance analysis still remains an open problem. In [4], Koetter and Vardy presented an asymptotically optimal MAS that maximizes the transmission rate for a given channel such that the probability of error can be made arbitrarily small as the code length goes to infinity. Multiplicity assignment optimization for finite length RS codes has been considered in [25–27] using numerical algorithms. In [28], a general framework has been studied for channels with additive cost [28]. In [27], MAS for general discrete memoryless channels (DMC) has been investigated and an upper bound based on a Chernoff bounding technique has been derived. However, the Chernoff-type bound [27] largely relies on numerical computations and gives little insight into the decoding region of ASD under a certain MAS. Besides, the bound becomes loose for practical high rate RS codes. More recently, the decoding region and typical error patterns of ASD with infinite cost over some basic DMC's have been studied independently in

[29][1] and [30]. Based on the analysis in [29], the performance of practical high rate RS codes can be tightly bounded over erasure channels. However, as suggested in [30], even with infinite cost, ASD has a significant gain over BM only for RS codes of low rates or when the number of candidate symbols at the decoder input is small.

From a practical perspective on the other hand, many techniques have been proposed for ASD, for example, [31–38] and references therein. We also refer interested readers to [39] for a comprehensive tutorial of the generic GS algorithm. Most of ASD schemes, such as the KV algorithm, can significantly outperform HDD for low rate RS codes. However, to achieve large coding gain for high rate practical RS codes, the complexity can be prohibitively large [34]. Large computational complexity and the limited performance gain for high rate RS codes of ASD becomes the main obstacle to steer ASD decoding towards an implementable alternative to conventional HDD.

F. Other Soft Decision Decoding Schemes for Reed-Solomon Codes

In this section, we review some other existing SDD techniques for RS codes.

1. Enhanced Algebraic Hard Decision Decoding

The idea to take advantage of the soft information of the received bits to improve the decoding performance of RS codes can be dated back to Forney's 1966 work [40]. In the original paper, Forney suggested successively erasing some of the unreliable symbols in the received signals and use algebraic HDD to decode. Thus, if twice the number of erroneous symbols plus the number of erased symbols does not exceed the designed minimum distance of the code, i.e., $2e+g < d_{min}$, we can recover the original codeword. Forney proved that GMD guarantees to return an estimated codeword.

[1] [29] is a conference paper which contains parts of this dissertation.

Since GMD has to run the generic HDD each time after erasing some symbols, the complexity of the GMD is about $d_{min}/2$ times as large as that of the generic HDD. Since GMD is based on the algebraic HDD and also takes advantage of the reliability information, it is called enhanced algebraic HDD.

Another enhanced algebraic HDD is Chase decoding [41]. The most popular version of Chase decoding, Chase type-II decoding (which is also usually abbreviated as Chase decoding) is also a reliability-assisted decoder, i.e., the decoder exhaustively flips up to d least reliable symbols and incorporates an algebraic HDD in each step. If the number of errors in the rest of the symbols is within the HDD error correction radius, the codeword can be recovered. Since Chase decoding involves exhaustively flipping symbols, its complexity is exponential in d. A hybrid of Chase type-II and GMD (Chase-GMD) has been proposed in [42]. Enhanced HDD usually gives a moderate performance improvement over HDD with reasonable complexity for practical applications.

2. Reliability Based Decoding

As shown in Section C, RS codes can be represented using their binary expansions. Efficient decoding methods for the general binary linear block codes such as the reliability based ordered statistics decoding (OSD) [43][44] can be applied to RS codes as well. The main idea of OSD is to propose reprocessing based on the most reliable basis (MRB) of the received signal. Since the reprocessing involves exhaustively flipping p bits in the MRB, the complexity increases exponentially with p. Nevertheless, with some small p, the OSD decoder provides good performance for short to medium length block codes. Recently, a variation of box and match (BMA) decoding [45] has been proposed by Fossorier and Valembois to trade-off memory for time complexity. Another variation of OSD which does not need large memory and turns out to be efficient has been proposed by Wu *et al.* [46] [47]. More recent works on BMA decoding for linear block codes can be found in [48] and [49].

3. Trellis Based Coset Decoding

All the soft decision decoding methods discussed above are list decoding based soft input hard output (SIHO) algorithms. In some situations, it is desirable to obtain soft output from the decoder. A typical example is when turbo equalization (TE) [50] is employed at the receiver and soft outputs from the decoder have to be fedback to the equalizer. Consequently, soft-input-soft-output (SISO) decoding algorithms for RS codes are of research interest. Though averaging over all returned codewords in the list can generate the soft output [51], a natural SISO decoder is still favorable in some applications.

In the early 90s, Vardy and Be'ery suggested a binary decomposition scheme of RS codes into BCH subfield subcodes and glue vectors [52], which essentially reduces the overall trellis complexity of MLD of RS codes. Some recent works [53] further reduce the complexity and modify the algorithm to be able to generate soft outputs efficiently. Nevertheless, this trellis based scheme still has an exponentially increasing complexity and works only for short RS codes or RS codes with small d_{min}.

G. Performance of Reed Solomon Codes under Maximum Likelihood Decoding

Maximum likelihood decoding (MLD) of RS codes is a non-trivial task, especially for long codes. It has recently been shown that MLD of RS codes is NP-complete [54]. Consequently, analytical bounds on the performance of RS codes under MLD are of interest as benchmarks for suboptimal decoders. However, the performance of the RS code under MLD using a particular binary image expansion is still difficult to analyze, since the weight enumerator of an RS code using a specific binary image expansion is in general unknown. In this section, we study the ensemble average performance of RS codes under MLD [55][56] using Divsalar bound [57], which is tighter than the union bound in the low signal to noise ratio (SNR) region. The ensemble average of the RS code is taken by averaging over all binary representations of the RS code expanded using all possible binary bases for each symbol.

Due to the maximum distance separable (MDS) property of RS codes, the symbol-level weight enumerator is well known [8] and given by:

$$R_w = \binom{N}{w}(q-1)\sum_{i=0}^{w-d_{min}}(-1)^i\binom{w-1}{i}q^{w-d_{min}-i} \tag{2.6}$$

where w is the symbol-level weight of a valid codeword and R_w is the corresponding weight enumerator at the symbol-level.

Whereas, the bit-level weight enumerator of an RS code is not unique, i.e., each symbol of the RS codeword can be expanded into different binary vectors using different bases. The binary image expansion of an RS code depends on the specific basis chosen to expand the m-dimensional binary vectors for each symbol. Though the weight enumerator given the specific basis is difficult to obtain, Retter [55] computed the averaged weight enumerator over the ensemble of all possible binary expansions for all symbols, i.e, of the generalized RS ensemble (GRSE). Recently, El-Khamy and McEliece [56] got exactly the same bit-level weight enumerator expression by assuming that each the bit-level Hamming weight of each symbol in the RS codeword is binomially distributed. The overall weight enumerator polynomial can then be expressed as:

$$G(x) = \sum_{i=0}^{Nm} G_i x^i = \sum_{w=0}^{N} R_w \left[\sum_{j=1}^{m} \frac{\binom{m}{j} x^j}{2^m - 1} \right]^w$$
$$= \sum_{w=0}^{N} \frac{R_w}{N^w} ((x+1)^m - 1)^w \tag{2.7}$$

Expanding $G(x)$, we can get:

$$G_i = \sum_{w=0}^{N} \frac{R_w}{N^w} \sum_{j=0}^{w} (-1)^{w-j} \binom{w}{j} \binom{jm}{i}$$
$$= \sum_{w=d_{min}}^{min(N,i)} \frac{R_w}{N^w} \sum_{j=0}^{w} (-1)^{w-j} \binom{w}{j} \binom{jm}{i} \tag{2.8}$$

The second equation holds by the observation that there is no non-zero codeword of symbol-level weight $w \leq d_{min}$ and there are no codewords of symbol-level weight $w > i$ contributing to the enumerator of bit weight i. Each code in the GRSE inherits all the properties of RS codes, e.g., designed minimum distance, symbol-level MDS property, etc.. Therefore, it is reasonable to evaluate the performance of the GRSE, which gives some idea about the performance of a specific RS code under MLD.

Here, we use standard bounding techniques to study the performance of the GRSE under MLD. Define the normalized weight $\delta = i/n$, the normalized weight enumerator $r(\delta) \triangleq (\ln G_i)/n$ and the normalized SNR per coded bit as $\rho = R\,\frac{E_b}{N_0}$. We have the union-type bound on the frame error rate (FER) as

$$FER \leq \sum_{i=d_{min}}^{n} e^{-nE(\rho,\delta)} \tag{2.9}$$

where the exponent term $E(\rho, \delta)$ depends on the specific bound chosen. For the standard union Chernoff bound, we have the exponent:

$$E(\rho, \delta) = -r(\delta) + \delta\rho \tag{2.10}$$

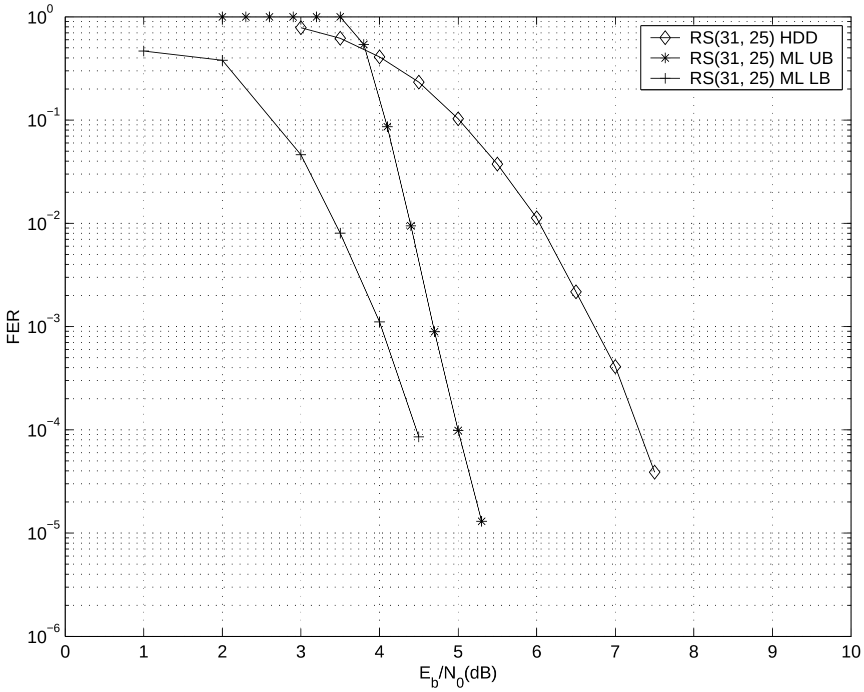


Fig. 1. Upper and Lower Bound on the Performance of an RS(31, 25) Code under MLD

In this dissertation, we choose a modified union-type bound derived by Divsalar [57], which is tight in the low SNR region, where standard union bound is loose. For this bound, the exponent is:

$$E(\rho,\delta) = -\gamma(\delta) + \frac{1}{2}\ln[\beta + (1-\beta)e^{2\gamma(\delta)}] + \frac{\delta\beta}{1-\delta(1-\beta)}\rho \tag{2.11}$$

where β is:

$$\beta = \sqrt{\rho\frac{1-\delta}{\delta}\frac{2}{1-e^{-2\gamma(\delta)}} + (\frac{1-\delta}{\delta})^2[(1+\rho)^2-1]} - \frac{1-\delta}{\delta}(1+\rho) \tag{2.12}$$

Since the above upper bound is an average over the ensemble of binary expansions, it is not clear *a priori*, how tight it is in predicting the performance of a specific RS code. Therefore, we also compute a simulation based lower bound for medium length codes. The lower bound is obtained using the following procedure: First, we run a soft decision decoder (e.g. some of the proposed decoding algorithms discussed in the following chapters). When the received vector is closer to the estimated codeword than to the transmitted codeword, an error is counted for the ML decoder. Otherwise, we assume that the ML decoder does not make an error. This provides a lower bound always, the tightness of which depends on how good the decoder is. The upper and lower bounds of an RS(31,25) code under MLD along with the performance of HDD are shown in Figure 1. It can be seen that the ML upper bound of the GRSE is tight in the high SNR region, which is 0.5dB away from the simulation-based ML lower bound of a specific RS code (expanded using a fixed basis) at an FER $= 10^{-4}$. Besides, the HDD performance is 2-3dB away from the optimal performance under ML decoding, which is consistent with the commonly believed potential performance gain using SDD over HDD. Hence, we will use the GRSE upper bound under the MLD as a performance benchmark for long RS codes where simulation based lower bounds are difficult to obtain.

We first investigate the performance of a widely used high rate code, i.e., RS (255,239). In Figure 2, we plot the performance of the upper bound over GRSE under MLD, HDD with error correction radius $t = (d_{min} - 1)/2$ and a hypothetical "genie decoder", which can correct up to $t = (d_{min} - 1)$ symbol errors. We can see that, the HDD is asymptotically 3dB worse than the performance under MLD (the largest gap is about 4dB, which appears at around an $FER = 10^{-20}$). The "genie decoder" is optimal for asymptotically large SNRs. However, this happens only at very low FERs (say, at an FER $= 10^{-200}$), which is impractical for most of the applications. For practical SNRs, it has approximately 2dB loss compared with the performance under ML decoding.

We further investigate a medium rate code RS(255,127) $R = 0.498 \approx 0.5$ in

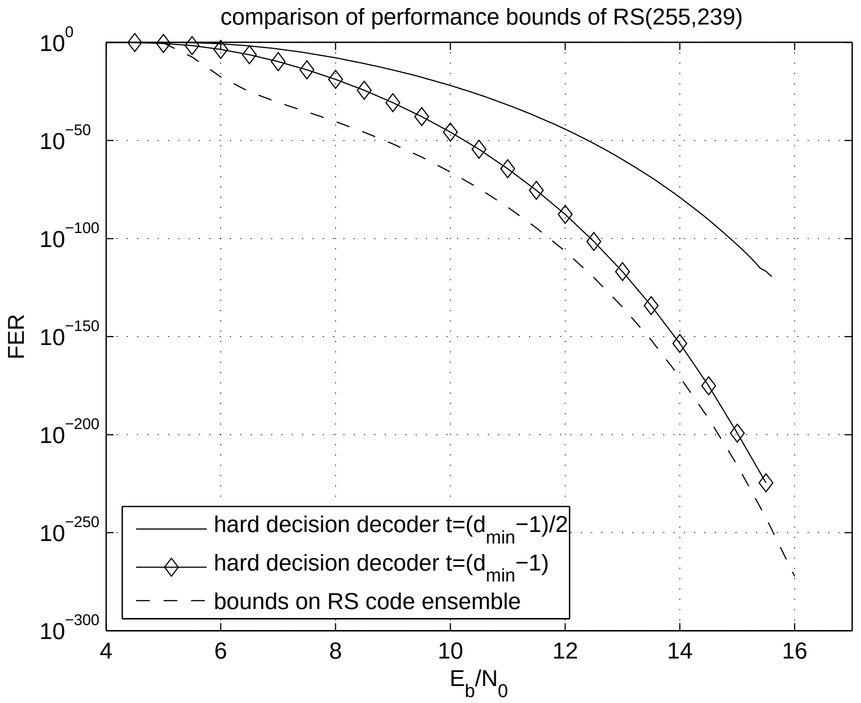


Fig. 2. Performance Comparison of an RS(255,239) Code under Different Decoders

Figure 3. We can see that the performance of the RS ensemble under MLD reaches an FER $= 10^{-4}$ at an $E_b/N_0 = 1.2$dB and outperforms the hypothetical decoder and HDD by 2.5dB and 5dB respectively. The performance of RS ensemble under MLD is only about 0.6 dB away from the sphere packing bound [58] (as can be seen from Figure 4), making it comparable to the best known turbo and LDPC codes. Note that for this code, all known decoders up to now are still away from the performance under MLD, making it difficult to obtain good simulation based lower bounds to estimate the MLD performance of the RS code.

The above examples reveal two important facts. First, the RS code itself is a good code, which can perform close to capacity under MLD as the codeword length goes long. Second, symbol-level bounded distance decoding does not fully exploit

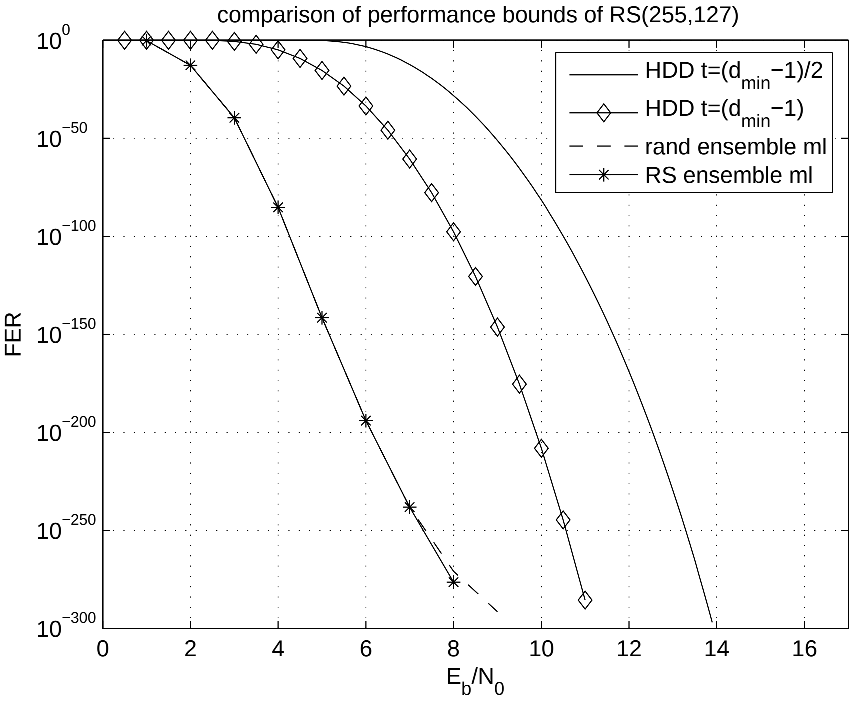


Fig. 3. Performance Comparison of an RS(255,127) Code under Different Decoders

the error correction capability of the code. Recently, there has been very significant developments in bounded distance decoding beyond half the minimum distance (see the seminal work [24][59]). However, the above example shows that even the "genie decoder", which decodes up to $t = d_{min} - 1$, performs far away from ML decoding. It suggests that the efficient RS decoder should be able to decode far beyond the minimum distance to take full advantage of the error correction capability of the code, since the number of low weight codewords can be very small. This motivates alternative design philosophies for RS soft-decision decoders, that is, decoding RS codes by taking advantage of bit-level soft information.

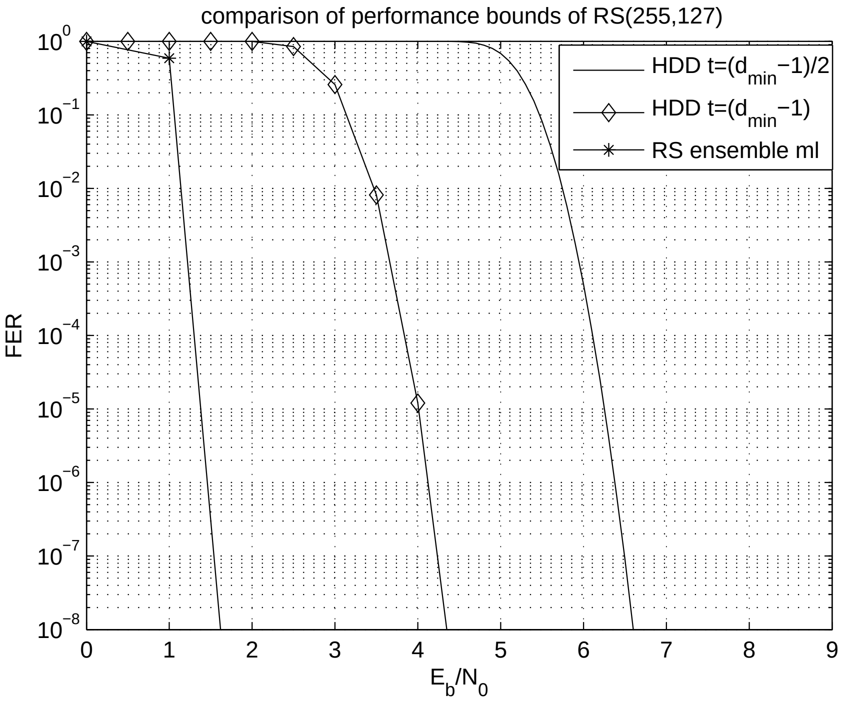


Fig. 4. Performance Comparison of an RS(255,127) Code under Different Decoders in the Practical SNR Region

CHAPTER III

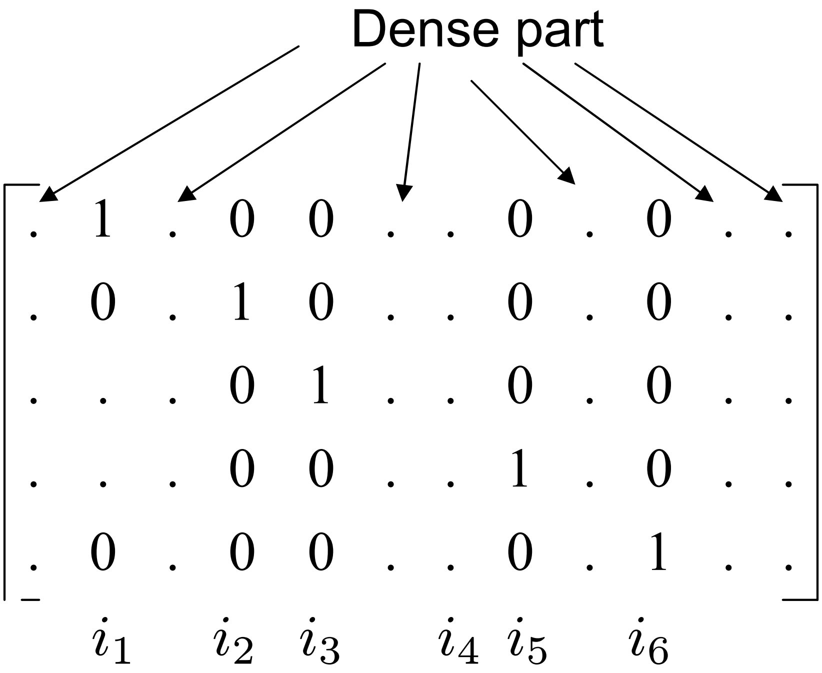
ITERATIVE SOFT-INPUT-SOFT-OUTPUT (SISO) DECODING OF HIGH DENSITY PARITY CHECK CODES BY ADAPTING THE PARITY CHECK MATRIX

It is commonly believed that BP decoding is not suitable for high density parity check (HDPC) codes [18], since iterative decoding can easily get stuck at pseudo-equilibrium points due to the large number of short cycles in the factor graph.

In this chapter, we present an SPA based SISO iterative decoding algorithm for HDPC codes. The main novelty in the proposed scheme is to adapt the parity check matrix at each iteration according to the bit reliabilities such that the unreliable bits correspond to a sparse submatrix and SPA is then applied to the adapted parity check matrix. The proposed algorithm can be geometrically interpreted as a two-stage gradient descent with an adaptive potential function. This adaptive procedure is crucial to the convergence behavior of the gradient descent algorithm since it prevents the gradient descent procedure from getting stuck at pseudo-equilibrium points and, hence, significantly improves the convergence behavior of the iterative decoder

In principle, the proposed algorithm can be applied to any linear block code; however, we restrict our attention to RS codes in this dissertation because of the practical interest in SDD of RS codes and the fact that the gain from this adaptive procedure is likely to be significant for codes with dense parity check matrices such as RS codes. Simulation results show that the proposed iterative decoding scheme performs well for RS codes with reasonable decoding complexity, even though their original parity check matrices are not sparse.

The rest of the chapter is organized as follows: The generic iterative decoding algorithm is presented in Section A. A geometric interpretation of the proposed



$$\begin{array}{cccccccccccc} . & 1 & . & 0 & 0 & . & . & 0 & . & 0 & . & . \\ . & 0 & . & 1 & 0 & . & . & 0 & . & 0 & . & . \\ . & . & . & 0 & 1 & . & . & 0 & . & 0 & . & . \\ . & . & . & 0 & 0 & . & . & 1 & . & 0 & . & . \\ . & 0 & . & 0 & 0 & . & . & 0 & . & 1 & . & . \\ & i_1 & & i_2 & i_3 & & i_4 & i_5 & & i_6 & & \end{array}$$

Fig. 5. Form of the Parity Check Matrix Suitable for Iterative Decoding Obtained through Row Operations

algorithm is given in Section B. Several variations of the generic algorithm are investigated in Section C. In Section D, simulation results of the proposed algorithm are presented and compared with popular RS soft decoding algorithms. Discussions and conclusions are presented in Section E.

A. Iterative Decoding Algorithm by Adapting the Parity Check Matrix

We will use underlined letters to denote vectors and bold face letters to denote matrices. Let $\underline{c} = [c_1, c_2, \ldots, c_n]$ be the binary representation of an RS codeword. In the description of the generic algorithm, we first assume that the bits are modulated using BPSK (with 0 mapped to +1 and 1 mapped to −1) and transmitted over an AWGN channel (extension to other channels is straightforward). The received vector is given by

$$\underline{y} = (-2\underline{c} + 1) + \underline{n}, \tag{3.1}$$

Thus, the initial reliability of each bit in the received vector can be expressed in terms of the log-likelihood ratios (LLR) as observed from the channel:

$$L^{(0)}(c_i) = \log \frac{P(c_i = 0|y_i)}{P(c_i = 1|y_i)}, \tag{3.2}$$

The proposed algorithm is composed of two stages: the matrix updating stage and the bit-reliability updating stage. In the matrix updating stage, the magnitude of the received LLRs $|L(c_i)|$ are first sorted and let $i_1, i_2, \ldots, i_{N-K}, \ldots, i_n$ denote the position of the bits in terms of ascending order of $|L(c_i)|$, i.e., the bit c_{i_1} is the least reliable and c_{i_n} is the most reliable. We begin with the original parity check matrix $\mathbf{H}_b$ and first reduce the i_1^{th} column of $\mathbf{H}_b$ to a form $[1\ 0 \ldots 0]^T$. Then, we try to reduce the i_2^{th} column of $\mathbf{H}_b$ to a form $[0\ 1\ 0 \ldots 0]^T$ and so on. We can be guaranteed to proceed until the $i_{(N-K)}^{th}$ column, since there are at least $(N-K)$ independent columns in $\mathbf{H}_b$. Then we try to reduce the i_{N-K+1}^{th} to $[\underbrace{0 \ldots 0}_{(N-K)} 1, 0, \ldots, 0]^T$. However, there is no guarantee we can do this. If we are unable to do so, we will leave that particular column and try to reduce $i_{(N-K+2)}^{th}$ column to the above form and so on. Finally, we can reduce $(n-k)$ columns among the n columns of $\mathbf{H}_b$ to be the identity matrix, since the matrix is $(n-k) \times n$ and is full rank. The matrix is thus reduced to a form as shown in Fig. 5. We denote the set of unreliable bits corresponding to the sparse submatrix as $\underline{B}_L$.

The proposed algorithm is iterative and during the j^{th} iteration, we have a vector of LLRs as:

$$\underline{L}^{(j)} = [L^{(j)}(c_1), L^{(j)}(c_2), \cdots, L^{(j)}(c_n)] \tag{3.3}$$

where initially $\underline{L}^{(0)}$ is determined from the channel output. Then, the parity check matrix is reduced to a desired form based on $\underline{L}^{(j)}$:

$$\mathbf{H}_b^{(j)} = \phi(\mathbf{H}_b, |\underline{L}^{(j)}|). \tag{3.4}$$

Henceforth, in the bit-reliability updating stage, the extrinsic LLR vector $\underline{L}_{ext}^{(j)}$ is first generated according to $\underline{L}^{(j)}$ using SPA [17] based on the adapted parity check matrix $\mathbf{H}_b^{(j)}$:

$$\underline{L}_{ext}^{(j)} = \psi(\mathbf{H}_b^{(j)}, \underline{L}^{(j)}) \tag{3.5}$$

That is for each bit, the extrinsic LLR is updated according to:

$$L_{ext}^{(j)}(c_i) = \sum_{\substack{l=1 \\ H_{li}=1}}^{n-k} 2\tanh^{-1}\left(\prod_{\substack{p=1 \\ p\neq i, H_{lp}=1}}^{n} \tanh\left(\frac{L^{(j)}(c_p)}{2}\right)\right) \tag{3.6}$$

The bit-reliability is then updated as:

$$\underline{L}^{(j+1)} = \underline{L}^{(j)} + \alpha \underline{L}_{ext}^{(j)} \tag{3.7}$$

where $0 < \alpha \leq 1$ is a damping coefficient. This is continued until a predetermined number of times $j_{max} = N_1$ or until all the parity checks are satisfied. A detailed description of the algorithm is given in Algorithm 1.

The proposed adaptive algorithm is inspired by the OSD [43]. However, instead of reprocessing the most reliable basis (MRB), we adapt the parity check matrix according to the bit reliability. It can also be viewed as a generalization of the iterative

Algorithm 1 Iterative Decoding Algorithm by Adapting the Parity Check Matrix

```
Step1. Initialization: set $\alpha$, $j_{max} = N_1$, $j = 0$ and the coded bits LLR from the
    channel: $\underline{L}^{(0)} = \frac{2}{\sigma^2}\underline{y}$
Step2. Reliability based parity check matrix adaptation: $\mathbf{H}_b^{(j)} = \phi(\mathbf{H}_b, |\underline{L}^{(j)}|)$.
    a) Order the coded bits according to the absolute value of the LLRs $|\underline{L}^{(j)}|$ and
    record the ordering indices.
    b) Implement Gaussian elimination to systematize the $(n-k)$ unreliable positions
    which are independent in the parity check matrix. (The submatrix can
    also be made to be degree-2 connected, see Section C-1).
Step3. Generating extrinsic information: Apply SPA to generate the extrinsic LLR
    for each bit using the adapted parity check matrix $\mathbf{H}_b^{(j)}$:
    $\underline{L}_{ext}^{(j)} = \psi(\mathbf{H}_b^{(j)}, \underline{L}^{(j)})$ (according to (3.6)).
Step4. Bit-level reliabilities Update:
    $\underline{L}^{(j+1)} = \underline{L}^{(j)} + \alpha \underline{L}_{ext}^{(j)}$, where $0 < \alpha \leq 1$.
Step5. Hard decision: $\hat{c}_i = \begin{cases} 0, & L^{(j+1)}(c_i) > 0; \\ 1, & L^{(j+1)}(c_i) < 0. \end{cases}$
Step6. Termination criterion: If all the checks are satisfied, output the estimated
    bits; else if $j = j_{max}$, declare a decoding failure; otherwise set $j \leftarrow j+1$ and go
    to Step2 for another iteration.
```

a posterior probability (APP) decoding algorithm based on a set of minimum weight parity check vectors by Lucas *et al.* [18]. In [18], the iterative algorithm is interpreted as a gradient descent. The adaptive algorithm generalizes the idea of gradient descent and extends it to be a two-stage gradient descent algorithm with an adaptive potential function. The damping coefficient α serves as the step size in the gradient descent process to control the dynamics of convergence. In the following section, we look into the geometric interpretation of this algorithm.

B. Geometric Interpretation of the Proposed Algorithm

In this section, a geometric interpretation of the proposed algorithm as a two-stage optimization procedure is presented. The idea of using optimization methods, such as gradient descent, to solve decoding problems can be dated back to Farrell *et al.* [60]. The belief propagation (BP) based algorithms by Gallager [2] and Pearl [61] were also shown to be special cases of the gradient descent algorithm. More similar to the bit reliability updating algorithm in this dissertation, Lucas *et al.* [18] proposed the APP decoding algorithm using the minimum weight parity check sets. They also interpreted the proposed APP algorithm as a gradient descent. Here, we generalize Lucas' idea and interpret the proposed adaptive algorithm as a two-stage gradient descent algorithm with an adaptive potential function.

Define the operator $\nu : [-\infty, +\infty] \rightarrow [-1, 1]$ as a mapping from LLR domain to tanh domain:

$$\nu(L) = \tanh\left(\frac{L}{2}\right) = \frac{e^L - 1}{e^L + 1} \tag{3.8}$$

where the mapping is one-to-one and onto.

It is immediate that the inverse operator $\nu^{-1} : [-1, +1] \rightarrow [-\infty, +\infty]$ can be expressed as:

$$\nu^{-1}(t) = \ln\left(\frac{1+t}{1-t}\right), \qquad t \in [-1, +1] \tag{3.9}$$

We apply the one-to-one tanh transform on the LLRs and get the reliability measure of the received signal in the tanh domain as:

$$\underline{T} = [T_1, T_2, \cdots, T_n] = [\nu(L(c_1)), \cdots, \nu(L(c_n))] \tag{3.10}$$

As in [17], we can measure the reliability of a parity check node in tanh domain as:

$$\nu(L(s)) = \nu(L(c_1 \oplus c_2 \oplus \cdots \oplus c_l)) = \prod_{p=1}^{l} \nu(L(c_p)). \tag{3.11}$$

Following the concept of generalized weighted syndrome proposed by Lucas *et al.* ((20) in [18]), we define a cost function J, which characterizes the reliability of the received vector $\underline{T}$ with a particular parity check matrix $\mathbf{H}_b$.

Definition 1 *Define the potential function J as:*

$$J(\boldsymbol{H}_b, \underline{\mathrm{T}}) = -\sum_{i=1}^{(n-k)} \nu(L(s_i)) = -\sum_{i=1}^{(n-k)} \prod_{j=1, H_{ij}=1}^{n} T_j \tag{3.12}$$

where J is a function of both the parity check matrix $\mathbf{H}_b$ and the received soft information $\underline{T}$.

The operator ν maps the original n-dimensional unbounded real space into an n-dimensional cube (since the output of the tanh function is confined to [-1, 1]). The potential function J is minimized iff a valid codeword is reached, that is all the checks are satisfied and $|T_j| = 1$ for $j = 1, \cdots, n$, where $J_{min} = -(n-k)$. Besides, points with all $|T_j| = 1$ correspond to vertex of the n-dimensional cube. Therefore, valid codewords lie on the minimum potential vertices of the n-dimensional cube. The decoding problem can be interpreted as searching for the most probable minimum potential vertex given the initial point observed from the channel.

Note that the potential function J is minimized iff a valid codeword is reached. It is quite natural to apply the gradient descent algorithm to search for the minimum potential vertex, with the initial value $\underline{T}$ observed from the channel. Consider the gradient of J with respect to the received vector $\underline{T}$. From (3.12), it can be seen that:

$$\nabla J(\mathbf{H}_b, \underline{T}) = \left(\frac{\partial J(\mathbf{H}_b, \underline{T})}{\partial T_1}, \frac{\partial J(\mathbf{H}_b, \underline{T})}{\partial T_2}, \cdots, \frac{\partial J(\mathbf{H}_b, \underline{T})}{\partial T_n} \right) \tag{3.13}$$

where the component wise partial derivative with respect to T_i is given by:

$$\frac{\partial J(\mathbf{H}_b, \underline{T})}{\partial T_i} = - \sum_{\substack{l=1 \\ H_{li}=1}}^{(n-k)} \prod_{\substack{p=1 \\ p \neq i, H_{lp}=1}}^{n} T_p \tag{3.14}$$

Thus, the gradient descent updating rule can be written as:

$$\underline{T}^{(j+1)} \leftarrow \underline{T}^{(j)} - \alpha \nabla J(\mathbf{H}_b, \underline{T}^{(j)}) \tag{3.15}$$

where α is a damping coefficient as in standard gradient descent algorithms to control the step width.

Note that the reliabilities in tanh domain are confined to $T_i \in [-1, 1]$. However, the updating rule (3.15) does not guarantee this. Therefore, we use the following modified updating rule to guarantee the validity of the updated values:

$$T_i^{(j+1)} \leftarrow \nu \left(\nu^{-1} \left(T_i^{(j)} \right) - \alpha \left[- \sum_{H_{li}=1} \nu^{-1} \left(\prod_{p \neq i, H_{lp}=1} T_p^{(j)} \right) \right] \right) \tag{3.16}$$

where $\nu^{-1}(x) = 2 \tanh^{-1}(x)$. Recall that the above non-linear updating rule is exactly the same as the Step 3-Step 4 in Algorithm 1.

When iterative decoding is applied to an HDPC code, with very high probability, the iterative algorithm will reach some local minimum points where $\nabla J(\mathbf{H}_b, \underline{T})$ is zero or is close to zero (since a few unreliable symbols will render the components of $\nabla J(\mathbf{H}_b, \underline{T})$ to be small or close to zero). We refer to these as pseudo-equilibrium points. Since gradient descent gets stuck at these points, while these points do not correspond to valid codewords.

From (3.12), we observe that since J is also a function of $\mathbf{H}_b$, different choices of the parity check matrices $\mathbf{H}_b$ (though span the same dual space), results in different potential functions J. More importantly, each $\mathbf{H}_b$ results in a different gradient $\nabla J(\mathbf{H}_b, \underline{T})$. The proposed algorithm exploits this fact and when a pseudo equilibrium point is reached, by adapting the parity check matrix based on the bit reliabilities, we switch to another $\mathbf{H}_b$ such that it allows the update in (3.16) to proceed rather than getting stuck at the pseudo-equilibrium point. However, note that the potential function that we want to minimize does not involve the Euclidean distance between the received codeword and current estimate at all. That is, the adaptive algorithm attempts merely to find a codeword that satisfies all the parity checks, without really enforcing that it be the one at minimum distance from the received word. Since small steps are taken in the gradient descent, very often we converge to the codeword at small distance from the received vector as well. However, there is no guarantee of convergence.

We use the following examples to show the operation of the adaptive algorithm and its difference from directly applying iterative decoding to an HDPC code.

Example 1: Consider codewords transmitted through a binary erasure channel (BEC). We first apply the gradient descent algorithm directly to HDPC codes (assume that each entry in the parity check matrix is i.i.d. with 0 or 1 equal probable) without reliability based adaptation. Assume that the erasure fraction is ϵ, therefore the number of erased bits is $n\epsilon$. Consider a particular parity check vector, any code bit will participate in that check with probability 1/2 (according to the i.i.d. equal probable assumption). A check is not erased iff all the participated bits are not erased. Therefore, the probability that a check is erased is

$$Prob\{L(s) = 0\} \doteq 1 - (\frac{1}{2})^{n\epsilon}, n \rightarrow \infty \tag{3.17}$$

Assume that we have r parity check vectors that the i^{th} bit participates in. The gradient with respect to the i^{th} coordinate becomes zero (i.e., $\frac{\partial J(\mathbf{H}_b, \underline{T})}{\partial T_i} = 0$) iff the extrinsic LLRs from all the checks it participates are erased. The probability of zero gradient in the i^{th} bit is:

$$\begin{aligned} Prob\{\frac{\partial J(\mathbf{H}_b, \underline{T})}{\partial T_i} = 0\} &\doteq [1 - 2^{-(n\epsilon - 1)}]^r \\ &\doteq 1 - r2^{-(n\epsilon - 1)} + o(2^{-(n\epsilon - 1)}) \\ &\doteq 1 - r2^{-(n\epsilon - 1)} \rightarrow 1, n \rightarrow \infty \end{aligned} \tag{3.18}$$

which suggests that unless the number of parity checks grows exponentially with $n\epsilon$, iterative decoding quickly gets stuck at a pseudo-equilibrium point.

On the other hand, for BEC, it is known that by adapting the parity check matrix corresponding to less reliable bits (i.e. the erased bits), ML decoding performance can be achieved [62] in one iteration. Notice that Gaussian elimination cannot proceed iff some of the columns corresponding to the erased bits are dependent. In this case, there is ambiguity between two or more valid codewords. In such a case, the ML decoder also fails.

In conclusion, for BEC, gradient descent without adaptation tends to get stuck at a pseudo-equilibrium point, while the reliability based adaptation will help gradient descent to converge to the ML solution in one iteration.

Example 2: The idea of reliability based parity check matrix adaptation can naturally be extended to AWGN channels and the insight remains the same. Though adapting the parity check matrix based on the channel output does not guarantee to converge to the ML decision for AWGN channels, it does avoid iterative decoding get-

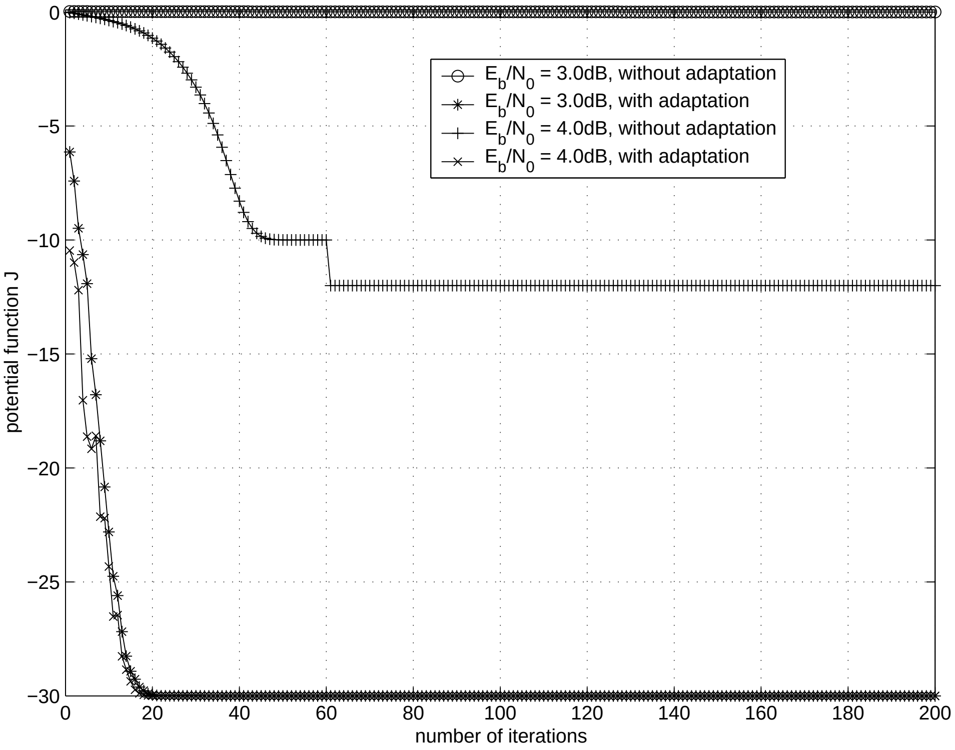


Fig. 6. Convergence Behavior of Iterative Decoding with and without Adaptation of an RS(31, 25) Code

ting stuck at pseudo-equilibrium points and thus improves the convergence behavior. We give a numerical example of the convergence behavior of iterative decoding of an RS(31,25) code in Figure 6. A typical realization of iterative decoding is simulated. The potential function J is plotted against the number of iterations. Since there are 30 parity checks for RS(31, 25), the minimum value of the potential function is $J = -30$ (corresponding to valid codewords). We can see that due to the high density of the parity check matrix of the RS code, iterative decoding without matrix adaptation (Algorithm 1 without Step 2) will easily get stuck at some pseudo-equilibrium. On the other hand, when the iterative algorithm is applied in conjunction with reliability based parity check matrix adaptation (Algorithm 1), the value of J quickly goes to the global minimum as the number of iteration increases. Consequently, reliability based parity check matrix adaptation improves the convergence behavior of iterative decoding significantly. We will show in Section D that the adaptive algorithm also largely improves the error performance.

C. Variations to the Generic Algorithms and Complexity Analysis

In this section, several variations of the proposed algorithm are discussed either to improve the performance or to reduce the decoding complexity.

1. Degree-2 Random Connection

One problem with the proposed approach is that each bit in the unreliable part $\underline{B}_L$ participates in only one check, it receives extrinsic information from one check only. If there is a bit error in the dense part participating in that check, the bit in $\underline{B}_L$ tends to be flipped and the decoder tends to converge to a wrong codeword. To overcome this drawback, we can reduce the matrix $\mathbf{H}_b$ to a form where the submatrix corresponding to the less reliable bits is sparse (say column weight 2 rather than 1). This can improve the performance since each less reliable bit now receives more extrinsic information while the submatrix corresponding to the unreliable bits still does not form any loops (i.e., unreliable bits themselves do not participate in any loops with each other). We can obtain this via a degree-2 random connection algorithm. The details are presented in Algorithm 2.

After the Deg-2 random connection, all the (n-k-1) positions in the parity check matrix are of degree-2 except the p_1^{th} column. The last column p_1 can be connected to some row or just left to be degree-1, which will not significantly affect the performance. This appears to improve the performance of the proposed algorithm especially in the

Algorithm 2 Deg-2 Random Connection Algorithm

```
Step1. Apply Gaussian elimination to the parity check matrix and obtain an identity
    matrix in the unreliable part.
Step2. Generate a random permutation of numbers from 1 to n-k.
    Record all the indices, i.e., $p_1, p_2, p_3, \cdots, p_{n-k}$.
Step3. Adapt the parity check matrix according to the follow rule:
    Add $p_{i+1}^{th}$ row to $p_i^{th}$ row, for i = 1 to n-k-1.
```

high SNRs.

2. Various Groupings of Unreliable Bits

Another variation that can help to further improve the performance is to run the proposed algorithm several times each time with the same initial LLRs from the channel but a different grouping of the less reliable bits. It is possible that some bits with $|L(c_j)|$ close to those in the unreliable set $\underline{B}_L$ are also of the wrong sign and vice-versa. Hence, we can run the proposed algorithm several times each time with a different grouping of the less reliable bits. That is, we can swap some bits in the reliable part with those in the unreliable part near the boundary and run the matrix adaptation all over again, which gives a new $\mathbf{H}_b$. We then run the proposed algorithm on that new matrix $\mathbf{H}_b$. Each time the proposed algorithm is run, a different estimate of codeword may be obtained due to the difference in the parity check matrix $\mathbf{H}_b$. All the returned codewords are kept in a list and finally the one that minimizes Euclidean distance from the received vector is chosen. We will see from simulation results that this method can significantly improve the asymptotic performance, but also increases the worst case complexity. Similar techniques have been used in conjunction with OSD by Fossorier [63] and Wu [47]. The way of grouping reliable bits used here is similar to the grouping scheme by Wu [47]. We refer interested readers to [47] for a detailed description and asymptotic performance analysis.

3. Incorporated Hard Decision Decoding

A hard decision decoder can be used during each iteration in the proposed algorithm to improve the performance and accelerate decoding as well. Since the HDD may return a codeword which is different from the ML codeword, we do not stop the decoder once a codeword is returned by the HDD. Rather, we still iterate up to a maximum number of iterations to obtain all the codewords returned by HDD during each iteration and finally pick up the most likely codeword. This guarantees to perform no worse than the proposed algorithm or HDD. In practice, error detection schemes such as CRC or other test techniques as discussed in [64] can serve as a practical stopping criterion to reduce the average decoding complexity. Combining the adaptive scheme with other SIHO algorithms such as the KV algorithm have recently been investigated by [65].

4. Partial Reliable Bits Updating

The complexity in the iterative decoding part can be further reduced via "partial reliable bits updating" scheme.

The main floating-point operation complexity comes from the computation of the extrinsic information in the reliable part (where the submatrix is dense). However, in the adaptation of the parity check matrix, only some bits in the boundary will be switched from the reliable part to the unreliable part. Therefore, in the bit reliability updating stage, we only update the bits in the unreliable set $\underline{B}_L$ and some reliable bits with $|L(c_j)|$ close to those in the unreliable set $\underline{B}_L$. For example, at each iteration, we may update the i_1^{th}, $\cdots$, i_{n-k+M}^{th} LLRs rather than all of them (where i_1 through i_n are sorted in ascending reliability). The number of bits in the reliable part M can be adjusted to control the complexity.

On the other hand, in the computation of the tanh of each check, we can also make approximations to reduce the complexity. For instance, min-sum can be used instead of SPA in Step 3 Algorithm 1 [66]. Furthermore, since the bit reliabilities are first ordered, the minimum of the absolute value of the LLRs in the dense part of the parity check matrix is known. Thus, we can approximate the tanh of all the bits in the reliable part using the tanh of the minimum value. This modification can significantly reduce the floating point operation complexity while retaining most of the performance gain.

More sophisticated updating schemes can also reduce the complexity of matrix adaptation. El-Khamy and McEliece proposed a scheme that adapts the parity check matrix from previous ones, which reduces the overall complexity by 75% [67].

5. Symbol-level Adaptation

Gaussian elimination requires serial update of the rows and is difficult to parallelize. Here we propose an alternative algorithm that is parallelizable. The idea is to take advantage of the structure of RS codes and adapt the parity check matrix at the symbol level. Let $S_L = \{i_1, i_2, \ldots, i_{(N-K)}\}$ be a set of $(N-K)$ least reliable symbols (symbol-level reliability can be computed by taking the tanh product of bit-level reliabilities or taking the minimum of the bit-level reliabilities). In order to update the parity check matrix at the symbol level, we need to find a valid parity check matrix for which the submatrix corresponding to the symbols in S_L is an identity matrix. The detailed procedure is as follows: first, the submatrix corresponding to the symbols in S_L is filled with an $(N-K) \times (N-K)$ identity matrix and the rest of the matrix with unknowns (erasure). The key idea is that computing the unknown symbols in the parity check matrix is equivalent to finding $(N-K)$ valid codewords of the dual code which will be the rows of the parity check matrix for the original code. For the j^{th} row, the i_j^{th} entry is 1 and the $i_1^{th}, i_2^{th}, \ldots, i_{j-1}^{th}, i_{j+1}^{th}, \ldots, i_{N-K}^{th}$ entries are 0s and all other entries are erasures E (i.e., all the positions corresponding to the reliable symbols are erased). Since the dual code is an $(N, N-K)$ RS code with $d_{min} = K+1$ and there are exactly K erasures in each row, Forney's algorithm [15][8] can be used to compute the values in the erased positions. Each decoded codeword corresponds to one row in the original parity check matrix. By repeating this procedure for all $(N-K)$ rows, we can get a systematic parity check matrix over $GF(2^m)$, where the submatrix corresponding to unreliable symbols is the identity matrix. Using the binary expansion, we can then get the binary parity check matrix and thereafter apply the SPA using it. Unlike Gaussian elimination, each entry of the parity check matrix can be computed independently and, hence, the whole procedure can be parallelized. This provides a computationally efficient way to obtain a parity check matrix in the desired form for hardware implementation. Related concepts such as re-encoding have also been used to reduce the complexity of KV decoding (see [33]).

6. SISO Decoding for Turbo Equalization Systems

With slight modification, the generic iterative decoding algorithm can be embedded in a turbo equalization (TE) system. In a TE system, the inputs to the decoder are obtained from the frontend SISO equalizer rather than directly observing from the channel. After several decoding iterations, the decoder outputs the soft information and feeds it back to the equalizer. The proposed TE algorithm based on this SISO algorithm is given in Algorithm 3.

Algorithm 3 Turbo Equalization Based on the ADP-SISO Algorithm

Step1. Initialization: set maximum turbo rounds M_2, $M = 0$ and the received value as observed from the channel.

Step2. Apply the BCJR algorithm (or other equalization algorithms) to the received values and the *a priori* LLRs to the equalizer (the initial *a priori* LLRs are set to be 0) to generate the overall LLR for each coded bit.

Step3. Subtract the *a priori* LLRs from the overall LLRs to get extrinsic LLRs.

Step4. Interleave the extrinsic LLRs and feed them as *a priori* LLRs to the SISO decoder. Run the proposed ADP-SISO decoder for a predetermined M_1 iterations and generate extrinsic LLRs from the decoder.

Step5. De-interleave the extrinsic LLRs from the decoder, feed them as *a priori* LLRs back to the equalizer, set $M \leftarrow M + 1$ and go to **Step2** for another TE round until $M = M_2$.

Step6. Make a hard decision based on the overall LLR for each bit. Output the estimated bits;

7. Computational Complexity

The proposed algorithm has a reasonable computational complexity. The reliability ordering and indexing process can be realized using standard quick sort algorithm, such as “mergesort”, with a complexity of $o(n \log_2 n)$. The Gaussian elimination of the parity check matrix is achieved in $o(n(n-k)^2)$ binary operations for bit-level adaptation. Degree-2 random connection scheme may further increase the complexity with the order about $o(n(n-k))$, which is negligible compared with the complexity of Gaussian elimination. If the parity check matrix is adapted from the previous ones, the complexity can further be reduced. For symbol-level adaptation, Gaussian elimination can be replaced with Forney's algorithm for computing the erased values, the complexity is about $o(K(N-K))$, while it facilitates parallel implementation.

In the iterative decoding part, since the parity check matrix is of low density in the unreliable part, the complexity is mainly in the high dense part and is about $o(k(n-k))$ real addition during each iteration (assuming that the function $\log(\tanh(L/2))$ is tabulated). If partial updating scheme is adapted, the complexity can be further reduced to be approximately $o((n-k))$, which is linear in the codeword length n. In conclusion, the over-all computational complexity is in polynomial time with respect to either n or d_{min}. Running many iteration rounds and outer rounds expands the complexity (the complexity will be expanded $N_1 \times N_2$ times), however, we have seen from the simulation results that even very few iterations produces significant improvement over algebraic HDD.

D. Simulation Results

In this section, simulation results of the proposed iterative decoding algorithm and its variations for RS codes over various channel models are presented.

The following notations will be used in the legends. ADP(N_1,N_2) refers to the proposed adaptive decoding scheme. N_1 refers to the maximum number of iterations of iterative decoding. N_2 refers to the number of decoding rounds with different groupings of the unreliable bits (see Section C-2). ADP & HDD refers to the proposed algorithm incorporated with an HDD (see Section C-3). SYM ADP refers to the proposed algorithm with symbol-level adaptation (see Section C-5). RED(M) ADP refers to the reduced complexity partial updating schedule with M bits in the reliable part to be updated (see Section C-4). Unless otherwise indicated, all the simulations adopt Deg-2 random connection (see Section C-1) to improve the asymptotic performance. The damping coefficient α is also specified on the plots. TE(M_1, M_2) refers to turbo equalization performance of the proposed adaptive scheme with M_1 SISO decoding iteration and M_2 TE iterations. For comparison, the simulation-based ML lower bounds and analytical ML upper bounds are also plotted in some figures. The details for obtaining the ML lower bound is described in Section G of Chapter II and in [18] as well. The ML upper bound will be discussed in details in the following subsection.

To speed up simulation, a genie aided stopping criterion scheme has been used, i.e., the decoding procedure stops when ADP & HDD gives the correct codeword. This is mildly optimistic as can be seen from the following argument. Assume that there is no genie, then the actual decoder will run a fixed number of N_1 iterations and may return a list of codewords (since the HDD may generate different codewords at different iterations). The actual decoder will pick the most likely codeword from the list. Thus, if the transmitted codeword is the most likely one, the result of the actual decoder will be the same as that of the genie aided decoder. Only when the transmitted codeword is not the most likely codeword, i.e., when the ML decoder would have made errors, the result of the actual decoder may be different from the genie aided decoder and, hence, the genie aided decoder may be optimistic. To understand this better consider the following relationship:

$$\begin{aligned} Pr\{\textbf{actual error}\} &= Pr\{\textbf{actual error}, \textbf{ML decision is correct}\} \\ &+ Pr\{\textbf{actual error}, \textbf{ML decision is in error}\} \\ &= Pr\{\textbf{genie error}, \textbf{ML decision is correct}\} \\ &+ Pr\{\textbf{actual error}, \textbf{ML decision is in error}\} \\ &\leq Pr\{\textbf{genie error}\} + Pr\{\textbf{ML decision is in error}\} \qquad (3.19) \end{aligned}$$

Therefore, we have:

$$\begin{aligned} Pr\{\textbf{genie error}\} &\leq Pr\{\textbf{actual error}\} \\ &\leq Pr\{\textbf{genie error}\} + Pr\{\textbf{ML decision is in error}\} \qquad (3.20) \end{aligned}$$

Whenever $Pr\{\textbf{ML decision is in error}\}$ is small compared to $Pr\{\textbf{genie error}\}$, $Pr\{\textbf{genie error}\}$ gives a fairly accurate estimate of $Pr\{\textbf{actual error}\}$.

1. AWGN Channels

We first present results for the RS (31,25) code over the AWGN channel in Fig. 7. For this code, standard belief propagation (BP) decoding (either with or without the damping coefficient, not plotted in the figure) has little gain (within 0.5 dB from algebraic HDD) due to the large number of short cycles. However, the proposed ADP(20,1) & HDD provides a 2.3 dB gain over HDD and an 1.0 dB over Chase-GMD(3) at an FER = 10^{-4}. Using the grouping method, the proposed ADP(20,3) & HDD can approach the ML lower bound within 0.25dB at an FER = 10^{-4}. The reduced complexity version RED(20) ADP(20,1) incurs 0.2dB performance loss compared with the generic ADP and outperforms MS ADP by about 0.5dB at an FER = 3×10^{-5}. The ML upper bound over RS averaged ensemble is also plotted for com-

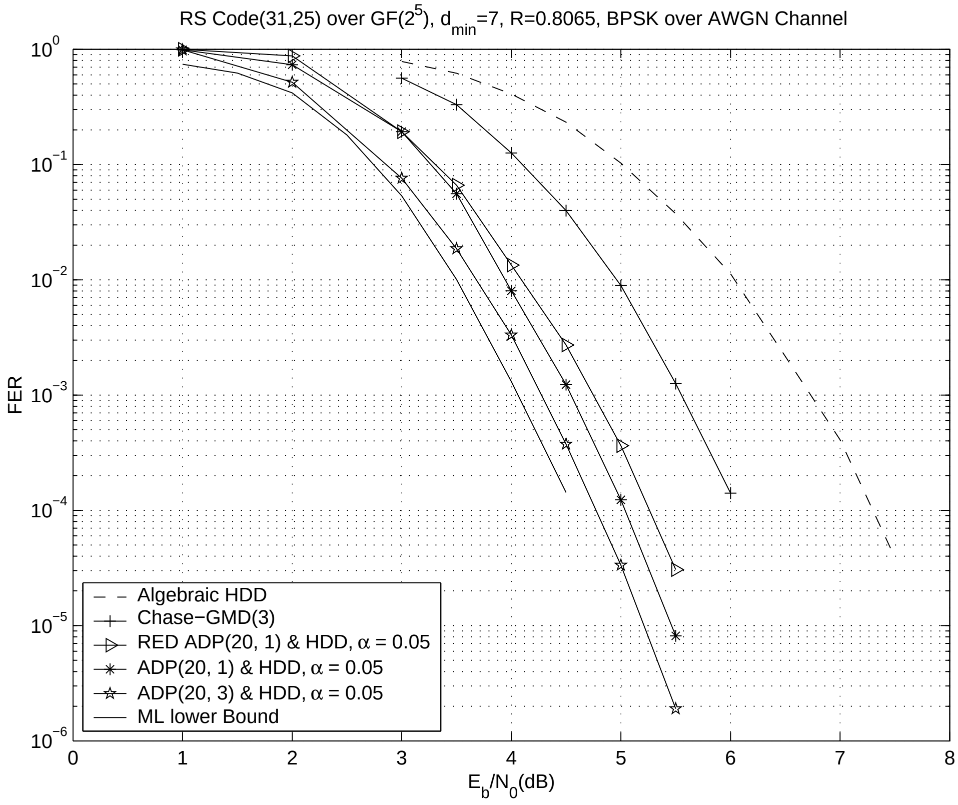


Fig. 7. Performance of an RS (31,25) Code over an AWGN Channel

parison. It can be seen that the ML upper bound is 0.5dB away from the ML lower bound at an FER $= 10^{-4}$ and these two bounds converge in the high SNR region.

Now we consider the (63,55) RS code. The performance is shown in Fig. 8. For this code, standard BP performs even worse than HDD (not plotted in the figure). However, the proposed algorithm ADP(5,1) & HDD provides 1.95 dB and 1.05 dB gain over algebraic HDD and Chase-GMD(3) at an FER $= 10^{-4}$. ADP(20,3) performs about 0.7 dB within the ML lower bound at an FER $= 10^{-4}$. It also provides another 0.3 dB gain over ADP(5,1). Similar to other gradient descent methods, the damping coefficient of the adaptive algorithm must be carefully chosen to control the updating

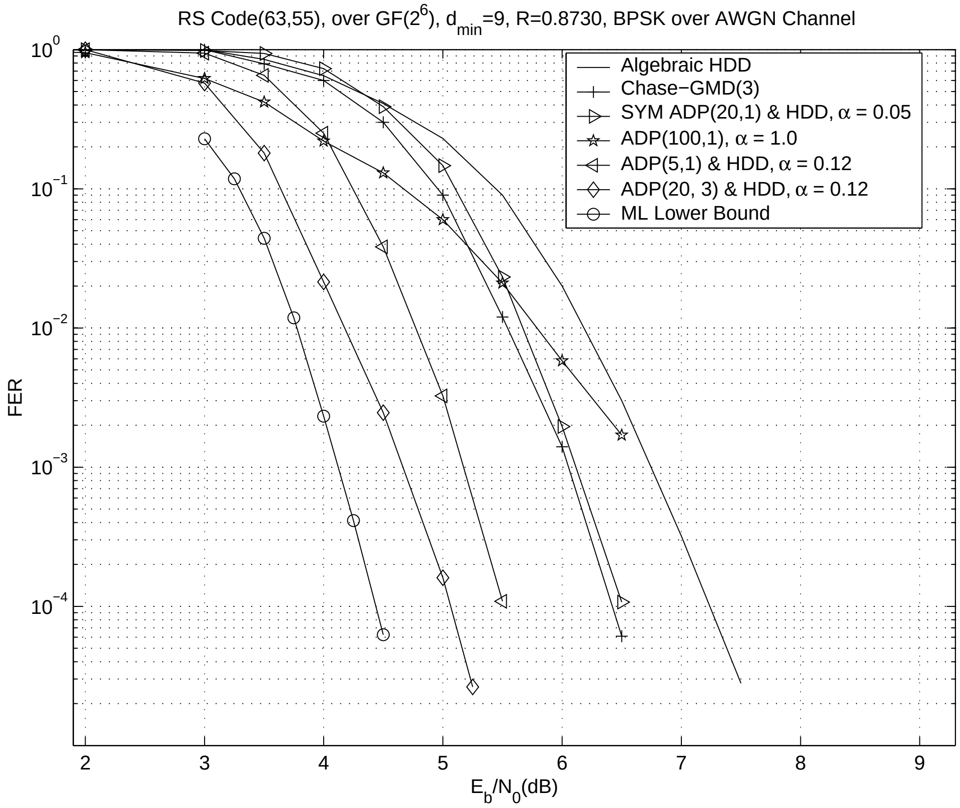


Fig. 8. Performance of an RS (63,55) Code over an AWGN Channel

step width. The performance curve of ADP(100,1) without damping or Deg-2 connection has a flat slope and the asymptotic gain diminishes, which is mainly due to the overshooting of the update scheduling such that the decoder tends to converge to a wrong codeword quickly. SYM ADP(20,1) & HDD also provides a non-trivial gain of about 0.7dB over HDD at an FER $= 10^{-4}$, which is comparable to Chase-GMD(3) while the complexity is significantly smaller.

Simulation results for the RS $(255, 239)$ code over the AWGN channel are shown in Fig. 9. When large complexity is tolerable, ADP(80, 50) & HDD outperforms the popular KV method proposed in [33] (with maximum multiplicity 100) by 1.0 dB

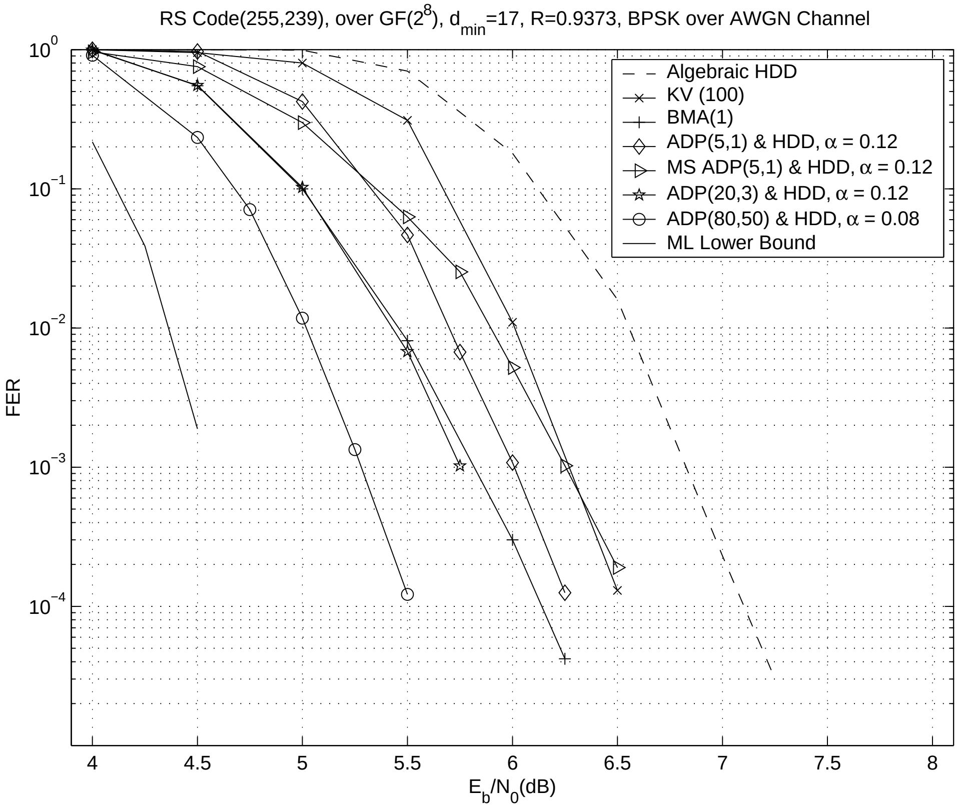


Fig. 9. Performance of an RS (255,239) Code over an AWGN Channel

and algebraic HDD by 1.65 dB respectively at an FER $= 10^{-4}$. We also compare this algorithm with BMA order-1 [45], ADP(80, 50) & HDD is also about 0.6 dB better than BMA (1) at an FER $= 10^{-4}$. Compared with the ML lower bound obtained by using a near ML decoding algorithm recently proposed in [68], the adaptive algorithm is still 0.6dB away from ML lower bound at an FER $= 10^{-3}$. With reasonable complexity, ADP(5,1) & HDD outperforms the KV(100) at an FER $= 10^{-4}$. Using the "min-sum" approximation, it will incur about 0.3dB loss at an FER $= 10^{-4}$. At the price of a slight increase in complexity, ADP(20,3) & HDD can provide comparable performance with BMA(1) at FER $= 10^{-4}$.

2. Rayleigh Fading Channels

Now we study the performance of the proposed iterative decoding of RS codes over Rayleigh fading channels. It is assumed that perfect channel state information is available at the receiver (CSIR). We first assume BPSK modulation where the coded bits are fully interleaved at the symbol-level, so that fading remains constant over one symbol but changes from symbol to symbol. The performance of an RS(31,15) code is shown in Fig. 10, the proposed algorithm ADP(40,1) & HDD outperforms algebraic HDD and GMD decoding by 6.5 dB and 3.3 dB respectively at an FER $= 10^{-4}$. ADP(40,3) & HDD can further improve the asymptotic performance. The performance of SYM ADP(40,1) & HDD is also plotted. We see that it also offers about 5 dB gain over HDD and 1.8 dB gain over GMD decoding respectively at an FER $= 10^{-4}$. Similar results are observed for long codes with rate $R = 0.5$; the performance of a shortened RS(128,64) over GF(256) is given in Fig. 11. The proposed decoding scheme provides several dB gain over HDD. This is a nontrivial gain considering the powerful burst error correction capability of HDD.

We also study the performance of RS coded modulation system over a symbol-level fully interleaved channel. We show the performance of a shortened RS(204,188) code with 256QAM modulation and gray mapping, which has similar settings as many existing standards, in Fig. 12. We can see from the figure that the proposed algorithm ADP(20,1) & HDD outperforms algebraic HDD by more than 7dB at an FER $= 10^{-3}$. Compared with KV decoder, there's also a 3 to 4dB gain. Though KV decoder takes the symbol-level soft information directly, its performance is mainly limited by the algebraic bounded distance decoding kernel.

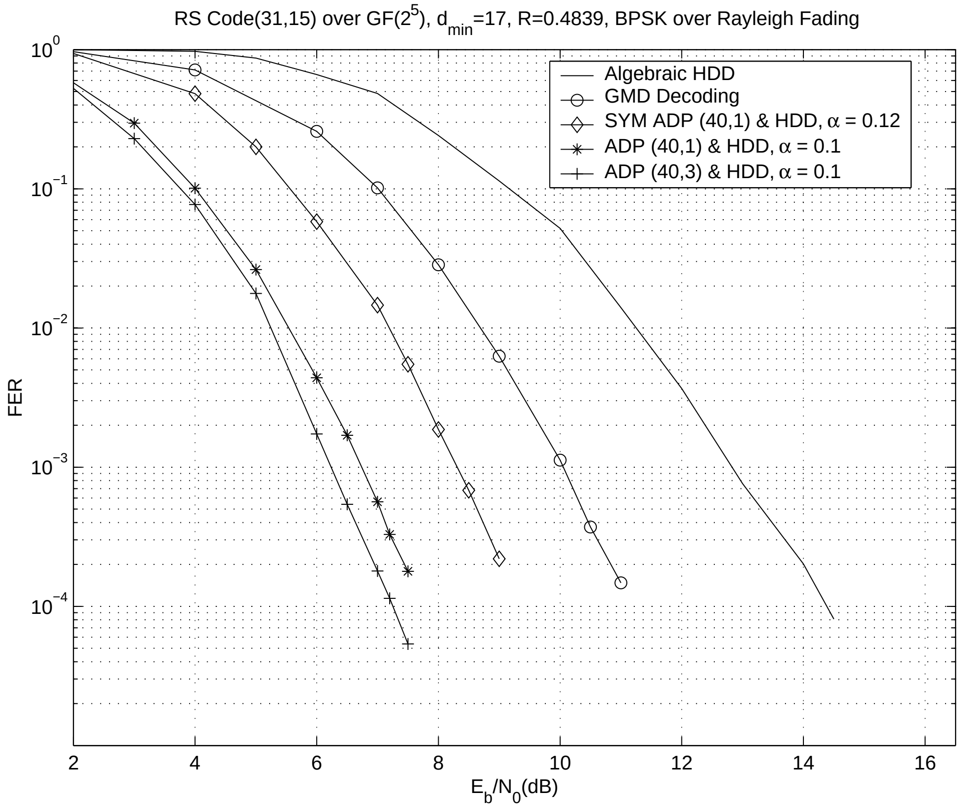


Fig. 10. Performance of an RS (31,15) Code over a Fully Interleaved Slow Fading Channel

3. Inter-Symbol Interference Channels

The TE performance of the proposed ADP-SISO algorithm over extended partial response-4 (EPR4) channels is also studied. The overall block length is approximately 20,000 bits and all the bits are fully interleaved at random such that incoming messages can be assumed to be independent for the outer RS code. More noticeable performance gain is observed for EPR4 channels. The performance of the RS(31,25) code under TE(5, 15) with $\alpha = 0.25$ ($\alpha = 0.2$ and $M_2 = 40$ is used for the last iteration) is shown in Figure 13. The performance curve shows a cliff region at $E_b/N_0 = 4.0dB$ and the performance in the high SNR region is almost identical to

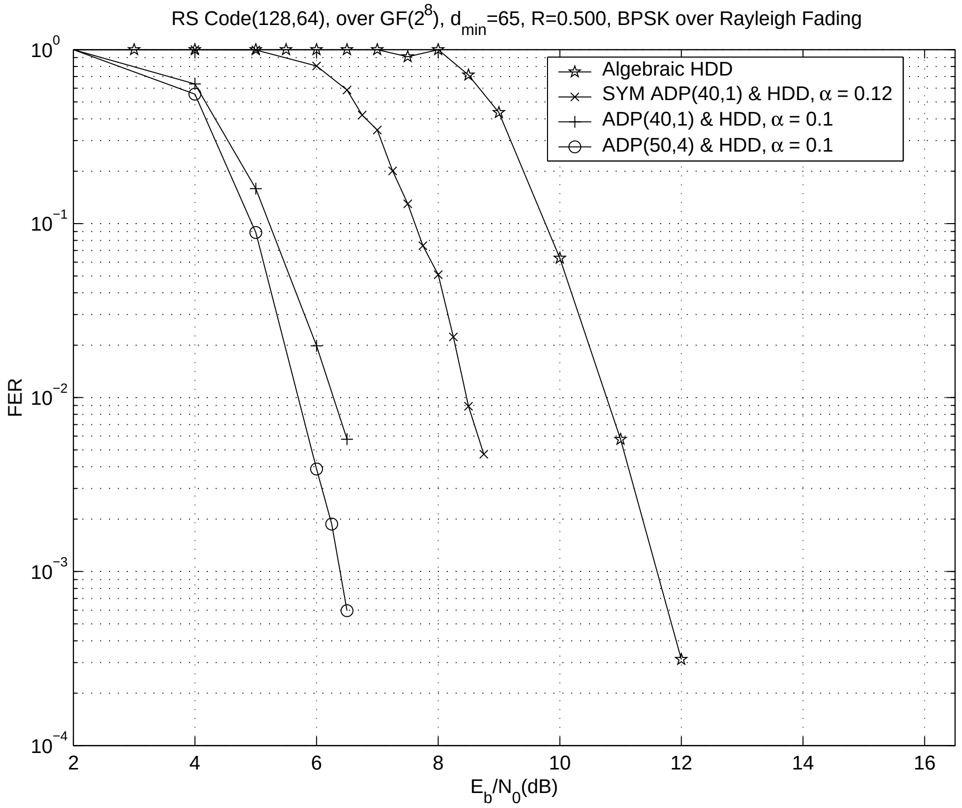


Fig. 11. Performance of an RS (128,64) Code over a Fully Interleaved Slow Fading Channel

the result of ADP(20,1) & HDD over the AWGN channel, which suggests that the proposed TE scheme mitigates the effect of ISI. It provides about 3.5dB gain over direct BCJR equalization followed by algebraic HDD and is within 0.5 dB away from ML lower bound over the AWGN channel at an FER $= 10^{-4}$. Similar results are also observed for long codes, RS(63,55) and RS(255,239) as shown in Figure 14. RS(63,55) under TE (5, 15) with $\alpha = 0.2$ ($M_2 = 40$ is used for the last iteration) also has a performance almost identical to that over the AWGN channel asymptotically. However, limited by the power of computer simulation, we are unable to simulate RS(255,239) under TE (5, 15) with $\alpha = 0.05$ ($\alpha = 0.2$ and $M_2 = 40$ is used for the last iteration)

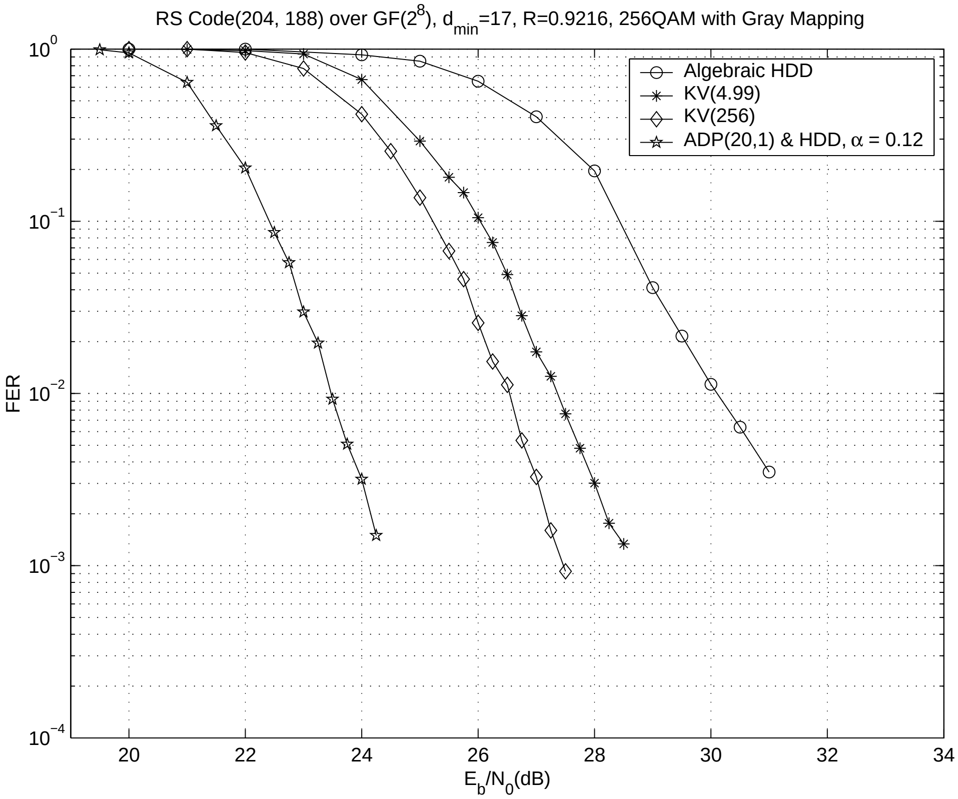


Fig. 12. Performance of an RS (204,188) Code Modulated Using 256-QAM over a Rayleigh Fading Channel

down to even lower FER to see whether it can converge to the performance curve over AWGN channels in the high SNRs. Whereas, in the waterfall region, TE(5,15) scheme already outperforms HDD about 1.7dB at an FER $= 10^{-3}$.

Moreover, the proposed adaptive iterative decoding algorithm also outperforms HDD for RS codes over practical perpendicular channels [69] even without interleaving and TE. As shown in Figure 15, over a perpendicular channel of normalized density $D_s = 2.5$ with or without jitter noise, however, the gain of ADP over HDD is less than observed as in the AWGN channel case. This suggests that the proposed bit-level iterative decoding algorithm is more effective for channels with memory when

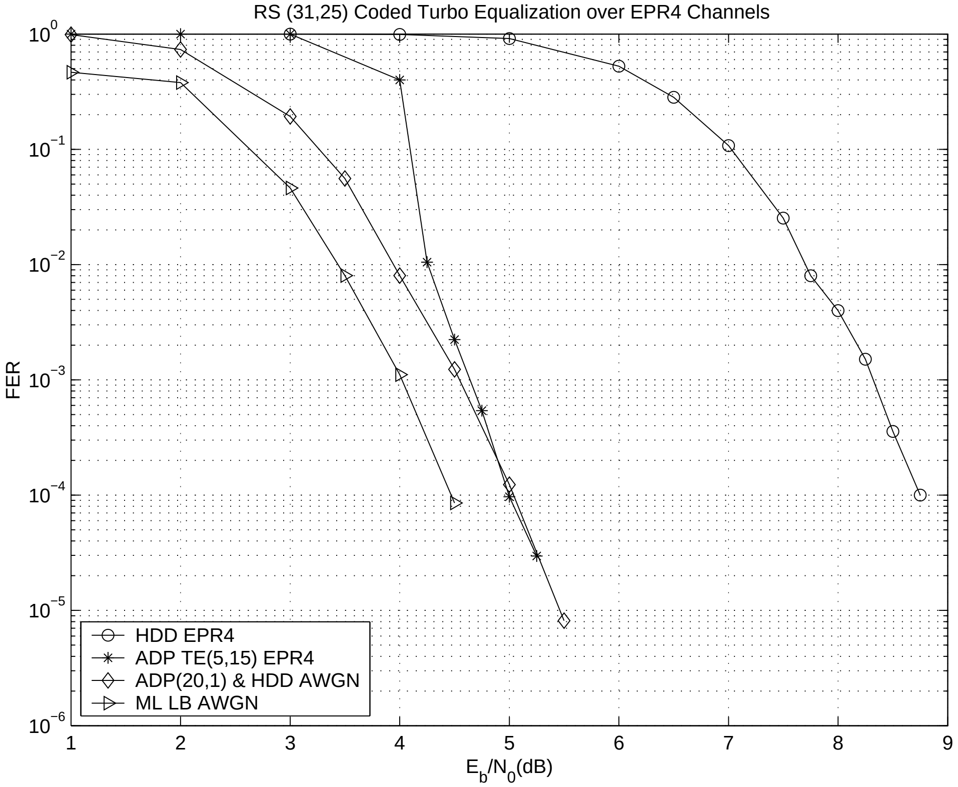


Fig. 13. TE Performance of an RS(31,25) Code over an EPR4 Channel

interleaving and iterative signal processing and decoding has been adopted.

E. Discussions and Conclusion

We first discuss some potential extensions of the adaptive algorithm. Firstly, the fraction of detectable error decreases as SNR increases as can be seen from Figure 16, i.e., the fraction of undetectable errors increases. The gain of the proposed scheme may be diminishing in the high SNRs for long codes as well. One observation of the failure mode of the proposed algorithm is that some errors considered to be “reliable” tend to pass a lot of wrong messages to the bits considered to be “unreliable” even though their actual LLR magnitudes are quite close, especially in the high SNR region.

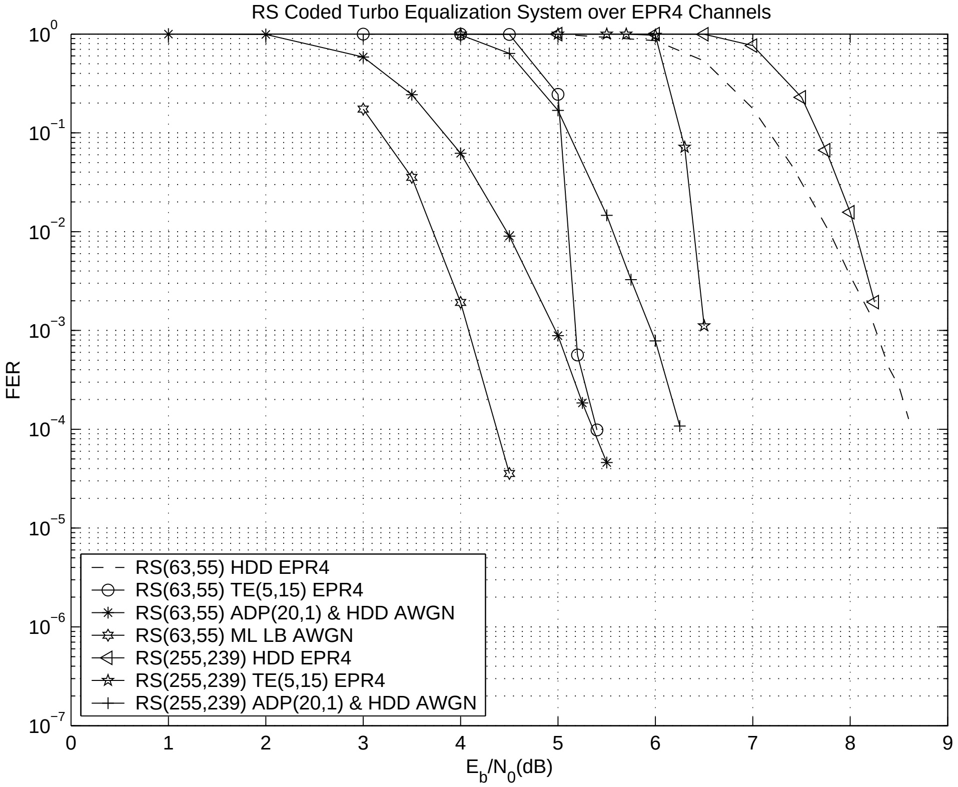


Fig. 14. TE Performance of an RS(63,55) Code and an RS(255,239) Code over an EPR4 Channel

One direction to further improve the decoding performance is to try to reduce the bit degree of "reliable" nodes in the boundary while slightly increase the density of the "unreliable" nodes so as to balance the variable node degrees between bits on the reliable/unreliable boundary. This will make part of the graph of the "reliable" bits be partially sparse and also increase the robustness of "unreliable" bits near the boundary.

Further improvement of the generic decoder without significantly increasing the complexity remains an challenging problem. It is favorable that the structure of the RS codes can be taken into account in conjunction with the adaptive algorithm.

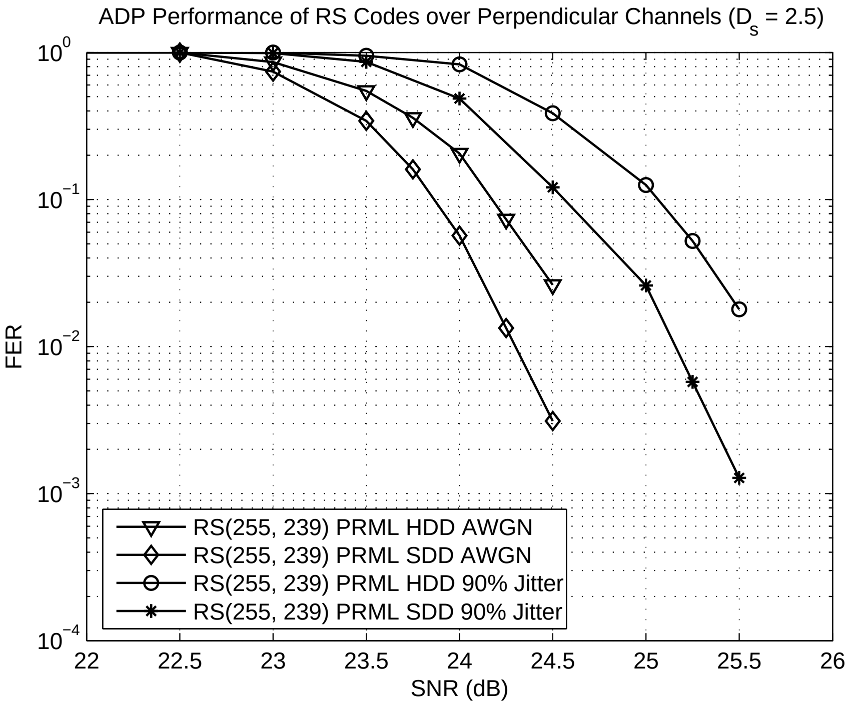


Fig. 15. Iterative Decoding Performance of an RS(255,239) Code over a Perpendicular Recording Channel ($D_s = 2.5$)

Therefore, Vardy and Be'ery's coset decomposition [52] seems to be a promising way to represent the $\mathbf{H}_b$ using a relatively sparse form. It is also natural to apply some more sophisticated decoding techniques (e.g. constructing some sub-trellis with reasonable complexity) and adopt the idea of the adaptive algorithm to improve the decoding performance. Secondly, from our simulation experience, when the channel has memory (say ISI channel or some FSK signaling), the performance gain of the adaptive algorithm (without iterative demodulation) diminishes. How to extend the adaptive scheme to detection and equalization such that they can generate good quality bit-level soft information is of interest. Thirdly, asymptotic performance analysis of the adaptive algorithm is also of practical value. Ahmed *et al.* [66] showed that

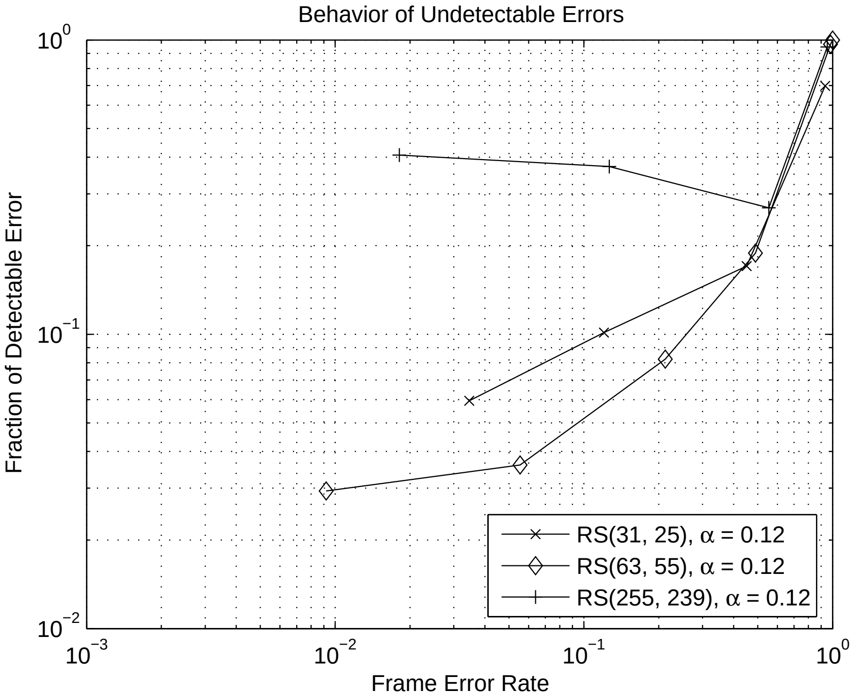


Fig. 16. Fraction of Detectable Errors as a Function of Frame Error Rate

using a certain probabilistic model, the performance of the adaptive algorithm under min-sum approximation (using min-sum rather than sum-product in Step 3 in Algorithm 1) can be evaluated using the OSD bound. However, the performance bounds for the exact scheme is still of interest.

In summary, we have presented a novel iterative SISO decoding algorithm for HDPC codes by adapting the parity check matrix. The proposed algorithm can be geometrically interpreted as a two-stage gradient descent algorithm with an adaptive potential function. Simulation results show that the proposed algorithm provides favorable performance gain for RS codes compared with known RS soft decoding methods over various channels for a wide range of code rate and code length. Besides, the proposed algorithm and its variations also provide flexible performance-complexity trade-off for different applications.

CHAPTER IV

ITERATIVE SOFT-DECISION DECODING OF CYCLIC CODES BASED ON STOCHASTIC SHIFT

In Chapter III, an iterative decoding scheme has been successfully applied to HDPC codes by reliability based parity check matrix adaptation at each decoding iteration. However, the parity check matrix adaptation procedure involves Gaussian elimination, which is undesirable for low complexity parallel implementation.

This chapter presents another iterative decoding method for RS codes. The proposed algorithm is a stochastic shifting based iterative decoding (SSID) algorithm which takes advantage of the cyclic structure of RS codes to prevent iterative decoding getting stuck. While the approach in Chapter III is a reliability based parity check matrix adaptation procedure, the adaptation in this chapter is restricted to be within the class of cyclic shifts of the parity check matrix. Consequently, a cyclic shift of the bit-level reliability values can be used to realize the desired adaptation, which is much less complex than the Gaussian elimination based adaptation as in Chapter III. It is shown that the performances of SSID is superior to many other popular soft decision methods for short RS codes. The generic SSID algorithm can naturally be extended to other class of cyclic codes, such as BCH codes as well.

The rest of the chapter is organized as follows: The iterative decoding algorithm is proposed in Section A. In Section B, simulation results of the proposed algorithm are presented. Conclusions are presented in Section C.

A. Proposed Iterative Decoding By Stochastic Shift

Suppose the coded bits are transmitted with BPSK modulation format (with 0 mapped to +1 and 1 mapped to −1, i.e., $\underline{x} = -2\underline{c} + 1$) over an AWGN channel,

$$\underline{y} = \underline{x} + \underline{n}, \tag{4.1}$$

Thus, the reliability of the received vector can be expressed in terms of their log-likelihood ratio (LLR) $\underline{L}$ (here we use underlined letters to denote vectors). The *a posteriori* LLR of each bit can be expressed as:

$$\underline{L}(x_i) = \log \frac{P(c_i = 0|\underline{y})}{P(c_i = 1|\underline{y})}, \tag{4.2}$$

Though the exact *a posteriori* LLR of each bit is difficult to obtain, for sparse graph codes, a good approximation can be obtained using the BP algorithm. By taking advantage of the cyclic property of RS codes, an SPA with a stochastic shifting schedule is proposed to prevent iterative decoding getting stuck. Let $\underline{L}^{(j)}$ denote the overall LLRs until the jth iteration. During the jth iteration, the SPA is used on the vector $\underline{L}^{(j)}$ to produce extrinsic information $\underline{L}^{(j)}_{ext}$. The LLR $\underline{L}^{(j+1)}$ is then updated according to:

$$\underline{L}^{(j+1)} = \underline{L}^{(j)} + \alpha \underline{L}^{(j)}_{ext}, \tag{4.3}$$

where $0 < \alpha \leq 1$ is a damping coefficient. The updated LLR $\underline{L}^{(j+1)}$ is cyclically shifted by θ symbols, where θ is a random integer uniformly distributed between $[0, N-1]$. Since RS codes are cyclic, the cyclically shifted version of $\underline{x}$ is a valid codeword. Hence, a shifted version of $\underline{L}^{(j+1)}$ can be thought of as the received signal when a shifted version of another valid codeword was transmitted. Therefore, another iteration of the SPA is performed with the shifted version of the LLR $\underline{L}^{(j+1)}$. Since the geometry of the factor graph with the shifted version is different from the previous ones, pseudo-equilibrium points can be avoided. We continue this procedure for a predetermined number of iterations or until all the parity checks are satisfied. When the maximum of j_{max} iterations is reached, another outer round, with a different realization of the random shifts and an increased α, begins with the original LLR from the channel.

Define $\psi(\underline{L})$ as an one iteration of the SPA algorithm function with the input LLR $\underline{L}$. Define $\underline{L}_\theta$ as a cyclic shift of the vector by θ symbols (Note that received symbols should be shifted at the symbol level). A detailed description of the algorithm is then given in Algorithm 4.

Kou *et al.* [70] also made use of the cyclic property of Geometry codes to construct redundant parity check matrix by cyclicly shifting parity check vectors, which is an exhaustive deterministic version of our method. Simulation results suggested that, the SSID based random updating scheme (RUS) outperforms the exhaustive parallel updating scheme (PUS). This is similar to the updating rules in a Hopfield network, where asynchronous and stochastic updating scheme outperforms synchronous updating scheme. The performance gain is believed to be mainly due to the stochastic shifting and multiple outer iteration rounds.

B. Experimental Results and Discussions

In this section, simulation results for decoding of RS codes based on the SSID algorithm are presented. The initial damping coefficient α_0 is selected to be 0.08 based on simulation.

Consider an RS(15,7) code and assume BPSK transmission over an AWGN channel. The performances of several updating schedules are shown in Fig. 17 along with the performance of the KV algorithm with a maximum multiplicty 4 taken from [33]. The updating schemes evaluated are: standard BP (300 iterations), RUS with a gradually changing damping coefficient (i.e., SSID), RUS with constant damping

Algorithm 4 SSID Algorithm for Cyclic Codes

```
Step1. Initialization: set $q = 0$, $j = 0$ and $\alpha_0$.
Step2. Set the coded bits LLR as observed from the channel: $L^{(0)}(x_i) = \frac{2}{\sigma^2} y_i$.
Step3. SPA: Feed the LLRs into the decoder and generate extrinsic LLRs for each bit using SPA:
    $\underline{L}_{ext}^{(j)} = \psi(\underline{L}^{(j)})$.
Step4. Parameter Update: Update the LLR of each bit:
    $\underline{L}^{(j+1)} = \underline{L}^{(j)} + \alpha \underline{L}_{ext}^{(j)}$.
    where $\alpha$ is a gradually increasing damping coefficient to control the updating step width.
Step5. Random Shifting: Cyclicly shift the LLRs by $\theta$ symbols and record the overall shift $\Theta$:
    $\underline{L}^{(j+1)} \leftarrow \underline{L}_{\theta}^{(j+1)}$.
Step6. Hard Decision: $\hat{c}_i = \begin{cases} 0, & L^{(j+1)}(x_i) > 0; \\ 1, & L^{(j+1)}(x_i) < 0. \end{cases}$
Step7. Termination Criterion: If all the checks are satisfied, stop iteration and go to Step9, else if $j = j_{max}$, go to Step8, otherwise set $j \leftarrow j + 1$ and go to Step3 for another SPA iteration.
Step8. Outer Round: If $q = q_{max}$, declare a decoding failure, otherwise set $q \leftarrow q + 1$ and $j = 0$, update the damping coefficient $\alpha = \alpha_0 + (q/(q_{max} - 1))(1 - \alpha_0)$ and go to Step2 for another outer round.
Step9. Extract Information Bits: Shift the decoded bits back to their original position and get the information bits from coded bits. $\underline{\hat{c}} = \underline{\hat{c}}_{(-\Theta)}$
```

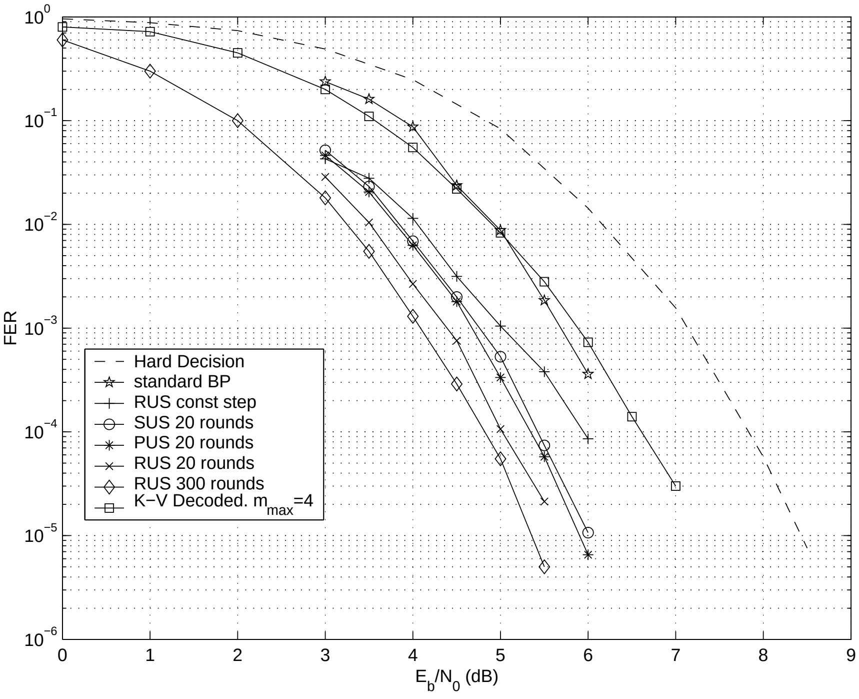


Fig. 17. Performance of an RS (15,7) Code over an AWGN Channel under Different Updating Schemes

coefficient, serial updating scheme (SUS), PUS with redundant checks. Note that all the above schedules set a maximum 30 SPA iterations and 20 outer rounds and another RUS (with 30 SPA iterations and 300 outer rounds) is proposed of the same complexity with the PUS scheme, which uses redundant checks.

We note that standard BP outperforms hard decision decoding by 1.4 dB at an FER of 10^{-3}. However, further improvement can be achieved by proper updating and scheduling. RUS with gradually increasing damping coefficient outperforms that with constant damping coefficient, since it keeps updating damping coefficient from being either too conservative or too aggressive. RUS outperforms both PUS and SUS with the same complexity by 0.5 and 0.3 dB respectively. This is due to the fact that RUS can reduce deterministic error patterns and therefore prevent iterative decoding from getting stuck. The best result can be achieved so far is RUS with 300 outer

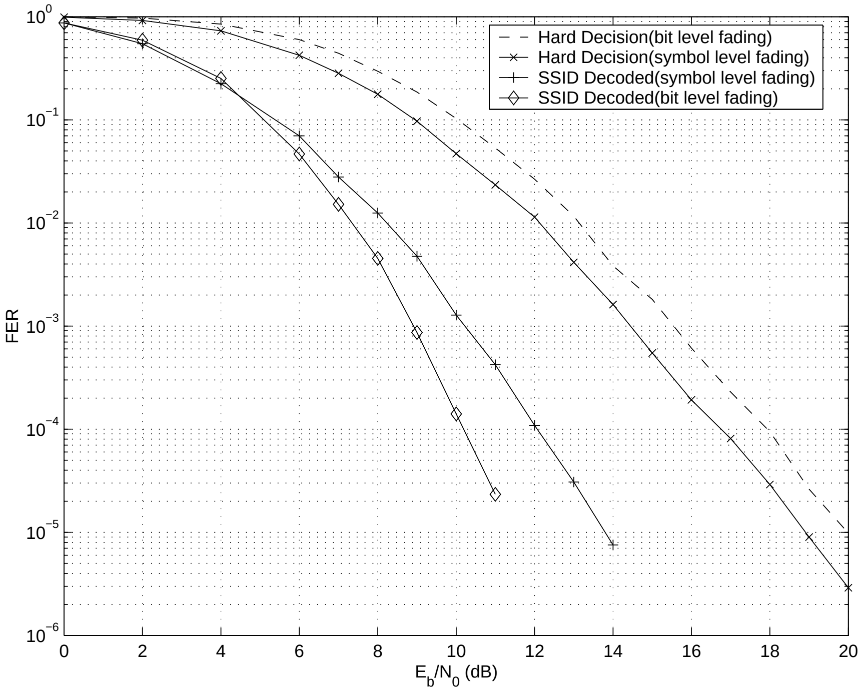


Fig. 18. Performance of an RS (15,7) Code over a Rayleigh Fading Channel Using SSID Decoding

rounds, which outperforms hard decision decoding by 3.1dB and the KV algorithm ($m_{max} = 4$) by about 2 dB at an FER of 10^{-5}.

An additional simulation of RS (15,7) is presented over a fully interleaved Rayleigh fading channel (the decoding scheme is proposed with 300 outer rounds and 30 SPA iterations). Fig. 18 suggests that the gain is even larger for fading channel, about 8.8dB for bit interleaving and about 5dB for symbol interleaving at an FER of 10^{-5}. This is mainly due to the poor performance of hard decision decoding over a fading channel.

We present results for the RS (31,25) code over an AWGN channel in Fig. 19. Several soft decision decoding methods are compared. For this code, standard BP algorithm has little gain due to the large number of short cycles. However, with SSID scheduling (with 200 outer rounds and 50 SPA iterations), the new method

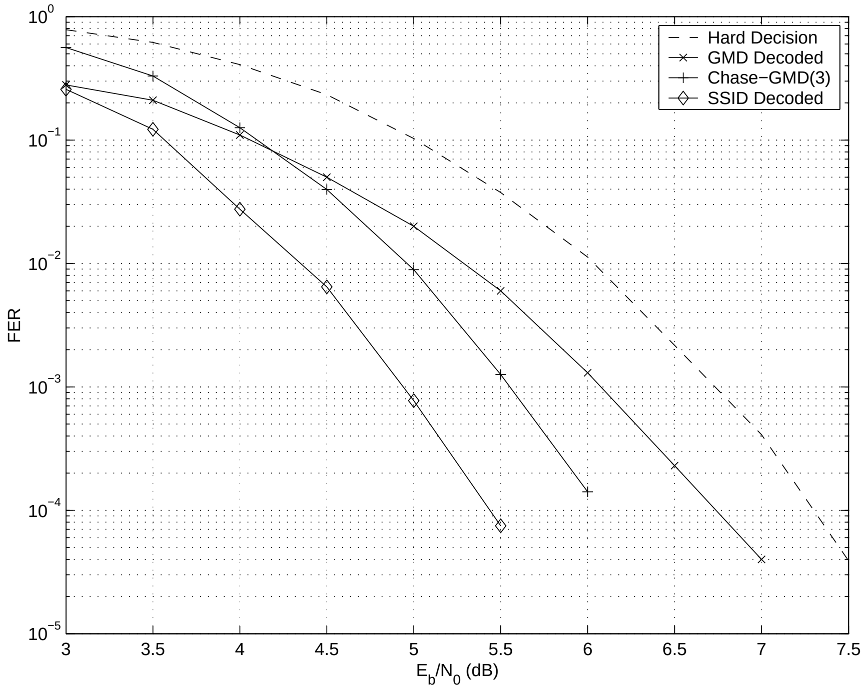


Fig. 19. Performance of an RS (31,25) Code over an AWGN Channel Using Different Decoding Schemes

outperforms Berlekamp & Massey (BM) decoding, Generalized Minimum Distance (GMD) decoding and combined chase & GMD decoding, by 1.9dB, 1.3dB and 0.63dB, respectively at an FER of 10^{-4}. As mentioned previously, the performance gain is due to the beyond bounded sphere decoding capability of the proposed algorithm.

Unfortunately, we notice that the soft decision gain of the new method still diminishes as the codeword length becomes long (for a (63,55) code, which is not shown here, the gain is only 0.6dB compared with hard decision at an FER of 10^{-3}). The reason for the performance degradation under the BP algorithm is mainly due to the fact that the parity check matrix has high density and iterative decoding gets stuck at equilibrium points regardless of the cyclic shift.

C. Conclusion

In this chapter, a stochastic shift based iterative decoding (SSID) algorithm for cyclic codes has been proposed. We have shown that a properly scheduled BP algorithm outperforms algebraic hard decision decoding and standard BP decoding for short RS codes. This iterative decoding method can be applied to other bit-level/symbol-level cyclic codes, such as BCH and Geometry codes, as well.

CHAPTER V

ALGEBRAIC SOFT-DECISION DECODING OF REED-SOLOMON CODES USING BIT-LEVEL SOFT INFORMATION

In this chapter, we study another advanced channel coding technique for RS codes, that is algebraic soft-decision (list) decoding (ASD). We propose a multiplicity assignment strategy (MAS) for ASD that provides a significant performance improvement over the BM algorithm even for high rate RS codes with a computational complexity that is practically affordable. In contrast to the popular view point that ASD is a symbol-level SDD scheme, we study the performance of ASD of RS codes using bit-level soft information. We show that carefully incorporating bit-level soft information in multiplicity assignment and interpolation is the key step to achieve most of the coding gain offered by ASD but still maintain a moderate complexity. Based on the analysis, a new SDD scheme is proposed for RS codes, which outperforms many existing ASD algorithms in the literature in terms of both performance and computational complexity.

The rest of this chapter is organized as follows: After a brief review of the background materials of ASD in Section A, we investigate the optimal MAS for ASD over erasure channels and binary symmetric channels (BSC) with infinite cost in Sections B and C. The corresponding decoding region of ASD is characterized and performance bounds are derived. It is shown that for practical high rate RS codes: over binary erasure channels (BEC), ASD has a significant gain over conventional BM decoding and most of the coding gain comes from appropriate multiplicity assignment to bit-level erasures; over BSC's, the gain of ASD over GS decoding is large only for short length or low rate RS codes. In Section D, the analysis is generalized to a mixed error and bit-level erasure channel and a simple MAS is proposed. In the infinite cost case, the decoding region of the proposed MAS is shown to approach the outer bound of the optimal MAS for practical medium to high rate RS codes. In the finite cost case, the decoding region of the proposed MAS is also characterized for any given multiplicity M. By treating erasures at the bit-level, the proposed MAS has a significantly larger decoding region than that of conventional BM decoding and more recent GS algorithm. Based on insights obtained from the performance analysis, in Section E, we develop a sequential MAS called bit-level generalized minimum distance (BGMD) decoding, which successively erases the least reliable bits (LRB). In spite of its simplicity, BGMD algorithm provides a significant gain over conventional BM decoding and compares favorably with many existing MASes of ASD and other RS SDD schemes over various channels of practical interests. Moreover, due to its simple structure, the decoding performance of BGMD for practical high rate RS codes can be tightly bounded using a standard ordered statistics bounding technique. BGMD upper bound suggests a significant gain over BM decoding in the high SNR region, where the evaluation of the performance is beyond the capability of computer simulation but of significant practical value. Simulation results are presented in Section F and conclusion is drawn in Section G.

A. Algebraic Soft Decision Decoding of Reed-Solomon Codes

In this section, we review some background materials of ASD of RS codes that are relevant to our proposed scheme. Underlined letters will be used to denote vectors and bold face letters will be used to denote matrices throughout this chapter.

1. Algebraic Soft-Decision Decoding

Let $\mathcal{A}(X,Y) = \sum_{i=0}^{\infty}\sum_{j=0}^{\infty} a_{i,j}X^iY^j$ be a bivariate polynomial over GF(q) and let w_x, w_y be nonnegative real numbers. The (w_x, w_y)-weighted degree of $\mathcal{A}(X,Y)$ (denoted as $\mathtt{deg}_{w_x,w_y}(\mathcal{A})$) is defined as the maximum over all numbers $iw_x + jw_y$ such that $a_{i,j} \neq 0$. The $(1,1)$ degree is usually referred to as the degree of the polynomial $\mathcal{A}(X,Y)$ (denoted as $\mathtt{deg}(\mathcal{A})$). The bivariate polynomial $\mathcal{A}(X,Y)$ is said to pass through a point (α,β) with multiplicity m (or equivalently, the point (α,β) is said to be a zero of multiplicity m of the polynomial $\mathcal{A}(X,Y)$), if $\mathcal{A}(X+\alpha, Y+\beta)$ contains a monomial of degree m and does not contain any monomials of degree less than m.

Suppose an RS codeword $\underline{\mathcal{X}} = (\mathcal{X}_1, \mathcal{X}_2, \cdots, \mathcal{X}_N)$ is modulated and transmitted through a memoryless channel and the decoder observes $\underline{\mathcal{Y}} = (\mathcal{Y}_1, \mathcal{Y}_2, \cdots, \mathcal{Y}_N)$ as the channel output. Following the setting in [4] and [27], we assume that $\mathcal{X}_i$'s are independent and uniformly distributed over $GF(q)$. In ASD, the *a posteriori* probability (APP) is computed as:

$$\pi_{i,j} = Pr\left(\mathcal{X}_j = \alpha_i | \mathcal{Y}_j\right), 1 \leq i \leq q, 1 \leq j \leq N \tag{5.1}$$

where $\{\alpha_1, \alpha_2, \cdots, \alpha_q\}$ are all possible elements in GF(q). Define the $q \times N$ reliability matrix $\mathbf{\Pi}$ as a matrix with entries $\{\pi_{i,j}\}$ as computed in (5.1). $\mathbf{\Pi}$ serves as a soft input to an ASD decoder. The generic algorithm of ASD as in [4] is described in the following 4 steps:

Multiplicity Assignment: Compute the multiplicity matrix $\mathbf{M}$ with integer entries $\{M_{i,j}\}$ based on the reliability matrix $\mathbf{\Pi}$ according to a particular multiplicity assignment strategy.

Interpolation: For a given multiplicity matrix $\mathbf{M}$ with entries $\{M_{i,j}\}$, construct a bivariate polynomial $\mathcal{A}$(X, Y) of minimal $(1, K-1)$-weighted degree that passes through each point (x_j, α_i) with multiplicity at least $m_{i,j}$, for $i = 1, 2, \cdots, q$ and $j = 1, 2, \cdots, N$.

Factorization: Find all polynomials $f(X)$ such that $(Y - f(X))$ divides $\mathcal{A}(X, Y)$ and $\mathtt{deg}(f(X)) < K$. Form a candidate codeword list by re-encoding all such polynomials $f(X)$.

Codeword Selection: Select the most likely codeword from the candidate codeword list as the decoder output. If there is no codeword in the list, a decoding failure is declared.

Intuitively, the idea of ASD is to take advantage of soft information and assign higher multiplicities to more probable symbols such that the decoder has a better chance to find the correct codeword.

2. Performance of Algebraic Soft-decision Decoding

Define the inner product between two matrices of the same dimensionality as:

$$\langle \mathbf{A}, \mathbf{B} \rangle \overset{\mathtt{def}}{=} \mathtt{trace}(\mathbf{A}\mathbf{B}^T) = \sum_{i=1}^{q} \sum_{j=1}^{N} a_{i,j} b_{i,j} \tag{5.2}$$

Let $\mathbf{1}$ be the all-one $q \times N$ matrix. Suppose the vector $\underline{\mathcal{X}}$ represents an RS codeword, let $[\underline{\mathcal{X}}]$ be the codeword matrix with entries $[\underline{\mathcal{X}}]_{i,j}$ defined as: $[\underline{\mathcal{X}}]_{i,j} = 1$ if $\mathcal{X}_j = \alpha_i$; $[\underline{\mathcal{X}}]_{i,j} = 0$, otherwise. As in [4], the score and cost are defined as follows.

Definition 2 *The sum of multiplicities in $\boldsymbol{M}$ that are assigned to the transmitted symbols is defined as the score:*

$$S = \langle \boldsymbol{M}, [\underline{\mathcal{X}}] \rangle \tag{5.3}$$

Definition 3 *The number of linear constraints imposed in order to satisfy the mul-*

tiplicities as specified by $\boldsymbol{M}$ *is defined as the cost:*

$$C = \frac{1}{2}\sum_{i=1}^{q}\sum_{i=1}^{N} M_{i,j}(M_{i,j}+1) = \langle \boldsymbol{M}, \boldsymbol{M+1}\rangle/2 \tag{5.4}$$

Similar to other list decoding algorithms, the probability of error of ASD can be upper bounded using the union bound:

$$P_{ASD} \leq P_{List} + P_{ML} \tag{5.5}$$

where P_{List} is the probability that the transmitted codeword is not in the list and P_{ML} is the probability that the maximum likelihood decision is not the transmitted codeword. Usually, $P_{List} \gg P_{ML}$ and, therefore, we will approximate $P_{ASD} \approx P_{List}$ throughout the rest of this chapter. In general, the decoding region of ASD is difficult to characterize and analytical computation of P_{List} is a tough problem. However, it is shown in [4] that ASD is guaranteed to return the transmitted codeword when the following sufficient condition is satisfied:

Lemma 1 *[4] Finite cost: A sufficient condition for the transmitted codeword to be in the list is:*

$$S > \min\left\{\delta \in \mathbb{Z} : \left\lceil\frac{\delta+1}{k-1}\right\rceil \left(\delta + 1 - \frac{(k-1)}{2}\left\lfloor\frac{\delta}{k-1}\right\rfloor\right) > C\right\} \tag{5.6}$$

The proof of Lemma 1 is given in Theorem 3 in [4]. The above sufficient condition can also be expressed as (see also [39]): the transmitted codeword will be in the list if

$$\begin{aligned} T(S) &> C \qquad (5.7)\\ \text{where } T(S) &= (a+1)\left[S - \frac{a}{2}(K-1)\right], a(K-1) < S \leq (a+1)(K-1), a = 0, 1, \cdots \end{aligned}$$

is a piecewise linear function.

Generally speaking, larger cost leads to better decoding performance, while it also increases complexity (though the performance of ASD does not monotonically improve as the cost increases). As the cost goes to infinity, we can further simplify the sufficient condition as:

Lemma 2 *[4] Infinite cost: The sufficient condition for ASD to list the transmitted codeword as $C \to \infty$ is:*

$$S \geq \sqrt{2(K-1)C} \tag{5.8}$$

See Corollary 5 in [4] for the proof.

Usually, the sufficient conditions (5.6) and (5.8) become tight approximations when N is large. With a little bit abuse of terminology, we define the decoding failure of ASD as follows:

Definition 4 *For ASD with finite cost, a received codeword is said to be decodable if and only if the sufficient condition (5.6) is satisfied. When the received codeword is not decodable, a decoding failure is declared.*

Definition 5 *For ASD with infinite cost, a received codeword is said to be decodable if and only if the sufficient condition (5.8) is satisfied. When the received codeword is not decodable, a decoding failure is declared.*

For the rest of the chapter, we approximate the actual decoding error probability of ASD by the probability of decoding failure defined in Definition 4 and Definition 5 for finite cost and infinite cost cases respectively. Practically speaking, though the decoder may still be able to list the transmitted codeword even when the sufficient condition is not satisfied, the probability is very small and the approximation is tight for long codes, which are used in many existing standards.

B. Performance Analysis of Algebraic Soft-decision Decoding over Erasure Channels

In this section, we consider MAS's for erasure channels and their corresponding performance analyses.

1. Algebraic Soft-decision Decoding over the Binary Erasure Channels (BEC)

We first consider the case when RS codewords are transmitted as bits through a BEC with erasure probability ϵ. Similar to the argument in [4], [27], [26], we assume that the symbols in a codeword are independent and identically distributed (i.i.d) with a uniform distribution over GF(q) during the multiplicity assignment stage. For BEC's, a natural MAS is to assign equal multiplicities to the equiprobable candidate symbols in the same coordinate. For further justification, please see Appendix A.

Consider the following definition: define each symbol with i-bit erasures as being of type i. Consequently, for a code over $GF(2^m)$, there are $(m+1)$ types of symbols. Let the number of symbols of type i in a received codeword be a_i. As discussed above, we will assign equal multiplicities to equiprobable candidates in the same coordinates. Moreover, we assume that equal multiplicities are assigned to all candidate symbols of the same type in a received codeword; whereas, the multiplicity assigned to type i may vary according to the received codeword. This assumption will be justified later.

Let m_i be the multiplicity assigned to each candidate symbol of type i. Thus, the total multiplicity assigned to one symbol of type i is $2^i m_i$. The score and cost are

$$\begin{aligned} S &= \sum_{i=0}^{m} a_i m_i && (5.9) \\ C &= \sum_{i=0}^{m} a_i 2^i \binom{m_i+1}{2} && (5.10) \\ C &= \frac{1}{2}\sum_{i=0}^{m} a_i 2^i m_i^2 (1+o(1)), m_i \to \infty && (5.11) \end{aligned}$$

The approximation in (5.11) becomes tight when m_i becomes large. We will derive an upper bound and a lower bound on the probability of decoding failure of ASD with infinite cost as defined in Definition 5. Furthermore, we consider ASD with infinite cost such that we can relax the multiplicities from being integers to real numbers. It is justified by the fact that rational numbers are dense on the real axis and they can always be scaled up to be integers with infinite cost. Hence any real numbers can be approximated arbitrarily close with rational numbers (see also [26]).

Following [4] [33], we define proportional multiplicity assignment strategy (PMAS) as follows:

Proportional Multiplicity Assignment Strategy: For a given total multiplicity per symbol M, PMAS assigns multiplicity proportional to the APP of that candidate symbol. That is the multiplicity we assign to symbol α_i in the j^{th} coordinate of the received vector is $\mathrm{M}_{i,j} = \lfloor \pi_{i,j}\mathrm{M} \rfloor$, where M is a predetermined real number.

PMAS defined above is equivalent to the simplified KV defined in [33]. Note that there is a quantization error, however, the multiplicity assignment is asymptotically proportional to the APP as the cost approaches infinity. We will show in the following that PMAS is optimal over the BEC with infinite cost:

Theorem 1 *The proportional multiplicity assignment strategy (PMAS) is optimal over the BEC regardless of the received signal. That is PMAS maximizes the score* *for a given cost over the BEC.*

Proof 1 *Assume that the received codeword has a_i symbols of type i, the MAS can be formulated as maximizing the score with a cost constraint. With infinite cost, the problem is expressed as:*

$$\begin{aligned} \max_{\{m_i\}} \quad S &= \sum_{i=0}^{m} a_i m_i \\ \textit{subject to} \quad C &\approx \frac{1}{2}\sum_{i=0}^{m} a_i 2^i m_i^2 \leq C_0 \end{aligned} \tag{5.12}$$

This is a standard optimization problem with a linear cost function and a quadratic constraint. Using a Lagrange multiplier, the new objective function becomes

$$\mathrm{L} = -\sum_{i=0}^{m} a_i m_i + \lambda\left(\frac{1}{2}\sum_{i=0}^{m} 2^i a_i m_i^2 - C_0\right) \tag{5.13}$$

Take the partial derivative with respect to m_i and set it to zero. We have:

$$\frac{\partial \mathrm{L}}{\partial m_i} = -a_i + \lambda 2^i a_i m_i = 0 \tag{5.14}$$

Therefore we have $m_i = \frac{2^{-i}}{\lambda}$, i.e., $m_i \propto 2^{-i}$, which proves that PMAS is optimal.

Note that m_i does not depend on a_i. Even without the assumption that equal multiplicities are assigned to candidate symbols of the same type, we still get $m_i = \frac{2^{-i}}{\lambda}$ for all type i candidates, i.e., PMAS is optimal over the BEC.

Since PMAS is optimal over the BEC, we will from now on assume that PMAS is used. Under PMAS, we assume that the total multiplicity for each symbol is M. Consequently, the score is $S_0 = \sum_{i=0}^{m} a_i 2^{-i}\mathrm{M} = \eta\mathrm{M}$ and the cost is $C_0 = \frac{1}{2}\sum_{i=0}^{m} a_i 2^{-i}\mathrm{M}^2(1 + o(1)) = \frac{1}{2}\eta\mathrm{M}^2(1 + o(1))$, where $\eta = \sum_{i=0}^{m} a_i 2^{-i}$ is a positive number. The sufficient condition of (5.8) becomes:

$$S_0 \geq \sqrt{2(K-1)C_0} \tag{5.15}$$

$$\eta > K-1 \tag{5.16}$$

When $K = 1$, under PMAS, $\eta > 0$, the transmitted codeword will always be on the decoding list. From now on, we only consider the case $K > 1$.

We study the worst case bit-level decoding radius of ASD under PMAS with infinite cost over the BEC. We need the following lemmas.

Lemma 3 *Over the BEC, if a received word is decodable under PMAS with infinite cost, it is always decodable if some of the erasures are recovered.*

Proof 2 *The proof is immediate by the fact that if some of the erasures are recovered, η will increase and as can be seen from (5.16), the decoding performance monotonically improves as η increases.*

Lemma 4 *Over the BEC, given f bit erasures, the worst case erasure pattern for ASD under PMAS with infinite cost is that all bits are spread in different symbols as evenly as possible. That is: $(N - f + \lfloor \frac{f}{N} \rfloor N)$ symbols contain $\lfloor \frac{f}{N} \rfloor$ bit erasures and $(f - \lfloor \frac{f}{N} \rfloor N)$ contain $\lceil \frac{f}{N} \rceil$ bit erasures.*

Proof 3 *Take two arbitrary symbols of type i and j, if we average the bit erasures between these two, we get two symbols of type $\lfloor \frac{i+j}{2} \rfloor$ and $\lceil \frac{i+j}{2} \rceil$. The updated η' can be expressed as:*

$$\eta' = \eta + 2^{-\lfloor \frac{i+j}{2} \rfloor} + 2^{-\lceil \frac{i+j}{2} \rceil} - 2^{-i} - 2^{-j} \leq \eta \tag{5.17}$$

Since $\eta \geq \eta'$ and again according to (5.16), the latter erasure pattern is worse. By repeating the above procedure, we can finally get the worst case erasure pattern of

ASD under PMAS, i.e., the bit erasures are spread as evenly as possible in different symbols.

According to Lemma 3 and Lemma 4, the bit-level decoding radius of PMAS can be characterized.

Theorem 2 *Over the BEC, the bit-level decoding radius of ASD under PMAS with infinite cost can be expressed as:*

$$\begin{aligned} f &< (i+1)N - 2^i(K-1), \\ \textit{for} \quad & 2^{-i} + \frac{1-2^{-(i+1)}}{N} \leq R < 2^{-(i-1)} + \frac{1-2^{-i}}{N}, i = 1, 2, \cdots, m \end{aligned}$$

Especially, for high rate RS codes, we have

$$f \quad < \quad 2N - 2(K-1), \ \textit{for } R \geq \frac{1}{2} + \frac{3}{4N} \tag{5.18}$$

Proof 4 *According to Lemma 4, the worst case erasure pattern is all erased bits are spread evenly over different symbols. First consider the case $f \leq N$, (5.16) becomes:*

$$(N-f) + \frac{1}{2}f > K - 1 \tag{5.19}$$

The corresponding rate region must satisfy the constraint that when $f = N+1$, in the worst case $\eta \leq K-1$. We get $K \geq \frac{1}{2}N + \frac{3}{4}$ in this case. Altogether, we get the bit-level decoding radius for the high rate case:

$$f < 2N - 2(K-1), \ \textit{when } R \geq \frac{1}{2} + \frac{3}{4N} \tag{5.20}$$

Similarly, when the decoding radius $(i-1)N < f \leq iN$, we must have $2^{-i}N + 1 - 2^{-(i+1)} \leq K < 2^{-(i-1)}N + 1 - 2^{-i}$, where $i = 1, 2, \cdots, m$. We can obtain the exact *decoding radius for all these cases:*

$$f < (i+1)N - 2^i(K-1), \text{ when } 2^{-i}N + 1 - 2^{-(i+1)} \leq K < 2^{-(i-1)}N + 1 - 2^{-i} \tag{5.21}$$

According to Theorem 2, any erasure pattern with $f < 2(N-K+1)$ is decodable. We can get an upper bound on the frame error rate (FER) of ASD under PMAS with infinite cost.

Corollary 1 *For RS codes of rate for* $R \geq \frac{1}{2} + \frac{3}{4N}$ *over the BEC, ASD under PMAS with infinite cost fails when there are more than* $2(N-K)+1$ *symbols containing erased bits.*

Proof 5 *The corollary follows from (5.18) and Lemma 3. If there are more than* $2(N-K)+1$ *symbols having erased bits, the most optimistic case is that these symbols are of type* 1*. Besides, due to (5.18), the sufficient condition is not satisfied and ASD fails as defined in Definition 5.*

Theorem 2 gives an upper bound on the FER performance over the BEC and Corollary 1 provides a lower bound. These bounds are shown in Figure 20 in conjunction with the union bound on the averaged FER of a maximum likelihood (ML) decoder over the RS ensemble [55]. Note that ASD has a significant performance gain over conventional BM erasure decoding. This is intuitive since we do not have to erase the whole symbol if some bits in the symbol are erased, which can be taken advantage of by ASD. It can be seen from the figure that for practical high rate long codes, both the upper and lower bounds are tight and they together accurately indicate the performance of ASD. Also note that the averaged performance of the ML decoder over the RS ensemble is very close to the capacity of the BEC, which shows that RS codes are good codes.

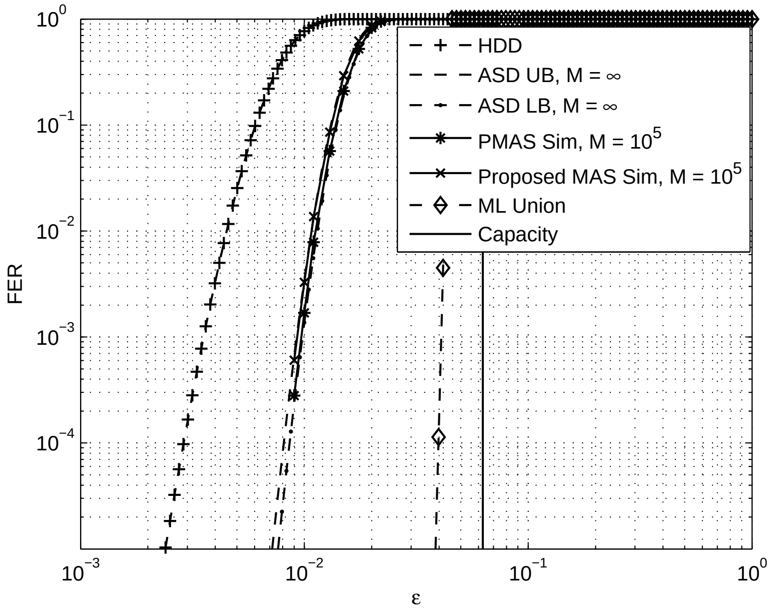


Fig. 20. Bounds and Simulation Results of ASD for an RS(255,239) over a BEC

The tightness of the upper bound and the lower bound of ASD motivates the following proposed MAS:

Proposed Multiplicity Assignment Strategy: In each received coordinate, we assign m_0 = M if that symbol does not contain erased bits, assign m_1 = M/2 if the symbol contains 1 bit-level erasure and do not assign any multiplicity for symbols containing more than 1 bit-level erasures, that is to set $m_j = 0, j = 2, \cdots, m$.

Since erasing 2 bits in the same symbol leads to the same score but less cost than 2 bits in two different symbols,the worst case erasure pattern of the proposed MAS for RS codes with $R \geq \frac{1}{2} + \frac{3}{4N}$ is that all bit-level erasures are spread in different symbols. According to Theorem 2, the proposed MAS can recover any bit-level erasures containing less than $2(N-K+1)$ bit erasures. Essentially, the proposed MAS takes care of the worst case erasure pattern only and it is asymptotically optimal in terms of achieving the largest worst case decoding radius. Consequently, the FER upper bound derived in Theorem 2 for $R \geq \frac{1}{2} + \frac{3}{4N}$ is also a valid upper bound for the proposed MAS. Though the proposed MAS is not optimal as PMAS, the loss is quite small by comparing the upper bound of the proposed MAS and the lower bound of PMAS for high rate RS codes. It can also be seen from Figure 20 that the simulation results of the proposed MAS and the optimal PMAS are very close.

2. Extension to 2^u-ary Erasure Channels

We extend the result in the previous subsection to 2^u-ary erasure channels, i.e. u coded bits are grouped together and transmitted using a 2^u-ary symbol (it can be QAM or PSK modulation format). The channel will erase the signal with erasure probability ϵ at 2^u-ary symbol-level. Practical channels of this model was discussed in [71].

In the previous subsection, we showed that PMAS is optimal for erasure channels. Clearly, all erasure patterns in this 2^u-ary erasure channel model is a subset of erasure patterns of BEC. Therefore, with infinite cost, PMAS is still optimal for this channel model. Here, we only consider the case when u divides m, i.e., $m = vu$. Thus, for each symbol, we have $(v + 1)$ types.

Lemma 5 *Over the 2^u-ary erasure channel, the worst case erasure pattern for ASD under PMAS with infinite cost is that all erasure events are spread in different symbols as evenly as possible.*

Proof 6 *Assume two RS symbols are of type i and j, we can average the erasure events between the two symbols, we have:*

$$\eta' = \eta + 2^{-\lfloor \frac{i+j}{2} \rfloor u} + 2^{-\lceil \frac{i+j}{2} \rceil u} - 2^{-iu} - 2^{-ju} \leq \eta \tag{5.22}$$

Similar to Lemma 4, spreading erasure events in different RS symbols evenly gives the worst case.

Theorem 3 *Over the 2^u-ary erasure channel, ASD under PMAS can guarantee to decode up to $f < (N-K+1)/(1-2^{-u})$ 2^u-ary symbol-level erasures if $R \geq 2^{-u} + \frac{1+2^{-2u}-2^{-u}}{N})$.*

Proof 7 *According to Lemma 5 and (5.16), spreading erasure events in different symbols is the worst case erasure pattern if $K \geq 2^{-u}N + 1 + 2^{-2u} - 2^{-u}$, that is $R \geq 2^{-u} + \frac{1+2^{-2u}-2^{-u}}{N}$. Thus $\eta = N - (1-2^{-u})f$. According to (5.16) when the following condition is satisfied:*

$$f < (N-K+1)/(1-2^{-u}) \tag{5.23}$$

ASD is guaranteed to decode the transmitted codeword.

Remark 1 *Note that (5.23) is a generalization of Theorem 2 (with $u = 1$ as a special case). As u becomes larger, the asymptotical 2^u-ary symbol-level decoding radius gets smaller. As $u \rightarrow m$, f gets close to conventional BM erasure decoding region.*

C. Performance Analysis of ASD over Binary Symmetric Channels (BSC)

In this section, we study the performance of ASD over BSC's. For BSC's, both the transmitted and the received symbols are in $GF(q)$, i.e., $\mathcal{X}_i \in GF(q)$ and $\mathcal{Y}_i \in GF(q)$, for $i = 1, \cdots, N$. In this case, bit-level reliabilities are the same for all received bits. However, symbol-level soft information can be utilized by ASD under proper MAS.

Let a candidate symbol in a particular coordinate be of type i if it differs from the received symbol in that coordinate by i bits and let a_i denote the number of coordinates that the transmitted symbol is of type i. Again, we assign equal multiplicities to candidate symbols of the same type, that is, we set $M_{i,j} = m_{k_{i,j}}$, where $k_{i,j}$ is the number of bit positions in which α_i and $\mathcal{Y}_j$ differ. That is, we assign m_0 to the received symbol, m_1 to all the m symbols which differ from the received symbol in 1-bit position and so on. Note that $M_{i,j}$ is the multiplicity assigned to α_i in the j^{th} coordinate and m_k is the predetermined multiplicity we will assign to candidate symbols of type k. However, unlike the BEC case, the type of the transmitted symbol is unknown at the receiver, the MAS optimization problem can not be formulated as in the BEC case. Therefore, we resort to the asymptotically optimal MAS, i.e., maximizing the bit-level decoding radius for the worst case error pattern. It can also be easily justified that non-uniform multiplicity assignment among symbols of the same type is strictly suboptimal in terms of achieving the largest decoding radius for the worst case error pattern, since the worst case error pattern will always correspond to the candidate symbols with smaller multiplicities. The MAS optimization problem can be formulated as a max-min problem over $\{a_i\}$ and $\{m_i\}$.

$$\begin{aligned} \max_{\{m_i\}} \min_{\{a_i\}} e &= \sum_{i=0}^{m} i a_i - 1 \qquad (5.24)\\ \text{s. t.} \sum_{i=0}^{m} a_i m_i &\leq \sqrt{2(K-1)N \sum_{i=0}^{m} \binom{m}{i} \frac{m_i^2}{2}} \\ \sum_{i=0}^{m} a_i &= N, \text{where } a_i \text{ are non-negative integers} \end{aligned}$$

The above optimization is still quite complicated, since a_i's are integers. Even if this condition is relaxed, the solution may only be obtained numerically, which does not give any insight into the exact decoding radius of ASD.

We first take one step back and consider a special case of BSC, called 1-bit flipped BSC, i.e., in each symbol, at most 1 bit is in error. By doing that, we only have to assign multiplicities to two types of symbols. The optimal MAS for this 1-bit flipped BSC is to assign M to the received vector and tM to 1-bit flipped neighbors. The asymptotically optimal decoding radius e_{max} and the corresponding optimal t can be computed in close forms. The derivations are given in Appendix B.

It should be noted that for the 1-bit flipped BSC, the performance improvement of ASD over GS is significant only when the rate is low. For instance, for $N = 255$, $K = 223$, ASD does not increase the bit-level error correcting radius, for $N = 255$, $K = 167$, ASD gives an extra error correction capability over GS decoding at the bit-level, for $N = 255$, $K = 77$, it corrects 7 more errors and for $N = 255$, $K = 30$, it corrects 45 more errors. For $K < 30$, all errors can be corrected for the 1-bit flipped BSC.

Now, we show that the above proposed MAS is also asymptotically optimal for RS codes over the BSC under certain conditions, which are satisfied for a wide range of code rates. We begin with the following Lemma.

Lemma 6 *Over the BSC, the worst case error pattern for the proposed ASD with infinite cost is all erroneous bits spread in different symbols, if the total number of bit errors $e \leq N$ and the optimal multiplicity coefficient $t \leq \frac{1}{2}$.*

Proof 8 *Assume $e \leq N$ bits get flipped by the channel. If bit errors are spread in different symbols, as the 1-bit flipped BSC case, the score can be expressed as:*

$$S = M[(N - e) + te] \tag{5.25}$$

The cost of ASD for the BSC does not change when the MAS is fixed. For a given number of bit errors, the worst case error pattern minimizes the score of the MAS. In the above MAS, multiplicities are assigned only to the received symbol and its 1-bit flipped neighbors only. Thus, a potentially worse error pattern than the 1-bit flipped

BSC is to group bits in some 1-bit-flipped symbols to reduce the score.

Let the worst case error pattern have e' symbols containing 1-bit error and e'' symbols containing more than 1-bit errors. Obviously, for symbols containing more than 2-bit errors, we can always further decrease the score of the proposed MAS for 1-bit flipped BSC by splitting these bit errors into one symbol containing 2-bit errors and the other containing the rest of the errors. Consequently, the worst case error pattern will contain symbols with at most 2-bit errors. We have $e' + 2e'' = e$. The score becomes:

$$S'' = M[(N - e' - e'') + te'] = M[(N - e) + et + e''(1 - 2t)] \tag{5.26}$$

When $t \leq \frac{1}{2}$, $S'' \geq S$, which proves that spreading all erroneous bits in different symbols is the worst case error pattern for the proposed ASD over the BSC.

Theorem 4 *In the infinite cost case, the asymptotically optimal MAS for the BSC is the same as the asymptotically optimal MAS for the 1-bit flipped BSC if the optimal decoding radius of the 1-bit flipped BSC e_{max} satisfies $e_{max} \leq N$ and the corresponding optimal multiplicity coefficient $t \leq \frac{1}{2}$. Besides, the bit-level decoding radius of the BSC is also e_{max}.*

Proof 9 *According to Lemma 6, over the BSC, all erroneous bits spread in different symbols is the worst case error pattern for the proposed ASD if $e \leq N$ and $t \leq \frac{1}{2}$, which is nothing but the 1-bit flipped BSC's. On the other hand, the proposed MAS derived in Appendix B is asymptotically optimal for the 1-bit flipped BSC, i.e., maximizing the worst case decoding radius e_{max}. Consequently, the proposed MAS will guarantee to decode all error patterns with no more than e_{max}-bit errors over BSC's as well.*

The error correction radius of the optimal MAS as a function of t over an RS(255,

55) code is given in Figure 21. It can be seen that the optimal MAS (which is achieved by $t = 0.2$) corrects 13 and 50 more bit-level errors than GS and BM in the worst case. Besides, we also plot bit-level radius of PMAS as a function of the crossover probability p_b of the BSC. Note that PMAS is not asymptotically optimal for the BSC here. Even though we choose p_b to maximize the bit-level radius (around $p_b = 0.13$), the bit-level decoding radius is still 1 bit smaller than that of the optimal MAS. The reason can be explained as follows: the worst case error pattern of this BSC is shown to be all bit-level errors spread in different symbols, thus, the asymptotically optimal MAS only has to assign multiplicities to symbols of type 0 and type 1. On the other hand, PMAS assigns multiplicities proportionally. Thus it also assigns multiplicities to candidate symbols with more than 1-bit error and unnecessarily spends more cost, which makes it suboptimal in terms of achieving the worst case bit-level decoding radius.

We consider the performance of this asymptotically optimal MAS using a toy example shown in Figure 22. Consider the performance of an RS(7, 3) code over the BSC. For this code, the decoding radii of conventional BM and GS algorithm are 2 and 3 respectively. Note that, the bit-level decoding radius of PMAS can also be optimized over the crossover probability. However, in this case, the optimal bit-level radius of PMAS is still 3, while the optimal MAS, on the other hand, can achieve bit-level decoding radius 4. We can see from Figure 22, the performance upper bound of ASD under the asymptotically optimal MAS outperforms GS and BM.

Remark 2 *Over BSC's, the gain of ASD with infinite cost over GS decoding is very little for practical high rate RS codes. Besides, simply increasing the bit-level decoding radius may not necessarily lead to better performance at a moderate FER level. Since GS is a symbol-level decoding algorithm, it may be able to correct more typical error*

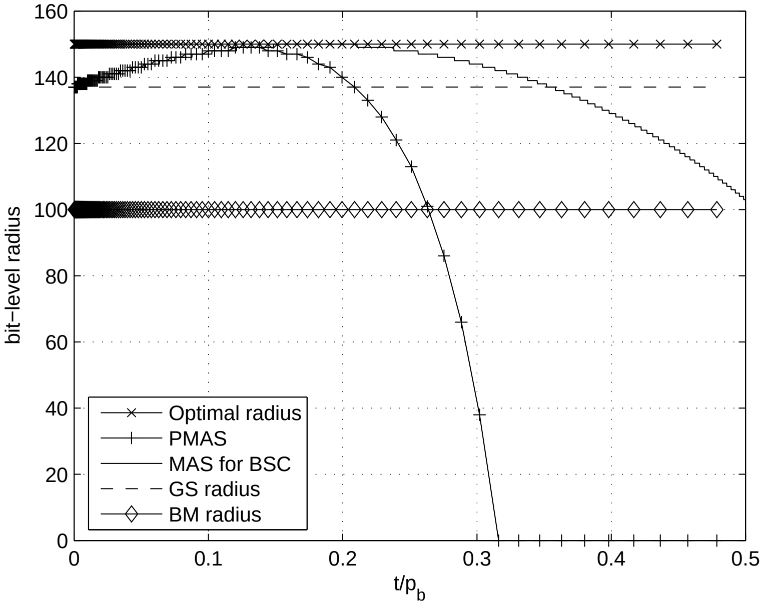


Fig. 21. Bit-level Decoding Radius of an RS (255,55) Code

patterns at a moderate FER level than a bit-level decoding algorithm with a slightly larger bit-level decoding radius and hence leads to a better performance at that FER level than ASD with the proposed asymptotically optimal MAS.

D. Bit-level Decoding Region of Algebraic Soft-decision Decoding Algorithm

In this section, we generalize the analysis in previous two sections to a bit-level error and erasure channel. A simple MAS is proposed and its properties are studied. The decoding region of the proposed ASD in terms of the number of errors e and the number of bit-level erasures f is investigated for both infinite and finite cost cases. Finally, we show that the decoding region of the proposed ASD monotonically enlarges as the multiplicity M increases.

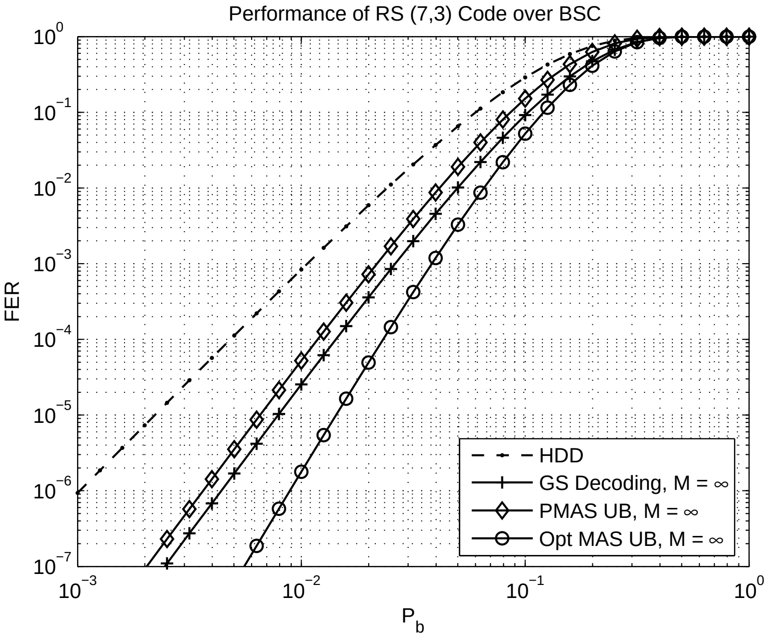


Fig. 22. Upper Bounds of ASD for an RS (7,3) Code over a BSC

1. Proposed MAS for a Mixed Bit-level Error and Erasure Channel

We first propose a MAS for the mixed channel, which is motivated by the analysis of the previous two sections. In Section B, a simple proposed MAS has been shown to have nearly the same performance as the optimal PMAS for high rate RS codes over BEC's. On the other hand, as shown in Section C, ASD even with an optimal MAS has hardly any gain over GS decoder for high rate RS codes over BSC's. Altogether, these results suggest that most of the gain of ASD for high rate RS codes is from taking care of 1-bit erased symbols. Therefore, we expect to obtain most of the gain of ASD over other channels by proper multiplicity assignment to at most 1-bit erased symbols. We have the following proposed MAS:

Proposed Multiplicity Assignment Strategy: In each received coordinate, we assign $m_0 = \text{M}$ if that symbol does not contain erased bits, assign $m_1 = \text{M}/2$ to each candidate if the symbol contains 1-bit erasure and do not assign any multiplicity for symbols containing more than 1-bit erasures, that is to set $m_j = 0, j = 2, \cdots, m$.

Under the proposed MAS, there are 5 types of symbols, which are listed below with their corresponding score per symbol S_B and cost per symbol C_B:

(T-1) correctly received symbol: $S_B = M$ and $C_B = \frac{M^2+M}{2}$

(T-2) symbol get erased at symbol-level: $S_B = 0$ and $C_B = 0$

(T-3) erroneous symbol without erasure: $S_B = 0$ and $C_B = \frac{M^2+M}{2}$

(T-4) 1-bit erased symbol without other errors: $S_B = \frac{M}{2}$ and $C_B = \frac{M^2+2M}{4}$

(T-5) 1-bit erased symbol with other errors: $S_B = 0$ and $C_B = \frac{M^2+2M}{4}$

As before, we first characterize the worst case error pattern for the proposed MAS, which dominates the performance for high rate RS codes. We first have the following lemma:

Lemma 7 *ASD under the proposed MAS fails over the mixed channel if* $S \leq M(K-1)$.

Proof 10 *When* $S \leq M(K-1)$, $T(S) \leq T(M(K-1))$. *Since* $T(M(K-1)) = (M+1)M(K-1)/2$ *and* $T(S)$ *is a convex function, it is easy to verify the following upper bound on* $T(S)$*:*

$$T(S) \leq \frac{1}{2}(M+1)S, \quad 0 \leq S \leq M(K-1) \tag{5.27}$$

Considering all types of symbols, we have $\frac{1}{2}(M+1)S_B \leq C_B$. *Therefore, for any received codeword, we have the following:*

$$T(S) \leq \frac{1}{2}(M+1)S \leq C \tag{5.28}$$

Next, we show that recovering bit-level erasures in error-free symbols improves the performance monotonically.

Lemma 8 *Over the mixed channel, suppose a received vector is decodable by ASD under the proposed MAS with multiplicity M. If any bit-level erasures in an error-free symbol is recovered, the received vector with recovered erasure is decodable by ASD under the same multiplicity assignment.*

Proof 11 *Recovering 1 bit-level erasure from an error-free symbol can be of the following 3 cases:*

1) From a symbol with more than 2-bit erasures: $\Delta S = 0$ and $\Delta C = 0$

2) From a symbol with exactly 2-bit erasures: $\Delta S = \frac{M}{2}$ and $\Delta C = \frac{M^2+2M}{4}$

3) From a symbol with exactly 1-bit erasure: $\Delta S = \frac{M}{2}$ and $\Delta C = \frac{M^2}{4}$

Case 1) is trivially decodable. For Case 2) and Case 3), let S, C and S', C' be scores and costs before and after erasure recovering. We have $a(K-1) \leq S \leq (a+1)(K-1)$, where a is an integer. Since $T(S) > C$ and according to Lemma 7, we must have $a+1 > M$. Since $T(S)$ is a piecewise linear function with monotonically increasing slope, we have:

$$\begin{aligned} T(S') &= T\left(S + \frac{M}{2}\right) \\ &\geq T(S) + (a+1)\frac{M}{2} \\ &> C(f) + \frac{M^2}{2} \\ &\geq C(f) + \frac{M^2+2M}{4} \geq C(f') \qquad (5.29) \end{aligned}$$

where (5.29) is due to the fact that $M \geq 2$ in the proposed MAS.

The following lemma shows that spreading bit-level erasures in different error-free symbols results in a worse performance than putting them in the same symbol.

Lemma 9 *Over the mixed channel, suppose a received vector has a symbol containing more than 1 bit-level erasures and we move 1 bit-level erasure from this symbol to a* *correctly received symbol. If the resulting vector is decodable using the proposed MAS with multiplicity M, then the original received vector is also decodable by ASD under the same multiplicity assignment.*

Proof 12 *Moving 1 bit-level erasure from a symbol with more than 1-bit erasure to a correctly received symbol can be of the following 3 cases:*

1) From a symbol with more than 2 erasures, $\Delta S = -\frac{M}{2}$ and $\Delta C = -\frac{M^2}{4}$

2) From a symbol with exactly 2 erasures and no errors, $\Delta S = 0$ and $\Delta C = \frac{M}{2}$

3) From a symbol with 2 erasures and some errors, $\Delta S = -\frac{M}{2}$ and $\Delta C = \frac{M}{2}$

Case 1) is nothing but adding 1-bit erasure in a correct symbol. As shown in Lemma 8, it results in no better performance. In Case 2) and Case 3), moving 1-bit erasure to correct symbols leads to no larger scores but a larger costs, therefore, it will also result in no better performance.

With the above lemmas, we now characterize the worst case error and erasure pattern.

Theorem 5 *Over the mixed channel, for any received codeword, the worst case error and erasure pattern for the proposed MAS is that all bit-level errors are spread in different erasure-free symbols and bit-level erasures are spread evenly in the rest symbols. Besides, if the worst case pattern with e errors and f erasures is decodable under the proposed MAS with multiplicity M, any received codeword with $e' \leq e$ bit-level errors and $f' \leq f$ bit-level erasures is decodable by ASD under the same multiplicity assignment.*

Proof 13 *In the worst case, errors should obviously be spread in different symbols. Besides, having erasures in erroneous symbols will lead to the same score, but a smaller cost. Hence, in the worst case, errors and erasures should also be spread in different* *symbols. If the number of errors e and the number of bit-level erasures f satisfy $e + f \leq N$, according to Lemma 9, putting erasures in a correctly received symbol is the worst case. Applying Lemma 9 recursively, in the worst case, bit-level erasures are spread in different symbols. If $e + f > N$, putting more than 2 bit-level erasures in the same symbol essentially reduces the number of bit-level erasures in error-free symbols and according to Lemma 8, it always leads to no worse performance. As a result, when $e + f > N$, in the worst case, we must have errors, 1-bit erased symbols and 2-bit erased symbols occupying all N coordinates of the received codeword.*

On the other hand, fewer errors will lead to better performance in the worst case. Erasures will only participate in error-free symbols in the worst case. According to Lemma 8, fewer bit-level erasures in error-free symbols leads to no worse performance. In conclusion, for any received codeword, the worst case is that all errors are spread in different erasure-free symbols and erasures are spread evenly in the rest symbols. Besides, reducing the number of bit-level errors e or the number of bit-level erasures f will not degrade the worst case performance.

Theorem 5 characterizes the worst case error and erasure pattern, which makes the decoding region analysis easier.

Corollary 2 *Over the mixed channel, the score and the cost of the proposed MAS with multiplicity M in the worst case with the number of bit-level errors e and bit-level erasures f.*

$$S = (N - e - f/2)\, M \tag{5.30}$$

$$C \leq (2N - f)\frac{M^2}{4} + N\frac{M}{2} \tag{5.31}$$

Proof 14 *The corollary is immediate by considering the worst case error and erasure pattern in Theorem 5 for both $e + f \leq N$ and $e + f > N$ cases.*

Corollary 3 *Over the mixed channel, the proposed ASD fails if* $f \geq 2(N-(K-1)-e)$

Proof 15 *The corollary is obtained by combining Lemma 7 and Corollary 2.*

Corollary 3 suggests that ASD under the proposed MAS fails before all error-free symbols get erased at the symbol-level. Besides, it also gives an outer bound on the decoding region of the proposed MAS. The exact decoding region of the proposed MAS will be studied in more details in the following subsection.

2. Infinite Cost Performance Analysis

Due to the simplicity, the decoding region of this proposed MAS for medium to high rate RS codes can be characterized analytically. First, we consider the infinite cost case.

Theorem 6 *Under the proposed MAS with* $M \to \infty$*, the decoding region over the mixed channel in terms of* e *and* f *when* $e+f \leq N$ *is:*

$$e < N - f/2 - \sqrt{(K-1)(N-f/2)} \tag{5.32}$$

Proof 16 *When* $e+f \leq N$*, in the worst case the score and the cost can be expressed as*

$$\begin{aligned} S &= (N-e-f/2)M & (5.33) \\ C &= 1/4M^2(1+o(1))(2N-f) & (5.34) \end{aligned}$$

Plugging in (5.8), we can get:

$$e < N - f/2 - \sqrt{(K-1)(N-f/2)} \tag{5.35}$$

According to Corollary 2, when $e + f > N$, (5.35) is still achievable and the actual decoding region can be larger. When $f = 0$, the above region becomes the maximum error correcting radius of GS decoding; when $e = 0$, we can obtain the worst case bit-level decoding radius derived in (5.18).

To get an idea on how good this proposed MAS is, we derive an outer bound on the optimal decoding region of ASD with infinite cost. Using a technique similar to that used in Section C, we first derive the optimal MAS over a 1-bit flipped or erased channel. That is, we assume in each symbol of the RS codeword, there is at most either 1 bit in error or at most 1-bit erasure. The derivation of the optimal decoding region for this channel is given in Appendix C. In general, the 1-bit flipped or erased channel is optimistic compared with the actual bit-level error and erasure channel. Hence, when $e + f \leq N$, the optimal decoding region of a 1-bit flipped or erased channel serves as an outer bound of the actual decoding region of a mixed error and bit-level erasure channel.

3. Finite Cost Performance Analysis

Consider, the proposed MAS with finite cost, in the simplest case, M = 2. That is, we assign $m_0 = 2$ to symbols without erasures; if there is 1 bit-level erasure, we assign $m_1 = 1$ to each candidate symbol; otherwise, we assign $m_i = 0, i = 2, 3, \cdots, m$. The decoding region is characterized in the following theorem.

Theorem 7 *Under the proposed MAS with $M = 2$, the decoding region of RS codes of rate $R \geq 2/3 + 1/N$ over the mixed channel is:*

$$e < \frac{1}{2}(N - K + 1) - 1/3f \tag{5.36}$$

Proof 17 *For $R \geq 2/3 + 1/N$, in the worst case, errors and erasures will not overlap.*

Hence, $S = 2(N - e - f/2)$ and $C = 3N - f$. We must have:

$$(a+1)(2(N-e-f/2) - a/2(K-1)) > 3N - f \tag{5.37}$$

$$a(K-1) < 2(N-e-f/2) \leq (a+1)(K-1), \text{ where } a \text{ is a non-negative integer} \tag{5.38}$$

For $a = 0, 1$, we get contradictions.

For $a \geq 3$, we get trivial bounds.

For $a = 2$, we obtain the decoding region:

$$e < \frac{1}{2}(N-K+1) - \frac{1}{3}f, \text{ for } (K-1)/N \geq 2/3 \tag{5.39}$$

Corollary 4 *For RS codes of rate $R < 2/3 + 1/N$, the decoding region over the mixed channel in Theorem 7 is achievable under the proposed MAS.*

Proof 18 *If $e + f \leq N$, when (5.36) is satisfied, we must have $T(S) > C$; if $e + f > N$, again due to Corollary 2, the above region is still achievable.*

We can also derive a decoding region of the proposed MAS with any given multiplicity M as follows:

Theorem 8 *Under the proposed MAS with a given multiplicity M, the decoding region in terms of the number of errors e and the number of bit-level erasures f for $e + f \leq N$ as:*

$$e < N - \frac{f}{2} - \frac{\hat{a}(\hat{a}+1)(K-1)/2 + C}{M(\hat{a}+1)} \tag{5.40}$$

with the cost $C = \frac{1}{2}(N-f)M(M+1) + f\frac{M}{2}(\frac{M}{2}+1)$ and $\hat{a} = \lfloor \frac{-1+\sqrt{1+\frac{8C}{K-1}}}{2} \rfloor$.

The derivation of Theorem 8 is provided in Appendix D. Similarly when $e + f > N$, the region in Theorem 8 is still achievable. Though the actual decoding region can

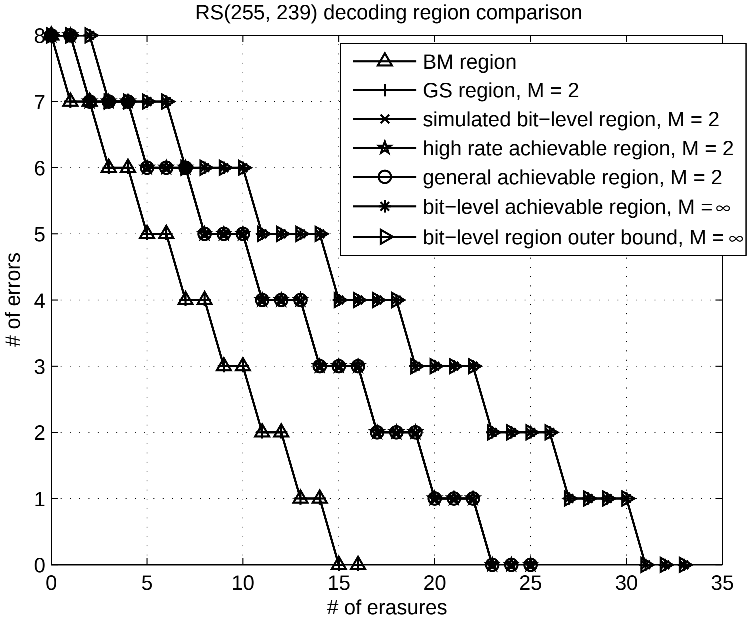


Fig. 23. Bit-level Decoding Region of ASD for an RS(255, 239) Code

be even larger for low rate RS codes, the study of which is beyond the scope of this dissertation.

We give some examples of the decoding region of RS codes over the mixed channel under different decoding schemes. In Figure 23, a high rate RS(255, 239) code is considered. When there is no erasure, the number of errors the proposed ASD can correct is the same as conventional BM and GS decoding. However, when considering bit-level erasures, the decoding region of the proposed ASD is significantly larger than BM and GS decoding. The maximum number of bit-level erasures it can correct is about 1.5 times that of BM and GS decoding at the bit-level. It is clear that the gain is contributed to treating erasures at the bit-level. In the infinite cost case, the proposed MAS achieves the outer bound of the decoding region.

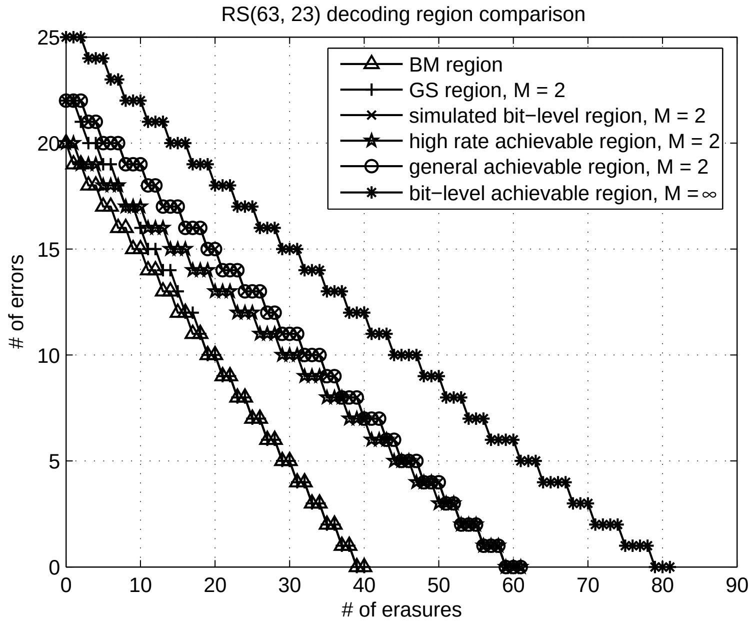


Fig. 24. Bit-level Decoding Region of ASD for an RS(63, 23) Code

In Figure 24, we show the decoding region of a low rate code RS(63, 23). In this case, the high rate achievable region in Corollary 4 becomes loose. On the other hand, the general decoding region derived in Theorem 8 still coincides with the actual decoding region (by checking the sufficient condition for each error and erasure pair). When there is no erasure, the maximum number of errors the proposed ASD can correct is the same as GS decoding. Again, since ASD can take advantage of the erasure information at the bit-level, the decoding region of the proposed ASD is strictly larger than the decoding region of GS with symbol-level error and erasure decoding. When $e + f > N$ in the infinite cost case, the outer bound becomes invalid. However, the achievable region in the infinite cost case is still a valid achievable region.

4. Monotonicity

In this subsection, we show the monotonicity of the decoding region of the proposed MAS as a function of multiplicity M over the mixed channel. It was shown by McEliece in [39], the error correction radius of GS algorithm is a monotonic function of multiplicity M. This monotonicity does not hold for ASD algorithms in general. However, the monotonicity result is of interest since it justifies that the asymptotical performance analysis by letting $M \rightarrow \infty$ is indeed the "best" achievable result and it also verifies that increasing the cost will lead to at least no worse performance.

We need the following property of the function $T(S)$:

Lemma 10 *$T((a+1)x) \geq \frac{a+2}{a}T(x)$, if $x \geq K-1$ and a is a positive integer.*

Proof 19 *This lemma is similar to Theorem A-1, (A-9) in [39]. Following similar ideas, we give a simpler proof. Since $x \geq K-1$, we have:*

$$\left(1+\frac{l}{a}\right)(K-1) \leq x \leq \left(1+\frac{l+1}{a}\right)(K-1) \quad \textit{for} \quad l = 0, 1, 2, \cdots, a-1 \tag{5.41}$$

$$(a+l)(K-1) \leq ax \leq (a+l+1)(K-1) \tag{5.42}$$

$T(ax)$ can be computed as:

$$T(ax) = (a+l+1)\left[ax - \frac{a+l}{2}(K-1)\right] \tag{5.43}$$

On the other hand, $(a+1)x$ is in the following range as:

$$\frac{a^2+(l+1)a+l}{a}(K-1) \quad \leq (a+1)x \leq \quad \frac{a^2+(l+2)a+(l+1)}{a}(K-1) \tag{5.44}$$

$$(a+l+1)(K-1) \quad \leq (a+1)x \leq \quad (a+l+3)(K-1) \tag{5.45}$$

Since $T(S)$ is a piecewise linear function with monotonically increasing slope, $T(S) \geq (i+1)(S - \frac{i}{2}(K-1))$ for any non-negative integer i. Hence, we have the following *lower bound on $T((a+1)x)$:*

$$T((a+1)x) \geq (a+l+2)\left[(a+1)x - \frac{a+l+1}{2}(K-1)\right] \tag{5.46}$$

Combining (5.43) and (5.46), we have the following:

$$\begin{aligned} T((a+1)x) - \frac{a+2}{a}T(ax) &\geq (a+l+2)\left[(a+1)x - \frac{a+l+1}{2}(K-1)\right] \\ &\quad -\frac{a+2}{a}(a+l+1)\left[ax - \frac{a+l}{2}(K-1)\right] & (5.47) \\ &\geq -lx + \frac{(a+l+1)l}{a}(K-1) & (5.48) \\ &= \frac{l}{a}\left[(K-1)(a+l+1) - ax\right] \geq 0 & (5.49) \end{aligned}$$

where the final step in (5.49) follows by the fact that $l \geq 0$ and $ax \leq (K-1)(a+l+1)$.

Theorem 9 *Over the mixed channel, if a received codeword is decodable using ASD with multiplicity M, it is decodable under multiplicity $M+2$ (M has to be even in the proposed MAS), which means the performance of ASD under the proposed MAS is monotonic with multiplicity M.*

Proof 20 *If a codeword is decodable with multiplicity M, we have $T(S(M)) > C(M)$, where $S(M)$ and $C(M)$ are score and cost with multiplicity M respectively. Considering all types of symbols in the received codeword, we have the following relationship:*

$$\begin{aligned} S(M+2) &= \frac{M+2}{M}S(M) & (5.50) \\ C(M+2) &\leq \frac{(M+2)(M+3)}{M(M+1)}C(M) & (5.51) \end{aligned}$$

If a received codeword is decodable, according to Lemma 7, we have $S(M) >$

$M(K-1)$. *Therefore:*

$$\begin{aligned}
T(S(M+2)) &= T((M+2)\frac{S(M)}{M}) \\
&\geq \frac{(M+3)(M+2)}{(M+1)M}T(S(M)) \qquad (5.52)\\
&> \frac{(M+3)(M+2)}{(M+1)M}C(M) \\
&\geq C(M+2) \qquad (5.53)
\end{aligned}$$

where (5.52) is obtained by applying Lemma 10 twice and (5.53) is due to (5.51).

Note that the monotonicity property holds for all RS codes regardless of the rate.

E. Bit-Level Generalized Minimum Distance Decoding Algorithm

In this section, we develop a practical SDD algorithm for RS codes, which is motivated by the analytical results in the previous sections.

1. The Generic BGMD Algorithm

As shown in Section 3, the proposed MAS has a significantly larger decoding region than conventional BM and GS decoding over a mixed error and bit-level erasure channel. This provides the intuition that properly treating erasures at the bit-level will also help in RS soft-decision decoding over other channels. An efficient way to utilize erasures over many channels is by ordering the reliability values of the received bits, treating the LRB's as erasures and running an error and erasure decoder successively, namely generalized minimum distance (GMD) decoding [40]. In each iteration, the decoder can decode erasures in the LRB's together with some extra errors in the remaining most reliable bits (MRB's) as long as the error and erasure (e, f) pair is within the decoding region of BM algorithm. Due to the similarity between the proposed algorithm and conventional symbol-level GMD for RS codes, it is called bit-level GMD (BGMD).

The generic algorithm of BGMD is described in Algorithm 5.

Remark 3 *In terms of implementation, BGMD does not need to run ASD algorithm many times. In fact, the interpolation part can be shared between different erasure patterns. Similar to the techniques proposed in [35–38], we can generate all the candidate codewords in one interpolation round by applying factorization in the intermediate steps during the interpolation procedure. Besides, factorization needs to be performed only at outer corner (e, f) points. For high rate RS codes, the number of "test erasure patterns" of BGMD is the same as conventional symbol-level GMD.*

2. Performance Analysis of BGMD

Due to the simple structure of BGMD, the performance of BGMD for practical high rate RS codes over an additive white Gaussian noise (AWGN) channel can be tightly bounded using ordered statistics techniques. Define D(M) as the decoding region of the proposed ASD over a mixed bit-level error and erasure channel, namely the set of error and erasure (e, f) pairs that is decodable by the proposed ASD with multiplicity M as specified in Theorem 8. Let $f_{max,M}$ and $e_{max,M}$ be the maximum number of errors and erasures respectively such that $(0, f_{max,M})$ and $(e_{max,M}, 0)$ are still in D(M). The FER of BGMD can be upper bounded by the FER performance of using a set of bit-level error and erasure decoders, each with different number of erased bits f (from 0 to $f_{max,M}$) in the LRB's and a different error correction capability e such that $(e, f) \in D(M)$. Note, however, D(M) is the worst case decoding region of the proposed ASD, BGMD can in fact correct even more number of errors and erasures if some of the errors and erasures overlap in some symbols. However, for high rate

Algorithm 5 Bit-level Generalized Minimum Distance Decoding Based on Algebraic Soft Decision Decoding for Reed-Solomon Codes

```
Step1. Initialization: set the initial iteration round $j = 1$ and generate the log
    likelihood ratio (LLR) for each coded bit based on the channel observation $y_i$:
    $L_i = \frac{P(c_i=0|y_i)}{P(c_i=1|y_i)}$, for $i = 1, 2, \cdots, n$
Step2. Reliability Ordering: order the coded bits according to the absolute value of
    the LLR's $\{|L_i|\}$ in ascending order and record the ordering indices $\{s_i\}$.
Step3. Hard Decision: $\hat{c}_i = \begin{cases} 0, & L_i > 0; \\ 1, & L_i \leq 0. \end{cases}$
Step4. Multiplicity Assignment:
    In each symbol of the estimated vector $\underline{\hat{c}}$ assign multiplicities according to:
    1) if no bit is erased, assign M to the received symbol;
    2) if there is 1-bit erasure, assign M/2 to each candidate symbol;
    3) if there is more than 1-bit erasure, do not assign any multiplicity.
Step5. Algebraic Soft Decision Decoding: Run ASD according to the multiplicity assignment
    determined in Step4.. Keep the generated codewords in the decoding list.
Step6. Successive Erasure Generation in the Least Reliable Bits: $\hat{c}_{s_j} = \epsilon$
Step7. Iteration: If $j \leq n - k$ and ASD is yet able to correct the current erasures
    given no error, set $j \leftarrow j + 1$ and go to Step4. for another decoding iteration.
Step8. Final Decision: Output the most likely codeword in the decoding list. If
    there is no codeword in the list, a decoding failure is declared.
```

RS codes, this upper bound becomes tight, since the worst case error and erasure pattern dominates. Performance analysis of BGMD then boils down to bounding the performance of a conventional GMD decoder for binary codes [72] with a skewed decoding region D(M). Hence, upper bounds of GMD for binary codes, such as the one derived in [72], are directly applicable to evaluating the performance of BGMD decoding. For readers' convenience, we give the detailed procedure to compute the FER upper bound on BGMD algorithm in Appendix E. For more comprehensive studies on this bound, we refer interested readers to [72] and [73] for applications to other ordered statistics based decoding algorithms.

Due to this upper bound, the performance of BGMD in high SNR's, where RS codes operate in many practical systems, can be predicted analytically, which is beyond the capability of computer simulation. As an example, performance bound of BGMD over a popular high rate RS(255, 239) is plotted in Figure 25. At an FER = 10^{-14}, the upper bound of BGMD with M = 2 has a 0.8dB and 0.3dB gain over conventional BM and GMD upper bound respectively. With asymptotically large cost M = ∞, the gain of BGMD upper bound over BM increases to 1dB at an FER = 10^{-14}. On the other hand, the performance of KV algorithm can not be simulated at such a low FER. Compared with another popular SDD algorithm, i.e., the box and match algorithm (BMA) order-1 with 22 bits in the control band [45], the upper bound of BGMD with M = 2 has a 0.2dB gain at this FER level with a much smaller complexity and memory consumption than BMA. In high SNR's, the upper bound of BGMD with M = 2 also has comparable performance to the upper bound of Chase type 2 decoding [41] with 16 test error patterns. The performance gap of BGMD to a genie decoder with decoding radius $t = N - K$ becomes smaller and smaller as SNR increases. Note that the actual performance of BGMD may be even better than that predicted by the upper bound as will be shown in the

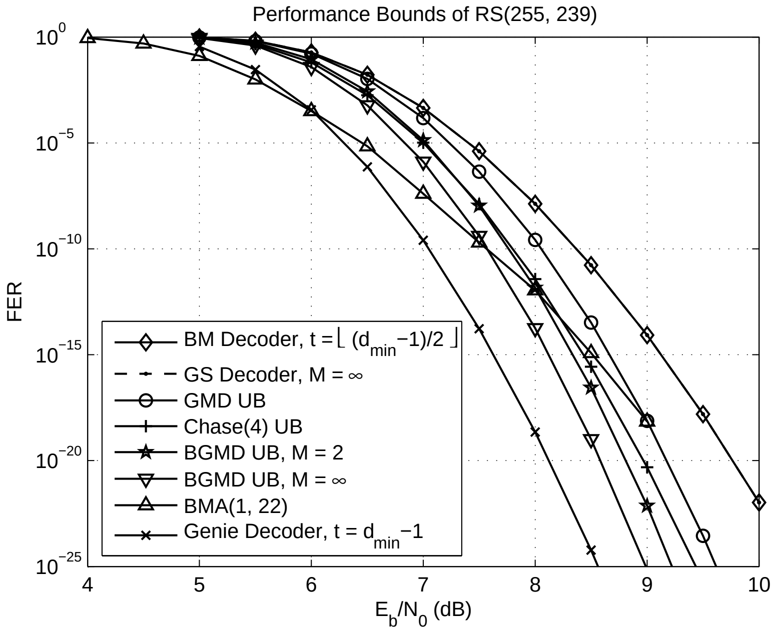


Fig. 25. Performance Bounds of Various Decodings for an RS (255, 239) Code over an AWGN Channel

simulation results in the following section. The FER upper bound on BGMD can be further tightened by considering the joint ordered statistics, which also increases computational complexity.

The generic BGMD algorithm can also be extended to incorporate Chase type decoding [37, 38, 41]. Under the proposed MAS, the corresponding performance can also be tightly upper bounded by similar bounding techniques using ordered statistics as shown in [73].

The proposed ASD algorithm can also be used for threshold based error and erasure decoding algorithm. We show the performance upper bounds of BMD based erasure decoding and ASD based erasure decodings for an RS(255, 239) over the AWGN channel in Figure 26. The erasure threshold is optimized numerically at each SNR point. It can be seen that bit-level ASD based erasure decoding significantly

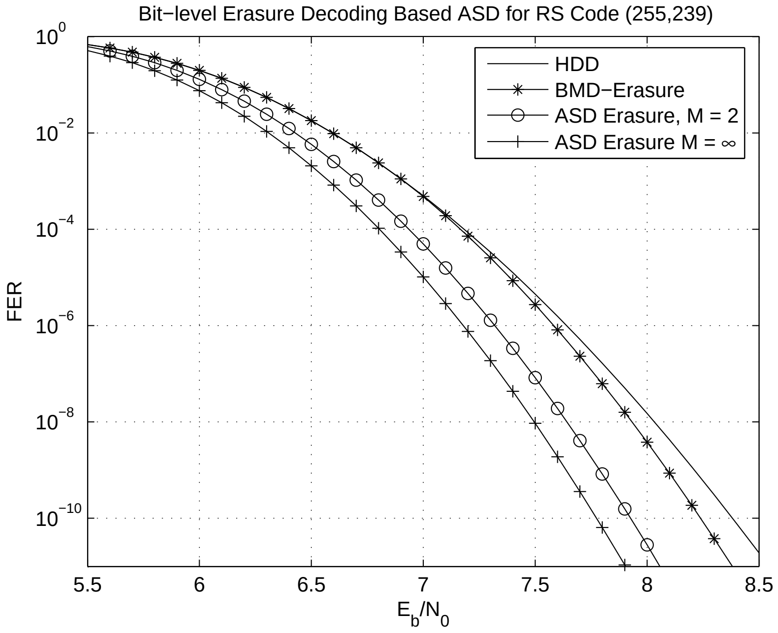


Fig. 26. Performance Bounds of Error and Erasure Decoding for an RS (255, 239) Code over an AWGN Channel

outperforms HDD, while the gain of BMD based erasure decoding over HDD is almost negligible. Compared with BGMD, the bit-level ASD based erasure decoding significantly reduce the computational complexity in reliability sorting and multiple factorization, while it also incurs slight performance degradation.

3. Discussions

We first discuss a counter-intuitive phenomenon of KV decoding, which was first observed in [74]. That is, KV decoder may fail even when the received vector does not contain any errors. We give an example as follows:

Example 1: Consider an RS(255, 239). Suppose in the received vector, no bit is in error. $255 \times 7 = 1785$ bits are perfectly received, i.e., the magnitude of their LLRs are all infinity. In each symbol, there is one bit that is corrupted by some noise, but still they have the correct signs. The probability that it is the transmitted bit is 0.7 and the probability that it is the wrong bit is 0.3. According to the KV algorithm, i.e., PMAS, we will have $S = 178.5M$ and $C \approx 73.95M^2$. It is easy to verify that even though there is no bit in error, the sufficient condition (5.8) will be violated. It can also be verified by actual simulation that KV will fail in some cases even when no bit is in error. In fact, this phenomenon was recently reassured in [75]. The analysis in [75] showed that under PMAS, the asymptotical decoding radius of ASD might be 0, which suggests the decoder can fail even though there is no error.

At first glance, this phenomenon seems counter-intuitive. It seems to suggest that soft information even degrades the performance. However, from the analysis in previous sections, we can get an intuitive and sensible interpretation. ASD in some sense treats weighted erasures, therefore, similar to erasure decoding over AWGN channels, in some cases, we may end up erasing too many correct bits and cause a decoding failure even though there is no error. On the other hand, since BGMD treats erasures according to the received reliability value and also erases bits successively, these abnormal cases will be excluded.

Besides, in general, the monotonicity of ASD is not guaranteed. For instance, it is observed in [33] that for the simplified KV algorithm, the decoding performance does not monotonically improve as the cost increases. For the proposed BGMD, on the other hand, as shown in the previous section, the decoding region will monotonically become larger as a function of the multiplicity number M.

The generic BGMD can naturally be generalized to take more than 1-bit erasures into account, which will be important in decoding medium to low rate RS codes. The associated performance bounds are also of great research interest, since for medium to low rate RS codes, the upper bound considering the worst case bit-level decoding region alone becomes loose.

F. Simulation Results

In this section, we show simulation results of the proposed BGMD over various communication channels. We will see that the proposed BGMD, though derived from a simple MAS, is superior to many existing MAS's which are far more complicated. Besides, in contrast to most MAS's in the literature, the ordered statistics based upper bound can accurately evaluate the actual performance of BGMD for many practical high rate RS codes.

In Figure 27, we plot the FER performance of an RS(31, 25) over an AWGN channel. BGMD ($M = 2$) outperforms conventional BM by 1.3dB at an $FER = 10^{-6}$. It also outperforms conventional symbol-level GMD by 0.6dB at an $FER = 10^{-5}$ and is slightly inferior to Combined Chase and GMD (CGA(3)), which has a much larger complexity, by 0.2dB. Compared with existing MAS's for ASD, it gives favorable performance as well. With $M = 2$, it even outperforms KV algorithm with $M = \infty$ by 0.5dB at an $FER = 10^{-6}$. With $M = \infty$, the performance of BGMD outperforms the performance of Gaussian approximation based MAS [25] and the performance of Chernoff technique based MAS [26, 76], which are far more complicated than the proposed BGMD in multiplicity assignment.

In Fig. 28 we evaluate the FER performance of a long code, RS(255,239) code. Again, BGMD ($M = 2$) outperforms GMD and is comparable to CGA(3). As the codeword length increases, KV algorithm becomes asymptotically optimal as shown in [4]. The performance of the proposed BGMD is still comparable to KV decoding. In the infinite cost case, the performance of BGMD ($M = \infty$) is slightly better than the performance of KV ($M = \infty$); in the finite cost case, BGMD ($M = 2$) even outperforms KV ($M = 4.99$). Besides, since BGMD only assigns multiplicities to symbols with at most 1-bit erasure, the memory consumption in storing the assigned

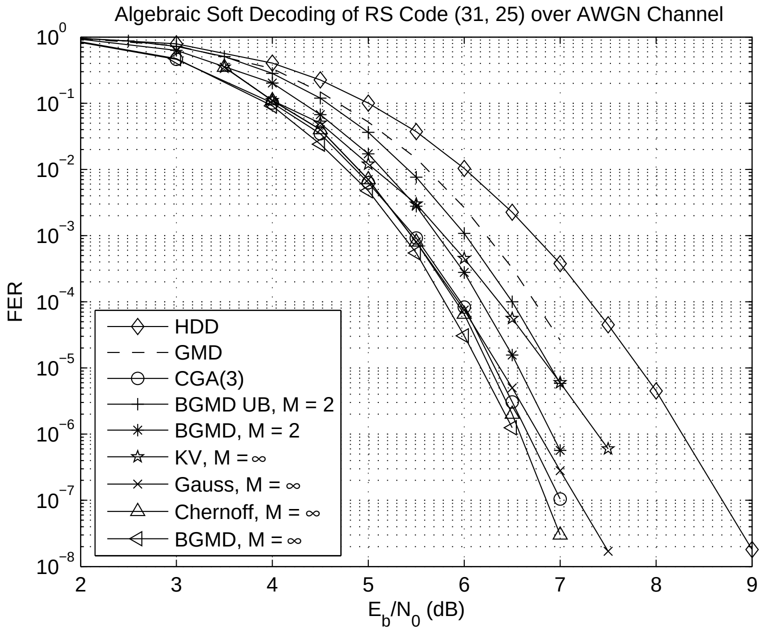


Fig. 27. Performance of ASD for an RS (31, 25) Code over an AWGN Channel

multiplicities is much smaller than KV. The upper bound is quite tight and it starts to outperform KV (M = 4.99) and is only 0.1dB inferior to the actual performance at an FER = 10^{-5}. As shown in Figure 25, it gives an estimate of the performance of BGMD in high SNR's as well.

Though the upper bound of BGMD is tight only for medium to high rate RS codes, the proposed BGMD algorithm actually provides even more significant coding gain for low rate RS codes. As shown in Figure 29, the performance of BGMD (M = 2) can outperform BM, GMD, CGA(3) decoding by a large margin for an RS(63, 12) code over an AWGN channel. The gain of BGMD over BM is about 2dB at an FER = 10^{-4}. In this case, CGA(3) is far more inferior to BGMD. BGMD (M = 2) has almost identical performance as KV (M = 4.99). While, in the infinite cost case, KV does have a 0.4dB gain over BGMD at an FER = 10^{-5}, which suggests that taking

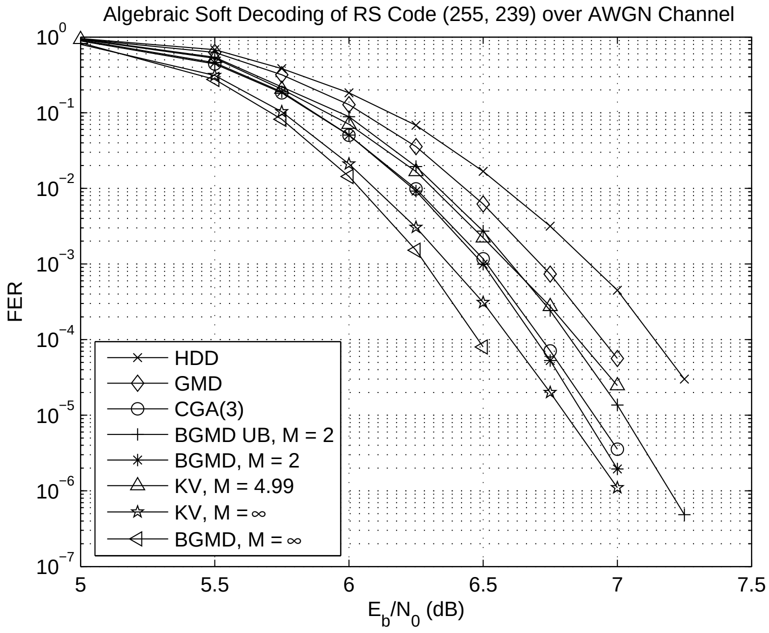


Fig. 28. Performance of ASD for an RS (255, 239) Code over an AWGN Channel

care of more than 1-bit-erased symbols might provide extra gains for low rate RS codes. It is an interesting open problem to develop such kind of MAS.

The gain of the proposed BGMD over BM and CGA becomes larger when the channel is similar to a BEC, say Rayleigh fast fading channels, since BGMD can correct a significantly larger number of bit-level erasures than conventional BM as discussed in Section B. As shown in Figure 30, the gain of BGMD (M = 2) is about 1.5dB compared with BM at an FER = 10^{-3}. As expected, the gain of BGMD over CGA(3) is more significant over the fading channel. Compared with KV (M = ∞), BGMD (M = 2) is slightly inferior to KV (M = ∞) in low SNR's, but it intersects KV (M = ∞) at an FER = 10^{-3} and performs better in high SNR's. BGMD (M = ∞) has a 0.75dB gain over KV (M = ∞) at an FER = 10^{-4}. The superior performance of BGMD seems to suggest that for high rate RS codes, efficiently taking advantage

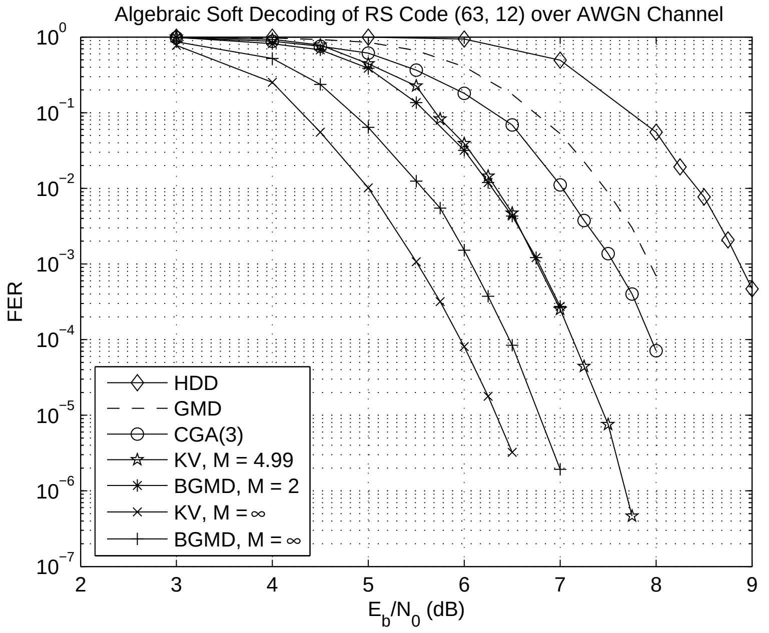


Fig. 29. Performance of ASD for an RS (63, 12) Code over an AWGN Channel

of bit-level erasures exploits most of the gain in ASD.

Performance of the proposed BGMD is also investigated over practical magnetic recording channels, that is, longitudinal channels (see Figure 31) and perpendicular channels (see Figure 32). More details of the channel model can be found in [69]. Similar performance gains of BGMD have also been observed over practical recording channels. BGMD (M = 2) outperforms conventional GMD and performs competitively with KV and CGA(3), which are much more complex. This superior performance of BGMD suggests that though RS codes are usually considered as a powerful burst error correction code, it is still beneficial to taking advantage of soft information at the bit-level even over practical magnetic recording channel models, where errors are usually bursty.

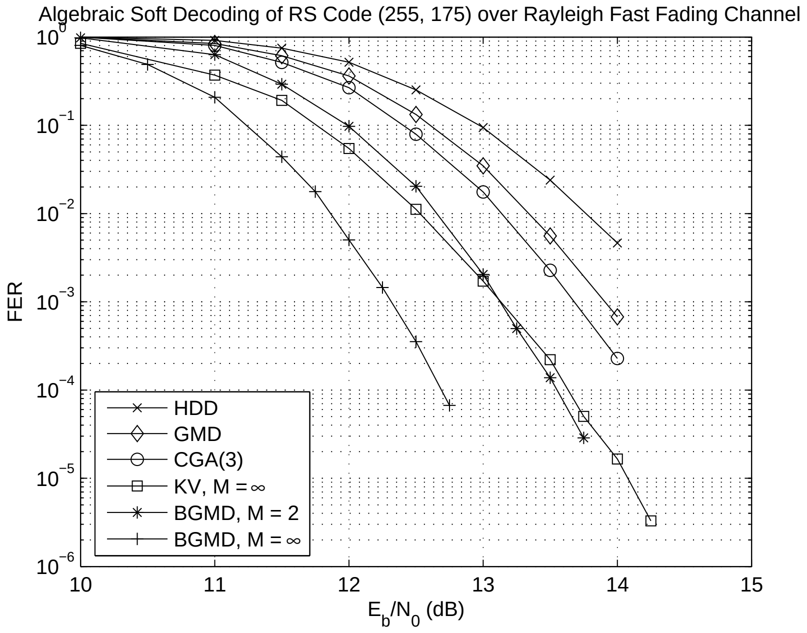


Fig. 30. Performance of ASD for an RS (255, 175) Code over a Rayleigh Fast Fading Channel

G. Conclusion

We have presented multiplicity assignment strategies and performance analyses of algebraic soft-decision decoding over erasure channels, binary symmetric channels and mixed error and bit-level erasure channels. Performance analysis motivates a simple sequential multiplicity assignment scheme, bit-level generalized minimum distance decoding. The proposed BGMD outperforms most of the MAS's in the literature for RS codes in a wide range of rates over various channels both in terms of performance and complexity. Due to its simplicity, the performance of BGMD can also be tightly bounded using ordered statistics based upper bounds even in high SNR's over an AWGN channel. The proposed BGMD has potential applications in decoding RS codes in practical recording systems and RS outer codes in concatenated systems.

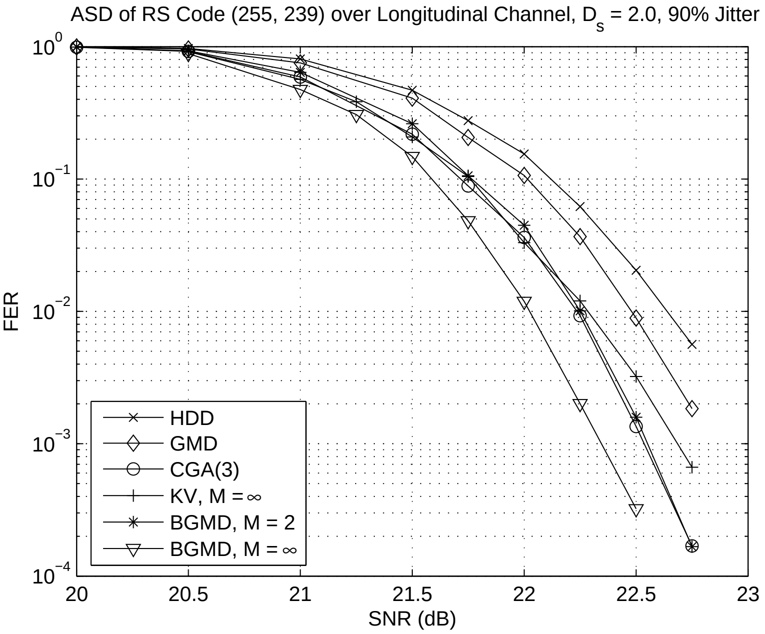


Fig. 31. Performance of ASD for an RS (255, 239) Code over a Longitudinal Recording Channel, (D_s = 2.0, 90% Jitter)

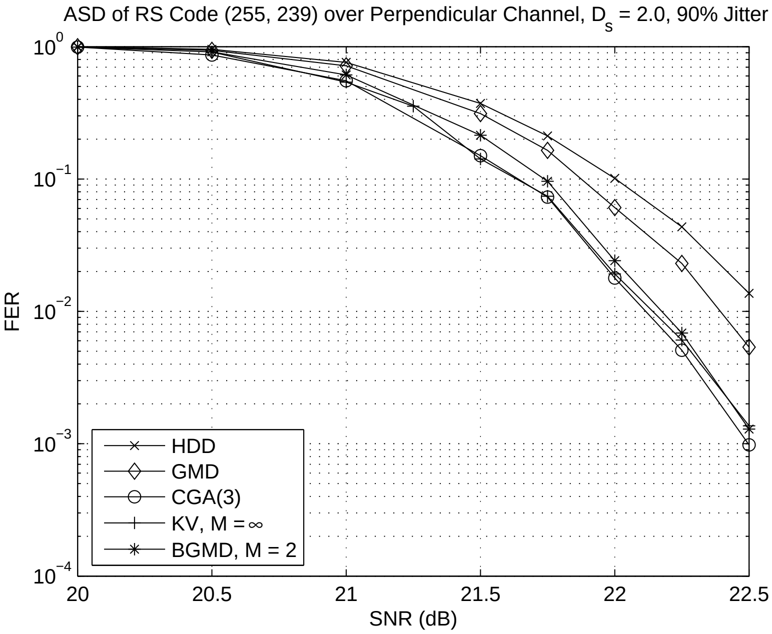


Fig. 32. Performance of ASD for an RS (255, 239) Code over a Perpendicular Recording Channel ($D_s = 2.0$, 90% Jitter)

CHAPTER VI

SUMMARY AND FUTURE WORKS

This dissertation studied several advanced channel decoding methods using bit-level soft information. In particular, we proposed bit-level iterative decoding for HDPC codes and bit-level algebraic soft-decision list decoding for RS codes. We have demonstrated that bit-level SDD is a more efficient way to exploit the potential gain of classical algebraic codes, such as RS codes than traditional symbol-level SDD schemes in terms of both performance and complexity. In this chapter, we summarize the main contributions of this dissertation and discuss potential future works.

A. Iterative Decoding of High Density Parity Check Codes

In Chapter III, a gradient descent based iterative algorithm for SISO decoding of HDPC codes has been proposed. The proposed iterative algorithm uses the SPA in conjunction with a binary parity check matrix adapted in each decoding iteration according to the bit-level reliabilities. For codes with non-sparse parity check matrices, this bit-level reliability based adaptation procedure significantly improves the convergence behavior of the gradient descent based iterative decoding algorithm compared to iterative decoding algorithms without adaptation. This algorithm is the first successful iterative decoding algorithm that can achieve a significant gain over conventional HDD for practical high rate long RS codes.

In Chapter IV, a stochastic shift based iterative decoding (SSID) scheme for cyclic codes has been proposed. In contrast to the adaptive scheme in Chapter III, we have shown that stochastic shift of the updated reliability values at each iteration can also prevent gradient descent based iterative decoding from getting stuck for short length cyclic codes. The stochastic shift based adaptation scheme is much less complex than the Gaussian elimination based adaptation scheme in Chapter IV.

B. Bit-level Algebraic Soft-decision List Decoding

In Chapter V, a novel multiplicity assignment strategy for ASD based on bit-level soft information has been presented. It is shown that by carefully incorporating bit-level soft information in multiplicity assignment and interpolation, ASD can significantly outperform HDD for practical high rate long RS codes even with a very small amount of complexity. More importantly, the proposed bit-level ASD is the first list decoding method whose performance is provably better than HDD even for high rate RS codes. The proposed bit-level ASD algorithm is potentially a feasible alternative to HDD in many practical systems.

C. Future Works

Bit-level advanced channel coding techniques developed in this dissertation also have several promising future research directions.

First, one possible extension of the proposed iterative decoding by adapting the parity check matrix is to use the adaptive algorithm to improve the iterative decoding performance of LDPC codes in the error floor region. The other potential application is to use the adaptive algorithm to help the convergence behavior of an LDPC code as a vector quantizer.

Second, iterative decoding of RS codes has wide applications in many concatenated systems. In the recent DVB-H standard, product RS codes are used for forward error control (FEC) over wireless channels. However, the performance of this concatenated system largely depends on the decoding scheme. Naively treating undecodable RS inner codewords as erasures might lead to very poor performance. Also, due to the MDS property of RS codes, efficient decoding algorithms for RS codes over non-ergodic channels such as orthogonal-frequency-division-multiplexing systems [77] and multiple antenna systems [78] are also of great research value. Effectively taking advantage of the soft-information at the bit-level can provide a significant gain in such systems.

Third, a promising future direction of the proposed bit-level ASD scheme is to implement ASD using classical BMD for errors and bit-level erasure decoding. Sidorenko [79] proposed using BMD to achieve the error correction radius of Sudan decoding. It will be very interesting to extend Sidorenko's result to more general ASD case for high rate RS codes. It is also of practical value to study decoding folded/interleaved RS codes [59] at the bit-level, which is of practical value in channels with memory.

Fourth, bit-level channel coding techniques also have applications in cross-layer receiver design. In packet oriented networks, erasure decoding are usually adopted for FEC. Unlike in wired case, where packet-erasures are mainly due to packet loss in the network and can not be recovered, in the wireless scenario, the soft information of each bit is usually available at the receiver. Treating all the bits in an undecodable packet as erasures is a waste of information. Therefore, it is beneficial to study bit-level soft decoding of many FEC codes adopted in the transport layer, such as RS codes and more recent digital fountain codes [80]. How to effectively take advantage of the soft-information at the bit-level and how to analyze the decoding performance is a challenging and rewarding open problem.

REFERENCES

[1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near shannon limit error-correcting and decoding: Turbo codes," in *Proc. of Int. Conf. Commun. (ICC)*, Geneva, Swizerland, May 1993, pp. 1064–1070.

[2] R.G. Gallager, *Low-Density Parity-Check Codes*, MIT Press, Cambridge, MA, 1963.

[3] M. Sudan, "Decoding of Reed-Solomon beyond the error-correction bound," *Journal of Complexity*, vol. 13, pp. 180–193, Sep. 1997.

[4] R. Koetter and A. Vardy, "Algebraic soft-decision decoding of Reed-Solomon codes," *IEEE Trans. Information Theory*, vol. 49, pp. 2809–2825, Nov. 2003.

[5] G. Reed and I. Solomon, "Polynomial codes over certain finite fields," *Journal of the Society for Industrial and Applied Mathematics*, vol. 8, pp. 300–304, Jun. 1960.

[6] R. C. Bose and D. K. Ray-Chaudhuri, "On a class of error correcting binary group codes," *Information and Control*, vol. 3, pp. 68–79, Mar. 1960.

[7] D. Gorenstein and N. Zierler, "A class of error correcting codes in p^m symbols," *Journal of the Society of Industrial and Applied Mathematics*, vol. 9, pp. 207–214, Jun. 1961.

[8] S. B. Wicker, *Error Control Systems for Digital Communication and Storage,* 1^{st} Edition, Prentice Hall, Upper Saddle River, NJ, USA, Jan 1995.

[9] W. Peterson, "Encoding and error-correction procedures for Bose-Chaudhuri codes," *IRE Transactions on Information Theory*, vol. 6, pp. 459–470, Sep. 1960.

[10] E. Berlekamp, "Nonbinary BCH decoding," *IEEE Transactions on Information Theory*, vol. 14, pp. 242, Mar. 1968.

[11] J. Massey, "Shift register synthesis and BCH decoding," *IEEE Transactions on Information Theory*, vol. 15, pp. 122–127, Jan. 1968.

[12] Y. Sugiyama, Y. Kasahara, S. Hirasawa, and T. Namekawa, "A method for solving kep equations for Goppa codes," *Information and Control*, vol. 27, pp. 87–99, 1975.

[13] R. E. Blahut, *Theory and Practice of Error Control Codes*, Addison-Wesley, Inc., Boston, MA, 1984.

[14] L. R. Welch and E. R. Berlekamp, "Error correction for algebraic block codes," Dec. 1986, US Patent 633 470.

[15] G. D. Forney, "On decoding BCH codes," *IEEE Transactions on Information Theory*, vol. 11, pp. 549–557, Oct. 1965.

[16] Shu Lin and D. J. Costello, *Error Control Coding: Fundamentals and Applications*, 1^{st} Edition, Prentice Hall, Upper Saddle River, NJ, USA, 1983.

[17] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Trans. Information Theory*, vol. 42, pp. 429–445, Mar. 1996.

[18] R. Lucas, M. Bossert, and M. Breitbach, “On iterative soft-decision decoding of linear binary block codes and product codes,” *IEEE Journal of Selected Areas in Communication*, vol. 16, pp. 276–296, Feb. 1998.

[19] J. S. Yedidia, J. Chen, and M. Fossorier, “Generating code representations suitable for belief propagation decoding,” in *Proc. Allerton*, Monticello, IL, Oct. 2002.

[20] J. S. Yedidia, W. T. Freeman, and Y. Weiss, “Constructing free energy approximations and generalized belief propagation algorithms,” Tech. Rep. 35, Mitsubishi Electrical Research Lab, Oct. 2002, available online at http://www.merl.com/papers/TR2002-35/.

[21] G. Ungerboeck, “Iterative soft decoding of Reed-Solomon codes,” in *Proc. ISTC*, Brest, France, Sep. 2003.

[22] J. S. Yedidia, “Sparse factor graph representations of Reed-Solomon and related codes,” in *Proc. ISIT*, Chicago, IL, Jun. 2004.

[23] W. Berlekamp, “Bounded distance+1 soft-decision Reed-Solomon decoding,” *IEEE Transactions on Information Theory*, vol. 42, pp. 704–720, May. 1996.

[24] V. Guruswami and M. Sudan, “Improved decoding of Reed-Solomon and algebraic-geometry codes,” *IEEE Trans. Information Theory*, vol. 45, pp. 1757–1767, Sep. 1999.

[25] F. Parvaresh and A. Vardy, “Multiplicity assignments for algebraic soft-decoding of Reed-Solomon codes,” in *Proc. ISIT*, Yokohoma, Japan, Jul. 2003, p. 205.

[26] M. El-Khamy and R. J. McEliece, “Interpolation multiplicity assignment algorithms for algebraic soft-decision decoding of Reed-Solomon codes,” in *Proc.*

DIMACS, 2005, vol. 68.

[27] N. Ratnakar and R. Koetter, "Exponential error bounds for algebraic soft-decision decoding of Reed-Solomon codes," *IEEE Trans. Information Theory*, vol. 51, pp. 3899–3917, Nov. 2005.

[28] R. Koetter and A. Vardy, "Decoding of Reed-Solomon codes for additive cost functions," in *Proc. ISIT*, Lausanne, Switzerland, Jun. 2002, p. 313.

[29] J. Jiang and K. R. Narayanan, "Performance analysis of algebraic soft decoding of Reed-Solomon codes over binary symmetric and erasure channels," in *Proc. ISIT*, Adelaide, Australia, Sep. 2005.

[30] J. Justesen, "Soft-decision decoding of RS codes," in *Proc. ISIT*, Adelaide, Australia, Sept. 2005.

[31] R. Koetter, "On algebraic decoding of algebraic-geometric and cyclic codes," Ph.D. dissertation, Univ. Linkoping, Sweden, 1996.

[32] R. Roth and G. Ruckenstein, "Efficient decoding of Reed-Solomon codes beyond half the minimum distance," *IEEE Transactions on Information Theory*, vol. 46, pp. 246–257, Jan. 2000.

[33] W. J. Gross, F. R. Kschischang, R. Koetter, and P. G. Gulak, "Applications of algebraic soft-decision decoding of Reed-Solomon codes," *IEEE Trans. Communication*, vol. 54, pp. 1224–1234, Jul. 2003.

[34] W. J. Gross, F. R. Kschischang, R. Koetter, and P. G. Gulak, "Towards a VLSI architecture for interpolation-based soft-decision Reed-Solomon decoders," *Journal of VLSI Signal Processing*, vol. 39, pp. 93–111, Jan.-Feb. 2005.

[35] A. Ahmed, R. Koetter, and N. Shanbhag, "Reduced complexity interpolation for soft decoding of Reed-Solomon codes," in *Proc. ISIT*, Chicago, IL, Jun. 2004.

[36] H. Xia and J. R. Cruz, "A reliability-based forward recursive algorithm for algebraic soft-decision decoding of Reed-Solomon codes," in *Proc. ISITA*, Parma, Italy, Oct. 2004.

[37] H. Xia, "Error-correction coding for high-density magnetic recording channels," Ph.D. dissertation, University of Oklahoma, Norman, OK, USA, 2004.

[38] J. Bellorado and A. Kavcic, "A low-complexity method for Chase-type decoding of Reed-Solomon codes," in *Proc. ISIT*, Seattle, WA, Jul. 2006.

[39] R. J. McEliece, "The Guruswami-Sudan decoding algorithm for Reed-Solomon codes," *IPN Progress Report*, , no. 42-153, May. 2003.

[40] G. D. Forney, "Generalized minimum distance decoding," *IEEE Trans. Information Theory*, vol. 12, pp. 125–131, Apr. 1996.

[41] D. Chase, "Class of algorithms for decoding block codes with channel measurement information," *IEEE Trans. Information Theory*, vol. 18, pp. 170–182, Jan. 1972.

[42] H. Tang, Y. Liu, M. Fossorier, and S. Lin, "Combining Chase-2 and GMD decoding algorithms for nonbinary block codes," *IEEE Communication Letters*, vol. 5, pp. 209–211, May. 2000.

[43] M. P. C. Fossorier and S. Lin, "Soft-decision decoding of linear block codes based on ordered statistics," *IEEE Trans. Information Theory*, vol. 41, pp. 1379–1396, Sep. 1995.

[44] T. Hu and S. Lin, "An efficient hybrid decoding algorithm for Reed-Solomon codes based on bit reliability," *IEEE Trans. Communications*, vol. 51, pp. 1073–1081, Jul. 2003.

[45] M. P. C. Fossorier and A. Valembois, "Reliability-based decoding of Reed-Solomon codes using their binary image," *IEEE Communication Letters*, vol. 7, pp. 452–454, Jul. 2004.

[46] Y. Wu, "Coding techniques for linear block codes with applications to fault identification," Ph.D. dissertation, University of Illinois Urbana Champaign, Urbana, IL, Aug. 2004.

[47] Y. Wu, R. Koetter, and C. Hadjicostis, "Soft-decision decoding of linear block codes using preprocessing," in *Proc. ISIT*, Chicago, IL, Jun. 2004.

[48] W. Jin and M. P. C. Fossorier, "Reliability-based soft-decision decoding with multiple biases," *IEEE Transactions on Information Theory*, vol. 53, pp. 105–120, Jan. 2007.

[49] W. Jin and M. P. C. Fossorier, "Enhanced box and match algorithm for reliability-based soft-decision decoding of linear block codes," in *Proc. Globecom*, San Francisco, CA, USA, Nov. 2006.

[50] J. Hagenauer, "The turbo principle: Tutorial introduction and state of the art," in *Proc. Int. Symp. Turbo Codes & Related Topics*, Brest, France, Sept. 1997, pp. 1–11.

[51] M. K. Cheng and P. H. Siegel, "Iterative soft-decision Reed-Solomon decoding on partial response channels," in *Proc. Global Comm. Conf.*, San-Francisco, CA, Dec. 2003.

[52] A. Vardy and Y. Be'ery, "Bit-level soft-decision decoding of Reed-Solomon codes," *IEEE Trans. Communications*, vol. 39, pp. 440–444, Mar. 1991.

[53] V. Ponnampalam and B. Vucetic, "Soft decision decoding of Reed-Solomon codes," *IEEE Trans. Communications*, vol. 50, pp. 1758–1768, Nov. 2002.

[54] V. Guruswami and A. Vardy, "Maximum-likelihood decoding of Reed-Solomon codes is NP-hard," TR 04-40, ECCC Report, April 2004.

[55] C. Retter, "The average weight-distance enumerator for binary expansions of Reed-Solomon codes," *IEEE Trans. Information Theory*, vol. 48, pp. 1195–1200, Mar. 2002.

[56] M. El-Khamy and R. J. McEliece, "Bounds on the average binary minimum distance and the maximum likelihood performance of Reed-Solomon codes," in *Proc. Allerton*, Monticello, IL, Oct. 2004.

[57] D. Divsalar, "A simple tight bound on error probability of block codes with application to turbo codes," TR 42-139, TMO Progress Report, Nov. 1999.

[58] S. Dolinar, D. Divsalar, and F. Pollara, "Code performance as a function of block size," TR 42-133, TMO Progress Report, May. 1998.

[59] F. Parvaresh and A. Vardy, "Multivariate interpolation decoding beyond the Guruswami-Sudan radius," in *Proc. Allerton*, Monticello, IL, Oct. 2004.

[60] K. Farrell, L. Rudolph, C. Hartmann, and L. Nielsen, "Decoding by local optimization (corresp.)," *IEEE Trans. Information Theory*, vol. 29, pp. 740–743, sep. 1983.

[61] R. J. McEliece, D. J. C. MacKay, and J. F. Cheng., "Turbo decoding as an instance of pearl's "belief propagation" algorithm," *IEEE Journal on Selected Areas in Comm.*, vol. 16, no. 2, pp. 140–152, Feb. 1998.

[62] P. Elias, "Coding for two noisy channels," in *Information Theory, Third London Symposium*, 1955.

[63] M. P. C. Fossorier, "Reliability-based soft-decision decoding with iterative information set reduction," *IEEE Trans. Information Theory*, vol. 48, pp. 3101–3106, Dec. 2002.

[64] M. P. C. Fossorier, "Iterative reliability-based decoding of low-density parity check codes," *IEEE Journal of Selected Areas in Communication*, vol. 19, pp. 908–917, May. 2001.

[65] M. El-Khamy and R. J. McEliece, "Iterative algebraic soft decision decoding of Reed-Solomon codes," in *Proc. ISITA*, Parma, Italy, Mar. 2004, pp. 1456 – 1461.

[66] A. Ahmed, R. Koetter, and N. R. Shanbhag, "Performance analysis of the adaptive parity check matrix based soft-decision decoding algorithm," in *Proc. Asilomar Conf. Signals, Systems, and Computers*, Pacific Grove, CA, Nov. 2004.

[67] M. El-Khamy and R. J. McEliece, "Iterative algebraic soft-decision list decoding of reedcsolomon codes," *IEEE Journal on Selected Areas in Communications*, vol. 24, pp. 481–490, Mar. 2006.

[68] W. Jin and M. P. C. Fossorier, "Iterative biased reliability-based decoding of binary linear codes," in *Proc. IEICE and SITA Joint Conference on Info. Theory*, Honolulu, HW, May. 2005.

[69] X. Yang and E. M. Kurtas, “Signal and noise generation for magnetic recording channel simulations,” *Coding and Signal Processing for Magnetic Recording Systems*, CRC Press, New York, pp. 6.1–6.20, 2004.

[70] Y. Kou, J. Xu, H. Tang, S. Lin, and K. Abdel-Ghaffar, “On circulant low density parity check codes,” in *Proc. ISIT*, Lausanne, Switzerland, Jun. 2002.

[71] R. D. Wesel, X. Liu, and W. Shi, “Trellis codes for periodic erasures,” *IEEE Trans. Communications*, vol. 48, pp. 938–947, Jun. 2000.

[72] D. Agrawal and A. Vardy, “Generalized minimum distance decoding in Euclidean space: Performance analysis,” *IEEE Trans. Information Theory*, vol. 46, pp. 60–83, Jan. 2000.

[73] M. P. C. Fossorier and S. Lin, “Error performance analysis for reliability based decoding algorithms,” *IEEE Trans. Information Theory*, vol. 48, pp. 287–293, Sep. 2002.

[74] U. Assaf, “Aspects in soft decision decoding of Reed-Solomon codes,” M.S. thesis, Israel Institute of Technology, Haifa, Israel, 2004.

[75] A. Duggan and A. Barg, “Performance analysis of algebraic soft-decision decoding of Reed-Solomon codes,” in *Proc. Allerton*, Monticello, IL, Sept. 2006.

[76] M. El-Khamy, R. J. McEliece, and J. Harel, “Performance enhancements for algebraic soft-decision decoding of Reed-Solomon codes,” in *Proc. ISIT*, Chicago, IL, Jun. 2004.

[77] Jun Zheng, “Analysis of coded OFDM system over frequency-selective fading channels,” M.S. thesis, Texas A & M University, July 2003.

[78] Jun Zheng, "Design and analysis of multiple antenna systems with practical channel state information assumptions," Ph.D. dissertation, University of California San Diego, July 2006.

[79] G. Schmidt, V. R. Sidorenko, and M. Bossert, "Decoding Reed-Solomon codes beyond half the minimum distance using shift-register synthesis," in *Proc. ISIT*, Seattle, WA, Jul. 2006.

[80] A. Shokrollahi, "Raptor codes," *IEEE Trans. on Information Theory*, vol. 52, pp. 2551–2567, Jun. 2006.

[81] A. Papoulis, *Probability, Random Variables and Stochastic Processes*, McGraw-Hill, 3rd edition, 1991, New York.

APPENDIX A

UNIFORM MULTIPLICITY ASSIGNMENT AMONG CANDIDATES SYMBOLS IN THE SAME COORDINATE

The optimal multiplicity assignment for a given channel observation $\underline{y}$ minimizes the probability of codeword error, that is:

$$\mathbf{M}_{opt} = \underset{\mathbf{M}}{\operatorname{argmin}} P\{\hat{C} \neq C|\underline{y}, \mathbf{M}\} \tag{A.1}$$

However it is highly nontrivial to take the structure of RS codes into account in the multiplicity assignment stage [4]. For instance, over BEC's and BSC's, the RS codeword has to be represented using its binary image expansion and then these bits are transmitted through the channel. However, even the weight enumerators of the binary image expansions of RS codes are in general not known [55], let alone taking the code structure into account in the MAS. Therefore, some assumptions are needed to make the problem analytically tractable. In the literature, the coset code argument [4, 26] and the dither argument [27] have been used to obtain the optimal MAS for the average ensemble performance over all RS coset codes (or, equivalently, all dither patterns).

Throughout this dissertation, we also use the coset code argument, that is, we assume the symbols in a transmitted codeword are independent and identically distributed (i.i.d) with a uniform distribution over GF(q) during the multiplicity assignment stage. However, the MAS is designed to be optimal for the worst case RS coset code. Consequently, it is natural to consider uniform multiplicity assignment among equiprobable candidates in the same coordinate. Since the channel only generates erasures, there is no preference for some candidates over others. Suppose non-uniform multiplicities are assigned, in the worst case, the transmitted symbol will correspond to the candidate with the least multiplicity and therefore, non-uniform multiplicity assignment will have a strictly smaller score than uniform multiplicity assignment with the same cost.

APPENDIX B

DERIVATION OF THE BIT-LEVEL RADIUS A 1-BIT FLIPPED BSC

Suppose there are $e \leq N$ 1-bit flipped symbols. In the MAS, we assign M to the received vector and tM to the 1-bit flipped neighbors. As $M \rightarrow \infty$, the score and cost are:

$$S = (N-e)M + eMt \tag{B.1}$$

$$C = \frac{N}{2}[M^2(1+mt^2)(1+o(1))] \tag{B.2}$$

Plugging the score and cost into (5.8), we get:

$$[(N-e)+et]M > \sqrt{(K-1)N(1+mt^2)}M \tag{B.3}$$

$$e < \frac{N - \sqrt{N(K-1)(1+mt^2)}}{1-t} \tag{B.4}$$

For RS codes of rate $R < \frac{1}{1+m} + \frac{1}{N}$, we have $(K-1)(1+m) < N$. Setting $t = 1$ in (B.3), the inequality becomes independent of e and is always satisfied. In this case, the transmitted codeword will always be on the list.

For higher rate RS codes, t is optimized to maximize the right hand side (RHS) of (B.4). This problem is equivalent to maximizing the slope between a given point $(1, N)$ and a point on the hyperbola $\frac{y^2}{N(K-1)} - mx^2 = 1$, within the range $0 \leq x \leq 1$ and $y \geq \sqrt{N(K-1)}$, which is nothing but the tangent to the hyperbola. For the tangential point (x_0, y_0), we have the following relationships:

$$\frac{y_0^2}{N(K-1)} - mx_0^2 = 1 \tag{B.5}$$

$$\frac{dy}{dx}\,|_{x=x_0} = N(K-1)m\frac{x_0}{y_0} \tag{B.6}$$

$$= \frac{N-y_0}{1-x_0} \tag{B.7}$$

From the above three equations, we can get:

$$y_0 = (K-1)(mx_0+1) \tag{B.8}$$

Plugging back to (B.5), we get

$$m\left[m(K-1)-N\right]x_0^2 + 2m(K-1)x_0 - \left[N-(K-1)\right] = 0 \tag{B.9}$$

Since we are only interested in $x_0 \in [0,1]$, it is easy to verify that in all cases, the solution of (B.9) will be of the following form:

$$x_0 = \frac{-m(K-1)+\sqrt{\Delta}}{m(m(K-1)-N)} \tag{B.10}$$

where $\Delta = (m(K-1))^2 + (N-K+1)(m^2(K-1)-mN)$. Note that the singular point $m(K-1)-N=0$ can be removed by taking the limit: $[m(K-1)-N] \to 0$. Combing (B.6) and (B.8), the optimal error correction radius is:

$$e_{max} < \frac{N}{\frac{1}{mx_0}+1} \tag{B.11}$$

where x_0 is computed in (B.10). The maximum e_{max} satisfying (B.11) is the error correction radius of ASD algorithm under the asymptotically optimal MAS over 1-bit flipped BSC and $t = x_0$ is the optimal multiplicity coefficient.

Moreover, $\sqrt{\Delta}$ can be further bounded as follows:

$$\sqrt{\Delta} < m(K-1)[1 + \frac{1}{2}\frac{(N-K+1)(m^2(K-1)-mN)}{m^2(K-1)^2}] \tag{B.12}$$

For high rate RS codes, $(N-K+1)(m^2(K-1)-mN) \ll (m(K-1))^2$, the left hand side (LHS) and RHS of (B.12) becomes very close and the upper bound on $\sqrt{\Delta}$ becomes tight. Plug (B.12) into (B.10):

$$\tilde{x}_0 = \frac{N-(K-1)}{2m(K-1)} \tag{B.13}$$

Plug (B.13) into (B.11), we finally get:

$$\tilde{e}_{max} = \frac{N(N-K+1)}{N+(K-1)} = \left[N - \sqrt{N(K-1)}\right]\left(1 + \frac{\sqrt{N(K-1)}-(K-1)}{N+(K-1)}\right) \tag{B.14}$$

Note that $\tilde{e}_{max} > e_{max}$ and it serves as an upper bound on the true decoding radius. However, (B.14) suggests that in the 1-bit flipped BSC case, the improvement of ASD over GS algorithm is very little for high rate RS codes. A similar result was independently obtained in [30].

APPENDIX C

DERIVATION OF THE DECODING REGION OF ASD OVER A 1-BIT FLIPPED OR ERASED CHANNEL

We consider the following MAS for the 1-bit flipped or erased channel: if the symbol does not contain erased bits, assign multiplicity M to the received symbol and Mt_1 to all 1-bit flipped neighbors. In the 1-bit erased symbols, we assign Mt_2 to both candidate symbols.

Suppose we have f erasures and e errors. The optimal MAS is such that given f, we want to maximize e.

In the infinite cost case, the score and the cost are:

$$S = (N-e-f)M + eMt_1 + fMt_2 \tag{C.1}$$

$$C = \frac{1}{2}\left[(N-f)(M^2+mM^2t_1^2)+2fM^2t_2^2\right](1+o(1)) \tag{C.2}$$

When a received vector is decodable in the infinite cost case, (5.8) has to be satisfied. We have:

$$e < \frac{N-f(1-t_2)-\sqrt{(K-1)\left[(N-f)(1+mt_1^2)+2ft_2^2\right]}}{1-t_1} \tag{C.3}$$

When $f = 0$, (C.3) reduces to (B.4). Here, we only consider the non-trivial case, $f > 0$. Define

$$J_1 = N-f(1-t_2)-\sqrt{(K-1)\left[(N-f)(1+mt_1^2)+2ft_2^2\right]} \tag{C.4}$$

$$J = \frac{J_1}{1-t_1} \tag{C.5}$$

We first maximize J_1 with respect to t_2. Take the derivative, we get:

$$g(t_2) = \frac{\partial J_1}{\partial t_2} = f - \frac{4(K-1)ft_2}{2\sqrt{(K-1)\left[(N-f)(1+mt_1^2)+2ft_2^2\right]}} \tag{C.6}$$

Note that $g(t_2)$ is a monotonically decreasing function, with $g(0) = f > 0$. Note that when $\lim_{t_2 \to \infty} g(t_2) > 0$ when $f > 2(K-1)$. This suggests the optimal MAS will have $t_2 \to \infty$. In this case, $S \approx fMt_2$, $C \approx fM^2t_2^2(1+o(1))$ and $S \geq \sqrt{2(K-1)C}$ will always be satisfied if $f > 2(K-1)$. Therefore, when $f > 2(K-1)$, $e + f = N$ errors and erasures can be recovered for the 1-bit flipped or erased channel, which is optimal. It can also be shown that when $f = 2(K-1) < N$, $e + f = N$ is also achievable by properly assigning multiplicities to symbols without erasure. This is not too surprising, since the 1-bit erased symbols are guaranteed to be error free and therefore, it worth putting more multiplicities on 1-bit erased symbols. For high rate RS codes, we have $2(K-1) > N$. Hence, $g(t_2)$ will have a unique zero in $t_2 \in [0, \infty)$, which maximizes J_1. Set $g(t_2) = 0$, we get:

$$t_2 = \sqrt{\frac{(N-f)(1+mt_1^2)}{4(K-1)-2f}} \tag{C.7}$$

$$J_1 = N - f - \sqrt{\left[(K-1)-f/2\right](N-f)(1+mt_1^2)} \tag{C.8}$$

J can thus be simplified as a function of t_1 only as:

$$J = A \times \frac{B - \sqrt{1+mt_1^2}}{1-t_1} \tag{C.9}$$

where

$$A = \sqrt{\left[(K-1)-f/2\right](N-f)} \tag{C.10}$$

$$B = \sqrt{(N-f)/\left[(K-1)-f/2\right]} \tag{C.11}$$

(C.9) has a similar structure to (B.4). When $f > 2(K-1) + \frac{4(K-1)-2N}{m-1}$, let $t_1 = 1$

and $t_2 = \sqrt{\frac{(N-f)(1+m)}{4(K-1)-2f}}$, the condition (5.8) will always be satisfied.

For $f \leq 2(K-1) + \frac{4(K-1)-2N}{m-1}$, we apply the same technique used in (B.4) here, i.e., to maximize the slope between the point $(1, B)$ and a point on the hyperbola $y^2 - mx^2 = 1$ will give the optimal multiplicity coefficient t_1:

$$t_{1,opt} = \frac{-m + \sqrt{m^2 + m(m - B^2)(B^2 - 1)}}{m(m - B^2)} \tag{C.12}$$

The optimal J as a function of f is:

$$J_{opt}(f) = (N - f)\frac{mt_{1,opt}}{mt_{1,opt} + 1} \tag{C.13}$$

Eventually, the optimal decoding region is:

$$e < J_{opt}(f) \tag{C.14}$$

Any received codeword with e-bit errors and f-bit erasures satisfying (C.14) is decodable by ASD under the optimal MAS with $t_{1,opt}$ and $t_{2,opt}$ as the optimal multiplicity coefficients respectively.

APPENDIX D

DERIVATION OF THE GENERAL DECODING REGION

Proof 21 *When $e + f \leq N$, the cost is*

$$C = (N - f)\frac{M(M+1)}{2} + f\frac{M(M+2)}{4} \tag{D.1}$$

which does not depend on the number of errors e. $T(S)$, as defined in (5.7), is a piecewise linear function with monotonically increasing slope. Since $T(S)$ is monotonic, we first determine the unique interval where $T(S)$ intersects C, i.e., $T(a(k-1)) \leq C \leq T((a+1)(K-1))$. Plugging (5.7) in $T(a(K-1)) \leq C$, we get an upper bound on a:

$$a \leq \frac{-1 + \sqrt{1 + \frac{8C}{K-1}}}{2} \tag{D.2}$$

with C defined in (D.1). The integer solution of a is:

$$\hat{a} = \lfloor \frac{-1 + \sqrt{1 + \frac{8C}{K-1}}}{2} \rfloor \tag{D.3}$$

The threshold of the score can then determined by

$$S^* = T^{-1}(C) = \frac{C}{\hat{a}+1} + \frac{\hat{a}}{2}(K-1) \tag{D.4}$$

where $T^{-1}(C)$ is the inverse function of $T(S)$.

The received codeword is decodable by ASD if $S > S^*$, where $S = (N - e - f/2)M$. Therefore, we have the final decoding region as follows:

$$e < N - \frac{f}{2} - \frac{\hat{a}(\hat{a}+1)(K-1)/2 + C}{M(\hat{a}+1)} \tag{D.5}$$

where $\hat{a}$ and C are defined in (D.3) and (D.1) respectively.

APPENDIX E

COMPUTATION OF THE FRAME ERROR RATE UPPER BOUND OF BGMD DECODING

We give a detailed description of the procedure to compute an upper bound on the FER of BGMD decoder. This upper bound is an extension of the GMD bound [72] for binary linear block codes with a bounded distance decoder (BDD). Without loss of generality, we assume that the all-zero codeword is transmitted. Assuming that BPSK is the modulation scheme and that a zero is mapped to a channel symbol +1, the received value for the ith bit is $r_i = 1 + n_i$, where $n_i \sim \mathcal{N}(0, N_0/2)$.

Let $f(x, N_0) = \frac{1}{\sqrt{\pi N_0}} e^{-\frac{x^2}{N_0}}$ be the probability density function (PDF) of a Gaussian random variable (RV) with mean zero and variance $N_0/2$. Then, the cumulative density function (CDF) of this Gaussian RV is given by:

$$Q(x, N_0) = \int_x^\infty f(t, N_0)dt \tag{E.1}$$

The probability that one bit is in error can therefore be expressed as:

$$P_b = Q(1, N_0). \tag{E.2}$$

Let f_α^e and f_α^c be the PDF's of $|r_i|$ given that $r_i \leq 0$ and $r_i > 0$, respectively. It is shown in [72] that f_α^e and f_α^c are given by

$$f_\alpha^e = \frac{f(x+1)}{Q(1, N_0)} u(x) \tag{E.3}$$

$$f_\alpha^c = \frac{f(x-1)}{1 - Q(1, N_0)} u(x) \tag{E.4}$$

where $u(x)$ is a step function.

Therefore, the corresponding CDF's are:

$$F_\alpha^e = \frac{Q(1,N_0) - Q(x+1,N_0)}{Q(1,N_0)} u(x) \tag{E.5}$$

$$F_\alpha^c = \frac{1 - Q(1,N_0) - Q(x-1,N_0)}{1 - Q(1,N_0)} u(x) \tag{E.6}$$

Assume there are i erroneous bits in the received vector. Order the received bits according to their reliability values in decreasing order. Let $\beta_j(i)$ be the j^{th} ordered reliability value in i erroneous bits. That is $\beta_1(i) \geq \beta_2(i) \geq \cdots \geq \beta_i(i)$. On the other hand, there are $n-i$ correct bits. Define $\gamma_l(n-i)$ as the l^{th} value after ordering. We have $\gamma_1(i) \geq \gamma_2(i) \geq \cdots \geq \gamma_{n-i}(n-i)$. The density of $\beta_j(i)$ and $\gamma_l(n-i)$ can be derived using the ordered statistics as in [81]:

$$f_{\beta_j(i)}(x) = \frac{i!}{(j-1)!(i-j)!} \left[1 - F_\alpha^e(x)\right]^{j-1} f_\alpha^e(x) \left[F_\alpha^e(x)\right]^{i-j} \tag{E.7}$$

$$f_{\gamma_l(n-i)}(x) = \frac{(n-i)!}{(l-1)!(n-i-l)!} \left[1 - F_\alpha^c(x)\right]^{l-1} f_\alpha^c(x) \left[F_\alpha^c(x)\right]^{n-i-l} \tag{E.8}$$

Hence, the probability that the event $\{\beta_j(i) \geq \gamma_l(n-i)\}$ occurs can be evaluated by the following double integral:

$$P\left(\beta_j(i) \geq \gamma_l(n-i)\right) = \int_0^\infty f_{\gamma_l(n-i)}(x) \int_x^\infty f_{\beta_j(i)}(y)\, dy\, dx \tag{E.9}$$

The performance of BGMD decoding can be bounded as follows:

$$P_{BGMD} \leq P_{ML} + P_{list} \approx P_{List} \tag{E.10}$$

P_{List} can be computed using the first order approximation in [72]. The basic idea is that for a specified number of errors in the received vector, the actual FER of BGMD is upper bounded by the FER of an error and erasure decoder with a fixed but optimized number of erasures in the LRB's. P_{List} can be expressed as:

$$\begin{aligned} P_{list} \leq & \sum_{i=e_{max,M}+1}^{f_{max,M}} P_b^i(1-P_b)^{n-i} \min_{(e,f)\in D(M)} P((e+1)\text{errors inthe (N-f) MRB's}) \\ & + \sum_{i=f_{max,M}+1}^{n} P_b^i(1-P_b)^{n-i} \end{aligned} \tag{E.11}$$

$$\begin{aligned} = & \sum_{i=e_{max,M}+1}^{f_{max,M}} P_b^i(1-P_b)^{n-i} \min_{(e,f)\in D(M)} P\left(\beta_{e+1}(i) \geq \gamma_{n-e-f}(n-i)\right) \\ & + \sum_{i=f_{max,M}+1}^{n} P_b^i(1-P_b)^{n-i} \end{aligned} \tag{E.12}$$

where D(M) is the set of all error and erasure pairs (e, f) that is within the decoding region of the proposed ASD for a specified multiplicity M, as characterized in Theorem 8. $f_{max,M}$ and $e_{max,M}$ are the maximum number of erasures and errors such that $(0, f_{max,M})$ and $(e_{max,M}, 0)$ still belong to D(M).

VITA

Jing Jiang received the B.S. degree from Shanghai Jiao Tong University in 2002. He received his Ph.D. degree in electrical engineering from Texas A&M University in August 2007. His general research interests lie in the areas of communication theory, channel coding, signal processing and information theory. Jing Jiang can be contacted at Department of Electrical and Computer Engineering, 214 Zachry Engineering Center, College Station, Texas, 77843-3128, USA.