NOTE TO USERS

This reproduction is the best copy available.

Efficient Soft Decoding Techniques for Reed-Solomon Codes

Farnaz Shayegh

A Thesis

in

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of Doctor of Philosophy at

Concordia University

Montréal, Québec, Canada

May 2010

Your file Votre référence
ISBN: 978-0-494-71151-4
Our file Notre référence
ISBN: 978-0-494-71151-4

ABSTRACT

Efficient Soft Decoding Techniques for Reed-Solomon Codes

Farnaz Shayegh, PhD.

Concordia University, 2010

The main focus of this thesis is on finding efficient decoding methods for Reed-Solomon (RS) codes, i.e., algorithms with acceptable performance and affordable complexity. Three classes of decoders are considered including sphere decoding, belief propagation decoding and interpolation-based decoding.

Originally proposed for finding the exact solution of least-squares problems, sphere decoding (SD) is used along with the most reliable basis (MRB) to design an efficient soft decoding algorithm for RS codes. For an (N, K) RS code, given the received vector and the lattice of all possible transmitted vectors, we propose to look for only those lattice points that fall within a sphere centered at the received vector and also are valid codewords. To achieve this goal, we use the fact that RS codes are maximum distance separable (MDS). Therefore, we use sphere decoding in order to find tentative solutions consisting of the K most reliable code symbols that fall inside the sphere. The acceptable values for each of these symbols are selected from an ordered set of most probable transmitted symbols. Based on the MDS property, K code symbols of each tentative solution can be used to find the rest of codeword symbols. If the resulting codeword is within the search radius, it is saved as a candidate transmitted codeword. Since we first find the most reliable code symbols and for each of them we use an ordered set of most probable transmitted symbols, candidate codewords are found quickly resulting in reduced complexity. Considerable coding gains are achieved over the traditional hard decision decoders with moderate increase in complexity.

Due to their simplicity and good performance when used for decoding low density parity check (LDPC) codes, iterative decoders based on belief propagation (BP) have also

been considered for RS codes. However, the parity check matrix of RS codes is very dense resulting in lots of short cycles in the factor graph and consequently preventing the reliability updates (using BP) from converging to a codeword. In this thesis, we propose two BP based decoding methods. In both of them, a low density extended parity check matrix is used because of its lower number of short cycles. In the first method, the cyclic structure of RS codes is taken into account and BP algorithm is applied on different cyclically shifted versions of received reliabilities, capable of detecting different error patterns. This way, some deterministic errors can be avoided. The second method is based on information correction in BP decoding where all possible values are tested for selected bits with low reliabilities. This way, the chance of BP iterations to converge to a codeword is improved significantly. Compared to the existing iterative methods for RS codes, our proposed methods provide a very good trade-off between the performance and the complexity.

We also consider interpolation based decoding of RS codes. We specifically focus on Guruswami-Sudan (GS) interpolation decoding algorithm. Using the algebraic structure of RS codes and bivariate interpolation, the GS method has shown improved error correction capability compared to the traditional hard decision decoders. Based on the GS method, a multivariate interpolation decoding method is proposed for decoding interleaved RS (IRS) codes. Using this method all the RS codewords of the interleaved scheme are decoded simultaneously. In the presence of burst errors, the proposed method has improved correction capability compared to the GS method. This method is applied for decoding IRS codes when used as outer codes in concatenated codes.

To Pejman

ACKNOWLEDGEMENTS

First and foremost, I would like to express my gratitude to my supervisor, Professor M. Reza Soleymani, for giving me the opportunity to work in his group, for his support and encouragement throughout my research not only as a professor but also as a friend. Without his guidance and teaching, I would not be able to complete this research.

I would like to thank all members of my Ph.D. defense committee: Prof. Fady Alajaji, Prof. Ali Ghrayeb, Prof. Yousef R. Shayan and Prof. Rajamohan Ganesan. I appreciate their valuable comments and all the time they spent to read this thesis. My especial thanks go to the external examiner, Prof. Fady Alajaji, of Queen's University at Kingston, for his suggestions for the future work and his technical insight. I also want to thank all current and previous members of the Wireless and Satellite Communications Laboratory for their help and support.

I wish to thank NSERC CRD as well as InterDigital Canada Ltd., PROMPT Quebec and FQRNT for their financial supports.

More than anyone else, I would like to thank my mother, Esmat and my late father, Manouchehr. I would not be where I am today without their love, encouragement and unlimited sacrifices. I only hope I have made their trials and support worthwhile. I hope this thesis would fulfill part of my father's wishes for me.

My thanks also go to my beloved sister, Elham, and my brothers Pedram and Shahram for their support.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

$RS(N, K)$	RS code with length N and dimension K
$GF(q)$	Galois field of size q
d_{min}	Minimum Hamming distance
R	Code rate
D	Support set of a Galois field
t	Error correction capability of a decoder
m	Multiplicity
G	Generator matrix of a linear block code
α's	Fading coefficients
G_{sys}	Systematic generator matrix
r	Sphere decoding search radius
e_c	Complexity exponent
ρ^{ch}	Channel reliabilities
$p(x)$	Primitive polynomial of $GF(2^p)$
c_p	Companion matrix corresponding to $p(x)$
H	Parity check matrix
π	Reliability matrix
M	Multiplicity matrix
ϵ	Erasure probability
J	Potential function
$IRS(N, K_1, ..., K_M)$	IRS code composed of M RS(N, K_i) code, $i = 1, ..., M$
$c(N, M, m)$	Interpolation cost
I	Ideal of polynomials
$Syl(A, B, x)$	Sylvester matrix of polynomials A and B with respect to x
$Res(A, B, x)$	Resultant of A and B with respect to x

LIST OF ABBREVIATIONS

RS	Reed-Solomon
T-DMB	Terrestrial Digital Multimedia Broadcasting
DVB	Digital Video Broadcasting
CD	Compact Discs
DVD	Digital Versatile Disc
3G	Third Generation
LDPC	Low Density Parity Check
SD	Sphere Decoding
BP	Belief Propagation
GF	Galois Field
MDS	Maximum Distance Separable
BM	Berlekamp-Massey
ML	Maximum Likelihood
GMD	Generalized Minimum Distance
OSD	Ordered Statistics Decoding
GS	Guruswami-Sudan
KV	Koetter-Vardy
ADP	Adaptive Parity Check
IRS	Interleaved Reed-Solomon
GC	Generalized Concatenated
AWGN	Additive White Gaussian Noise
BPSK	Binary Phase Shift Keying
QAM	Quadrature Amplitude Modulation
SNR	Signal-to-Noise Ratio
MIMO	Multiple Input Multiple Output
i.i.d.	Independent and Identically Distributed

ZF	Zero Forcing
LLR	Log-Likelihood Ratio
HD	Hard Decision
MRB	Most Reliable Basis
BEC	Binary Erasure Channel
gcd	greatest common divisor
TSB	Tangential Sphere Bound
SPB	Sphere Packing Bound

Chapter 1

Introduction

Reed-Solomon (RS) codes were invented in 1960 by Irving S. Reed and Gustave Solomon [1] and separately by Arimoto [2]. They are non-binary linear block codes with many interesting properties such as random error correcting capability, burst error correcting capability and erasure recovery capability. RS codes have been widely used in commercial applications including optical and magnetic storage systems, satellite and deep-space communication and mobile data communication. They have been employed in various digital communication standards including terrestrial digital multimedia broadcasting (T-DMB) [3] and digital video broadcasting (DVB) [4]. Data storage devices such as compact discs (CDs) and digital versatile discs (DVDs) have concatenated RS codes [5] and RS product codes [6] as their error correcting codes. Today, storage systems are implemented in many devices including digital music players, cell phones, digital cameras, high definition TVs and so many more. RS codes concatenated with convolutional codes are the standard channel codes for satellite transmission. They have also been widely used in wireless communication systems because of their ability to correct burst errors. They are outer codes in the third generation (3G) wireless standard and CDMA2000 [7]. As another application, paper bar codes such as Postbar use RS error correction to correct for encoding errors on paper [8].

Recently, modern coding techniques such as LDPC codes [9] and Turbo codes [10] have become popular because of their capacity approaching capabilities. The problem with

these codes is the uncertainty of their performances at high signal to noise ratios (SNR's) especially in applications where burst noise is present. On the other hand, classic RS codes have very good burst error correction capability and their performance at high SNR's is determined. Also, there have been considerable recent developments in decoding RS codes [11] [12] [13]. Therefore, RS codes remain very relevant today. If modern codes were to be used in communication standards, RS codes will still be needed as outer codes to cure their error floor problems.

Despite their wide areas of applications, soft decision decoding of RS codes still represents an open issue. In this thesis, we present efficient decoding techniques for RS codes including a soft decision decoder based on sphere decoding [14], two iterative techniques based on belief propagation (BP) decoding [9] and a collaborative interpolation decoder based on Guruswami-Sudan decoding method [11].

In this chapter, we present RS codes including their definition, characteristics and decoding. A comprehensive literature review is given on soft decision decoding techniques used for RS codes. Finally, the organization of the thesis concludes the chapter.

1.1 Reed-Solomon Codes

Denote the ring of polynomials over the Galois field $GF(q)$ in a variable X by $F_q[X]$. The support set of $GF(q)$ is defined as the set of all its non-zero elements:

$$D = \{x_1, x_2, ..., x_N\} \subset GF(q) \tag{1.1}$$

where $N = q - 1$. RS codes are obtained by evaluating certain subspaces of $F_q[X]$ in the support set D. Specifically, an RS code of length N and dimension K is defined as follows:

$$C_q(N, K) = \tag{1.2}$$
$$\{(f(x_1), f(x_2), ..., f(x_N)) | x_1, x_2, ..., x_N \in D, f(X) \in F_q[X], degf(X) < K\}$$

where 'deg' refers to the degree of a polynomial. The minimum distance of this RS code is $d_{min} = N - K + 1$. It should be noted that RS codes satisfy the Singleton bound [15] with

equality and are therefore maximum distance separable (MDS). In this thesis we especially consider RS codes over $GF(2^p)$ with $p \geq 3$. This means that every symbol of an RS codeword can be represented with p bits.

The first efficient decoding algorithm for RS codes has been proposed by Berlekamp [16] and Massey [17]. Today, it is known as the Berlekamp-Massey (BM) algorithm [6]. The error correction capability of any Reed-Solomon code is determined by its redundancy $N - K$. Using the BM decoder, any combination of errors and erasures in an RS codeword can be corrected as long as the following inequality is satisfied:

$$2E + S \leq N - K. \tag{1.3}$$

Here, E is the number of symbol errors and S is the number of symbol erasures in the RS codeword. RS codes are especially well-suited to applications where errors occur in bursts. This is because multiple bit errors in a symbol are only considered as a single error.

1.2 Literature Review on Reed-Solomon Decoding

Decoding RS codes is the problem of reconstructing univariate polynomials from their noisy evaluations. This is a task best performed using maximum likelihood (ML) decoding. Guruswami and Vardy have shown that the ML decoding of RS codes is NP-complete [18]. They have considered the following problem:

Problem: ML decoding of RS codes.
Instance: An integer $p > 0$, a set $D = \{x_l, x_2, ..., x_N\}$ of N distinct elements of Galois field $GF(2^p)$, a positive integer K, a target vector $y \in GF(2^p)$ and an integer $w > 0$.
Question: Is there a codeword $c \in C_{2^p}(N, k)$ such that the Hamming distance $d(c, y) \leq w$?

They have proved that the above problem is NP-complete meaning although we can quickly verify any given solution to such a problem, there is no known efficient method to find a solution in the first place. In fact, the main characteristic of NP-complete problems is that

no fast solution to them is known. That is, as the size of the problem increases, the time required to solve the problem using any currently known algorithm increases very quickly.

As mentioned before, BM algorithm [6] has been the first efficient hard decision decoding method for RS codes. It involves calculating the syndromes, finding the error locator polynomial and finally determining the error values. The complexity of BM algorithm is $O(N \times d_{min})$. It is capable of successfully decoding any hard decision vector with a Hamming distance no greater than half of the minimum distance of the code (d_{min}) from the transmitted codeword. The error correction capability of BM algorithm for an (N, K) RS code over $GF(2^p)$ is defined as

$$t = \left\lfloor \frac{N-K}{2} \right\rfloor. \tag{1.4}$$

The probability of error and failure of the BM algorithm for RS codes with BPSK modulation is

$$P_e = 1 - \sum_{j=0}^{t} \binom{N}{j} S^{N-j}(1-S)^j \tag{1.5}$$

where

$$S = \begin{cases} (1 - Q(\sqrt{2\gamma R}))^p, & \text{AWGN Channel} \\ (1 - \sqrt{\frac{\gamma R}{1+\gamma R}})^p, & \text{Flat Fading Channel (fading known at the receiver).} \end{cases} \tag{1.6}$$

Here, γ is the signal to noise ratio, R is the code rate and S is the probability of receiving one symbol of the RS codeword correctly.

As mentioned at the beginning, RS codes have the capability of erasure recovery. BM algorithm is able to perform error-erasure decoding. In the presence of v erasures, the probability of error and failure of BM method is

$$P_{e-erasure} = 1 - \sum_{j=0}^{t_1} \binom{N-v}{j} S^{N-v-j}(1-S)^j \tag{1.7}$$

where $t_1 = \lfloor \frac{N-K-v}{2} \rfloor$.

Lots of works on the improvement of BM hard decision decoding have been done such as Euclid's algorithm for the determination of the error locator polynomial [19] and

the work of Berlekamp-Welch that allows for decoding without the need for syndrome computations [20].

Although hard decision decoding methods have low complexity, they do not use the channel reliability information which causes considerable performance loss. The best performance possible is that of ML soft decision decoding. As discussed earlier, the complexity of ML decoding is prohibitive especially for long RS codes that are used in practical systems. So, one of the current issues about RS codes is to find decoding methods with performance as close as to that of the ML decoding method and at the same time keep the complexity affordable. In the following, we will give a comprehensive review on soft decision decoding methods for RS codes.

First we mention two old and still popular soft decision decoding methods for RS codes. In 1966, Forney invented generalized minimum distance (GMD) decoding [21]. In GMD decoding, based on channel reliabilities, some symbols may be erased and then fed into an algebraic error-erasure decoder. Different erasing patterns are considered using an erasure-choosing algorithm which relies on soft information from the channel. The algorithm is terminated when a codeword satisfying a certain distance criterion is found. In 1972, Chase [22] proposed a soft decoding algorithm for general block codes. In Chase decoding, a test set of hard decision vectors is developed using reliability information. To construct the test set η least reliable coordinate locations are determined and vectors with all possible symbols in these locations are considered. Each of these vectors is applied to the hard decision decoding algorithm. Among all the successfully decoded codewords, the one with the highest a posteriori probability is chosen as the decoder output. Chase decoding provides moderate coding gain over hard decision decoding and its complexity is exponentially increasing in η. Both Chase and GMD decoding methods provide moderate coding gain over hard decision decoders with reasonable complexity. Other related soft decoding algorithms include the Chase II algorithm [22], the simple extension of Chase algorithm [23] and the combined Chase II-GMD algorithm [24].

Vardy and Be'ery [25] have proposed ML decoding of RS codes using their binary image expansion by decomposing RS codes into BCH subfield subcodes. However, their

algorithm is practically applicable only to small RS codes.

The ordered statistics decoding (OSD) algorithm by Fossorier and Lin [26] has first been proposed for binary linear block codes. It sorts the received bits with respect to their reliabilities. Then, the columns in the generator matrix corresponding to the most reliable bits are reduced to an identity submatrix. Using order$-w$ reprocessing, up to w bits are systematically flipped on the most reliable (information) basis (MRB). The modified generator matrix is then used to generate (permuted) codewords using these most reliable basis. Using the binary image of RS codes, OSD method and its variations [27] [28] can be used for RS codes. OSD based algorithms are efficient for practical RS codes. However, to improve the performance, w should be increased resulting in high complexity.

In 1999, Guruswami and Sudan [11] showed that hard decision decoding of RS codes beyond their traditional capability (half the minimum distance) is possible. Their method, now called the GS algorithm, is a polynomial time algebraic list decoding method. It interpolates a bivariate polynomial $Q(x, y)$ from the channel output that passes through all received values with multiplicity at least m. The y-linear factors of this polynomial contain all the codewords within a decoding radius $t > \frac{d_{min}}{2}$ from the hard decision vector. It is shown that GS algorithm can correct any fraction of $\tau \leq 1 - \sqrt{R}$ errors for an RS code of rate R.

In 2003, Koetter and Vardy [29] developed an algebraic soft decision decoding (ASD) method for RS codes, now called the KV algorithm. It is based on GS algorithm but it uses the reliability information at the channel output to construct $Q(x, y)$. While GS algorithm forces $Q(x, y)$ to pass through all received values with equal multiplicity, KV method allows $Q(x, y)$ to pass through received values with a multiplicity dependent on each coordinate's reliability. KV algorithm has been shown to outperform GS method at a comparable decoding complexity. There are alternative ASD algorithms with better performance compared to KV method. They include the Gaussian approximation algorithm by Parvaresh and Vardy [30] and the algorithm by El-Khamy and McEliece [31] based on Chernoff bound.

Soft decision decoding of linear block codes based on sphere decoding (SD) [32]

[14] has been investigated in [33] and [34]. As we know, ML decoding of error correcting codes is actually a search for the closest point to the received vector in the lattice formed by the symbol space of the code. SD is a method that can perform ML decoding without an exhaustive search over the entire lattice. Its search for the closest point to the received vector is limited to only those lattice points that fall within a sphere centered at the received vector. This results in a considerable reduction in complexity of ML decoding. In [33], the concept of sphere decoding for joint ML detection and decoding of linear block codes on Gaussian vector channels has been investigated.

Iterative belief propagation (BP) decoding is the basis for another class of RS soft decoders. BP decoding has first been proposed by Gallager [9] for decoding low density parity check (LDPC) codes. Although iterative BP decoding in its standard form is not suitable for high density parity check codes such as RS codes, lots of attempts have been made to adopt BP decoding for RS codes [35] [36] [37]. The main problem is the large number of short cycles in the factor graph [38] of RS codes which causes correlation between the messages and error propagation. The first successful iterative decoding method for RS codes was proposed by Jiang and Narayanan in 2004 [12]. It is referred to as the adaptive parity check (ADP) algorithm and uses the binary image of the RS parity check matrix to implement BP decoding. In ADP algorithm, each BP iteration is run on an adapted parity check matrix with columns corresponding to the least reliable independent bits reduced to an identity submatrix. At the end of each BP iteration, hard decisions are made from the updated reliability information and passed to a hard decision decoder. ADP method compares favorably with other soft decision decoding algorithms for RS codes. In 2006, Elkhamy et. al. [13] used ADP algorithm to improve the reliability of the symbols at the beginning of the KV algorithm. Their algorithm has impressive coding gains over previously known soft decision decoding algorithms for RS codes. The problem with the ADP method is the high complexity of adopting the parity check matrix at each BP iteration. Recently, more decoding methods based on BP algorithm have been proposed for general linear block codes [39–43]. In [44] multiple-bases belief propagation for linear block codes with dense parity check matrices has been proposed. It makes use of the fact that a code

has many structurally diverse parity check matrices, capable of detecting different error patterns.

1.3 Thesis Outline

So far, we have reviewed RS codes, their properties and exiting soft decision decoding techniques for them. The rest of the thesis is organized as follows:

In Chapter 2, "Background", the required background information that is the basis for the work done in this thesis is presented. We investigate three classes of low complexity decoding methods including sphere decoding (SD), message passing decoding and interpolation-based decoding. The basis of sphere decoding is explained and further is used in Chapter 3 to implement a soft decision decoder for RS codes. Then, a subclass of message passing decoding, belief propagation, is presented. Later in Chapter 4, we introduce BP based algorithms for efficient iterative decoding of RS codes. Finally, the GS interpolation decoding algorithm is explained in detail. In Chapter 5, the basis of GS algorithm is used for collaborative decoding of interleaved Reed-Solomon (IRS) codes.

In Chapter 3, "Efficient Soft Decoding of RS Codes based on Sphere Decoding", a novel soft decision decoding method for RS codes using sphere decoding is proposed. With sphere decoding, instead of considering all of the possible transmitted codewords to determine the most probable one, one can only consider the codewords whose distance from the received signal is smaller than a specific search radius. This results in a considerable reduction in the complexity. Because we need to find points inside the sphere that are also valid codewords of an (N, K) RS code, the sphere decoder algorithm first selects a tentative solution consisting of the K most reliable and independent code symbols whose distance from the corresponding symbols in the received vector is less than the search radius. The acceptable values for each of these K code symbols are determined based on the ordered set of most probable transmitted symbols. Each time K code symbols are selected using the sphere decoder, they are re-encoded and if the resulting codeword is within the search radius, we add it to the list of the candidate transmitted codewords. The ordering that was

discussed earlier will help finding the candidate codewords quickly. Our method results in considerable improvement of the performance of RS codes compared to the hard decision decoding with a moderate increase in complexity. The performance is also superior or comparable to some popular soft decision decoding methods.

In Chapter 4, "Efficient Iterative Techniques for Soft Decision Decoding of RS Codes", two new iterative soft decision decoding methods for RS codes are proposed. These methods are based on bit level belief propagation decoding. In order to make BP decoding effective for RS codes, we use an extended binary parity check matrix with a lower density and reduced number of 4-cycles compared to the original binary parity check matrix of the code. In the first proposed method, we take advantage of the cyclic structure of RS codes. Based on this property, we can apply the belief propagation algorithm on any cyclically shifted version of the received symbols with the same binary parity check matrix. For each shifted version of received symbols, the geometry of the factor graph will change and deterministic errors can be avoided. This method results in considerable performance improvement of RS codes compared to hard decision decoding. The performance is also superior to some popular soft decision decoding methods. The second method is based on information correction in BP decoding. It means that we determine least reliable bits and by changing their channel information, the convergence of the decoder is improved. Compared to the first method, this method needs less BP iterations (less complexity) but its performance is not as good.

In Chapter 5, "Collaborative Algebraic Decoding of Interleaved RS Codes", we derive and analyze an algorithm for collaborative decoding of heterogeneous Interleaved Reed-Solomon (IRS) codes. They are generated by interleaving several codewords from different RS codes with the same length over the same Galois field. The basis of the decoding algorithm is similar to the GS decoding method. However, here multivariate interpolation is used in order to decode all the codewords of the interleaved scheme simultaneously. In the presence of burst errors, we show that the error correction capability of this algorithm is larger than that of independent decoding of each codeword using the standard GS method. In the latter case, the error correction capability is equal to the decoding radius

of the GS algorithm for the RS code with the largest dimension. Also, generalized concatenated (GC) codes using IRS codes as their outer codes and binary linear block codes as their inner codes are considered. Assuming ML decoding of the inner code, we derive upper and lower bounds for the word error probability of GC codes over AWGN channel with BPSK modulation for both cases of independent and collaborative decoding of the outer IRS codes. We show that collaborative decoding provides considerable coding gain compared to independent decoding.

Finally, in Chapter 6, "Conclusions and Future Works", we summarize the contributions of this thesis and give suggestions for future works.

Chapter 2

Background

In this chapter, we investigate three classes of complexity-reducing methods. They include sphere decoding (SD), message passing decoding and interpolation-based decoding. In the next chapters, we use these methods to develop efficient decoding techniques for RS codes.

Sphere decoding is a low complexity method used to find the exact solution of least-squares problems. In communication systems, it has been used for multiple input multiple output (MIMO) detection [45], joint detection and decoding of linear block codes over Gaussian vector channels [33] and so on. In the first part of this chapter, we explain the basis of sphere decoding (SD). The application of SD for MIMO detection is investigated and a new method for reducing the complexity of SD is proposed.

Message passing decoding [38] is a general technique to compute marginal functions in a factor graph. Belief propagation is a subclass of message passing decoding and has been used in coding theory for decoding LDPC codes, Turbo codes and so on. In the second part of this chapter, belief propagation iterative decoding of binary linear block codes is explained. The binary image of RS codes is introduced and a popular iterative RS decoding method is presented.

The first interpolation-based decoding method used for RS codes is the Berlekamp-Massey (BM) algorithm [6]. It decodes the hard decision received signal using univariate polynomial interpolation. The Guruswami-Sudan (GS) method [11] can provide higher

correction capability compared to the BM algorithm using bivariate polynomial interpolation. The Koetter-Vardy (KV) method [29] can incorporate the soft information from the channel into the GS method to improve the performance of decoding. In the third part of this chapter, we briefly explain both the GS method and the KV method.

2.1 Sphere Decoding

The concept of sphere decoding (SD) in mathematics has been first introduced in [14]. It is used to calculate vectors of short length in a lattice. While, the standard methods use a reduction procedure followed by considering all vectors in a suitable box, the SD method only looks for those vectors lying in a suitable ellipsoid having a much smaller volume than the box. Therefore, sphere decoding is more efficient than the standard methods.

The SD method have been used to find the exact solution of least-squares problems in many applications. In communication systems, we usually deal with a system of linear equations where the coefficients of the matrix and the given vector are real numbers but those of the unknown vector are integers. In cases like this, the least-squares problem is reduced to:

$$\underset{s \in Z^m}{\operatorname{argmin}} \|x - Hs\|^2 \tag{2.1}$$

where x is the given $n \times 1$ real vector, H is the $n \times m$ real matrix, s is the unknown $m \times 1$ integer vector and Z^m denotes the m-dimensional integer lattice. Since s spans an m-dimensional lattice Z^m, Hs spans a skewed lattice. Therefore, the integer least-squares problem is to find the closest lattice point of this skewed lattice to x.

The basic idea of sphere decoding is to search over only lattice points s that lie in a certain sphere of radius d around the given vector x (Figure 2.1). This reduces the search space and, hence, the required computations [32]. Using sphere decoding, the least-squares problem is reduced to

$$\|x - Hs\|^2 \leq d^2. \tag{2.2}$$

In order to do sphere decoding, asuming $n \geq m$, at first QR factorization of H is

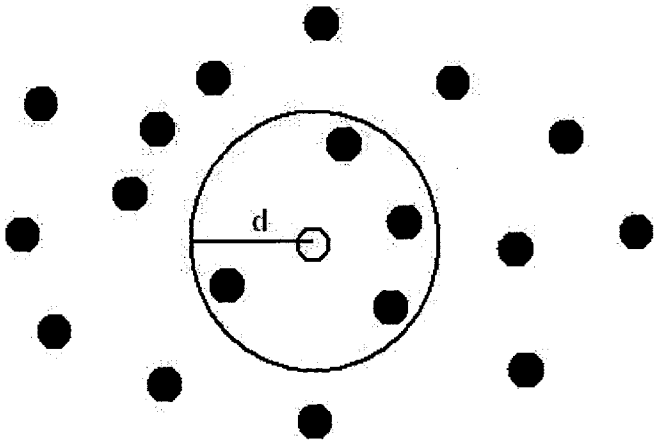

Figure 2.1: The basic idea of sphere decoding

performed:

$$H = Q \begin{bmatrix} R \\ 0_{(n-m)\times m} \end{bmatrix} \tag{2.3}$$

where, Q is an $n \times n$ orthogonal matrix and R is an $m \times m$ upper triangular matrix. After replacing the QR factorization of H in Equation (2.2) and some mathematical simplifications, the problem in Equation (2.2) becomes equivalent to

$$\|R(s - \widehat{s})\|^2 \leq d^2 - \|x\|^2 + \|H\widehat{s}\|^2 = d_m^2. \tag{2.4}$$

Here, $\widehat{s} = R^{-1}Q_1^* x$, Q_1 is the first m orthonormal columns of Q and '$*$' denotes Hermitian matrix transposition. The above equation can be written as

$$\begin{aligned} & r_{m,m}^2(s_m - \hat{s}_m)^2 + r_{m-1,m-1}^2(s_{m-1} - \hat{s}_{m-1} + \frac{r_{m-1,m}}{r_{m-1,m-1}}(s_m - \hat{s}_m))^2 \\ & + \ldots \leq d_m^2 \end{aligned} \tag{2.5}$$

where $r_{i,j}$ is the $(i,j)^{th}$ entry of R. The first term of the above inequality depends only on s_m, the second term on s_m and s_{m-1}, and so on. A necessary condition for Hs to lie inside the sphere is that $r_{m,m}^2(s_m - \hat{s}_m)^2 \leq d_m^2$. From this condition, a bound for s_m is found:

$$\left\lceil \hat{s}_m - \frac{d_m}{r_{m,m}} \right\rceil \leq s_m \leq \left\lfloor \hat{s}_m + \frac{d_m}{r_{m,m}} \right\rfloor. \tag{2.6}$$

Here, $\lceil . \rceil$ and $\lfloor . \rfloor$ denote respectively rounding to the nearest larger and the nearest smaller element in the set of numbers that spans the lattice. Of course, this bound is not sufficient. For every element s_m of this bound, we can define $d^2_{m-1} = d^2_m - r^2_{m,m}(s_m - \hat{s}_m)^2$ and use the first two terms in Equation (2.5) in order to find a bound for s_{m-1}:

$$\left\lceil \hat{s}_{m-1} - \frac{r_{m-1,m}}{r_{m-1,m-1}}(s_m - \hat{s}_m)) - \frac{d_{m-1}}{r_{m-1,m-1}} \right\rceil \leq s_{m-1} \leq \left\lfloor \hat{s}_{m-1} - \frac{r_{m-1,m}}{r_{m-1,m-1}}(s_m - \hat{s}_m)) + \frac{d_{m-1}}{r_{m-1,m-1}} \right\rfloor. \tag{2.7}$$

This process can be continued in a similar fashion for s_{m-2}, s_{m-3} and so on until obtaining all lattice points inside the sphere satisfying Equation (2.2). Among these points, the one that minimizes $\|x - Hs\|^2$ is the solution to the least-squares problem. The complete algorithm of sphere decoding is given in [32].

In order to reduce the complexity of sphere decoding, each time a point is found inside the sphere we can replace our search radius with the distance of that point from the received vector. Adjusting the search radius results in considerable complexity reduction because each time a point is found inside the sphere we reduce our search radius and it is clear that with smaller d we have less complexity.

In the following, we consider a popular application of the SD method for MIMO detection. We also propose a new modified SD method with less complexity compared to original sphere decoding. Later in Chapter 3, we use this new technique for efficient soft decision decoding of RS codes.

2.1.1 MIMO Detection using Sphere Decoding

Standard maximum likelihood (ML) detection of MIMO signals is very complex. Sphere decoding has been considered as a reduced complexity ML detector for MIMO systems [45]. For an $M \times N$ MIMO system, the received signal can be written as

$$x = Hs + v \tag{2.8}$$

where s is the $M \times 1$ transmitted vector with entries from a complex-valued constellation, H is the $N \times M$ channel matrix with independent and identically distributed (i.i.d) entries from

a circularly symmetric complex Gaussian distribution with zero mean and unit variance, v is the $N \times 1$ noise vector with entries from a zero mean complex Gaussian distribution with variance of σ^2 and x is the $N \times 1$ received vector.

The goal of a MIMO detector is to find the transmitted vector s knowing the received vector x and the channel Matrix H. To state the ML detection as an integer least-squares problem, we first find the real-valued equivalent of the equation $x = Hs + v$. To this end, let $m = 2M, n = 2N$ and

$$s = \left[Re(s)^T \; Im(s)^T\right]^T \tag{2.9}$$

$$x = \left[Re(x)^T \; Im(x)^T\right]^T \tag{2.10}$$

$$v = \left[Re(v)^T \; Im(v)^T\right]^T \tag{2.11}$$

$$H = \begin{bmatrix} Re(H) & Im(H) \\ -Im(H) & Re(H) \end{bmatrix}. \tag{2.12}$$

Then the real valued equivalent of our equation can be written as $x = Hs + v$ with the new x, s, v and H. It can be seen that the vector dimensions have been doubled in the new equation.

ML detector checks all of the possible transmitted vectors to find the one that minimizes $\|x - Hs\|^2$. So, its complexity is exponential in the number of transmit antennas and constellation points. However, the SD method only yields the set of points s such that $\|x - Hs\|^2 \leq r^2$. The search radius r has to be selected very carefully. If r is too small the complexity will be very low but we may obtain no points inside the sphere. If r is too large there will be too many points inside the sphere and the complexity remains exponential in size. The search radius r can be chosen based on the statistical properties of the noise. From Equation (2.8), $\|v\|^2 = \|x - Hs\|^2$ is a χ^2 random variable with n degrees of freedom. So the radius is selected to be a scaled variance of the noise $r^2 = \alpha n \sigma^2$ (for a properly chesen α) in such a way that with a high probability at least one point is found inside the sphere [32]. The probability of finding at least one point inside the sphere is [32]

$$P_{fp} = \int_0^{\frac{\alpha n}{2}} \frac{\lambda^{n/2-1}}{\Gamma(n/2)} e^{-\lambda} d\lambda \tag{2.13}$$

where 'Γ' denotes the Gamma function.

2.1.2 Ordered Sphere Decoding

Here, a reduced complexity sphere decoding method [46] is explained. First, a linear receiver such as zero forcing (ZF) [47] is used to provide an estimate for the response of the MIMO detector. This estimate is used as a reference signal by the SD method. Then each time we find a bound for an element of the transmitted vector s, we sort the elements of this bound based on their distance from the corresponding element of the reference signal such that we always start from the most probable element and we continue the same way [46]. So, we always start from the output of the linear receiver that is a good initial point and in many cases the ML point. This method combined with adjusting the search radius results in considerable complexity reduction because the candidate responses are found very quickly due to ordering.

2.1.3 Ordered Sphere Decoding with Channel Ordering

In this section, we propose a new low complexity sphere decoding method. We suppose that the number of transmit antennas are equal to the number of received antennas ($M \times M$ MIMO) and the coded data in the transmitter is modulated by a 2^p-QAM modulation.

Using the real-valued received signal x and the real-valued channel matrix H, a linear receiver such as zero forcing (ZF) provides a soft-output estimate for the response of the MIMO detector,

$$s_{zf} = H^t x \tag{2.14}$$

where H^t is the pseudo inverse [48] of the channel matrix. We are looking for the real-valued transmitted vector $\hat{s}$. For each element of $\hat{s}$, we find an ordered set of all possible transmitted elements $\left\{\hat{s}_i^{(1)}, \hat{s}_i^{(2)}, ..., \hat{s}_i^{(p)}\right\}$, $i = 1, ..., m$, based on their distance from the corresponding element of s_{zf} such that

$$\left\|\hat{s}_i^{(1)} - s_{zf}(i)\right\| \leq \left\|\hat{s}_i^{(2)} - s_{zf}(i)\right\| \leq ... \leq \left\|\hat{s}_i^{(p)} - s_{zf}(i)\right\|. \tag{2.15}$$

From these ordered sets, we can define a reliability measure for each element of the transmitted vector:

$$LLR(\hat{s}_i) = \left\| \hat{s}_i^{(2)} - s_{zf}(i) \right\| - \left\| \hat{s}_i^{(1)} - s_{zf}(i) \right\|. \tag{2.16}$$

In order to reduce the complexity of detection, we start detecting the transmitted vector $\hat{s}$ from its most reliable element. To this end, we arrange the reliabilities in an increasing order. This ordering will define a permutation λ. The elements of the reference signal s_{zf} and also the columns of the channel matrix H are permuted according to λ:

$$s_{zf}^{ord} = s_{zf}(\lambda), \tag{2.17}$$

$$H^{ord} = H(:, \lambda). \tag{2.18}$$

Therefore, the detected vector will also be permuted according to the same permutation such that $\hat{s}_1 = \lambda(\hat{s})$. In the end, the actual transmitted vector can be obtained by permuting the components of the detected vector using the inverse permutation λ^{-1}.

Now, we can detect the transmitted vector using the ordered sphere decoder with the new channel matrix H^{ord} and the new reference signal s_{zf}^{ord}. The proposed sphere decoding algorithm is summarized in Figure 2.2.

In order to reduce the complexity, each time we find a vector $\hat{s}_1$ inside the sphere, we replace the search radius r with $\left\| x - H^{ord}\hat{s}_1 \right\|$. After finishing the algorithm, we have a list of permuted candidate transmitted vectors. We will choose the one that minimizes $\left\| x - H^{ord}\hat{s}_1 \right\|$ as the response of sphere decoding. The actual transmitted vector is

$$\hat{s} = \lambda^{-1}(\hat{s}_1). \tag{2.19}$$

In this method, we start detecting the transmitted vector from its most reliable element and for each element, we start from the most probable transmitted symbol based on the information from the reference signal. This kind of ordering will help finding the candidate transmitted vectors quickly. By finding the candidate transmitted vectors early and replacing the search radius by their distances from the received vector, there will be less points inside the sphere that satisfy $\left\| x - H^{ord}\hat{s} \right\|$. This results in reducing the complexity

Data: Received signal x, permuted reference signal s_{zf}^{ord}, permuted channel matrix H^{ord} and the search radius r.

Result: A list of permuted signals $\hat{s}_1$'s inside the sphere of radius r around the received signal x.

begin

Initializations: $m = 2M$, $k = m$, $d(k) = r$, $A = [00...0]_{1\times m}$.

1 [Q R]: QR factorization of H^{ord}.

2 $s_2(k) = s_{zf}^{ord}(k)$

3 $z = d(k)/R(k,k)$

4 $ub(k) = \lfloor min(abs(z) + s_2(k), sqrt(2^p) - 1) \rfloor$

5 $lb(k) = \lceil max(-abs(z) + s_2(k), -(sqrt(2^p) - 1)) \rceil$

6 Bound for $\hat{s}_1(k)$: $cd = lb(k) : 2 : ub(k)$

7 $Lx(k) = length(cd)$

8 Sorting the elements of cd based on their distances from $s_{zf}^{ord}(k)$: Permutation λ_k

9 $cd^{ord}(k,:) = cd(\lambda_k)$

10 **if** $k = m + 1$ **then**
  stop.

11 **if** $A(k) + 1 \leq Lx(k)$ **then**
  $\hat{s}_1(k) = cd^{ord}(k, A(k) + 1)$, $A(k) = A(k) + 1$.
else
  Go to 14.

12 **if** $k = 1$ **then**
  save $\hat{s_1}$

13 **else**
  $k = k - 1$,
  $s_2(k) = s_{zf}^{ord}(k) - \sum_{l=k+1}^{m}(R(k,l)/R(k,k)) \times (\hat{s}_1(l) - s_{zf}^{ord}(l))$
  $d(k)^2 = d(k+1)^2 - R(k+1,k+1)^2 \times (\hat{s}_1(k+1) - s_2(k+1))^2$
  Go to 3.

14 $k = k + 1$, $A(1 : k - 1) = zeros(1, k - 1)$, Go to 10.

end

Figure 2.2: Ordered Sphere Decoding with Channel Ordering

of the sphere decoder specially for low signal to noise ratios without compromising the performance of ML detection.

2.1.4 Simulation Results

A 4×4 MIMO system is simulated assuming a systematic feedback convolutional code (rate 1/2) and 16-QAM modulation. At the receiver, the SD method is used for MIMO detection and a Viterbi decoder [49] is used to decode the convolutional code (see Figure 2.3). We have used random interleaver/deinterleaver. In the transmitter, the interleaver reorders input symbols using random permutation and in the receiver, the deinterleaver restores ordering of input symbols using inverse permutation. Sphere decoding is used to perform ML detection of MIMO signals with reduced complexity. For standard ML detection, all the possible transmitted signals ($16^4 = 2^{16} = 65536$) should be considered to find the one with the highest probability of happening. Therefore, the standard ML detection algorithm has high complexity. Using sphere decoding, the number of the lattice points that should be considered for detection is really less than that of the ML detecting method resulting in lower complexity.

In Figure 2.4, the complexity of different SD methods are compared with each other and also with that of the standard ML detection method. The total complexity is characterized using the average number of floating point operations. As a complexity measure, instead of the complexity itself, it is useful to use the complexity exponent e_c [50] such that

$$\text{Average \# of floating point operations} = m^{e_c} \tag{2.20}$$

where $m = 2M$ and M is the number of transmit antennas.

In our simulations, the scale α is selected such that $P_{fp} = 0.9999$ (Equation (2.13)) and from that the search radius will be 5.66. For all of the three SD methods, the search radius is adjusted every time a point is found inside the sphere. From Figure 2.4, the complexity of the standard SD method is lower than that of the standard ML method but in low signal to noise ratios (SNR's) the complexity is still high. The ordered SD method has lower complexity compared to the standard SD method. Finally, the ordered SD method

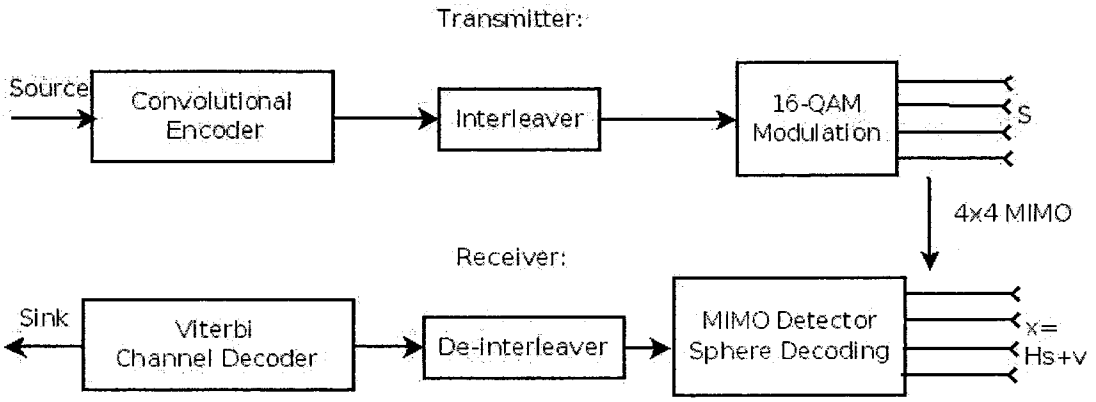

Figure 2.3: The block diagram of the MIMO system with sphere decoding as the MIMO detector.

with channel ordering has even lower complexity especially at low SNR's. An interesting property of all these SD methods is that the complexity reduces as the SNR increases. All of these three SD methods have the same performance as the standard ML method.

2.1.5 Discussions and Conclusions

Sphere decoding has been introduced as an efficient method for finding the exact solution of least-squares problems. The application of sphere decoding for MIMO detection has been investigated. The ordered SD method has been presented as a method with lower complexity compared to the standard SD method. We have also proposed a modified ordered SD method based on channel ordering. From simulation results, our proposed method has lower complexity compared to the ordered and the standard SD methods especially at low SNR's. Sphere decoding can be used to perform efficient MIMO detection without compromising the performance of ML detection.

As we mentioned at the beginning of this chapter, sphere decoding can be used to solve general least-squares problems. In Chapter 3, we use the idea of our modified ordered SD method based on channel ordering and we propose a new efficient method for decoding RS codes.

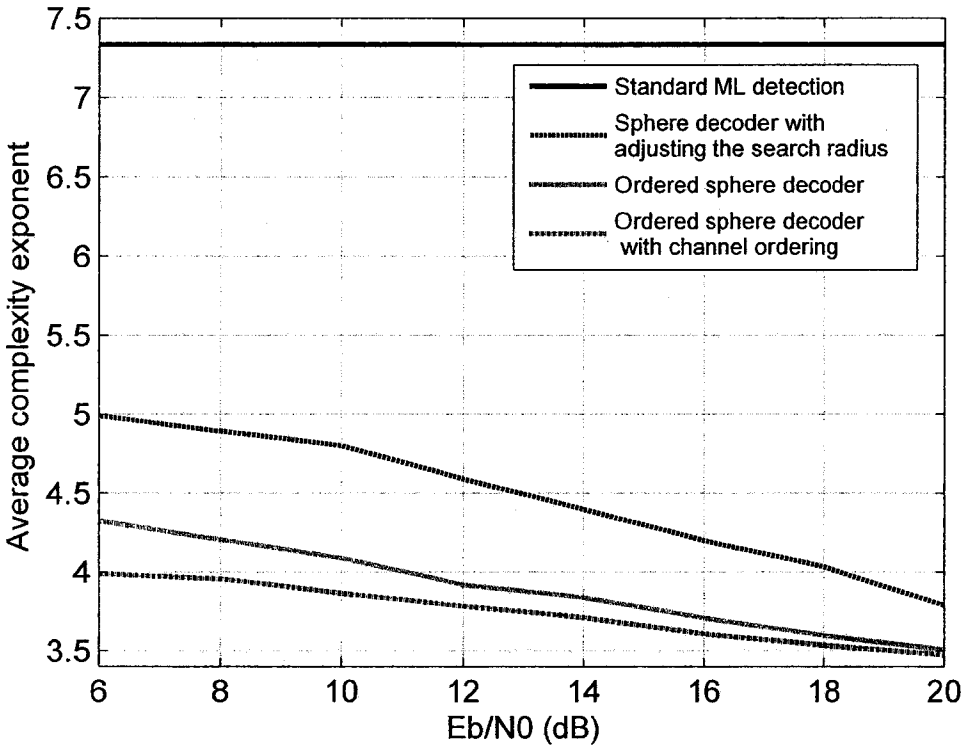

Figure 2.4: The average complexity exponent of different sphere decoding methods used for MIMO detection.

2.2 Message Passing Decoding

A factor graph is a bipartite graph that shows how a global function of many variables can be factored into a product of local functions. The sum product algorithm [38] uses a simple computational rule and performs distributed message passing in the graph to compute various marginal functions (exactly or approximately). Specific examples of factor graphs are Bayesian networks [51], Markov random fields [52] and Tanner graphs [53]. Specific examples of the sum product algorithm are the forward/backward algorithm [54], the Viterbi algorithm [49], the iterative turbo decoding algorithm [10], Pearl's belief propagation algorithm [55] for Bayesian networks and so on.

In coding theory, the first appearance of the sum product algorithm is Gallager's decoding method [9] for low density parity (LDPC) codes, now called belief propagation (BP) decoding. Tanner [53] generalized Gallager's bipartite graph approach to low complexity codes.

It has been shown that iterative decoders perform close to the Shannon capacity for long codes with sparse factor graphs. Therefore, it would be ideal if Reed-Solomon codes are suitable for this class of decoders.

In this section, we first explain BP decoding for general linear block codes. Then, we investigate the application of BP decoding for RS codes and introduce one of the most successful modified BP methods, called the ADP method, for decoding RS codes.

2.2.1 The Basis of BP Decoding for Linear Block Codes

The bipartite graph [38] of an (n, k) linear block code is formed using its parity check matrix H which is an $(n-k) \times n$ matrix (Figure 2.6). In this graph, there are two types of nodes: $(n-k)$ check nodes and n variable nodes. For any codeword c of this code we have $Hc^T = 0$ where 'T' denotes the transpose operation. This equation specifies the set of linear constraints satisfied by the codeword bits. In the bipartite graph, the set of variable nodes represents the codeword bits and the set of check nodes represents the set of parity-check constraints satisfied by the codeword bits. There is also a set of edges that

connect every check node with all the variables nodes involved in its check equation. We denote the number of the check equations that a variable node $i, i = 1, 2, ..., n$ is involved with by dv_i and refer to it as the degree of that variable node. Also, we denote the number of the variable nodes that are involved in a check equation $i, i = 1, 2, ..., (n-k)$ by dc_i and refer to it as the degree of that check node. The (edge) degree distributions of the code are defined as

$$\lambda(x) = \sum_{i=1}^{dv_{max}} \lambda_i x^{i-1},$$
$$\rho(x) = \sum_{i=1}^{dc_{max}} \rho_i x^{i-1}. \tag{2.21}$$

Here λ_i (ρ_i) is equal to the fraction of edges that connect to variable (check) nodes of degree i.

BP decoding is an iterative decoding method that receives the reliabilities of codeword bits from the channel and performs message passing (from variable nodes to check nodes and vice versa) using the bipartite graph of the code to update the reliability information based on the parity check constraints.

In order to explain the algorithm of BP decoding, we define Nc_i as the set of variable nodes participating in check equation i and Nv_j as the set of check nodes that variable node j is involved with. The log-likelihood ratio (LLR) of the jth variable node given the information about all parity check nodes except node i is shown by $Q_{i,j}$ and the LLR that check node i is satisfied when variable node j is fixed to 0 and 1 respectively is shown by $R_{i,j}$. Given the vector ρ^{in} of initial LLR's, the BP algorithm (Figure 2.5) outputs the extrinsic LLR's ρ^x [13].

The stopping criterion could be when all the checks are satisfied or when we reach the maximum number of iterations. Using the extrinsic LLR's at the output of the algorithm, we can perform hard decision decoding using the BM algorithm to decode the received signal.

Data: Received LLR's ρ^{in} and the parity check matrix H.
Result: Updated LLR's ρ^{x}.
begin
  $\forall(i,j), H(i,j) = 1$:
1 Initialization: $Q_{i,j} = \rho_j^{in}$
2 **while** *stopping criterion is not met* **do**
3 Horizontal step (check nodes updates):
   $R_{i,j} = 2tanh^{-1}(\Pi_{k \in Nc_i \backslash j} tanh(Q_{i,k}/2))$
4 Vertical step (variable nodes updates):
   $Q_{i,j} = \rho_j^{in} + \sum_{k \in Nv_j \backslash i} R_{k,j}$
5 $\rho_j^x = \sum_{k \in Nv_j} R_{k,j}, j = 1, 2, ..., n$
end

Figure 2.5: Belief Propagation Decoding

2.2.2 Standard Belief Propagation decoding of RS Codes

The parity check matrix of an (N, K) RS code over the Galois field $GF(2^p)$ can be represented by

$$\mathbf{H} = \begin{pmatrix} 1 & \alpha & \alpha^2 & ... & \alpha^{(N-1)} \\ 1 & \alpha^2 & \alpha^4 & ... & \alpha^{2(N-1)} \\ & & & ... & \\ 1 & \alpha^{(N-K)} & \alpha^{2(N-K)} & ... & \alpha^{(N-K)(N-1)} \end{pmatrix}. \tag{2.22}$$

where α is a primitive element of $GF(2^p)$. For any codeword c of the RS code, $Hc^T = 0$. Since any element $\beta \in GF(2^p)$ has a p tuple representation, we can show any codeword of length N in binary form as

$$c_b = (c_{1,1}, c_{1,2}, ..., c_{1,p}, c_{2,1}, ..., c_{2,p}, ..., c_{N,1}, ..., c_{N,p}). \tag{2.23}$$

For decoding RS codes using belief propagation, we consider RS codes over an extension field of $GF(2)$. We denote a primitive polynomial of $GF(2^p)$ over $GF(2)$ by $p(x) = a_0 + a_1 x + ... + a_{p-1}x^{p-1} + x^p$. We also suppose that α is a root of $p(x)$ and therefore a primitive element in $GF(2^p)$. For $p(x)$, there is a $p \times p$ companion matrix which is

given as

$$\mathbf{c_p} = \begin{pmatrix} 0 & \dots & 0 & a_0 \\ & & & a_1 \\ & I_{p-1} & & \vdots \\ & & & a_{p-1} \end{pmatrix} \tag{2.24}$$

where I_{p-1} is a $(p-1) \times (p-1)$ identity matrix [56]. A field isomorphism can be defined by the mapping $\alpha^i \rightarrow c_p^i$, $i = \{0, 1, ...\}$. Based on this mapping, each element of the parity check matrix of the code is replaced with a $p \times p$ binary matrix resulting in a binary parity check matrix H_b of size $(N-K)p \times Np$. Such a mapping results in $H_b c_b^T = 0$. From this binary parity check matrix, a bipartite graph with Np variable nodes and $(N-K)p$ check nodes can be formed for the binary image of RS codes. Using the BP algorithm in Figure 2.5, BP decoding can be used for RS codes.

Standard BP iterative decoding is not suitable for high density parity check codes such as RS codes, because for these codes, the large number of short cycles (Figure 2.6) in the factor graph will cause correlation between the messages and consequently error propagation. Low reliable, erroneous bits can significantly affect the value of high reliable bits and extra errors might be generated. Due to the message passing in BP decoding, these error are propagated and won't let the decoder converge to a codeword. Based on this fact, different modified BP decoding methods have been proposed to overcome the problem of short cycles in the bipartite graph of RS codes. The most popular BP based iterative decoding method for RS codes is the adaptive parity check (ADP) method which is explained in the next section and will be used as a reference throughout the thesis (Chapter 4).

2.2.3 Adaptive Parity Check Algorithm for Decoding RS Codes

The first successful iterative decoding method for RS codes was proposed by Jiang and Narayanan in 2004 [12]. This method is referred to as the adaptive parity-check (ADP) algorithm. In this algorithm, BP is run on the parity check matrix after reducing its independent columns corresponding to the least reliable bits to an identity submatrix. Since

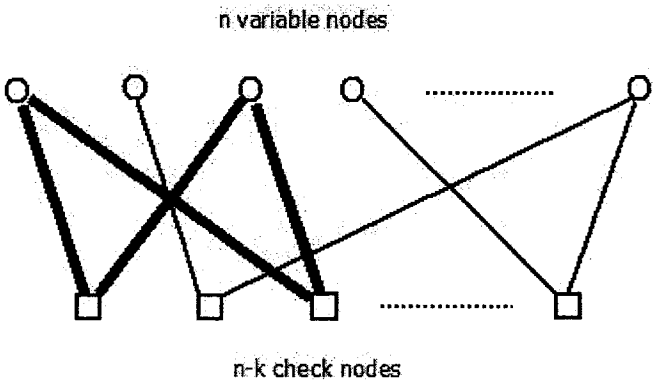

Figure 2.6: Bipartite graph of an (n,k) linear block code. A cycle of length 4 is also shown in this graph using solid lines.

$(N-K)p$ least reliable bits are not involved in any cycles, the performance of BP decoding is improved. The ADP algorithm is described in Figure 2.7.

In the original ADP method, the decoder D in ADP algorithm (Figure 2.7) is one of the following:

1- HD: Perform hard decisions (HD) on the updated LLR's. If the results satisfies the parity check equations, then a decoding success is signaled.

2- BM: Run the Berlekamp-Massey (BM) algorithm [6] on the LLR's after hard decisions. If the BM algorithm finds a codeword, a decoding success is signaled.

The stopping criterion in ADP algorithm (Figure 2.7) is when a decoding success is signaled by the decoder D or when the number of iterations is equal to the maximum number of iterations. Of course the performance largely depends on the decoder D and the stopping criterion used. For example, in [13], Koetter-Vardy (KV) soft decision decoder [29] has been used as the decoder D in ADP algorithm (Figure 2.7) resulting in impressive coding gains over previously known soft decision decoding algorithms for RS codes.

2.2.4 Discussions and Conclusions

The basis of BP decoding for general linear binary block codes has been explained. Then, the binary image of RS codes has been introduced in order to perform iterative BP decoding for these non-binary block codes. The difficulties of BP decoding of RS codes have been

Data: Channel LLR's ρ^{ch}, the binary parity check matrix H_b, the ADP damping factor $0 < \alpha \leq 1$ and the number of inner iterations It_H
Result: A list of candidate codewords $\hat{c}$.
begin
1 Initialization step: $\rho^s = \rho^{ch}$
2 **while** *Stopping criterion not satisfied* **do**
3 Sort ρ^s in ascending order of magnitude and store the permutation λ. $\rho^{in} = \lambda(\rho^s)$
4 Rearrange the columns of H_b based on λ: $\mathrm{H}_P = H_b(:, \lambda)$.
5 Gaussian elimination (GE) on H_P from left to right such that the first independent $(N-K)p$ columns in H_P are reduced to an identity sub-matrix $\rightarrow \hat{H}_P$
6 Run BP decoding algorithm (Figure 2.5) with inputs $\hat{H}_P$ and ρ^{in} for a maximum number of It_H iterations. The output: extrinsic LLR's ρ^x.
7 Update the LLR's: $\rho^q = \rho^{in} + \alpha\rho^x$ and $\rho^s = \lambda^{-1}(\rho^q)$.
8 Decode ρ^s using the decoding algorithm D to find a candidate codeword $\hat{c}$.
end

Figure 2.7: Adaptive Parity-Check (ADP) Decoding Method for RS Codes

discussed and a popular modified BP based method (ADP) for effective decoding of RS codes has been presented.

The disadvantage of the ADP method is the high complexity of performing Gaussian elimination on the binary parity check matrix at every iteration. In Chapter 4, we use the basis of BP decoding to develop efficient iterative decoding methods for RS codes.

2.3 Interpolation-based Decoding of RS Codes

Decoding RS codes is equivalent to the problem of reconstructing univariate polynomials from their noisy evaluations. This is a task best performed using ML decoding. However, as mentioned in Chapter 1, ML decoding of RS codes is NP complete. Conventional Berlekamp-Massey (BM) decoding [6] tries to solve this problem using univariate polynomial interpolation. If a codeword $c = (f(x_1), f(x_2), ..., f(x_N))$ of an (N, K) RS code over $GF(q)$ was transmitted and a vector $y = (y_1, y_2, ..., y_N) \in GF(q)^N$ was received, BM algorithm tries to construct a univariate polynomial of degree less than K that passes

through as many as possible of the received points $y_1, y_2, ..., y_N$.

Guruswami and Sudan [11] have improved the performance of the BM algorithm by introducing bivariate polynomial interpolation. Their method is referred to as the GS method. In general, the hard decision decoding task consists of finding the codeword $c \in RS(N, K)$ such that the Hamming weight $wt(e)$ of the error vector $e = y - c$ is minimized. The BM algorithm is used to do this if $wt(e) < \frac{d_{min}}{2}$ with $d_{min} = N - K + 1$ the minimum distance of the code. The GS method is a polynomial-time algorithm that achieves error correction substantially beyond half the minimum distance of the code.

In the following, we describe the GS decoding algorithm. Then an algebraic soft decision decoding method, called the KV method [29], that is based on the idea of the GS method is explained.

2.3.1 Guruswami-Sudan (GS) Decoding of RS Codes

Given the hard decision received vector $y = \{y_1, y_2, ..., y_N\} \in GF(q)$ and the corresponding support set $D = \{x_1, x_2, ..., x_N\} \subset GF(q)$, we consider the set of pairs $P = \{(x_1, y_1), (x_2, y_2), ..., (x_N, y_N)\}$ as points in a two-dimensional space. The GS algorithm has two major steps:

1. Interpolation step: Given the set P and a positive integer m, compute a nontrivial bivariate polynomial $Q_P(X, Y)$ of minimal $(1, K-1)$- weighted degree that passes through all the points in P with multiplicity at least m. Koetter's interpolation algorithm [57] can be used to find $Q_P(X, Y)$.

2. Factorization step: Given the bivariate polynomial $Q_P(X, Y)$, identify all polynomials $f(X)$ of degree less than K such that $Q_P(X, f(X)) = 0$. The output of the algorithm is a list of the codewords that correspond to these polynomials. This task is best performed using the Roth-Ruckenstein algorithm [58].

Among the list of the codewords obtained in the factorization step, the one with lowest Hamming distance from the received vector y is chosen as the response of the GS

decoding algorithm. The error correction capability of the GS method is

$$t = \left\lfloor N - N\sqrt{\frac{K-1}{N}(1+\frac{1}{m})} - \frac{1}{m} \right\rfloor. \tag{2.25}$$

As the multiplicity $m \to \infty$, the algorithm corrects any fraction of $\tau \leq 1 - \sqrt{\frac{K-1}{N}}$ erroneous positions.

2.3.2 Koetter-Vardy (KV) Algebraic Soft Decision Decoding

In many situations, the decoder can be supplied with probabilistic reliability information concerning the received symbols. A decoding algorithm that utilizes such information is generally referred to as a soft decision decoding algorithm. In this section, we present a soft decision decoder called the Koetter-Vardy (KV) method. It is based on the GS algorithm but it uses the reliability information at the channel output to construct $Q_P(X,Y)$. While the GS algorithm forces $Q_P(X,Y)$ to pass through all received values with equal multiplicity, the KV method allows $Q_P(X,Y)$ to pass through received values with a multiplicity dependent on each coordinate's reliability.

Given the vector $y = \{y_1, y_2, ..., y_N\}$ observed at the channel output, we compute

$$\pi_{i,j} = Pr(c_j = \alpha_i | y_j), \qquad i = 1, 2, ..., q, j = 1, 2, ..., N \tag{2.26}$$

where α_i's are the elements of the Galois field $GF(q)$. Let π be the $q \times N$ matrix with entries $\pi_{i,j}$ defined above. π is called the reliability matrix and is considered as the input to the KV soft decision decoding algorithm.

A soft decision decoder works directly with the probabilities compiled in the reliability matrix π. If the decoder is algebraic, it must somehow convert these probabilities into algebraic conditions. The algebraic KV method converts the reliability matrix into a choice of interpolation points and their multiplicities [29].

In order to work with the interpolation points and their multiplicities, Koetter and Vardy have introduced the multiplicity matrix. A multiplicity matrix is a $q \times N$ matrix M with nonnegative integer entries $m_{i,j}$. Thus the first step of the decoding algorithm consists

of computing the multiplicity matrix from the reliability matrix [29]. From there, the soft decision decoder proceeds as in the GS method [11]:

1. Soft interpolation step: Given the point set D and the multiplicity matrix $M = [m_{i,j}]$, compute a nontrivial bivariate polynomial $Q_M(X, Y)$ of minimal $(1, K-1)$-weighted degree that has a zero of multiplicity at least $m_{i,j}$ at the point (α_i, y_j) for every i, j such that $m_{i,j} \neq 0$.

2. Factorization step: is identical to the factorization step of the GS algorithm, described in the previous section.

The KV soft decoding algorithm outperforms the GS hard decoding method by a substantial margin. In the next chapters, we use the KV method as a reference for comparison.

2.3.3 Discussions and Conclusions

We have introduced interpolation based decoding of RS codes including the hard decision GS method and the soft decision KV method. Using the soft information from the channel, the KV method has shown considerable coding gain compared to the GS method.

In Chapter 5, we use the idea of the GS method and propose a collaborative decoding strategy for interleaved RS codes. In the presence of burst errors, this collaborative method provides higher error correction capability compared to the GS method.

Chapter 3

Efficient Soft Decoding of RS Codes based on Sphere Decoding

In this chapter, a novel soft decision decoding method for RS codes based on sphere decoding [32] [14] is proposed. In the presence of error correcting coding, the symbol space forms a sparse lattice and ML decoding is actually a search for the closest point in the sparse lattice to the received vector. Sphere decoding is a complexity reducing method that can solve the closest point search without performing an exhaustive search over the entire lattice. It only considers those lattice points that fall within a sphere centered at the received vector and among them identifies the one with minimum distance from the received vector which is actually the ML point. This results in a considerable reduction in complexity. In [33], the concept of sphere decoding for joint ML detection and decoding of linear block codes on Gaussian vector channels has been investigated. The basis of sphere decoding and two methods for reducing its complexity have been explained in Chapter 2, Section 2.1.

In this chapter, we use sphere decoding and the most reliable basis (MRB) to design a decoding algorithm for RS codes on AWGN and Rayleigh fading channels. Two types of ordering are used to make the sphere decoding faster and therefore less complex. In our proposed algorithm, we try to find those lattice points that fall within a sphere centered at the received vector and also are valid codewords. In order to consider only the lattice

points that are valid codewords of an (N, K) RS code, the search using the sphere decoder first selects a tentative solution consisting of the K most reliable basis (code symbols) (MRB) whose distance from the corresponding symbols in the received vector is less than the search radius. The acceptable values for each of these K code symbols are determined based on the ordered set of most probable transmitted symbols. Since RS codes are maximum distance separable (MDS), each time the sphere decoder selects K code symbols, they can be used to find the rest of RS symbols. If the resulting codeword is within the search radius, it is saved as a possible transmitted codeword. In the end, a list of codewords inside the sphere is found. Among these codewords, the one with minimum Euclidean distance from the received vector is chosen as the output of the decoding algorithm.

The search radius of sphere decoding should be selected carefully. We set the search radius to be the distance between the hard-decision decoded codeword (with Berlekamp-Massey (BM) algorithm [6]) and the received signal. Our algorithm works based on the Euclidean distance which means that we start from the hard-decision decoded codeword and we try to find more probable codewords with smaller distances from the received signal. In cases where the BM decoding is not successful, the radius has to be selected large enough in such a way that with high probability at least one codeword fits inside the sphere.

For short RS codes, our proposed algorithm can perform ML decoding with considerable reduction in the complexity. However, for long RS codes, in order to reduce the decoding complexity, we have to apply some limitations on our algorithm which lead to a suboptimum decoder with moderate complexity and very good performance compared to the hard decision decoding.

The rest of this chapter is organized as follows. In Section 3.1, the system model for RS encoding and transmission is introduced. Then, the algorithm for soft decision decoding of RS codes using sphere decoding is presented in Section 3.2. The complexity analysis of the proposed method is provided in Section 3.3. The modified version of the proposed method for the case of BPSK modulation is presented in Section 3.4. We also propose bit-level sphere decoding of RS codes using their binary image in Section 3.5. Simulation results and discussions are provided in Section 3.6 where the performance of

different RS codes over different channels with different modulations is considered. We explore the amount of coding gain that our algorithm provides on different channels. Finally, conclusions are presented in Section 3.7.

3.1 System Model and Problem Statement

We consider an (N, K) RS code over Galois field $GF(2^p)$ with $N = 2^p - 1$. A vector b of K information symbols is encoded using the systematic generator matrix of the code, denoted by G, to form a codeword c of length N. The encoding process is given by

$$c = bG \tag{3.1}$$

where the matrix multiplication is done over $GF(2^p)$. To simplify the explanation of our decoding algorithm in the next sections, we choose the QAM modulation with a constellation size equal to the size of the Galois Field. Later, in Section 3.4, we consider the case of BPSK modulation. The modulated signal x can be written as

$$x = f(c) \tag{3.2}$$

where f denotes the mapping from the code symbols to the constellation symbols. Here, we consider both the AWGN channel and the Rayleigh fading channel. The received signal at the output of the Rayleigh fading channel can be written as

$$y_i = \alpha_i x_i + z_i, \qquad i = 1, 2, ..., N \tag{3.3}$$

where x is the $1 \times N$ transmitted vector with entries from a complex-valued 2^p-QAM constellation, α_i, $i = 1, 2, ..., N$ are independent and identically distributed (i.i.d.) fading coefficients from a circularly symmetric complex Gaussian distribution with zero mean and unit variance, v is the $1 \times N$ noise vector with entries from a zero mean complex Gaussian distribution with variance σ^2 and y is the $1 \times N$ received vector. For an AWGN channel, we simply write $y_i = x_i + z_i$, $i = 1, 2, ..., N$.

For ML decoding of the Reed-Solomon code, we should solve the following problem:

$$\underset{b \in GF(2^p)^K}{\operatorname{argmax}} \; p(y|b) \tag{3.4}$$

which is equivalent to

$$\underset{b \in GF(2^p)^K}{\operatorname{argmin}} \; \|y - Hx\|^2 \tag{3.5}$$

where $x = f(bG)$. So, for ML decoding we should consider all the possible information vectors to determine the most probable one which is of prohibitive complexity especially for long RS codes that are used in practical systems. One of the popular methods to reduce the complexity of ML decoding is sphere decoding. In the following section, we introduce an algorithm using sphere decoding for soft decision decoding of RS codes.

3.2 RS Decoder Algorithm using Sphere Decoding

In this section, using the basis of sphere decoding, we present a soft decoding method for RS codes. Assuming the fading coefficients are known at the receiver, our proposed algorithm is explained with the following steps:

Step 1: The hard-decision decoded codeword is generated using the Berlekamp-Massey (BM) algorithm [6].

Step 2: The received vector y has N elements. For each of them, a set of probable transmitted symbols $(x_i's)$ is determined. When short RS codes are considered, for each element of the received vector, we consider all of the possible transmitted symbols. However, when the RS code is long, in order to avoid high complexity, only a small set of most probable transmitted symbols is considered for each received element. This can be done by solving

$$\|y_i - \alpha_i x_i\| \leq r_e. \tag{3.6}$$

The radius r_e can be adjusted in order to consider the desired number of transmitted symbols. The set of candidate symbols for each received element will be ordered from the most probable one to the least probable one. For example, the ordered set with s elements

corresponding to y_i is

$$\left\{x_i^{(1)}, x_i^{(2)}, ..., x_i^{(s)}\right\} \tag{3.7}$$

such that

$$\left\|y_i - \alpha_i x_i^{(1)}\right\| \leq \left\|y_i - \alpha_i x_i^{(2)}\right\| \leq ... \leq \left\|y_i - \alpha_i x_i^{(s)}\right\|. \tag{3.8}$$

Step 3: Similar to binary codes [22] [26], here we can use only the two most probable symbols from step 2 to define the reliability of each received element as

$$LLR(y_i) \approx \left\|y_i - \alpha_i x_i^{(2)}\right\|^2 - \left\|y_i - \alpha_i x_i^{(1)}\right\|^2. \tag{3.9}$$

Using this reliability measure, the symbols of the received signal are arranged in decreasing order of reliability. This ordering defines a permutation λ_1. So, the new received signal can be written as [26]

$$w = \lambda_1(y). \tag{3.10}$$

Since RS codes are maximum distance separable, the first K elements of w correspond to the most reliable and independent positions of the RS codeword. They are called the most reliable basis (MRB).

Step 4: The columns of the generator matrix G of the code and also the fading coefficients are also permuted according to λ_1:

$$G_1 = \lambda_1(G), \tag{3.11}$$

$$\alpha^1 = \lambda_1(\alpha) \tag{3.12}$$

where α is the $1 \times N$ vector of fading coefficients. If we denote the code generated by G with C and the one generated by G_1 with C_1, we have $C_1 = \lambda_1(C)$.

Step 5: Because of the MDS property of RS codes, every K columns of G_1 are independent. Therefore, we can convert G_1 to the systematic form G_{sys} using elementary row operations. This way, the first K columns corresponding to the most reliable and independent positions of the codeword are reduced to an identity matrix.

Step 6: Now, we are ready to decode the received signal using sphere decoding. Our algorithm works based on the Euclidean distance. It means we try to find probable

transmitted codewords ($\hat{c}$'s) with small Euclidean distance from the received signal such that

$$\|y - \alpha \cdot \hat{x}\|^2 \leq r^2 \tag{3.13}$$

or equivalently

$$\left\|w - \alpha^1 \cdot \hat{x}_1\right\|^2 \leq r^2 \tag{3.14}$$

where $\hat{x} = f(\hat{c})$, $\hat{x}_1 = f(\hat{c}_1)$ and '$\cdot$' denotes the component-wise multiplication. When the hard decision BM algorithm [6] in step 1 is successful, we denote the hard-decision decoded codeword by c_{HD} and its modulated version by x_{HD}, then the search radius r of sphere decoding is set to

$$r = \|y - \alpha \cdot x_{HD}\| . \tag{3.15}$$

However, when the BM algorithm is not successful, the radius should be selected in such a way that with high probability at least one codeword fits inside the sphere. From Equation (3.3), $z_i = y_i - \alpha_i x_i$, $i = 1, ..., N$ and each z_i has the variance of σ^2. If r^2 is selected as a scaled variance of the noise, we can be confident that at least one codeword fits inside the sphere. Therefore, we choose

$$r = 2 \times \sqrt{N\sigma^2}. \tag{3.16}$$

A higher value should be chosen for r in situations that no codeword exists inside the sphere with the above radius. We have confirmed with simulation that these situations have a really low (almost zero) probability of happening.

From Equation (3.14), we have

$$\sum_{k=1}^{N} \left|w_k - \alpha_k^1 \hat{x}_1(k)\right|^2 \leq r^2. \tag{3.17}$$

We are looking for $\hat{x}_1$'s that satisfy the above equation and are also valid RS codewords. Therefore, we first try to find the first K components of $\hat{x}_1$ considering only the first K terms of the above inequality,

$$\left|w_1 - \alpha_1^1 \hat{x}_1(1)\right|^2 + \left|w_2 - \alpha_2^1 \hat{x}_1(2)\right|^2 + ... + \left|w_K - \alpha_K^1 \hat{x}_1(K)\right|^2 \leq r^2. \tag{3.18}$$

Each time we find the K components of $\hat{x}_1$ that satisfy (3.18), they can be encoded using G_{sys} to find the rest of RS symbols (based on the MDS property of RS codes). Equation (3.18) is similar to, but simpler than Equation (2.5). The first term only depends on $\hat{x}_1(1)$, the second term only on $\hat{x}_1(2)$ and so on. Considering the first term of this inequality,

$$\left|w_1 - \alpha_1^1 \hat{x}_1(1)\right|^2 \leq d(1)^2 \tag{3.19}$$

with $d(1) = r$, we try to find $\hat{x}_1(1)$. In order to do this, the ordered set of most probable transmitted symbols $\left\{x_1^{(1)}, x_1^{(2)}, ..., x_1^{(s)}\right\}$ from step 2 is considered. The elements of this set that satisfy (3.19) are considered as the acceptable values for $\hat{x}_1(1)$. Because of the ordering in step 2, these acceptable values are ordered from the most probable one to the least probable one. For each of them we can define $d(2)^2 = d(1)^2 - \left|w_1 - \alpha_1^1 \hat{x}_1(1)\right|^2$ and use the first two terms in inequality (3.18) in order to find acceptable values for $\hat{x}_1(2)$:

$$\left|w_2 - \alpha_2^1 \hat{x}_1(2)\right|^2 \leq d(2)^2. \tag{3.20}$$

Similarly, we use the ordered set of most probable transmitted symbols from step 2, $\left\{x_2^{(1)}, x_2^{(2)}, ..., x_2^{(s)}\right\}$, and the elements of this set that satisfy (3.20) are considered as the acceptable values for $\hat{x}_1(2)$. We continue this process for $\hat{x}_1(3)$, $\hat{x}_1(4)$ and so on until we obtain all the lattice points inside the sphere that satisfy (3.18). The whole process is shown in Figure 3.1. From this figure, when we select a value for $\hat{x}_1(j)$ at step j, we go to step $j+1$. For each acceptable value of $\hat{x}_1(j)$, there is a list of acceptable values for $\hat{x}_1(j+1)$ at step $j+1$. Each time we go from step $j-1$ to j, we start with the first acceptable value for $\hat{x}_1(j)$ which has the highest probability of being sent. Then we continue with less probable ones. In general when there is no more acceptable value for $\hat{x}_1(j)$ at step j, we go back to step $j-1$ and find the next acceptable value that has not been checked yet for $\hat{x}_1(j-1)$. Then, we return to step j.

Each of the lattice points that satisfies (3.18) contains K symbols of an RS codeword. These K symbols can be encoded using G_{sys} to find the other $N-K$ symbols. If the resulting codeword $\hat{c}_1$ satisfies (3.14), it will be saved as a candidate transmitted codeword. A summary of sphere decoding algorithm for RS codes is given in Figure 3.2.

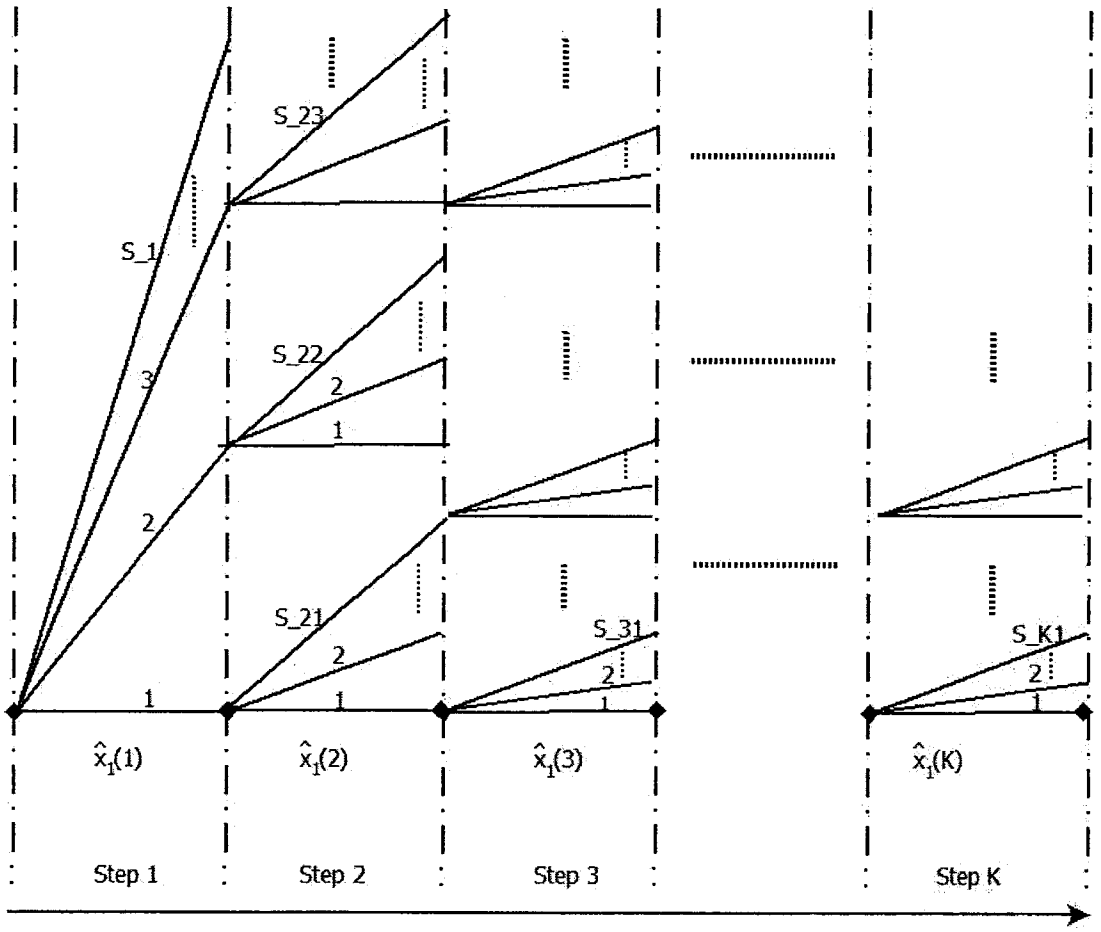

Figure 3.1: Sphere decoding of RS codes. The process of finding the first K elements of the permuted transmitted signal $\hat{x_1} = f(\hat{c_1})$ that satisfy the following inequality: $\sum_{k=1}^{K} |w_k - \alpha_k^1 \hat{x_1}(k)|^2 \leq r^2$. The index '1' on the lines in each step indicates the highest probable transmitted symbol, the index '2' the second highest probable transmitted symbol and so on.

Data: permuted fading coefficients α^1, systematic permuted generator matrix G_{sys}, search radius r, permuted received signal w and $s(k)$ the number of the elements of the ordered set corresponding to $w(k)$.

Result: A list of permuted codewords $\hat{c}_1$'s inside the sphere of radius r around the permuted received signal w.

begin

1 Initialization: $k = 1$, $d(k) = r$, $A = [00...0]_{1\times K}$.

2 **if** $k = 0$ **then**
  stop.
  else
  Go to 3.;

3 **if** $A(k) + 1 \leq s(k)$ **and** $\left|w_k - \alpha_k^1 x_k^{A(k)+1}\right|^2 \leq d(k)^2$ **then**
  $\hat{x}_1(k) = x_k^{A(k)+1}$, $A(k) = A(k) + 1$.
  else
  Go to 6.;

4 **if** $k = K$ **then**
  $\hat{b}_1 = f^{-1}(\hat{x}_1(1:K))$, $\hat{c}_1 = \hat{b}_1 G_{sys}$.
  if $\|w - \alpha^1 \cdot f(\hat{c}_1)\|^2 \leq r^2$ **then**
  save $\hat{c}_1$.
  else
  $k = k + 1$,;
  $d(k)^2 = d(k-1)^2 - \left|w_{k-1} - \alpha_{k-1}^1 \hat{x_1}(k-1)\right|^2$.;

5 Go to 2.;

6 $k = k - 1$, $A(k+1:K) = zeros(1, K-k)$, Go to 2.

end

Figure 3.2: Sphere decoding of RS codes

In order to reduce the complexity of sphere decoding algorithm (Figure 3.2), each time we find a codeword $\hat{c_1}$ inside the sphere, we replace the search radius r with $\|w - \alpha^1 \cdot f(\hat{c_1})\|$ and change vector d in the algorithm accordingly. After finishing the algorithm, we have a list of permuted candidate codewords that satisfy (3.14). We will choose the one that minimizes $\|w - \alpha^1 \cdot f(\hat{c_1})\|$ as the response of sphere decoding. The actual transmitted codeword will be

$$\hat{c} = \lambda_1^{-1}(\hat{c}_1). \tag{3.21}$$

The ordering that was discussed earlier will help us to find the candidate codewords quickly, because we first decode the K most reliable code symbols and for each of them we start from the most probable transmitted symbol. By finding the candidate codewords early and replacing the search radius by their distance from the received vector, there will be less choices of the first K elements of $\hat{x_1}$ that satisfy $\sum_{k=1}^{K} |w_k - \alpha_k^1 \hat{x_1}(k)|^2 \leq r^2$. This results in reducing the complexity of the algorithm.

Another feature of the proposed algorithm is that when the BM algorithm or the sphere decoder finds the correct transmitted codeword, the search radius will be the distance of that codeword from the received vector and therefore the algorithm can not find any codeword closer to the received vector and stops quickly.

If each ordered set in step 2 contains all of the possible transmitted symbols, the above algorithm can do ML decoding. This is perfect for short RS codes. However, it is still complex for long RS codes. In order to avoid high complexity, we have to apply some limitations on our algorithm which lead to a suboptimum decoding method. These limitations are:

1. For each received symbol, we only consider a small set of most probable transmitted symbols (for example s symbols) instead of all the possible symbols.

2. We denote the number of the elements of the ordered set corresponding to $w(k)$ with $s(k)$ which at the beginning is the same for all k's and is equal to s. However, if $\left|w_k - \alpha_k^1 x_k^{A(k)+1}\right|^2 > d(k)^2$ (line 3 of SD algorithm in Figure 3.2), it means that in most of the cases we should not go beyond $x_k^{A(k)}$ and we can set $s(k) = A(k)$. If in

some cases $s(k) = 0$, it means that there will be no other points inside the sphere and we will stop the algorithm.

3. As we discussed above, the kind of ordering we used in our algorithm helps us to find the candidate codewords quickly. Every lattice point satisfying (3.18) contains K code symbols that can be used to find an RS codeword. If this RS codeword lies inside the sphere, it is considered a candidate transmitted codeword. Therefore, for signal to noise ratios (SNR's) of our interest, if we don't find the best codeword after considering a specific number of codewords (lattice points satisfying (3.18)), it means that we have bad noise realizations. In such cases, it is not worth to search for more points inside the sphere and we stop the algorithm. Then, among the candidate codewords found by the algorithm we choose the best one. In cases this limitation is applied on the decoder, the specific number of lattice points (codewords) considered by the algorithm before being stopped is determined from simulation.

By introducing these limitations, we have made a trade-off between the performance and the complexity of our proposed method. It means that adding any of these limitations will degrade the performance of decoding from ML decoding but will also reduce the complexity considerably. Compared to hard-decision decoding, our method can improve the performance of RS codes considerably with a moderate increase in complexity.

3.3 Complexity Analysis of the Proposed Algorithm

The proposed algorithm for decoding an (N, K) RS code assuming 2^p-QAM modulation ($N = 2^p - 1$) consists of the following steps where the required complexity for each of them is considered:

1. Finding the ordered sets for all received symbols: Using Equation (3.6), by adjusting r_e, we can find the desired s most probable code symbols. Calculating the probability of s symbols for all of the N received elements has the time complexity of $O(Ns)$.

Using "Mergesort" [26], sorting s symbols for all of the N received elements has the time complexity of $O(Nslog_2 s)$.

2. Sorting the received elements based on their reliabilities: Calculating the reliabilities for N received elements has the time complexity of $O(N)$. Using "Mergesort", ordering of the received sequence based on reliabilities is achieved with about $Nlog_2 N$ floating point operations [59].

3. Systematizing the generator matrix of the code using elementary row operations in such a way that the first K columns corresponding to the most reliable and independent positions of the codeword are reduced to an identity matrix: At this step, the required number of finite field operations over $GF(2^p)$ is [60] [26]

$$O((min\{N-K,K\})^2 \times N). \tag{3.22}$$

As we can see, the complexity of the this step of the algorithm which is a Gaussian-elimination process is dominant compared to the first two steps.

4. Sphere decoding: we represent the complexity of sphere decoding using the average number of codewords (lattice points satisfying (3.18)) that are considered with our proposed algorithm. It should be mentioned that by increasing the SNR, the average number of considered codewords and consequently the complexity of the method are decreased. This is because at higher SNR's, the first K most reliable positions of the codeword have higher reliabilities which makes it faster for the sphere decoder to find the acceptable codewords especially the correct transmitted codeword. As mentioned before, after finding each acceptable codeword, the radius of the sphere is reduced which makes the sphere decoder even faster.

   In Section 3.6 that we present simulation results, we provide the average number of the codewords considered by the decoder for different SNR's.

3.4 RS Decoder Algorithm using Sphere Decoding for BPSK Modulation

In this section, we consider the case of BPSK modulation. For simplicity of explanation, we choose the AWGN channel.

Since any element $\beta \in GF(2^p)$ has an p tuple representation, we can show any codeword of length N in binary form:

$$c_b = (c_{1,1}, c_{1,2}, ..., c_{1,p}, c_{2,1}, ..., c_{2,p}, ..., c_{N,1}, ..., c_{N,p}). \tag{3.23}$$

Assuming BPSK modulation, the transmitted signal can be written as $x = -2c_b + 1$. The received signal of length Np at the output of an AWGN channel is

$$y = x + n \tag{3.24}$$

where entries of n are from a zero mean Gaussian distribution with variance σ^2. The reliability of the received vector can be expressed in terms of the log likelihood ratios (LLR's) that are given by $\rho^{ch} = 2y/\sigma^2$.

The proposed decoding algorithm has been described assuming 2^p-QAM modulation. We have to make some changes to our algorithm in order to be used for BPSK modulation. An RS codeword contains non-binary symbols each having p bits. So, we have to divide the received sequence into N groups each containing p elements. For each group, we determine an ordered set of most probable sequences with p elements from $\{+1, -1\}$. For example, if we denote a group with g, its ordered set will be $\{g_1, g_2, ..., g_s\}$ such that

$$\|g - g_1\|^2 \leq \|g - g_2\|^2 \leq ... \leq \|g - g_s\|^2 \tag{3.25}$$

where the maximum value for s is 2^p. From this list we can also define the reliability of each group as

$$LLR(g) = \|g - g_2\|^2 - \|g - g_1\|^2. \tag{3.26}$$

Based on these ordered sets and the reliabilities, our proposed decoding algorithm can be applied on BPSK modulated RS codes over the AWGN channel. However, as we will

see in simulation results, the average number of the codewords (lattice points satisfying (3.18)) considered by the algorithm is rather high specially at low signal to noise ratios. This results in high complexity. Also, the coding gain is not as good as the case of 2^p-QAM modulation.

Here, we explain what might causes these problems. With BPSK modulation, we have the reliabilities for bits at the receiver. We group every p reliabilities to calculate the reliability for every symbol. We arrange the reliabilities of symbols for soft decision decoding. Therefore, there might be low reliable bits in a high reliable symbol and vice versa. As a result, the algorithm has to consider a lot of lattice points inside the sphere before finding the correct transmitted codeword. Also, there is a higher chance that we do not find the correct codeword at all (assuming some of the three limitations introduced in section 3.2 are applied on the decoding method) which degrades the performance. In order to improve the performance of decoding for the case of BPSK modulation, we propose to use the binary image of the RS code.

3.5 Bit-level Decoding of RS Codes using Sphere Decoding

In this section, we explain how to use the binary image of RS codes for bit-level decoding in cases that we have BPSK modulation. The parity check matrix of an (N, K) RS code over $GF(2^p)$ is denoted by H. For any codeword c of the RS code, $Hc^T = 0$.

Here for the decoding, we consider RS codes over an extension field of $GF(2)$. From Chapter 2, Section 2.2, each element of the parity check matrix of the code can be replaced with a $p \times p$ binary matrix resulting in a binary parity check matrix H_b of size $(N-K)p \times Np$ [56]. If the binary form of the codeword c is denoted by c_b, we have $H_b c_b^T = 0$.

The binary parity check matrix H_b can be converted to the systematic from using elementary row operations such that

$$H_{b-sys} = \left[P | I_{(N-K)p}\right] \tag{3.27}$$

where P is an $(N-K)p \times Kp$ matrix and I is an $(N-K)p \times (N-K)p$ identity matrix.

Then the generator matrix of the binary image of the RS code can be written as

$$G = \left[I_{Kp} | P^T\right]. \tag{3.28}$$

We use this binary generator matrix and using the same algorithm based on sphere decoding we decode the received vector to a binary image codeword. The difference here is that the code length is Np and the code dimension is Kp. Also, since the binary image of the RS code is no longer an MDS code, step 5 of the algorithm has to be modified as follows.

Step 5: The reliability of the ith element of y corresponds to the ith column of the code generator matrix G. In this step, we rearrange the columns of G_1 to generate another matrix G_2 such that the first Kp columns of G_2 be the Kp linearly independent columns of G_1 with largest associated reliability values and also maintain the decreasing order of their reliability values. The remaining $(N-K)p$ columns of G_1 are also arranged in the order of decreasing their associated reliability values and form the remaining $(N-K)p$ columns of G_2. This process defines another permutation λ_2 [26]. So, the new received signal, the new code generator matrix and the new fading coefficients are

$$v = \lambda_2(w), \tag{3.29}$$

$$G_2 = \lambda_2(G_1), \tag{3.30}$$

$$\alpha^2 = \lambda_2(\alpha^1). \tag{3.31}$$

If we denote the code generated by G_2 with C_2, we have $C_2 = \lambda_2(\lambda_1(C))$. Now, we are ready to convert G_2 to the systematic form G_{sys} using elementary row operations. This way, the first Kp columns corresponding to the most reliable and independent positions of the codeword are reduced to an identity matrix. The required binary operations for this process can be shown as

$$O((min\{(N-K)p, Kp\})^2 \times Np). \tag{3.32}$$

In SD algorithm (Figure 3.2), instead of w and α^1, we should use v and α^2. The permuted candidate codewords are also denoted by $\hat{c}_2$'s. Finally, the actual transmitted codeword is $\hat{c} = \lambda_1^{-1}(\lambda_2^{-1}(\hat{c}_2))$.

3.6 Simulation Results and Discussions

In this section, we explore the amount of soft-decision gain that our algorithm provides on different channels. In Sections 3.6.1 and 3.6.2, Rayleigh fading channels and AWGN channels are considered respectively.

We compare our algorithm with the BM hard decision decoding [6] method and also the algebraic soft decision decoding algorithm proposed by Koetter and Vardy [29] that will be mentioned by the KV algorithm. We especially use the simulation results of the KV algorithm from [61] and [13] for comparison. The BM hard decision decoding has the time complexity of $O(Nd_{min})$ where $d_{min} = N - K + 1$ is the minimum distance of the RS code. To have an idea about the complexity of KV algorithm, it should be mentioned that it has four major steps. Calculating the reliability matrix has a time complexity of $O(N^2)$. Multiplicity assignment has also a time complexity of $O(N^2)$. Solving the interpolation problem has the best complexity of $O(N^2\lambda^4)$ [62] where $\lambda = (-1 + \sqrt{1 + 8\,|M|\,/N})/2$ and $|M|$ is the interpolation cost of the multiplicity matrix M. Finally, efficient factorization algorithm proposed by Roth and Ruckenstein [58] has a time complexity of $O((l \times log^2 l)K(N + l \times logq))$ where $q = N + 1$ and l is an upper bound on the KV list size and is determined by λ.

In order to have an idea of the complexity of the proposed algorithm, we refer the reader to Section 3.3. The total complexity of the proposed decoding algorithm is described using the average number of floating point operations. As a complexity measure, instead of the complexity itself, it is useful to use the complexity exponent e_c [50] where

$$\text{Average \# of floating point operations} = N^{e_c} \tag{3.33}$$

and N is the code length.

For each simulation in this section, we provide a table of the average number of the codewords that are checked by the sphere decoder for different SNR's. Also, the total complexity exponent of the proposed algorithm is measured for different SNR's and compared with that of the KV method.

3.6.1 Rayleigh Fading Channel

1- RS(15,11) code and 16-QAM modulation: In Figure 3.3, we compare the performance of different decoding algorithms. We use our proposed decoding algorithm for both ML and suboptimum decoding. In suboptimum decoding, we only apply the second limitation mentioned before on our algorithm. Suboptimum decoding provides slightly better performance compared to the KV algorithm while ML decoding using our algorithm provides about 1 dB coding gain compared to the KV algorithm with $m_{max} = 100$. We should mention that $m_{max} = \lfloor \lambda \rfloor$ and the higher its value, the higher the complexity of the KV method [61]. At codeword error rates of 10^{-3} and lower, ML decoding using the proposed algorithm provides a coding gain of more than 7.5 dB compared to the conventional hard decision decoding.

The average number of the codewords that are checked by the algorithm for both cases of ML and suboptimum decoding is given in Table 3.1. As we can see, adding the second limitation to the algorithm degrades the performance but at the same time reduces the number of codewords that should be checked by the algorithm which means less complexity. In Figure 3.4, the total complexity exponent of the proposed method for both ML and suboptimum decoding has been compared with that of the KV algorithm for two cases ($m_{max} = 4, 100$). As we can see, our algorithm provides better performance and at the same time less complexity compared to the KV method. An interesting property of our method is that the complexity decreases as the SNR increases.

Table 3.1: Average number of the codewords that are considered for ML and suboptimum decoding of RS(15,11) with 16-QAM modulation on a Rayleigh fading channel

Eb/N0 (dB)	5	7	9	11	13	15
Average number of the codewords, ML decoding	753	375	133	18	8	3
Average number of the codewords, Suboptimum decoding	104	51	29	15	5	1

2- RS(255,239) code and 256-QAM modulation: As we mentioned before, in order to avoid increase in decoding complexity of this long RS code, we apply all of the three

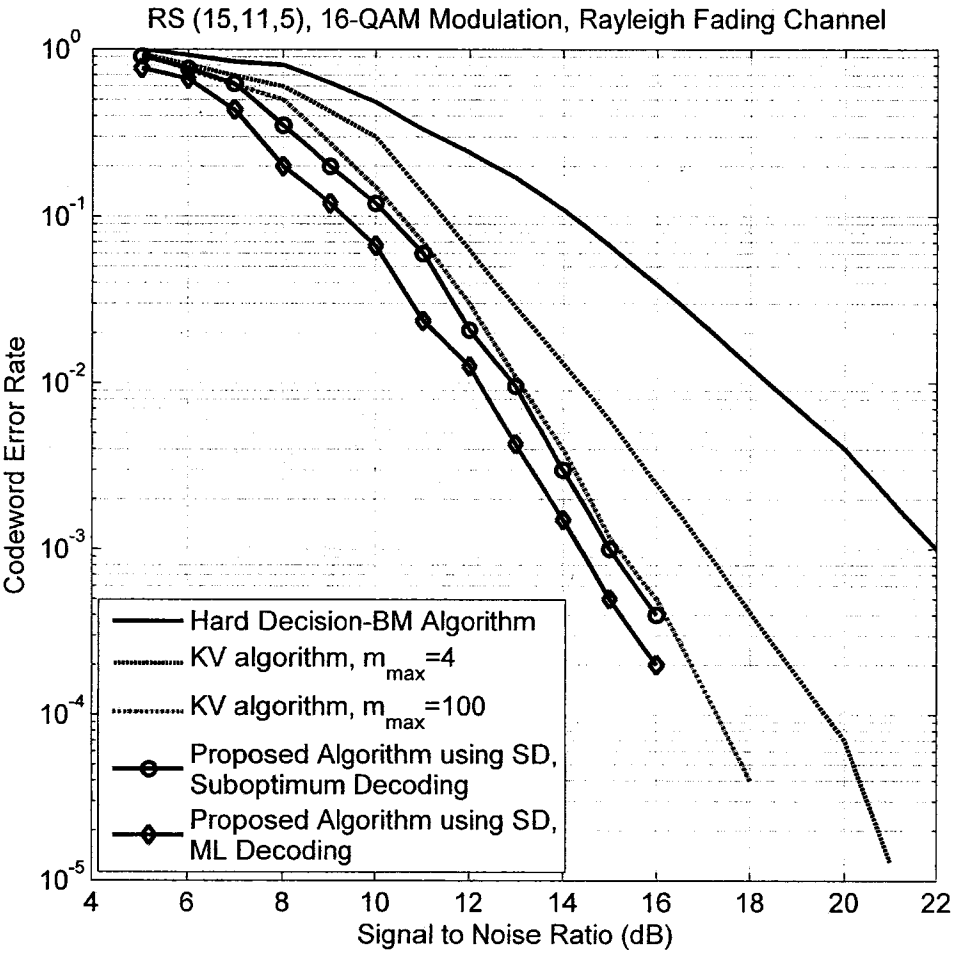

Figure 3.3: Performance of the proposed algorithm for ML and suboptimum decoding of RS (15,11) with 16-QAM modulation on a Rayleigh fading channel.

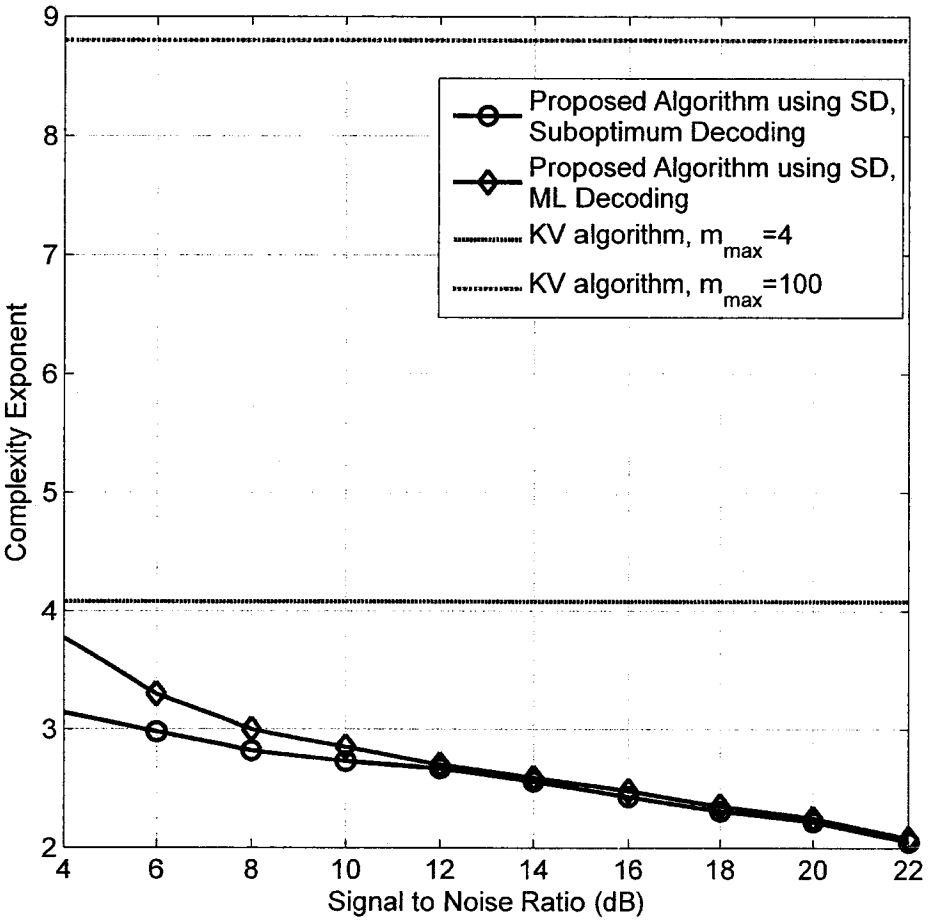

Figure 3.4: Complexity exponent of the proposed algorithm for ML and suboptimum decoding of RS (15,11) with 16-QAM modulation on a Rayleigh fading channel.

limitations discussed before on our algorithm. For each received element we only consider a small set of most probable transmitted symbols (4 in this case). The number of codewords (lattice points) considered by the algorithm before being stopped is limited to 8000. Performance of this code is shown in Figure 3.5. At codeword error rates of 10^{-3} and lower, suboptimum decoding using the proposed algorithm provides a coding gain of more than 3 dB compared to the BM hard decision decoding method, 0.3 dB compared to the KV method with $m_{max} = 100$ and 1.3 dB compared to the KV method with $m_{max} = 4$.

Table 3.2 shows the average number of the codewords that are considered by the algorithm. Our proposed method can do suboptimum soft decoding for this long RS code by checking a relatively small number of codewords which results in moderate complexity. In Figure 3.6, the total complexity exponent of the proposed method has been compared with that of the KV algorithm for two cases ($m_{max} = 4, 100$). Except for low SNR's, our method has lower complexity than the KV method with $m_{max} = 4$.

Table 3.2: Average number of the codewords that are considered for suboptimum decoding of RS(255,239) with 256-QAM modulation on a Rayleigh fading channel

Eb/N0 (dB)	23	24	25	26	27
Average number of the codewords	4038	1144	508.21	107.6	40.92

3.6.2 AWGN Channel

In this section, we consider the case where BPSK modulated bits are transmitted over the AWGN channel.

1- RS(15,11) code: In Figure 3.7, we compare the performance of different decoding algorithms. Here, we only apply the third limitation discussed before on our algorithm. The number of codewords (lattice points) considered by the algorithm before being stopped is limited to 10000. We can see from Figure 3.7 that at codeword error rate of 10^{-3}, symbol-level suboptimum decoding provides more than 0.5 dB coding gain compared to the asymptotic performance of the KV algorithm and 1.7 dB compared to the conventional hard decision decoding. In this figure, the performance of bit-level sphere decoding using

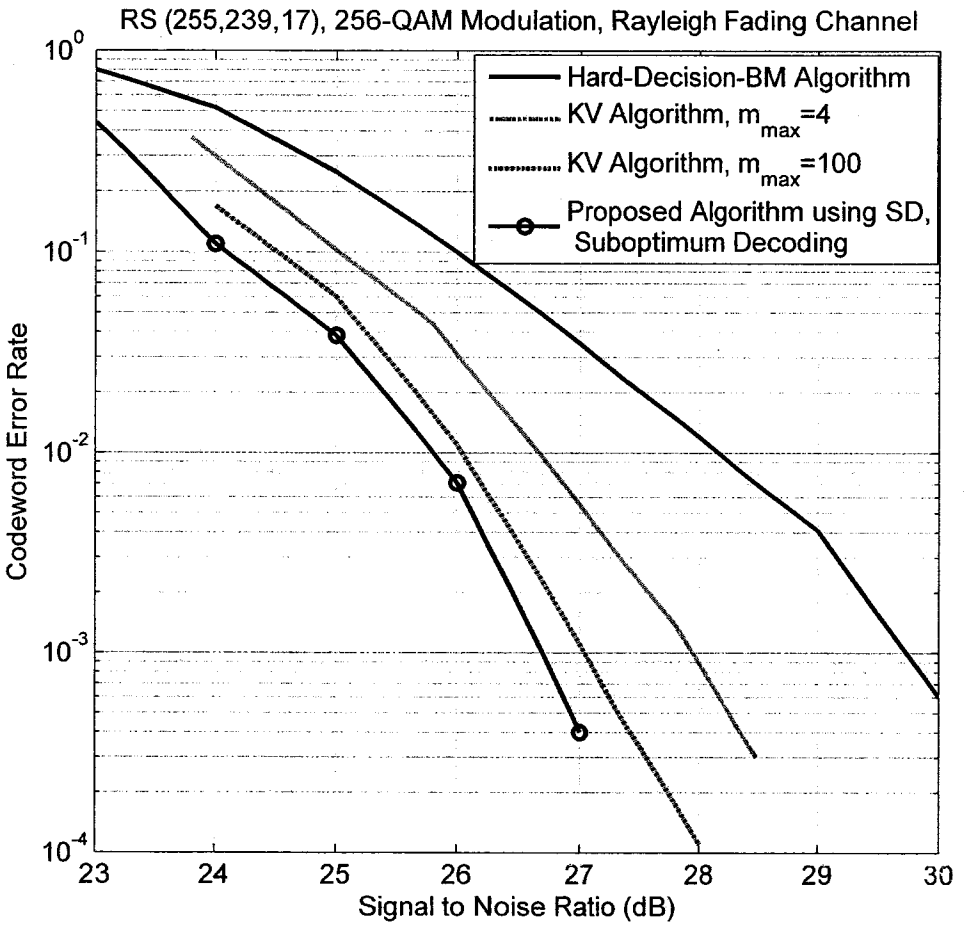

Figure 3.5: Performance of the proposed algorithm for suboptimum decoding of RS (255,239) with 256-QAM modulation on a Rayleigh fading channel.

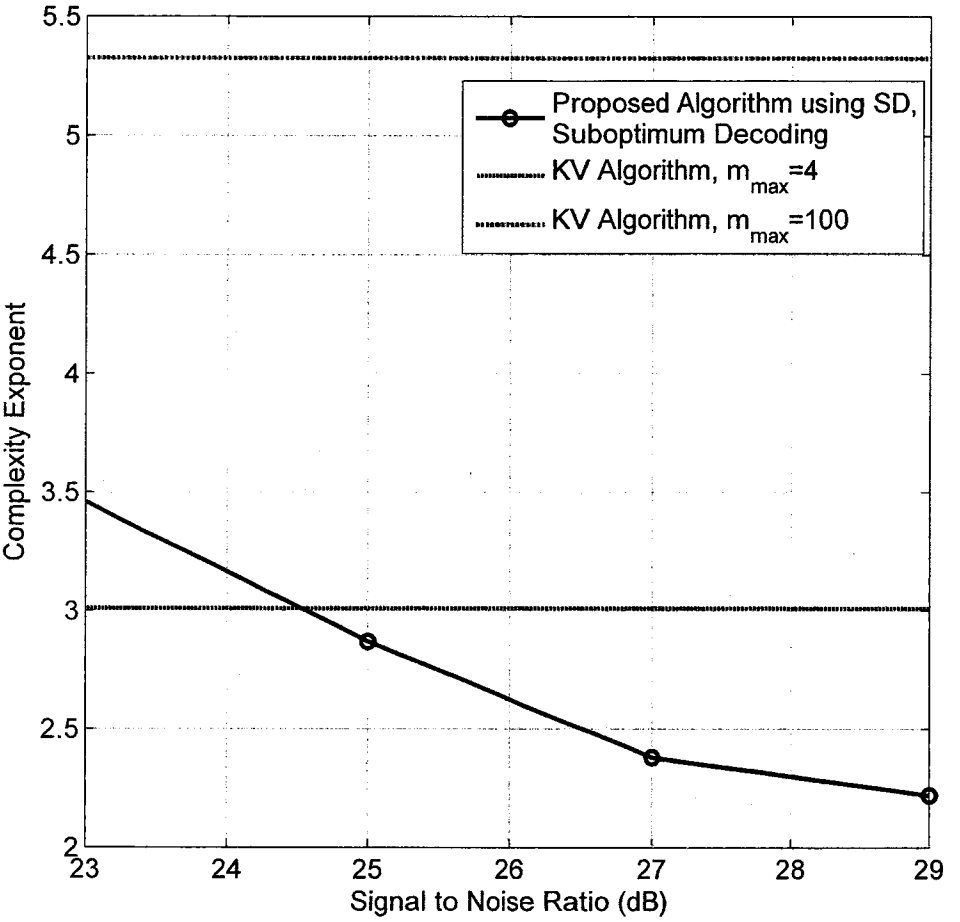

Figure 3.6: Complexity exponent of the proposed algorithm for suboptimum decoding of RS (255,239) with 256-QAM modulation on a Rayleigh fading channel.

the binary image of the code is also shown. We can see that bit-level decoding provides a coding gain of 0.7 dB compared to the symbol-level decoding at codeword error rates of 10^{-3} and lower. Actually, the performance of bit-level sphere decoding is 0.2 dB away from the best performance possible which is of ML simulation.

From Table 3.3, the bit-level decoding method requires to check considerably less codewords compared to the symbol-level decoding specially at low signal to noise ratios. Therefore, in the case of BPSK modulation, bit-level decoding using the binary image of RS codes provides better performance and at the same time less complexity compared to symbol-level decoding. The total complexity exponent (Figure 3.8) of both symbol-level and bit-level decoding is less than that of the KV method with $m_{max} = 11$.

Table 3.3: Average number of the codewords that are considered for suboptimum decoding of RS(15,11) code with BPSK modulation on an AWGN channel

Eb/N0 (dB)	3	4	5	6	7
Symbol-level decoding	4314.1	1279	180.55	20.91	3.38
Bit-level decoding	1517.2	576.12	120.5	18.5	3.2

2- RS(31,25) code: In Figure 3.9, we compare the performance of different decoding algorithms. In suboptimum decoding of this code using the proposed algorithm, we apply the second and the third limitations discussed before on our algorithm. The number of codewords (lattice points) considered by the algorithm before being stopped is limited to 20000. Symbol-level decoding provides slightly better performance (about 0.12 dB coding gain) compared to the KV algorithm. Also at codeword error rates of 10^{-3} and lower, it provides a coding gain of about 1 dB compared to the conventional hard decision decoding. Bit-level decoding has better performance compared to symbol-level decoding (about 0.6 dB at codeword error rate of 10^{-3}).

From Table 3.4, the number of codewords that are considered by symbol-level decoding is very high at low signal to noise ratios. Again, using the bit-level decoding, we need to check much less codewords meaning less complexity. Except for low SNR's, the total complexity exponent (Figure 3.10) for both symbol-level and bit-level decoding is less than that of the KV method with $m_{max} = 7$.

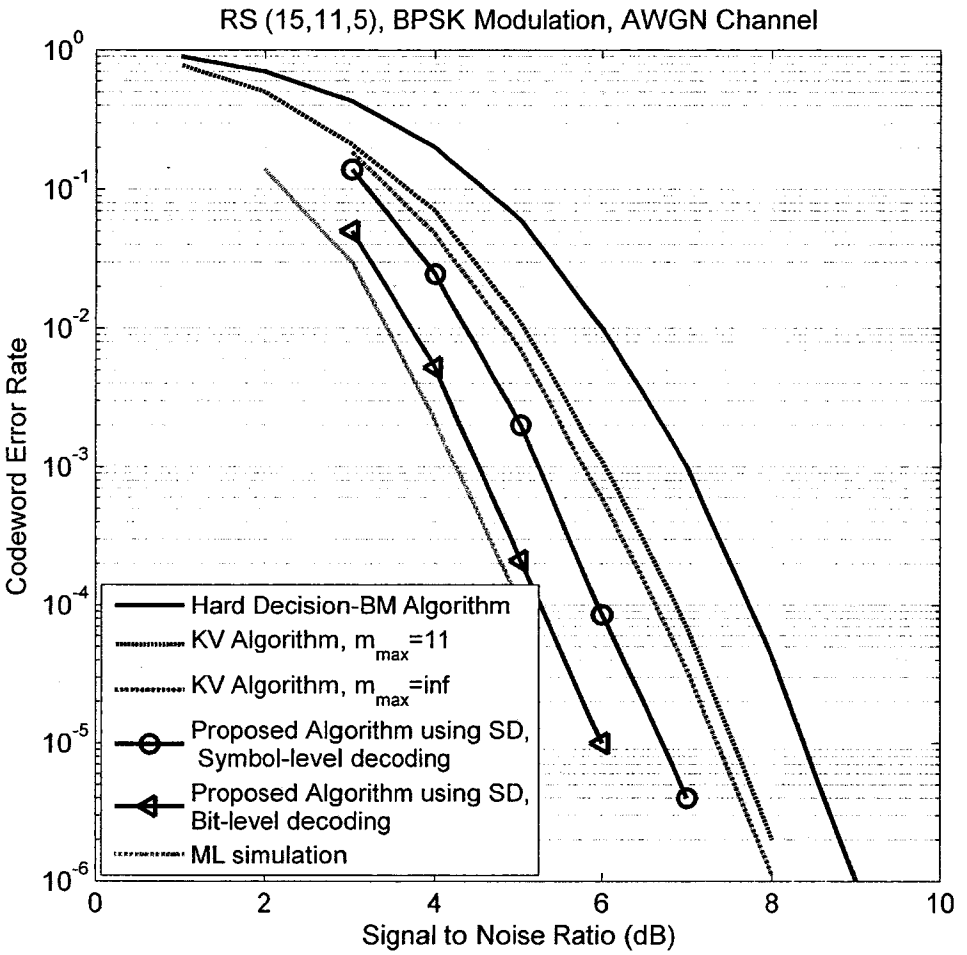

Figure 3.7: Performance of the proposed algorithm for suboptimum decoding of RS (15,11) code with BPSK modulation on an AWGN channel.

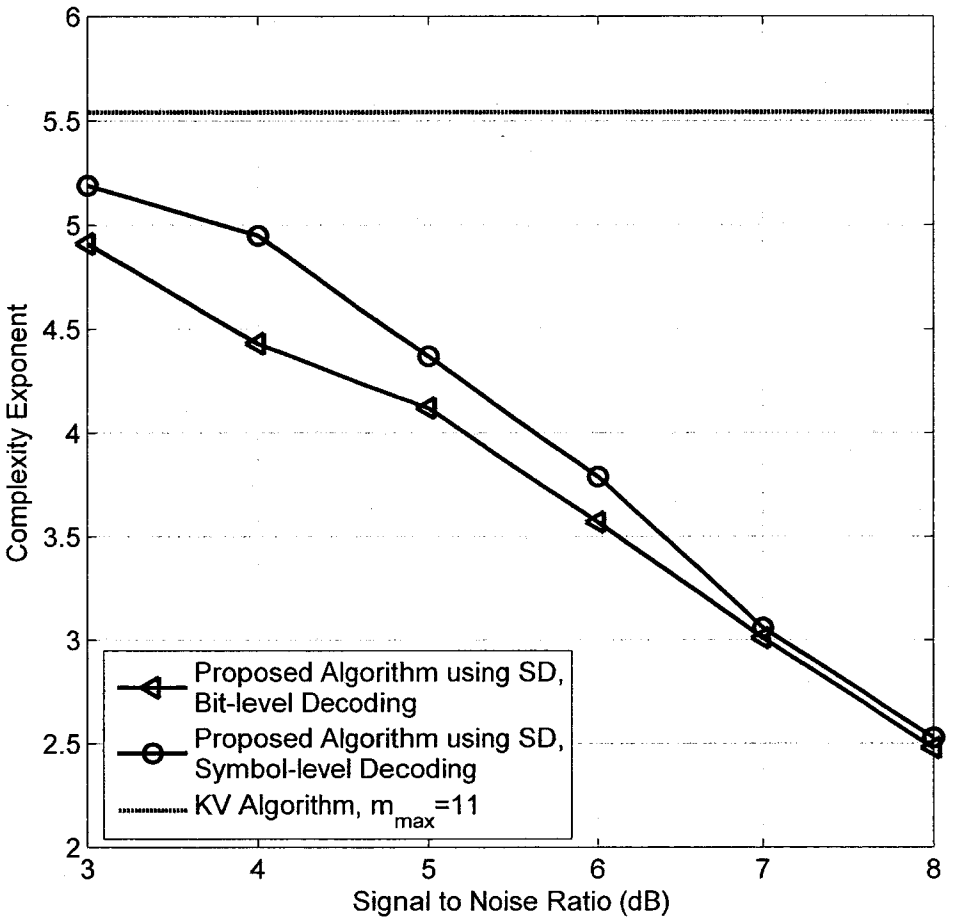

Figure 3.8: Complexity exponent of the proposed algorithm for suboptimum decoding of RS (15,11) with BPSK modulation over an AWGN channel.

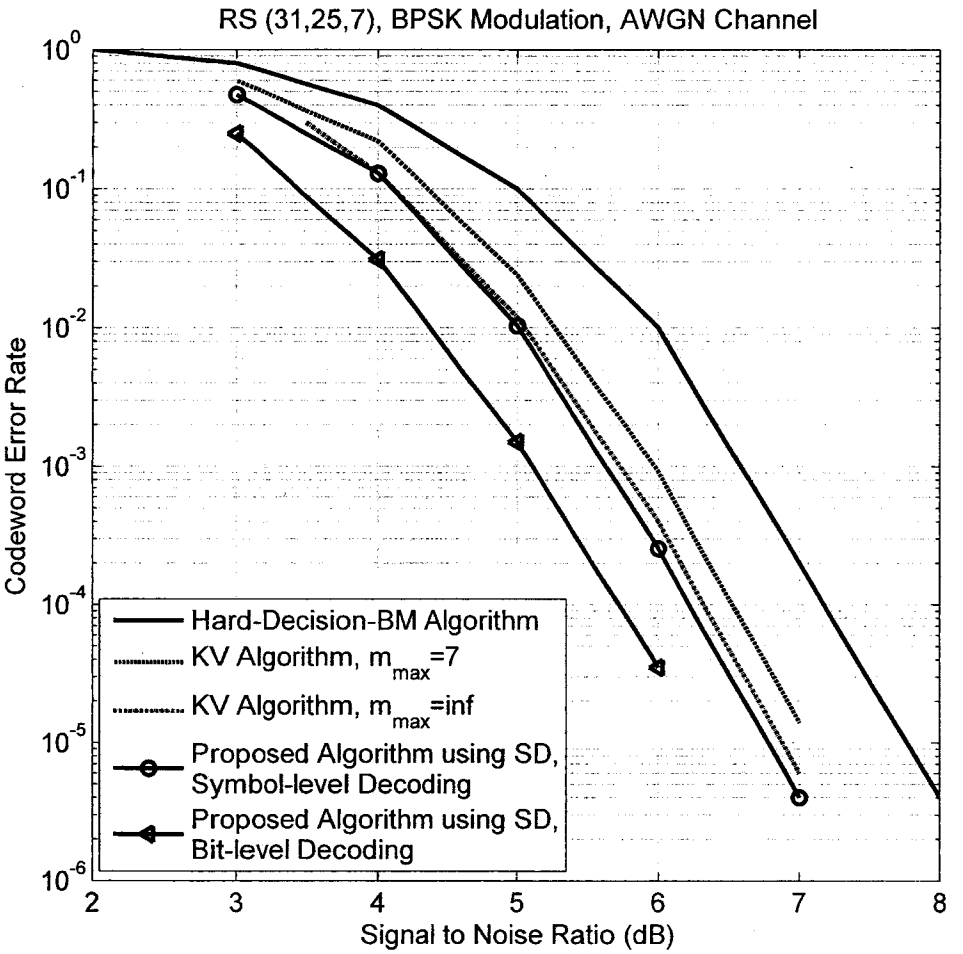

Figure 3.9: Performance of the proposed algorithm for suboptimum decoding of RS (31,25) with BPSK modulation on an AWGN channel.

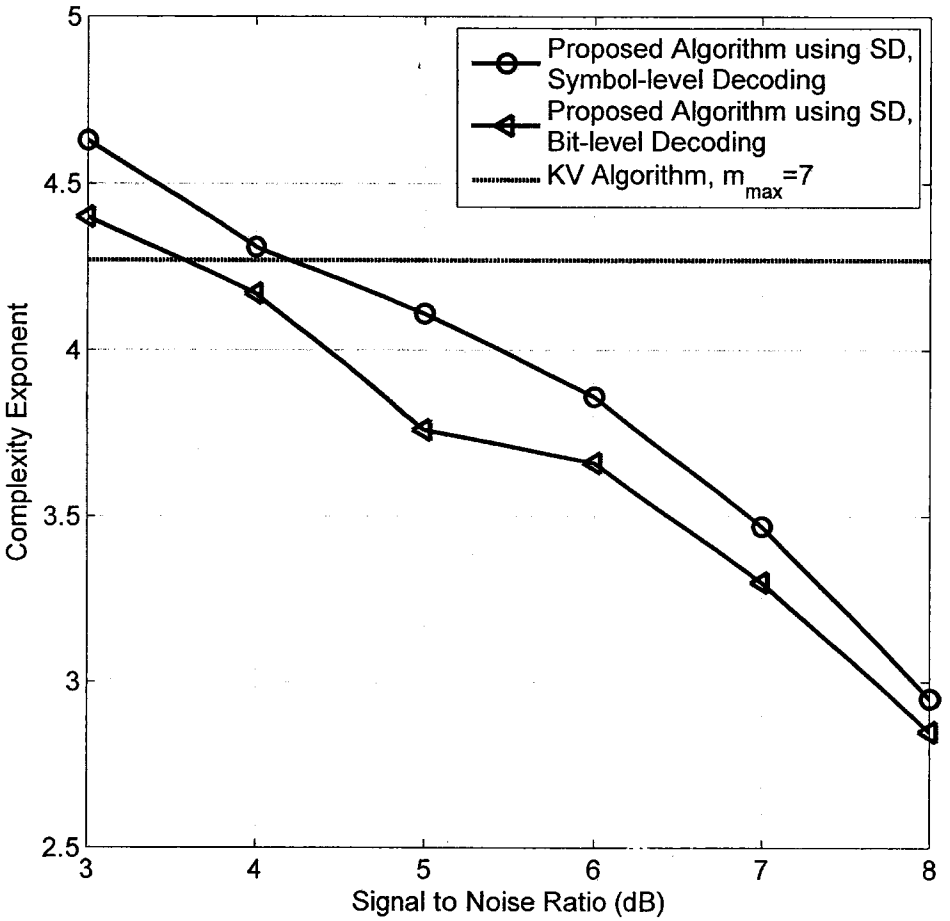

Figure 3.10: Cmplexity exponent of the proposed algorithm for suboptimum decoding of RS (31,25) with BPSK modulation on an AWGN channel.

Table 3.4: Average number of the codewords that is considered for suboptimum decoding of RS(31,25) with BPSK modulation on an AWGN channel

Eb/N0 (dB)	3	4	5	6	7
Symbol-level decoding	11720	4210	546.06	71.55	12.55
Bit-level decoding	3220.1	986	276.13	52.3	11.32

We can see that our algorithm using sphere decoding can provide equal or in some cases better performance compared to the KV algorithm with less complexity. In all of the considered cases, by applying the proper limitations discussed in section 3.2, we could provide acceptable performance with moderate complexity.

3.7 Conclusion

We have used the basis of sphere decoding to form an efficient soft decision decoding algorithm for RS codes. The concept of most reliable basis (MRB) and two types of ordering have been used in our algorithm in order to increase the speed of sphere decoding. Our original algorithm can be used for ML decoding. However, its complexity is too much for long RS codes. So, we have introduced three limitations to be applied on our algorithm in order to reduce its complexity. In fact, we have made a trade-off between the performance and the complexity of our proposed method. Depending the code, the modulation and the channel, we have applied one, two or all of the limitations on our algorithm in order to have good performance and at the same time moderate complexity. For the considered cases, it has been shown that our method can provide considerable coding gain compared to the hard decision decoding with a moderate increase in complexity. Its performance is also comparable or in some cases even better than the KV algorithm. In the case of BPSK modulation, we have seen from simulation results that bit-level sphere decoding using the binary image of the code can provide better performance and less complexity compared to symbol-level decoding.

Chapter 4

Efficient Iterative Techniques for Soft Decision Decoding of RS Codes

Belief Propagation (BP) iterative decoding has first been proposed for decoding low-density parity check (LDPC) codes [9]. It has been shown that iterative decoders for very long codes with sparse factor graphs can achieve performances close to Shannon capacity [63]. Therefore, it would be ideal if RS codes are suitable for this class of decoders. The basis of BP decoding has been explained in Chapter 2, Section 2.2. For belief propagation decoding of RS codes, they should be considered over an extension field of $GF(2)$. The binary image of RS codes has also been described in Chapter 2, Section 2.2.

As discussed in Chapter 2, standard BP iterative decoding is not suitable for high density parity check codes like RS codes. For these codes, there are a large number of short cycles in the factor graph [38] which cause correlation between the messages and error propagation. In [37], the cyclic structure of RS codes is used and BP decoding is applied to a random shift of the received vector at each iteration to avoid error propagation. For short RS codes, the coding gain is significant but it diminishes for long codes. Another bit-level decoding method based on belief propagation has been proposed in [35], but it is only efficient for very low rate RS code. In [36], an algorithm for removing all the 4-cycles in the factor graph of linear block codes has been introduced. This method

improves the suitability of iterative decoders for short low rate RS codes. The adaptive parity-check (ADP) algorithm [12] has been the first successful bit-level iterative decoding method for RS codes. In ADP, in order to make BP decoding effective for a dense RS parity check matrix, Gaussian elimination is performed on the binary parity check matrix before each iteration such that the variables of lowest reliability connect into the graph only once. This algorithm has very good performance compared to other soft decision decoding algorithms for RS codes. The problem with the ADP algorithm is that the required Gaussian elimination of the parity check matrix at each iteration is very complex. In [44], multiple-bases belief propagation for linear block codes with dense parity check matrices has been proposed. It makes use of the fact that a code has many structurally diverse parity check matrices, capable of detecting different error patterns. Other BP based decoding methods for general linear block codes have been proposed in [39–43].

In this chapter, for efficient iterative decoding of RS codes, we use a fixed binary parity check matrix. This matrix is the extended version of the original parity check matrix with lower density and less number of 4-cycles [64]. This parity check matrix representation is better suited for iterative decoders. Parity check matrix extension is only applied in the receiver for the decoding without affecting the RS code itself (i. e. the transmitter). Even using the extended parity check matrix, the performance of standard BP decoding is not very good. In order to improve the performance of standard BP decoding, we propose two new bit level iterative soft decision decoding methods for RS codes using the fixed extended binary parity check matrix.

The first algorithm uses the cyclic structure of RS codes. Based on this property, we can apply the BP algorithm on any cyclically shifted version of the received symbols with the same binary parity check matrix. For an (N, K) RS codeword, each of N shifted versions of the received symbols leads to a different distribution of reliability values and deterministic errors can be avoided. Simulation results demonstrate that the performance of this method is superior or comparable to some popular methods including ADP method.

The second algorithm uses the information correction in BP decoding. Based on the

updated reliabilities at the end of each iteration, we can determine the bits with lowest reliabilities such that changing their channel information improves the convergence of the decoder to a codeword [65]. In this method, a few steps of information correction are performed which help improving the performance of normal BP decoding. The performance of this method is not as good as the first algorithm, but it is less complex because it needs less number of BP iterations.

The rest of this chapter is organized as follows. In section 4.1, the system model is introduced. In section 4.2, we first briefly review the BP decoding. Then we explain low density extended binary parity check matrices for RS codes. After that we investigate the performance of BP decoding with different binary parity check matrices over binary erasure channels (BEC). In section 4.3, using the extended parity check matrix, two algorithms for soft decision decoding of RS codes based on belief propagation are presented. First, iterative decoding based on the cyclic structure of RS codes is introduced. We also give a geometric interpretation of the proposed algorithm. Then, iterative decoding using information correction is introduced. In section 4.4, we present the complexity analysis of the proposed algorithms. Simulation results and discussions are in section 4.5 where the performance of different RS codes over the AWGN channel is considered. We explore the amount of soft-decision gain that our algorithms provide for different RS codes. Finally, conclusions are presented in section 4.6.

4.1 System Model

We use an (N, K) RS code over Galois field $GF(2^p)$ where $N = 2^p - 1$. Denoting the parity check matrix of the code by H, for any codeword c of the RS code, we have $Hc^T = 0$. Since any element $\beta \in GF(2^p)$ has a p tuple representation, we can show any codeword c of length N in binary form as

$$c_b = (c_{1,1}, c_{1,2}, ..., c_{1,p}, c_{2,1}, ..., c_{2,p}, ..., c_{N,1}, ..., c_{N,p}). \tag{4.1}$$

We assume BPSK modulation $x = -2c_b + 1$ over an AWGN channel. So, the received signal is

$$y = x + n \tag{4.2}$$

where n is the AWGN vector with variance σ^2. The reliability of the received vector can be expressed in terms of the log likelihood ratios (LLR's) that are given by $\rho^{ch} = 2y/\sigma^2$.

For the decoding, we consider the binary image of RS codes. As discussed in Chapter 2, Section 2.2, for a primitive element α of $GF(2^p)$, there is a $p \times p$ binary companion matrix c_p (Equation (2.24)) [56]. A field isomorphism can be defined by the mapping $\alpha^i \rightarrow c_p^i, i = \{0, 1, ...\}$. Based on this mapping, each element of the parity check matrix of the code is replaced with a $p \times p$ binary matrix resulting a binary parity check matrix H_b of size $(N - K)p \times Np$. Such a mapping results in $H_b c_b^T = 0$. This equation specifies the set of linear constraints satisfied by the codeword bits. These constraints can be represented using a bipartite graph [38] where the set of variable nodes represents the codeword bits and the set of check nodes represents the set of parity-check constraints satisfied by the codeword bits. There is also a set of edges that connect every check node with all the variables nodes involved in its check equation.

Using the binary image of RS codes, standard BP decoding (Figure 2.5) can be used for decoding the received signal to a binary RS codeword.

4.2 Iterative Decoding of RS Codes using Belief Propagation

4.2.1 Standard Belief Propagation decoding of RS Codes

In the bipartite graph of the binary image of an (N, K) RS code, there are $(N - K)p$ check nodes and Np variable nodes. Given the vector ρ^{ch} of initial channel LLR's and the bipartite graph defined by H_b, the BP algorithm updates the reliability information of the bits for an specific number of iterations (Figure 2.5). The stopping criterion could be when all the checks are satisfied or when we reach the maximum number of iterations. Standard

BP iterative decoding is not suitable for RS codes. The binary parity check matrix of these codes is very dense and there are a large number of short cycles in the factor graph which cause correlation between the messages and consequently error propagation.

4.2.2 Low Density Parity Check Matrices for RS Codes

The total number of short cycles in the factor graph of RS codes increases exponentially with the parity check matrix density. The density of the binary image of the parity check matrix given in Equation (2.22) is around 50% for different RS codes which leads to a large number of short cycles. So, one step is to find a low density binary parity check matrix for an (N, K) RS code. This is equivalent to finding the $(N-K)p$ codewords of low Hamming weight that span the binary image of the dual code of the RS code which is also an RS code. We may construct these low weight codewords by considering the subcodes of the binary image of the dual code consisting of codewords that are nonzero only on a subset of the Np coordinate positions [64]. In order to construct these subcodes, for a given parameter $s < p$, we define a support set R as

$$R = \{i_1, i_2, ..., i_s\} \subset \{1, 2, ..., p\}\,. \tag{4.3}$$

We look for any codeword $b = \{b_{1,1}, ..., b_{1,p}, b_{2,1}, ..., b_{2,p}, b_{N,1}, ..., b_{N,p}\}$ of the dual code which is zero outsize the defined support set R:

$$b_{i,j} = 0, (i, j) \in \{1, 2, ..., N\} \times \bar{R} \tag{4.4}$$

where $\bar{R} = \{1, 2, ..., p\} \setminus R$. Codewords satisfying the above constrains (if they exist) may be obtained from the appropriate linear combinations of the rows of H_b [64]. We repeat this procedure for different support sets and among the codewords we obtain, we choose the ones with lowest weight that span the binary image of the dual code [64].

If the low weight codewords don't span the binary image of the dual code, we arrange the rows of the systematic H_b in an increasing order of their Hamming weight. We start from the first row and select the ones that help to complete the span of the dual code. The

density of the new parity check matrix for different RS codes is around 30% which leads to less short cycles compared to the original H_b.

4.2.3 Extended Parity Check Matrix for RS Codes

In order to reduce the number of 4-cycles in the factor graph of the RS code, we can extend the low density parity check matrix by rows and columns [64]. At first, we select the two rows (checks) with maximum variable nodes in common denoted by row i and row j. We form a new row with ones in variable overlap positions of the two checks. In order for this new row to be a check equation, we have to add a new variable (column) with single one at the new row and zero elsewhere to the parity matrix to ensure even parity of the new check. We add this new check to row i and row j and replace them with the results [64]. By doing so, all the 4-cycles among variables associated with row i, row j and this new row will be eliminated. This process has been shown in Figure 4.1. Then we select next two checks with maximum variable overlap and repeat the previous procedure to add a new row and a new column. After completing this process, the extended parity check matrix will have $re = (N-K)p+(N-K)p/2$ rows and $ne = Np+(N-K)p/2$ columns. The density of the new extended parity check matrix is around 15% for different RS codes which means considerably less short cycles. Since we never actually transmit these new variables, we consider them as erasures (zero LLR's) during the decoding. Iterative BP decoding using this extended binary parity check matrix is called phantom decoding [64].

4.2.4 Performance Analysis over Binary Erasure Channels

Because of its simplicity, we investigate the performance of belief propagation decoding of RS codes over a binary erasure channel (BEC) with erasure probability equal to ϵ. Due to the binary form of the messages, BP decoding on the erasure channel can be done much easier as follows [66]:

1. Set the values of all check nodes to zero.

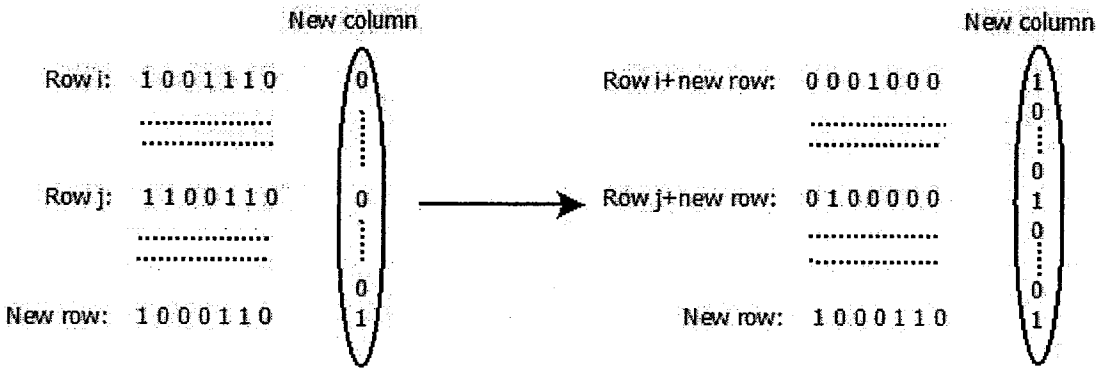

Figure 4.1: Extending the binary parity check matrix by adding rows and columns.

2. If a variable node is received correctly, add its value to the values of all adjacent check nodes (binary addition) and remove the variable node and all its edges from the graph. Repeat this for all variable nodes.

3. If there is a check node with degree one, substitute its value into the value of its unique neighbor (variable node). Repeat step 2 for this variable node and repeat step 3. This step is repeated until all the erased bits are recovered. If at some point there is no check node with degree one, the algorithm can not proceed further causing decoding failure.

Similar to low density parity check (LDPC) codes, for a given degree distribution pair (λ, ρ) from Equation (2.21) and $\epsilon \in [0,1]$, we define $f(\epsilon, x) = \epsilon\lambda(1-\rho(1-x))$. Then the threshold is defined as [67]

$$\epsilon^*(\lambda,\rho) = sup\{\epsilon \in [0,1] : x = f(\epsilon,x) \text{has no solution } x \text{ in } (0,1]\}\,. \tag{4.5}$$

A critical point is a point at which $f(\epsilon^*, x) - x$ tangentially touches the horizontal axis. It has been proved [67] that for unconditionally stable graphs (for which $\lambda'(0)\rho'(1) < 1$) with only one critical point (x^*), the block error probability curves of a code of fixed length N can be approximated in the waterfall region (around the asymptotic threshold) by

$$P_B = Q(\frac{\sqrt{N}(\epsilon^* - \beta N^{-2/3} - \epsilon)}{\alpha})(1 + O(N^{-1/3})) \tag{4.6}$$

where Q is the Q-function. Knowing the degree distributions $\lambda(x)$ and $\rho(x)$, we can compute the covariance term α and the shift parameter β [67].

In defining the threshold, it is supposed that the messages coming from the check nodes are independent as well as the messages coming from the variable nodes. However, when there are cycles in the factor graph, this assumption is not always true. So, the results in this section are not exact but they give us a good estimate to investigate the performance of BP decoding of RS codes over the BEC.

It should be mentioned that the performance of BP decoding of a specific code depends largely on the parity check matrix that is used for decoding. Here, we consider three different parity check matrices for RS(31,25):

1. The original binary parity check matrix with density around 50% and $\epsilon^* = 0.065$.
2. The low density parity check matrix with density around 30% and $\epsilon^* = 0.0911$.
3. The extended parity check matrix with density around 15% and $\epsilon^* = 0.155$.

The performance of BP decoding of RS(31,25) over the BEC using these three different binary parity check matrices has been shown in Figure 4.2. For the case of extended parity check matrix, newly added variable nodes are considered as erasure and therefore the erasure probability used in the formula is more than the real erasure probability. Even doing so, the performance of BP decoding using the extended parity check matrix is still much better than the others. As it can be seen from this figure, by reducing the density of the parity check matrix and therefore the number of cycles, the erasure probability threshold increases and the performance of BP decoding improves largely. Therefore, in the next section that we propose new methods for iterative decoding of RS codes, we use the extended binary parity check matrix for BP decoding.

4.3 Efficient Iterative Decoding of RS Codes

Using the extended parity check matrix discussed previously, we can perform phantom decoding [64] of RS codes using standard BP iterations. However, the performance will be

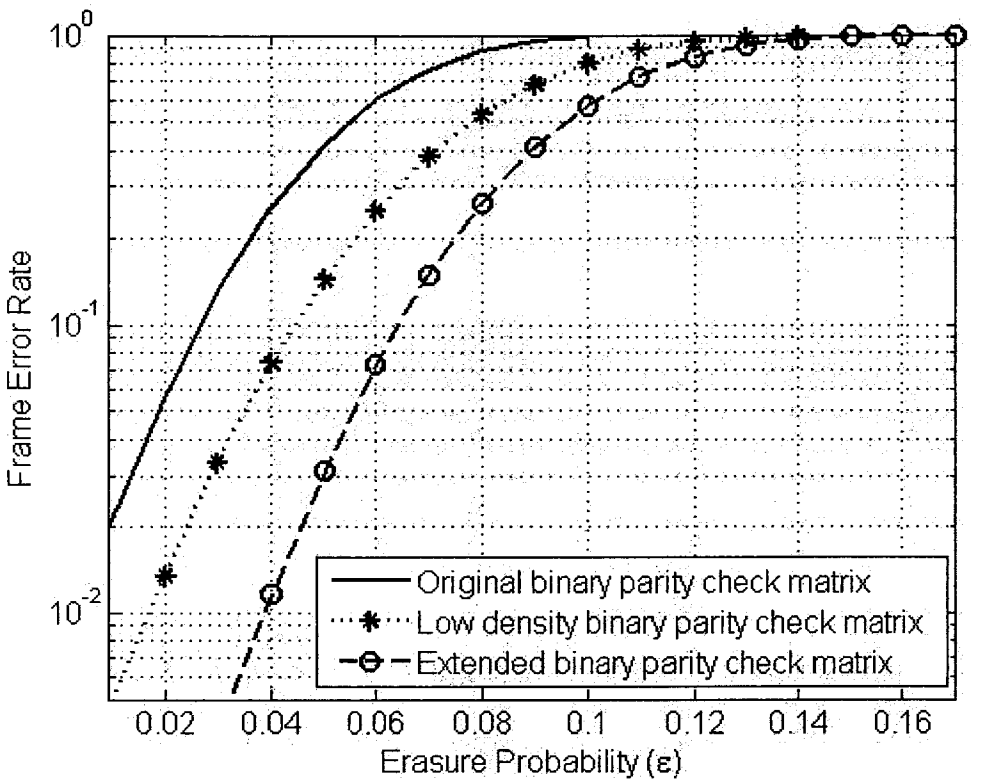

Figure 4.2: Performance of belief propagation decoding of RS(31,25) over the BEC using three different binary parity check matrices.

far away from the performance of the ML decoder because there are still a large number of cycles in the factor graph of the code. Therefore, in this section we present two methods to improve the performance of phantom decoding.

4.3.1 Method A: Iterative Decoding of RS Codes Based on Their Cyclic Structure

Since RS codes are cyclic, any cyclically shifted version of a codeword c is also a valid codeword. Therefore, we can consider a cyclically shifted version of the received signal y by φ symbols ($\varphi \in [0, N-1]$) as the received signal when a shifted version of c by φ symbols which is also a valid codeword has been transmitted. Based on this fact, we can apply the BP algorithm on any cyclically shifted version of the received reliabilities with the same binary parity check matrix [37]. In the end, the updated reliabilities are shifted back to their original positions.

Since for each cyclically shifted version of the reliabilities, their values are differently distributed, some deterministic errors can be avoided. So, the idea is to have outer rounds during the decoding. During each outer round, a different cyclically shifted version of the received reliabilities is generated and then used as the input to the BP decoding algorithm. Denoting the low density extended parity check matrix by H, the proposed algorithm is summarized in Figure 4.3. The maximum number of outer rounds, c_{it-max}, can be chosen to be equal $N-1$ such that all the possible cyclically shifted versions of the received reliabilities are considered as the input of the BP decoding algorithm. Most of the time, our algorithm finds the response very soon (during the first or the second outer round) and is terminated. So, there is no need to complete all of the outer rounds.

In the proposed algorithm, we use both BP decoding and hard decision BM decoding [6]. At the end of each BP iteration, the hard decision version of updated reliabilities is used as the input to the BM decoding algorithm. Even in situations that BM algorithm can recover a codeword, we do not stop the algorithm just save that codeword. In the end, if the BP decoding was not successful, we choose one of the saved codewords with

Data: The low density extended binary parity check matrix H, channel reliabilities $\rho^{ch} = 2y/\sigma^2$, the damping factor θ, maximum outer rounds c_{it-max} and maximum inner BP iterations BP_{it-max}.

Result: A list of candidate transmitted codewords.

begin

1 Initialization step: $c_{it} = 0$, $BP_{it} = 0$.
Outer rounds:

2 **while** $c_{it} \leq c_{it-max}$ **do**

3 $\varphi = c_{it}$.

4 Cyclically shifting the channel LLR's by φ symbols: $\rho^{ch-shift} = \rho^{ch}_{\varphi}$.

5 $\rho^{BP_{it}} = [00...0]_{1\times(Np+(N-K)p/2)}$ and $\rho^{BP_{it}}(1:Np) = \rho^{ch-shift}$.
Inner BP iterations:

6 **while** $BP_{it} \leq BP_{it-max}$ **do**

7 Belief Propagation: feed $\rho^{BP_{it}}$ and H into the BP algorithm (Figure 2.5) and generate extrinsic LLR's for each bit: ρ^{x}.

8 Update the LLR of each bit:
$\rho^{BP_{it}+1}(c_i) = \rho^{BP_{it}}(c_i) + \theta\rho^{x}(c_i)$.

9 Hard Decisions, for $i = 1, 2, ..., Np$:
$\tilde{c_i} = \begin{cases} 0 & \rho^{BP_{it}+1}(c_i) > 0 \\ 1 & \rho^{BP_{it}+1}(c_i) < 0 \end{cases}$.

10 **if** $\tilde{c}$ *satisfies all the check equations* **then**
shift the decoded bits back to their original position: $\tilde{c} = \tilde{c}(-\varphi)$, save $\tilde{c}$, terminate the algorithm and go to line 13.
else
BM hard decision decoding [6]: $c_{\tilde{BM}} = BM(\tilde{c})$.
if *a decoding success was signaled* **then**
$c_{\tilde{BM}} = c_{\tilde{BM}}(-\varphi)$ and save $c_{\tilde{BM}}$

11 $BP_{it} = BP_{it} + 1$.

12 $c_{it} = c_{it} + 1$, $BP_{it} = 0$

13 Among all the codewords that have been saved throughout the algorithm, choose the one with minimum Euclidean distance from the received vector.

end

Figure 4.3: Method A: Iterative Decoding of RS Codes based on Their Cyclic Structure

minimum Euclidean distance from the received vector as the response of the proposed iterative decoding method. Therefore, the block error rate of the method can be written as

$$P_B = P_{B-BM} \times P_{B-BP} \tag{4.7}$$

where P_{B-BM} is the block error rate of BM decoding method and P_{B-BP} is the block error rate of BP decoding with cyclic shifting of reliabilities.

In order to reduce the average number of required BP iterations, a stopping criterion is introduced. We define a radius $r = \sqrt{Np\sigma^2}$. Each time there is a decoding success at the output of the BM decoding method (Figure 4.3), we check the distance of that codeword from the received vector, $d = \|\tilde{x}_{BM} - y\|$, where $\tilde{x}_{BM} = -2\tilde{c}_{BM} + 1$. If $d <= r$, we terminate the algorithm and go to line 13 of Method A (Figure 4.3). We refer to this process as the extra stopping criterion ($d \leq r$). Of course, by doing so, the performance of our proposed algorithm will be slightly worse than before. Therefore, we have a trade-off between the performance and the complexity.

As we will see in simulation results, the performance of iterative decoding using the cyclic property of RS codes is very good and even for short RS codes close to the ML performance. However, the disadvantage is that for low signal to noise ratios we need to perform a large number of BP iterations leading to high complexity. In the next section, we present another method for efficient decoding of RS codes. Although the performance of this new method is not be as good as method A, but it requires much less BP iterations.

4.3.1.1 Analysis of Cyclic Shifting: Geometric Interpretation of Method A

We define the potential function J as [12] [68]

$$J(H,T) = -\sum_{j=1}^{n_e-k_e} \gamma_j = -\sum_{j=1}^{n_e-k_e} \prod_{p=1,H(j,p)=1}^{n_e} T_p \tag{4.8}$$

where J is a function of both the $(n_e - k_e) \times n_e$ parity check matrix H and the received soft information T such that

$$T = [T_1, T_2, ..., T_{n_e}] = [v(\rho(c_1)), ..., v(\rho(c_{n_e}))]\,. \tag{4.9}$$

Here, the operator $v : [-\infty, \infty] \rightarrow [-1, 1]$ is a mapping from the LLR domain to the tanh domain:

$$v(\rho) = \tanh(\frac{\rho}{2}) = \frac{1 + e^{\rho/2}}{1 - e^{\rho/2}}. \tag{4.10}$$

When all the checks are satisfied, we have reached a valid codeword and the potential function J is minimized. In this case, $|T_j| = 1$ for $j = 1, ..., n_e$ and $J_{min} = -(n_e - k_e)$. In the case of RS codes, even using the extended parity check matrix, the density is still high and applying the iterative decoding might lead to some local minimum points called pseudo-equilibrium points. These point are not corresponded to valid codewords and the iterative algorithm gets stuck at them. Actually, there are a few unreliable bits which do not let J to be minimum.

Because J is a function of both H and T (bit reliabilities), different arrangements of the reliabilities with the same H result in different values for the potential function J. The proposed algorithm uses this fact and when a pseudo-equilibrium point is reached, we change the arrangement of bit reliabilities using cyclic shifting by φ symbols ($\varphi = 0, ..., N - 1$). This way, the update might proceed rather than getting stuck at the pseudo-equilibrium point. In Figure 4.4, we have shown the potential function J versus the number of iterations for three outer rounds of method A while decoding RS(15,11) with BPSK modulation over the AWGN channel with $\frac{Eb}{N0} = 4$ dB. As it can be seen in this figure, during the first outer round, BP decoding gets stuck in a pseudo-equilibrium point and can not converge to a codeword. However, by cyclically shifting the reliabilities by one symbol during the second round, BP decoding converges to the correct codeword. Here, for RS(15,11), the extended parity check matrix is an $r_e \times n_e$ matrix with $re = (N-K)p + (N - K)p/2$ and therefore $J_{min} = -r_e = -24$ that has been reached during the second outer round.

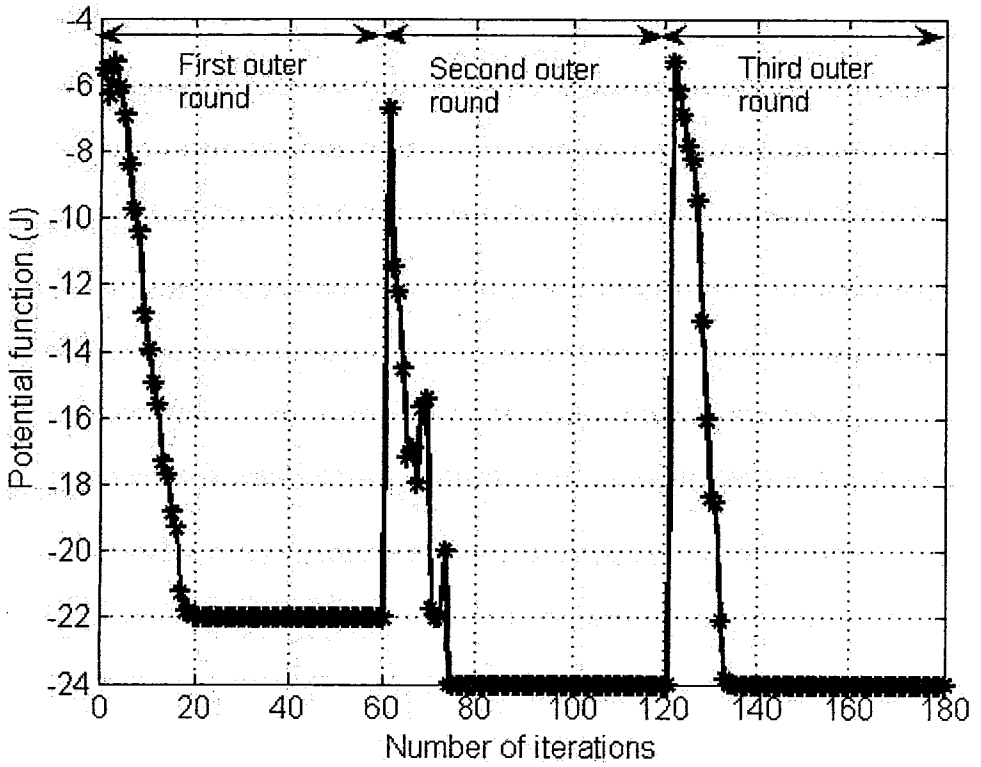

Figure 4.4: Potential function versus the number of iterations for three outer rounds of method A while decoding RS(15,11) with BPSK modulation over the AWGN channel with $\frac{Eb}{N0} = 4$ dB.

4.3.2 Method B: Iterative Decoding of RS Codes Based on Information Correction

In this section, we introduce another iterative method to improve the standard BP decoding of RS codes. As we mentioned before, even using the extended binary parity check matrix, there will be a big gap between the performance of BP decoding of RS codes and that of ML decoding. This is because of the large number of 4-cycles in the factor graph of the code.

The new method is based on information correction in the BP decoding algorithm [65]. In this method, first normal BP decoding is performed for BP_{it-max} iterations. During

these iterations, the average LLR's are recorded as

$$\rho_{sum}^{BP_{it-max}}(c_j) = \frac{1}{BP_{it-max}} \sum_{i=1}^{BP_{it-max}} \rho^i(c_j), \qquad j = 1, ..., n_e \tag{4.11}$$

where $\rho^i(c_j)$ is the reliability of the jth bit after i iterations and $n_e = Np + \frac{(N-K)p}{2}$. If a codeword is reached during these iterations, we terminate the algorithm. However, if decoding was not successful at the end of BP_{it-max} iterations, IC_{max} steps of information correction are performed:

- Step 1. Based on the average LLR's, we select the least reliable bit:

$$c_p = \underset{c_j, j=1,...,n_e}{\operatorname{argmin}} \left| \rho_{sum}^{BP_{it-max}}(c_j) \right|. \tag{4.12}$$

  Changing the channel information of the selected bit very likely improves the convergence of the decoder to a codeword. Therefore, we perform two tests by setting $\rho^{ch}(c_p) = \pm inf$ and for each case we continue BP iterations with the new channel reliabilities for additional BP_{add} iterations. During these iterations, we also record the average LLR's:

$$\rho_{sum}^{BP_{add}}(c_j) = \frac{1}{BP_{add}} \sum_{i=BP_{it-max}+1}^{BP_{it-max}+BP_{add}} \rho^i(c_j), \qquad j = 1, ..., n_e. \tag{4.13}$$

  At the end of each test, we select one least reliable bit based on the average LLR's. We assume these bits are at positions h and k for the case $\rho^{ch}(c_p) = +inf$ and $\rho^{ch}(c_p) = -inf$ respectively.

- Step 2. At the second step, we perform four tests by setting $\rho^{ch}(c_p) = +inf, \rho^{ch}(c_h) = \pm inf$ and $\rho^{ch}(c_p) = -inf, \rho^{ch}(c_k) = \pm inf$. For each of the new channel reliabilities, we continue BP iterations from BP_{it-max} for additional BP_{add} iterations and record the average LLR's based on Equation (4.13). Finally, four least reliable bits are selected.

  In general, step j of the algorithm for $2 < j \leq IC_{max}$ will be as follow:

- Step j. For each of the 2^{j-1} bits selected at step $j-1$, two tests are performed. For each of the 2^j tests, BP iterations are continued with the new channel reliabilities for additional BP_{add} iterations and 2^j bits are selected.

If a valid codeword is reached at any step, the decoding is terminated. Here, similar to method A, at the end of each BP iteration, we perform BM decoding using the updated reliabilities. If the BM decoding is successful, we do not stop the algorithm just save that codeword. In the end, we choose one of the saved codewords with minimum Euclidean distance from the received vector as the decoding response.

Using the information correction strategy, we can enforce the correct values on the selected bits and eliminate pseudo-codewords. Therefore, the chance of BP iterations to converge to a codeword is improved significantly. It should be noted that by increasing the signal to noise ratio, the information correction technique is more effective because there are less bit errors at the channel output.

As we mentioned before, there are considerable short cycles even in the factor graph of the extended parity check matrix. Therefore, there will be correlation between the messages and the values of high reliable bits may be affected significantly by the values of low reliable bits. To avoid this, we can select the highest reliable bits using a threshold value τ:

$$C_h = \left\{ c_j, \left| \rho^{ch}(c_j) \right| > \tau \right\}. \tag{4.14}$$

We then fix the values of these selected bits using hard decision and do not update them during the BP iterations:

$$\rho^{ch}(c_j) = sign(\rho^{ch}(c_j)) \times inf, \qquad \forall c_j \in C_h. \tag{4.15}$$

If the threshold is selected appropriately, the selected bits will have correct channel information with high probability. This way, high reliable correct bits are safe and their values are not affected in any ways.

In method B, we also add the extra stopping criterion used in method A in order to reduce the number of required BP iterations.

4.3.2.1 Analysis of Information Correction: Geometric Interpretation of Method B

Here, we use the potential function J defined in (4.8). It is a function of both the parity check matrix H and the received soft information. In method B, at each step of information correction, the reliabilities of some bits are changed. Different reliabilities result in different values for the potential function J. Therefore, when a pseudo-equilibrium point is reached, changing the bit reliabilities allows the update to proceed. In Figure 4.5, we have shown the potential function J for three steps of information correction while decoding RS(31,25) with BPSK modulation over the AWGN channel with $\frac{Eb}{N0} = 4$ dB. As it can be seen in this figure, during initial iterations and also the first step of information correction, BP decoding gets stuck in a pseudo-equilibrium point and can not converge to a codeword. However, in the second step of information correction, BP decoding converges to the correct codeword. Here, for RS(31,25), the extended parity check matrix is an $r_e \times n_e$ matrix with $re = (N-K)p + (N-K)p/2$ and therefore $J_{min} = -r_e = -45$ that has been reached during the second step of information correction.

4.4 Complexity Analysis of the Proposed Algorithms

For both algorithms, the extended parity check matrix is used. The complexity of belief propagation in each iteration is proportional to the number of nonzero elements in the parity check matrix of the code. Because the density of the extended binary parity check matrix is around 15%, the time complexity of each BP iteration is

$$O(0.15 \times re \times ne) \tag{4.16}$$

where $re = (N-K)p + (N-K)p/2$ and $ne = Np + (N-K)p/2$.

In the ADP method, the original parity check matrix with the density around 50% is used. Each iteration involves $O(Np \times log_2(Np))$ floating point operations for sorting, $O(Np \times min(K^2p^2, (N-K)^2p^2))$ binary operations for Gaussian elimination and $O(0.5 \times (N-K)p \times Np)$ floating point operations for BP. Therefore, the complexity of each iteration of our methods is much less than the ADP method.

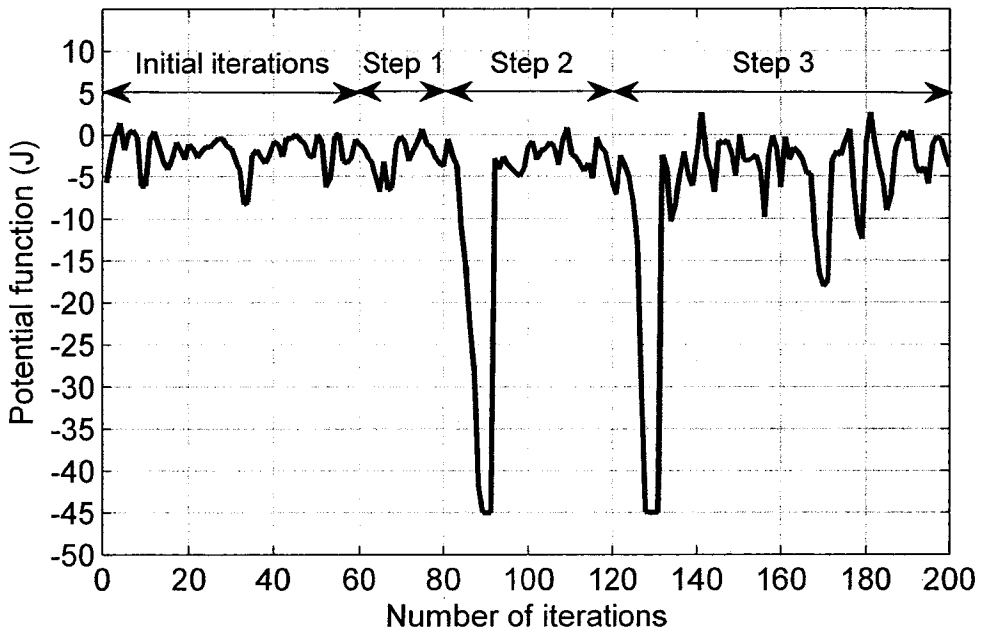

Figure 4.5: Potential function versus the number of iterations for three steps of information correction of method B while decoding RS(31,25) with BPSK modulation over the AWGN channel with $\frac{Eb}{N0} = 4$ dB.

The maximum number of BP iterations in method A based on cyclic shifting of RS codes is

$$c_{it-max} \times BP_{it-max}. \tag{4.17}$$

However, most of the times, we find the response very soon (during the first or the second outer round) and we terminate the algorithm. So, the actual complexity will be much less than the maximum complexity.

The maximum number of BP iterations in method B based on information correction is

$$BP_{it-max} + BP_{add} \times \left(\sum_{j=1}^{IC_{max}} 2^j \right). \tag{4.18}$$

Again, most of the times, we find the response very soon and the actual complexity is much less than the maximum complexity.

In the next section that we present simulation results for different RS codes, we

provide tables of the average number of required BP iterations for different signal to noise ratios.

4.5 Simulation Results and Discussions

In this section, simulation results for decoding RS codes using the proposed algorithms are presented. For all the simulations, we assume BPSK transmission over AWGN channel. In method A, the maximum number of BP iterations for the inner rounds is set to 60. So, the maximum number of iterations will be $60N$. For method B based on information correction, the maximum number of initial BP iterations is set to $BP_{it-max} = 60$. We will have 4 steps of information correction and $BP_{add} = 10$. So, the maximum number of iterations using Equation (4.18) will be 360. It is clear that the maximum number of iterations in the second algorithm is much less than the first one.

We compare our algorithms with the Berlekamp-Massey (BM) hard decision decoding method [6] and also the algebraic soft decision decoding method proposed by Koetter and Vardy [29] [13] that will be mentioned as the KV algorithm. We will also compare our results with those of phantom decoding [64] and ADP method [12]. It is also important to compare our algorithms with the best performance possible, which is that of the ML decoder. The weight enumerator of an RS code under a specific binary image expansion is not known. The averaged ensemble of an RS code can be found by averaging over all possible binary expansions [69]. The averaged binary weight enumerator can then be used by Divsalar simple bound [70] to bound the ML error probability [12].

4.5.1 RS(15,11) code

In Figure 4.6, we have shown the performances of our proposed methods. From this figure, at codeword error rate of 10^{-3}, decoding using method A provides a coding gain of more than 1.5 dB compared to the asymptotic performance of the KV algorithm, 2.7 dB compared to the BM algorithm and 1.7 dB compared to phantom decoding. This method

Table 4.1: The average number of required BP iterations for RS(15,11)

Eb/N0 (dB)	2	3	4	5
Method A	422	174	52	6
Method A with extra stopping criterion ($d \leq r$)	253	97	24	5
Method B with extra stopping criterion ($d \leq r$)	99	47	13	2

has also about 0.5 dB coding gain compared to the performance of ADP-BM (5,1) [12]. The performance of method B has also been shown in this figure which is almost the same as method A with extra stopping criterion ($d \leq r$). It has 1.4 dB coding gain compared to phantom decoding. In order to consider the complexity, Table 4.1 shows the average number of required BP iterations for both methods. By comparing the complexity and the performance of method A with and without extra stopping criterion, we see that adding the extra stopping criterion ($d \leq r$) causes about 0.2 dB performance loss and at the same time reduces the average number of required BP iterations considerably. Therefore, for longer RS codes, we always add this extra stopping criterion. From Table 4.1, we can also see that the complexity of method B is much less than method A. Finally from Figure 4.6, the performance of method A is very close to the performance of ML simulation.

4.5.2 RS(31,25) code

From Figure 4.7, decoding using algorithm A with the extra stopping criterion ($d \leq r$) provides a coding gain of more than 1.5 dB compared to the asymptotic performance of the KV algorithm, about 2.5 dB compared to the BM algorithm and 1 dB compared to phantom decoding at codeword error rate of 10^{-4}. Also, method A has about 0.5 dB coding gain compared to ADP-BM (5,1) and 0.2 dB compared to ADP-BM (20,1). From this figure, at codeword error rate of 10^{-4}, method B has 0.5 dB coding gain compared to phantom decoding. The performance of method B is not as good as method A and it is close to the performance of ADP-BM (5,1). However, as we can see in Table 4.2, the complexity of method B is much less than method A especially at low signal to noise ratios.

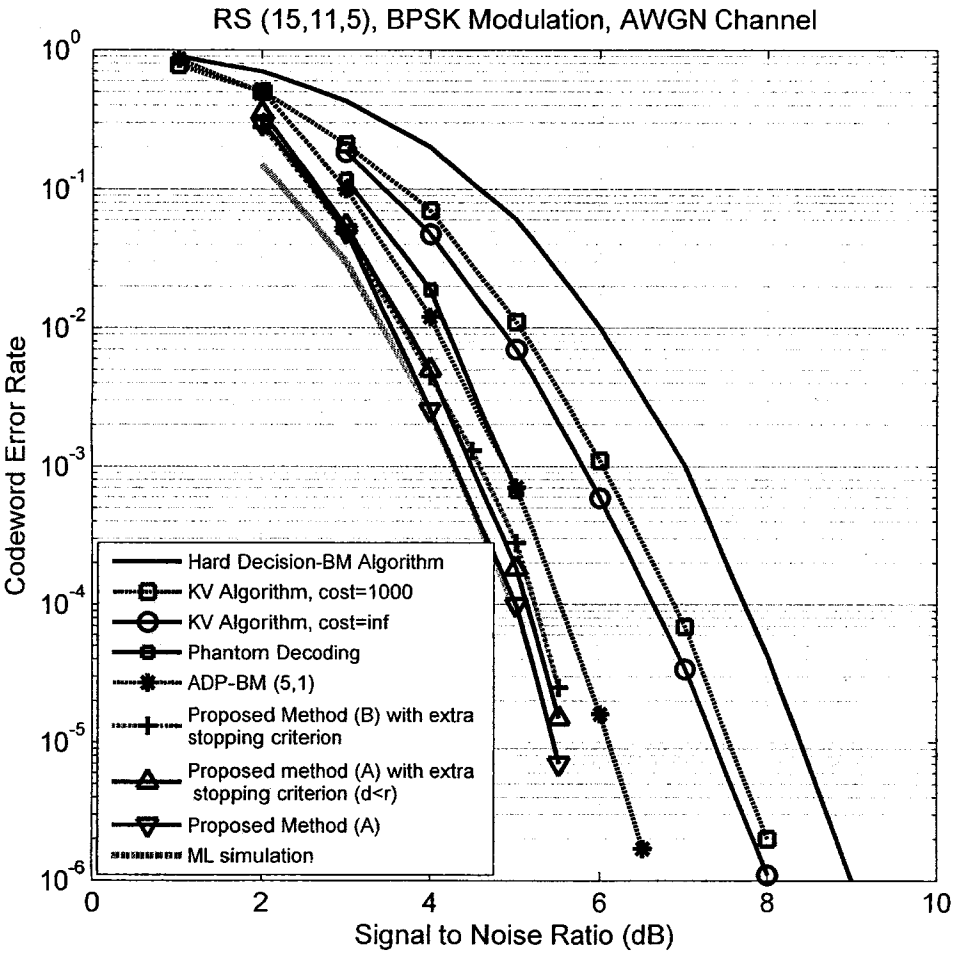

Figure 4.6: Performance of the proposed algorithms (A and B) for RS(15,11)

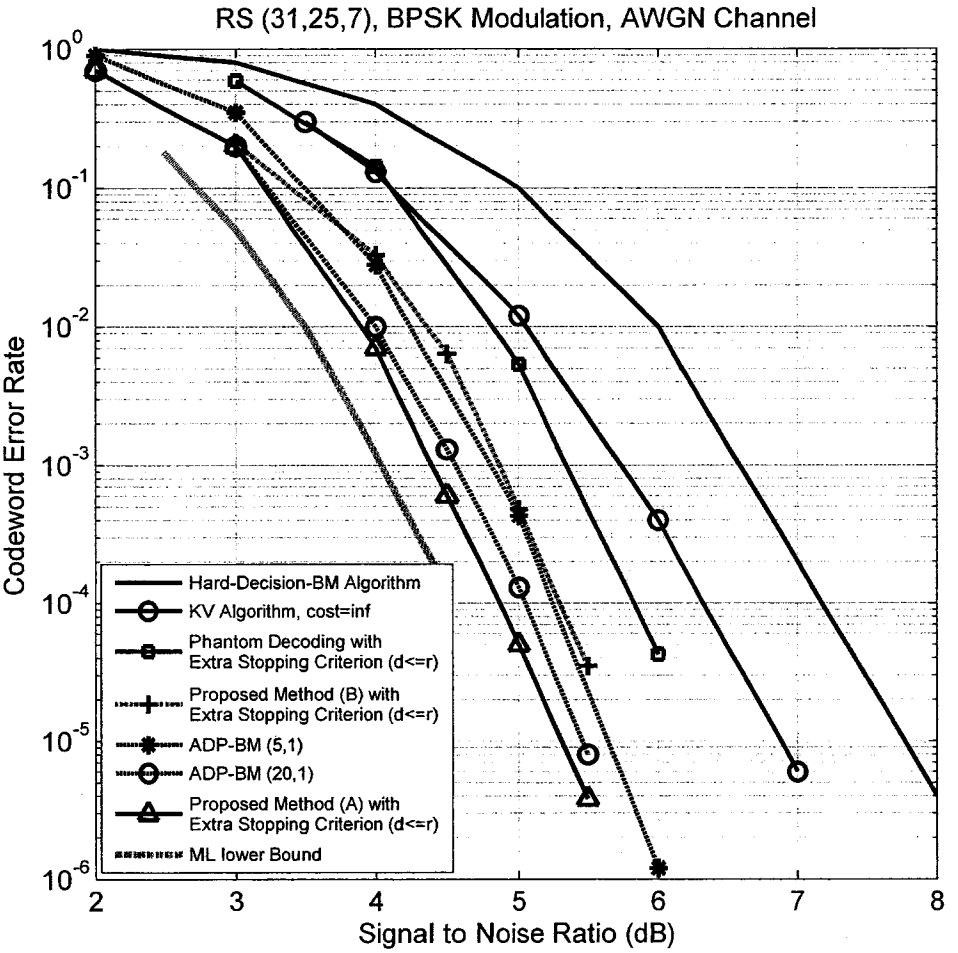

Figure 4.7: Performance of the proposed algorithms (A and B) for RS(31,25)

Table 4.2: The average number of required BP iterations for RS(31,25)

Eb/N0 (dB)	3	4	5	5.5
Method A with extra stopping criterion ($d \leq r$)	1139	331	41	11
Method B with extra stopping criterion ($d \leq r$)	164	102	22	7

4.5.3 RS(63,55) code

From Figure 4.8, at codeword error rate of 10^{-3}, method A provides better performance (about 0.2 dB coding gain) compared to ADP-BM (5,1) algorithm. Also it provides a coding gain of about 1.75 dB compared to the BM algorithm and 0.75 dB compared to phantom decoding. The performance of method A is about 0.25 dB away from the performance of ADP-BM (20,3). Method B has about 1.35 dB coding gain over the BM algorithm and 0.3 dB over phantom decoding.

Table 4.3: The average number of required BP iterations for RS(63,55)

Eb/N0 (dB)	4	4.5	5	5.5
Method A with extra stopping criterion ($d \leq r$)	2012	1089	373	29
Method B with extra stopping criterion ($d \leq r$)	205	147	90	42

4.5.4 RS(255,239) code

From Figure 4.9, at codeword error rate of 10^{-4}, method A provides better performance (about 0.15 dB coding gain) compared to ADP-BM (5,1) algorithm. Also it provides a coding gain of about 1.05 dB compared to the BM algorithm, 0.5 dB compared to the KV algorithm and 0.7 dB compared to phantom decoding. Method B has about 0.65 dB coding gain over BM algorithm, 0.2 dB over phantom decoding and its performance is very close to the performance of KV algorithm. The performance of method A is about 0.25 dB away from the performance of ADP-BM (20,3) at codeword error rate of 10^{-3}. However, as discussed in Section 4.4, the complexity of our methods is much less than the ADP method

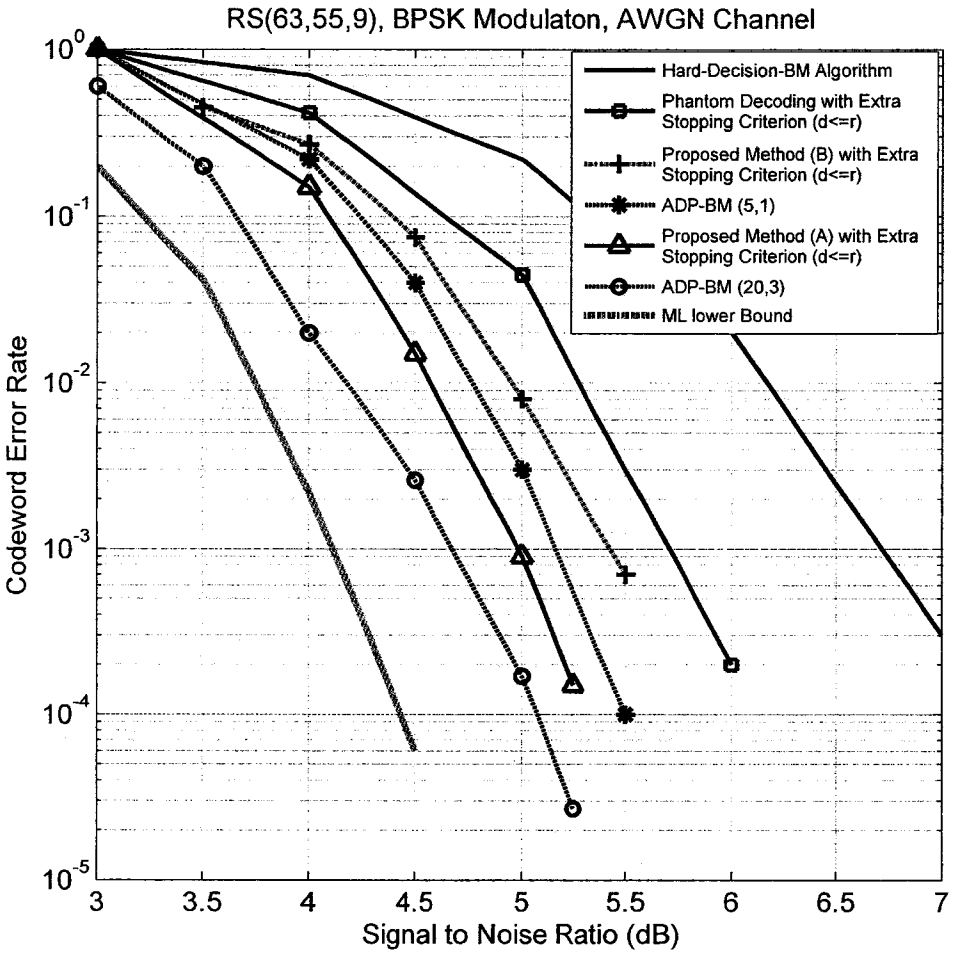

Figure 4.8: Performance of the proposed algorithms (A and B) for RS(63,55)

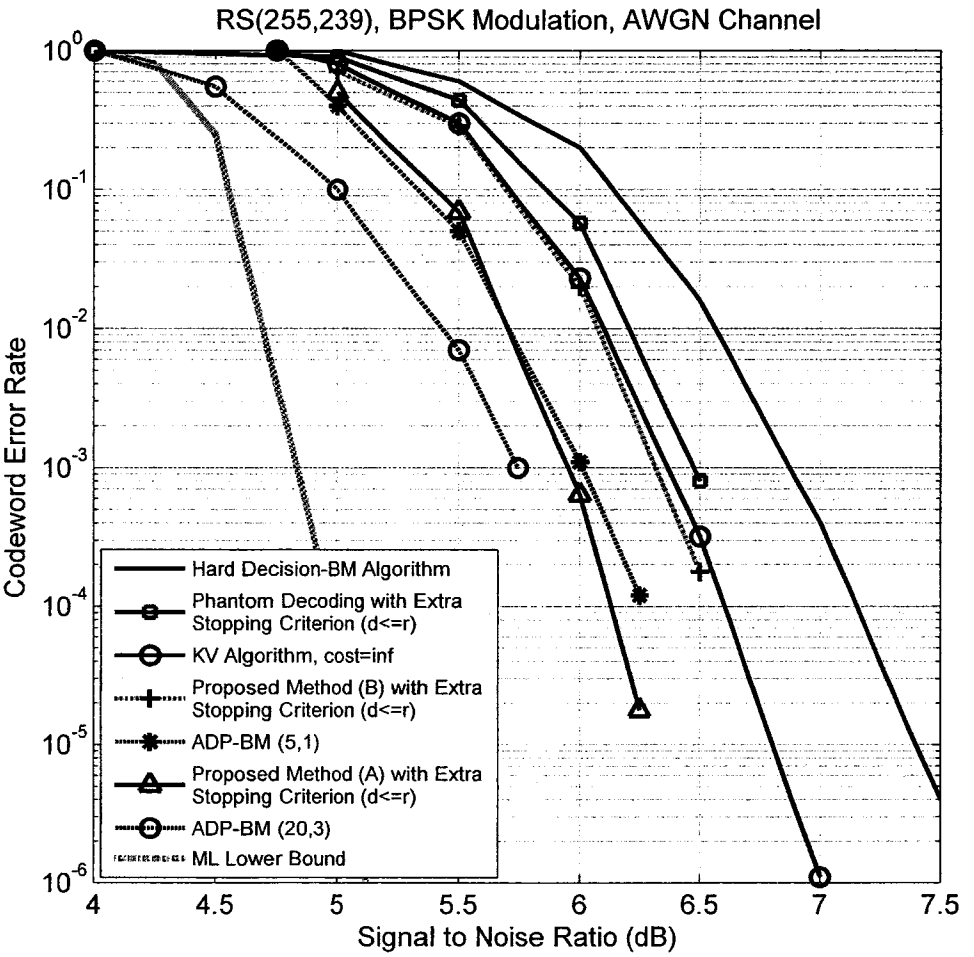

Figure 4.9: Performance of the proposed algorithms (A and B) for RS(255,239)

mainly because we do not perform Gaussian eliminations at every iteration.

From Tables 4.1, 4.2, 4.3 and 4.4, we can see that by increasing the signal to noise ratio, the number of required BP iterations and therefore the complexity of the iterative algorithms are reduced. Also, the average number of required BP iterations is much less than the maximum number of BP iterations for both methods. The other result is that, keeping the same number of information correction steps in method B, the larger the code length, the higher the difference between the performances and also complexities of method A and method B.

Table 4.4: The average number of required BP iterations for RS(255,239)

Eb/N0 (dB)	5	5.5	6	6.5
Method A with extra stopping criterion ($d \leq r$)	8894	1986	58	3
Method B with extra stopping criterion ($d \leq r$)	264	158	17	2

4.6 Conclusion

We have proposed two soft decision decoding algorithms (A and B) based on bit level belief propagation decoding for RS codes. The advantage of our methods over the ADP method is that they work with the fixed parity check matrix. In both methods, we have used an extended binary parity check matrix with lower density and reduced number of 4-cycles compared to the original binary parity check matrix of the code. We have investigated the performance of normal BP decoding with different binary parity check matrices over a binary erasure channel and realized that the extended binary parity check matrix provides better performance. Method A is based on the cyclic structure of RS codes and its geometric interpretation has also been presented. Simulation results have shown that method A has significant coding gain over hard decision decoding. Its performance is also superior to some other popular soft decision decoding methods including the KV method and the ADP method. Method B is based on information correction in BP decoding. Compared to method A, method B needs less BP iterations but its performance is not as good. We have also presented complexity analysis for both methods.

Chapter 5

Collaborative Algebraic Decoding of Interleaved RS Codes

Interleaved Reed-Solomon (IRS) codes have a wide range of applications in data processing, data transmission and data storage systems. IRS codes have been the topic of several studies recently [71–74]. They are generally effective in applications where burst errors happen at the channel and affect all words of the interleaved scheme simultaneously. In a similar scenario, IRS codes can be used as outer codes in Generalized Concatenated (GC) codes for channel models with statistically independent random errors. Proposed by Blokh and Zyablov [75], GC codes consist of a number of outer codes whose code symbols are protected by an inner code. Using IRS codes as outer codes of a GC code, the inner decoder generates correlated burst errors at the input of the outer RS decoders.

In this chapter, we consider general IRS codes where each codeword is an $M \times N$ matrix consisting of M rows (codewords) from M RS codes of length N and dimensions K_1, K_2, ..., K_M. If all the M dimensions are equal, the code is called a homogeneous IRS code, otherwise, it's called a heterogeneous IRS code.

In traditional applications, each RS codeword of an IRS codeword is decoded independently. The classic method for hard decision decoding of RS codes is Berlekamp-Massey (BM) algorithm [6] with error correction capability equal half the minimum distance of the code. Collaborative decoding of IRS codes based on BM algorithm has been proposed in [76] assuming errors in the received signal occur in bursts. This collaborative decoding strategy locates the errors jointly in all RS codewords instead of locating them independently in the several words. Up to t errors can be located uniquely, in many cases even if t is larger than half the minimum distance of the RS code with the largest dimension.

In 1999, Guruswami and Sudan (GS) [11] proposed a new method to improve the error correcting capability of RS codes beyond their traditional capability (half the minimum distance). Later, Parvaresh and Vardy [77] proposed multivariate interpolation decoding of RS codes based on GS algorithm and showed that if errors happen simultaneously for multiple codewords of an RS code, errors beyond GS algorithm can be corrected.

In this chapter, we derive and analyze an algorithm for collaborative decoding of heterogeneous IRS codes in the presence of burst errors based on multivariate interpolation decoding of RS codes [77]. Similar to GS algorithm, our method has two steps: interpolation and factorization. We find the error correction capability of the proposed algorithm and show that it is larger than the decoding radius of GS algorithm for the RS code with the largest dimension. Then, we analyze the performance of concatenated codes using IRS codes as their outer codes. We derive upper and lower bounds for the word error probability of GC codes over AWGN channel with BPSK modulation for both cases of independent and collaborative decoding of the outer IRS codes. We will show that using collaborative decoding, the word error probability is better compared to the case of independent decoding.

The rest of this chapter is organized as follows. In section 5.1, interleaved Reed-Solomon codes are introduced. The proposed collaborative decoding of IRS codes based on GS algorithm [11] is explained in detail in section 5.2 including the interpolation step and the factorization step. The error correction capability of the proposed method is also

derived in this section. In section 5.3, the use of IRS codes in concatenated codes is discussed and the performance of GC codes over AWGN channel with BPSK modulation is considered. Also, lower and upper bounds on the performance of GC codes are derived. Numerical results and discussions are presented in section 5.4. Finally, conclusions are given in section 5.5.

5.1 Interleaved Reed-Solomon Codes

As discussed in Chapter one, an RS code of length N and dimension K over the Galois field $GF(q)$ with support set $D = \{x_1, x_2, ..., x_N\} \subset GF(q)$ is defined as

$$C_q(N,K) = \tag{5.1}$$
$$\{(f(x_1), f(x_2), ..., f(x_N)) | x_1, x_2, ..., x_N \in D, f(X) \in F_q[X], degf(X) < K\}$$

where $F_q[X]$ is the ring of polynomials over the Galois field $GF(q)$ in a variable X. RS codes are maximum distance separable (MDS) and therefore from any set of K correct symbols, an RS codeword can uniquely be reconstructed. An interleaved Reed-Solomon code is now obtained by taking M Reed-Solomon codes over the same Galois field $GF(q)$ and grouping them row-wise into a matrix. The codewords of an IRS code are matrices whose rows are the codewords of the Reed-Solomon codes. We denote these M Reed-Solomon codes by $RS^{(1)}$, $RS^{(2)}$, ..., $RS^{(M)}$ where $RS^{(i)} = RS(N, K_i, d_i)$, $i = 1, 2, ..., M$ and $d_i = N - K_i + 1$. Now, an interleaved RS code denoted by $IRS(N, K_1, K_2, ..., K_M)$ is defined as

$$IRS(N, K_1, K_2, ..., K_M) = \left\{ \begin{pmatrix} c^1 \\ c^2 \\ \cdots \\ c^M \end{pmatrix}, c^i \in RS^{(i)}, i \in 1, ..., M \right\}. \tag{5.2}$$

If all the M Reed-Solomon codes are equivalent, i.e., $RS^{(1)} = RS^{(2)} = ... = RS^{(M)}$, the IRS code is called homogeneous. Otherwise, we say that the IRS code is heterogeneous. Figure 5.1 shows a typical IRS codeword.

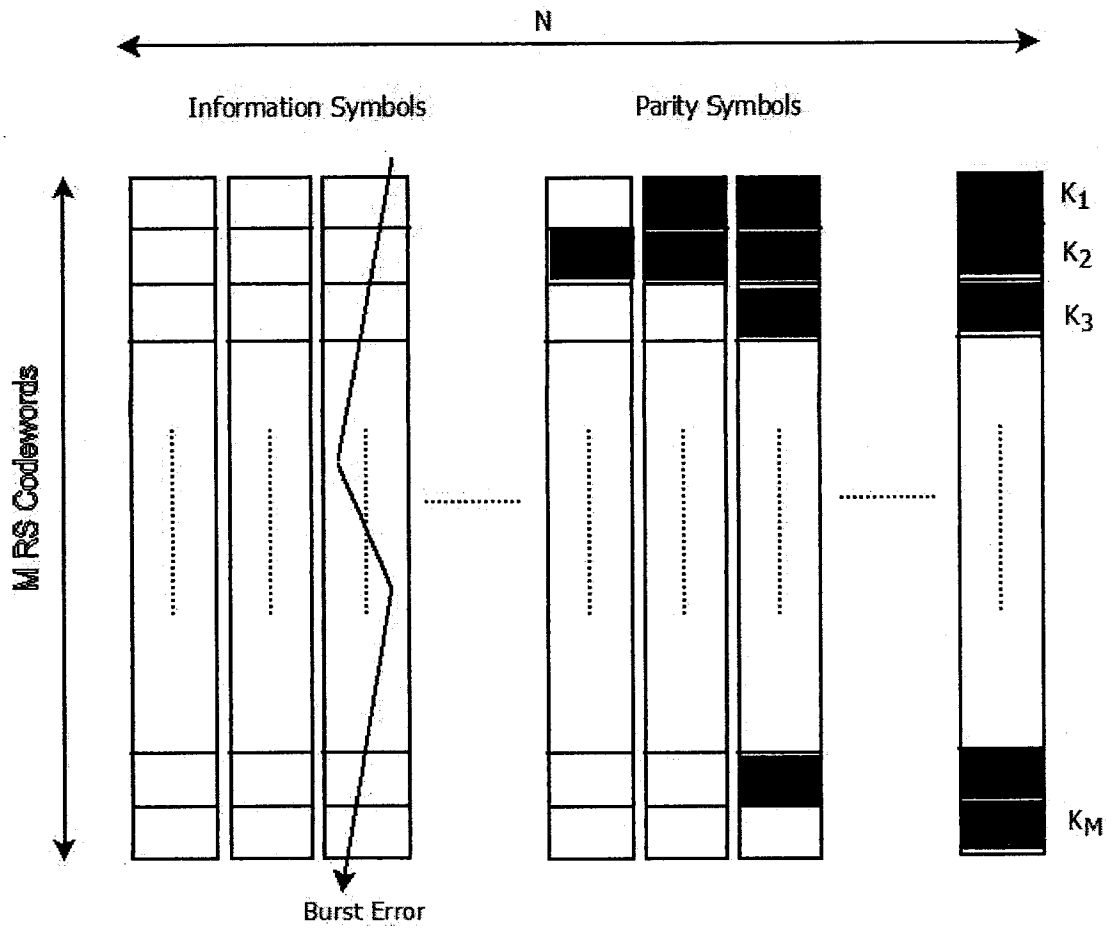

Figure 5.1: Interleaved Read-Solomon Code

5.2 Collaborative Interpolation Decoding of IRS Codes

The basis of GS decoding algorithm [11] has been explained in Chapter 2, Section 2.3. In this section, we use the basis of GS algorithm and introduce collaborative decoding of IRS codes assuming burst errors. In this method instead of decoding each RS codeword independently, we try to decode all the codewords simultaneously. In the end, we show that we can increase the error correction capability of GS algorithm using this method of decoding.

Suppose that one codeword $(c^1, c^2, ..., c^M)^T$ of an $IRS(N, K_1, K_2, ..., K_M)$ code, corresponding to evaluations of the polynomials $f^1(X)$, $f^2(X)$, ..., $f^M(X)$ over $GF(q)$ with degrees less than K_1, K_2, ..., K_M respectively, is transmitted over a hard decision channel and received as

$$\begin{pmatrix} y^1 \\ y^2 \\ \cdots \\ y^M \end{pmatrix} = \begin{pmatrix} c^1 \\ c^2 \\ \cdots \\ c^M \end{pmatrix} + \begin{pmatrix} e^1 \\ e^2 \\ \cdots \\ e^M \end{pmatrix}. \tag{5.3}$$

where c^i, y^i and e^i, $i = 1, ..., M$ are vectors with N elements (symbols). In the case of IRS codes, one codeword which is a matrix is transmitted across the channel column-by-column. In the presence of burst errors in the channel, we can assume there are at most t synchronized errors meaning the error matrix has at most t nonzero columns. In this case, the M Reed-Solomon codewords may have erroneous symbols at the same positions (columns).

Each codeword can be decoded separately at the decoder using the GS algorithm. From Chapter 2, Section 2.3, the asymptotic ($m \to \infty$) error correcting capability of the GS decoder for an (N, K_i) RS code is $t_i = \left\lfloor N(1 - \sqrt{(K_i - 1)/N}) \right\rfloor$. Therefore, if each codeword of the IRS code is decoded independently using GS algorithm, the error correcting capability for the IRS code is

$$t_g = min\{t_i, i = 1, 2, ..., M\} = \left\lfloor N(1 - \sqrt{(\overline{K} - 1)/N}) \right\rfloor \tag{5.4}$$

where $\overline{K} = max\{K_i, i = 1, 2, ..., M\}$. However, since errors happen at the same positions, a collaborative decoding strategy can be applied to decode all the codewords at the same time. This may allow for correcting up to t synchronized errors with $t > t_g$.

For collaborative decoding, we follow the basis of GS algorithm. First, we show each of the received vectors $y^i, i = 1, 2, ..., M$ with $\{y_1^i, y_2^i, ..., y_N^i\}$. Given the support set $D = \{x_1, x_2, ..., x_N\}$ of $GF(q)$, we consider the following set of points P in an $(M+1)$-dimensional space:

$$P = \left\{(x_1, y_1^1, y_1^2, ..., y_1^M), (x_2, y_2^1, y_2^2, ..., y_2^M), ..., (x_N, y_N^1, y_N^2, ..., y_N^M)\right\}. \tag{5.5}$$

The collaborative decoding algorithm has two major steps (interpolation and factorization) that will be explained in the following.

5.2.1 Interpolation step

In this step, given the point set P in $GF(q)^M$ and a positive integer m, we try to compute a nontrivial $(M+1)$-variate polynomial $Q_P(X, Y^1, Y^2, ..., Y^M)$ of minimal $(1, K_1 - 1, K_2 - 1, ..., K_M - 1)$-weighted degree over $GF(q)$ that passes through all the points in P with multiplicity at least m.

The weighted degree of a polynomial can be defined as the weighted degree of its leading monomial. The $(1, K_1 - 1, K_2 - 1, ..., K_M - 1)$-weighed degree of the monomial $X^{(i_0)}Y^{1(i_1)}...Y^{M(i_M)}$ is defined as

$$_W deg X^{(i_0)}Y^{1(i_1)}...Y^{M(i_M)} = i_0 + (K_1 - 1)i_1 + ... + (K_M - 1)i_M. \tag{5.6}$$

The weighted degree can be extended to monomial ordering $\prec_w$ if augmented with the lex order [62]. We define w-lex order as

$$X^{(i_0)}Y^{1(i_1)}...Y^{M(i_M)} \prec_w X^{(j_0)}Y^{1(j_1)}...Y^{M(j_M)} \tag{5.7}$$

if either $i_0 + (K_1 - 1)i_1 + ... + (K_M - 1)i_M < j_0 + (K_1 - 1)j_1 + ... + (K_M - 1)j_M$ or $i_0 + (K_1 - 1)i_1 + ... + (K_M - 1)i_M = j_0 + (K_1 - 1)j_1 + ... + (K_M - 1)j_M$ and $i_0 < j_0$ or

$i_0 + (K_1 - 1)i_1 + ... + (K_M - 1)i_M = j_0 + (K_1 - 1)j_1 + ... + (K_M - 1)j_M$ and $i_0 = j_0$ and $i_1 < j_1$ or $\cdots$.

In order to find the $(M + 1)$-variate polynomial Q_P, we use the Hasse derivatives [62]:

$$\delta_{a_0,a_1,...,a_M}[Q_P(X, Y^1, Y^2, ..., Y^M)] = \sum_{i_0=a_0}^{\infty} ... \sum_{i_M=a_M}^{\infty} \binom{i_0}{a_0} ... \binom{i_M}{a_M} q_{i_0 i_1 ... i_M} X^{(i_0-a_0)} Y^{1^{(i_1-a_1)}} ... Y^{M^{(i_M-a_M)}} \tag{5.8}$$

where $q_{i_0 i_1 ... i_M}$ is the coefficient of the monomial $X^{(i_0)} Y^{1^{(i_1)}} ... Y^{M^{(i_M)}}$ of Q_P. It is proved that Q_P passes through $(x_0, x_1, ..., x_M)$ with multiplicity m if

$$\delta_{a_0,a_1,...,a_M}[Q_P]|_{(x_0,x_1,...,x_M)} = 0, \forall (a_0, a_1, ..., a_M) : 0 \leq a_0 + a_1 + ... + a_M < m. \tag{5.9}$$

Lemma 1.

$$_W deg Q_P(X, Y^1, Y^2, ..., Y^M) \leq \left\lceil \sqrt[M+1]{N(K_1 - 1)...(K_M - 1)m(m + 1)...(m + M)} \right\rceil. \tag{5.10}$$

Proof. Equation (5.9) is actually a system of linear constraints on the coefficients of Q_P. Based on this equation, in order for Q_P to pass through each point of P with multiplicity m, $\binom{m+M}{M+1}$ constraints should be satisfied. Because P has N points, the total number of constraints will be

$$c(N, M, m) = N \binom{m + M}{M + 1}. \tag{5.11}$$

We mention $c(N, M, m)$ as the interpolation cost. We denote the number of $(M + 1)$-variate monomials with weighted degree at most σ by $Num(\sigma)$. We can say that there exists a polynomial Q_P of weighted degree at most σ that passes through all the points in P with multiplicity at least m if

$$Num(\sigma) > c(N, M, m). \tag{5.12}$$

In order for the weighted degree of a monomial shown in Equation (5.6) to be smaller than σ, $\{i_0, i_1, ..., i_M\}$ should satisfy the following inequalities:

$$i_0 \geq 0, i_1 \geq 0, ..., i_M \geq 0, i_0 + (K_1 - 1)i_1 + ... + (K_M - 1)i_M \leq \sigma. \tag{5.13}$$

The above ranges define a pyramid with $M+1$ sides in the $M+1$ dimensional space. We can say that

$$Num(\sigma) > volume(pyramid) = \frac{1}{(M+1)!} \frac{\sigma^{M+1}}{(K_1 - 1)(K_2 - 1)...(K_M - 1)}. \tag{5.14}$$

Therefore, we can say that Q_P exists if $volume(pyramid) \geq c(N, M, m)$. Using equations (5.11) and (5.14), we have

$$\sigma \geq \left\lceil \sqrt[M+1]{N(K_1 - 1)...(K_M - 1)m(m+1)...(m+M)} \right\rceil. \tag{5.15}$$

Knowing that σ is an upper bound for the weighted degree of Q_P, the upper bound in Equation (5.10) satisfies Equation (5.15) with equality which proves the Lemma. ■

Theorem 1. Suppose that one codeword $(c^1, c^2, ..., c^M)^T$ of an $IRS(N, K_1, K_2, ..., K_M)$ code, corresponding to evaluations of the polynomials $f^1(X)$, $f^2(X)$, ..., $f^M(X)$ over $GF(q)$ with degrees less than K_1, K_2, ..., K_M respectively, is transmitted over a hard decision channel and at most t synchronized errors happen. The expression $P(X) = Q_P(X, f^1(X), f^2(X), ..., f^M(X)) = 0$ is satisfied if

$$t \leq t_{max} = \left\lfloor N - N \sqrt[M+1]{\frac{K_1 - 1}{N}\frac{K_2 - 1}{N}...\frac{K_M - 1}{N}(1 + \frac{1}{m})(1 + \frac{2}{m})...(1 + \frac{M}{m})} - \frac{1}{m} \right\rfloor. \tag{5.16}$$

Proof. Since there are at most t synchronized errors, $P(X)$ has at least $m(N-t)$ zeros. Also, the degree of $P(X)$ cannot exceed ${}_W deg Q_P(X, Y^1, Y^2, ..., Y^M)$. From these two facts and using the bound in Equation (5.10), in order for $P(X)$ to be an all zero polynomial, the following condition should be satisfied based on the fundamental theorem of algebra:

$$\left\lceil \sqrt[M+1]{N(K_1 - 1)...(K_M - 1)m(m+1)...(m+M)} \right\rceil - 1 \leq m(N-t). \tag{5.17}$$

From the condition in Equation (5.17), the expression in (5.16) is concluded. ■

t_{max} in Equation (5.16) is called the error correction capability of multivariate interpolation algorithm. The algorithm that is used to find $Q_P(X, Y^1, Y^2, ..., Y^M)$ is explained in detail in the following section.

Lemma 2. For an $IRS(N, K_1, K_2, ..., K_M)$ code, assuming the asymptotic case of $m \to \infty$, if we denote the error correction capability of multivariate interpolation algorithm with t_{max} and the error correction capability of independent decoding with t_g, we have

$$t_{max} > t_g. \tag{5.18}$$

Proof. As $m \to \infty$, the error correction capability of multivariate interpolation algorithm can be written as

$$t_{max} = \left\lfloor N - N \sqrt[M+1]{\frac{K_1 - 1}{N}\frac{K_2 - 1}{N}...\frac{K_M - 1}{N}} \right\rfloor. \tag{5.19}$$

The expression for t_g is also given in Equation (5.4). From these two formulas and using $\overline{K} = max\{K_i, i = 1, 2, ..., M\}$, we have

$$t_{max} \geq \left\lfloor N - N \sqrt[M+1]{\frac{\overline{K} - 1}{n}\frac{\overline{K} - 1}{N}...\frac{\overline{K} - 1}{N}} \right\rfloor \geq \left\lfloor N - N\left(\frac{\overline{K} - 1}{N}\right)^{\frac{M}{M+1}} \right\rfloor$$
$$> \left\lfloor N(1 - \sqrt{(\frac{\overline{K} - 1)}{N}}) \right\rfloor = t_g. \blacksquare \tag{5.20}$$

We should mention that the lower the rates of RS codes in the structure of an IRS code, the higher the difference between t_{max} and t_g.

5.2.1.1 Collaborative Interpolation Algorithm

We use the basis of Koetter's interpolation algorithm [57] with some modifications and extend the algorithm proposed in [77]. The inputs to the algorithm are the point set P, multiplicity m and the weighted-degree monomial order defined in Equation (5.6). The

output will be a Grobner basis for the ideal of all the (M+1)-variate polynomials over $GF(q)$ that pass through all the points in P with multiplicity m. We denote this ideal with $I_m(P)$. We first initialize the Grobner basis:

$$G^{(0)} = \{G_1^0, G_2^0, ..., G_L^0\} =$$
$$\left\{ Y^{1^{(a_1)}} ... Y^{M^{(a_M)}} : \forall (a_1, a_2, ..., a_M) \in N^M \quad \text{such that} \quad \sum_{i=1}^{M} (k_i - 1)a_i < \sigma_{min} \right\} \tag{5.21}$$

where

$$\sigma_{min} = min \left\{ \sigma : \sigma > \left\lceil \sqrt[M+1]{N(K_1 - 1)...(K_M - 1)m(m+1)...(m+M)} \right\rceil \right\} \tag{5.22}$$

The interpolation algorithm has $c(N, M, m)$ iterations to impose each of the $c(N, M, m)$ linear constraints one at a time. At iteration l of the algorithm corresponding to the point $(x_i, y_i^1, ..., y_i^M) \in P$, we have to perform the following steps:

- Compute discrepancies $\Delta_j = \delta_{a_0, a_1, ..., a_M}[G_j^{l-1}]|_{(x_i, y_i^1, ..., y_i^M)}$, $j = 1, 2, ..., L$. If $\Delta_j = 0$ for all $j = 1, 2, ..., L$, stop.

- Among the set $G^{(l-1)} = \{G_1^{l-1}, G_2^{l-1}, ..., G_L^{l-1}\}$, find the least weighted degree (with respect to $\prec_\omega$) polynomial with nonzero discrepancy denoted by G_t^{l-1}.

- For all $j = 1, 2, ..., L$ except $j = t$:

$$G_j^l = G_j^{l-1} - \frac{\Delta_j}{\Delta_t} G_t^{l-1} \qquad \text{"no increase in the weighted degree"} \tag{5.23}$$

  For $j = t$:

$$G_t^l = (X - x_i) G_t^{l-1} \qquad \text{"weighted degree increased by 1"} \tag{5.24}$$

By the end of $c(N, M, m)$ iterations, the set $G = \{G_1, G_2, ..., G_L\}$ returned by the above iterative interpolation algorithm is a Grobner basis for $I_m(P)$. After arranging the polynomials in G in an increasing order with respect to $\prec_\omega$, we can take the least weighted degree polynomial $G_1(X, Y^1, ..., Y^M)$ as the interpolation polynomial $Q_P(X, Y^1, ..., Y^M)$.

5.2.2 Factorization step

In this section, given the multivariate polynomial $Q_P(X, Y^1, ..., Y^M)$, we try to identify all polynomials $f^1(X)$, $f^2(X)$, ..., $f^M(X)$ of degrees less than K_1, K_2,..., K_M respectively such that

$$Q_P(X, f^1(X), f^2(X), ..., f^M(X)) = 0. \tag{5.25}$$

The output is a list of the codewords corresponding to these polynomials. The problem here is the difficulty of such task especially for large M. From [77], we present a method for the case $M = 2$ which with proper changes can be extended for $M \geq 3$.

In order to explain the factorization process, first we need to define the resultant of two polynomials. We assume two polynomials $A(x)$ and $B(x)$ over a field such that

$$A(x) = a_0 x^l + \cdots + a_l, \qquad a_0 \neq 0$$
$$B(x) = b_0 x^m + \cdots + b_m, \qquad b_0 \neq 0.$$

The Sylvester matrix of A and B with respect to x, denoted by $Syl(A, B, x)$, is the following $(l + m) \times (l + m)$ matrix [78]:

$$Syl(A, B, x) = \begin{pmatrix} a_0 & 0 & \cdots & 0 & b_0 & 0 & \cdots & 0 \\ a_1 & a_0 & \ddots & \cdots & b_1 & b_0 & \ddots & \cdots \\ a_2 & a_1 & \ddots & 0 & b_2 & b_1 & \ddots & 0 \\ \vdots & & \ddots & a_0 & \vdots & & \ddots & b_0 \\ & \vdots & & a_1 & & \vdots & & b_1 \\ a_{l-1} & & & & b_{m-1} & & & \\ a_l & a_{l-1} & & \vdots & b_m & b_{m-1} & & \vdots \\ 0 & a_l & \ddots & & 0 & b_m & \ddots & \\ \vdots & \ddots & \ddots & a_{l-1} & \vdots & \ddots & \ddots & b_{m-1} \\ 0 & \cdots & 0 & a_l & 0 & \cdots & 0 & b_m \end{pmatrix}. \tag{5.26}$$

The resultant of A and B with respect to x is the determinant of the Sylvester matrix [78],

$$Res(A, B, x) = det(Syl(A, B, x)). \tag{5.27}$$

Assuming A and B both have positive degrees, there exist two polynomials $C(x)$ and $D(x)$ over the same field as A and B such that [78]

$$Res(A, B, x) = AC + BD. \tag{5.28}$$

Using the Grobner basis obtained in the interpolation part, we take two least weighted degree polynomials $G_1(X, Y^1, Y^2)$ and $G_2(X, Y^1, Y^2)$ and find their resultants [78]:

$$\begin{aligned} R_1(X, Y^1) &= Res(G_1, G_2, Y^2), \\ R_2(X, Y^2) &= Res(G_1, G_2, Y^1). \end{aligned} \tag{5.29}$$

From Equation (5.28), if $G_1(X, f^1(X), f^2(X)) = 0$ and $G_2(X, f^1(X), f^2(X)) = 0$, then

$$R_1(X, f^1(X)) = 0, \tag{5.30}$$

$$R_2(X, f^2(X)) = 0. \tag{5.31}$$

Therefore, we can use Roth-Ruckenstein algorithm [58] to factor $R_1(X, Y^1)$ in order to find all polynomials $f^1(X)$ with degrees less than K_1 and factor $R_2(X, Y^2)$ in order to find all polynomials $f^2(X)$ with degrees less than K_2.

As we mentioned before, if we only use $G_1 = Q_P$ for the decoding, the error correction capability will be t_{max} from Equation (5.16). However, for the factorization process described here, we also use G_2. In order for $G_2(X, f^1(X), f^2(X)) = 0$, the weighted degree of G_2 should satisfy $\delta_2 - 1 \leq m(N - t)$ based on what was explained in the proof of Theorem 1. Here, we extend Lemma 5 in [77] to find a bound for δ_2.

Lemma 3. Let δ_1 and δ_2 denote the weighted degrees of $G_1(X, Y^1, Y^2)$ and $G_2(X, Y^1, Y^2)$ respectively, then:

$$\frac{\delta_2^3 - (\delta_2 - \delta_1)^3}{3!(K_1 - 1)(K_2 - 1)} \leq c(N, M, m) = N\binom{m+2}{2+1} \tag{5.32}$$

Proof. The deltaset Δ of an ideal I of polynomials is defined as the set of all monomials that are not the leading monomials of the polynomials in the ideal. If $I_m(P)$ denote the

ideal of all the (M+1)-variate polynomials ($M = 2$) over $GF(q)$ that pass through all the points in P with multiplicity m, then $|\Delta(I_m(P))| = c(N, M, m)$ (for proof, see [79]).

From **Lemma 1**, for $M = 2$, the number of monomials with weighted degree smaller than δ is approximately equal to $\frac{\delta^3}{3!(K_1-1)(K_2-1)}$. Therefore, the number of monomials with weighted-degree smaller than δ_2 in $\Delta(I_m(P))$ is almost equal to the number of monomials with weighted-degree smaller than δ_2 ($\frac{\delta_2^3}{3!(K_1-1)(K_2-1)}$) minus the monomials that G_1 carves out the deltaset ($\frac{(\delta_2-\delta_1)^3}{3!(K_1-1)(K_2-1)}$). For more details, see the proof of **Lemma 5** in [77]. ■

Now, we are ready to find the error correction capability of collaborative decoding of a heterogeneous IRS code with two RS codewords using the factorization process explained above.

Theorem 2. Suppose that one codeword $(c^1, c^2)^T$ of an $IRS(N, K_1, K_2)$ code, corresponding to evaluations of the polynomials $f^1(X)$, $f^2(X)$ with degrees less than K_1, K_2 respectively, is transmitted over a hard decision channel and at most t synchronized errors have happened. The expressions $G_1(X, f^1(X), f^2(X)) = 0$ and $G_2(X, f^1(X), f^2(X)) = 0$ are satisfied if

$$t \leq t_{max2} = \left\lfloor N - \frac{N}{2}\left[\frac{K_1-1}{N}\frac{K_2-1}{N}(1+\frac{1}{m})(1+\frac{2}{m})\right]^{1/2} \left(1+\sqrt{\frac{4}{3}\left[\frac{K_1-1}{N}\frac{K_2-1}{N}(1+\frac{1}{m})(1+\frac{2}{m})\right]^{-1/2}-\frac{1}{3}}\right)-\frac{1}{m}\right\rfloor. \quad (5.33)$$

Proof. Using Equation (5.32) and the bound for δ_1 expressed in Equation (5.10), we find a bound for δ_2 as follow:

$$\delta_2 \leq \frac{\delta_1}{2}\left(1+\sqrt{\frac{4}{3}\frac{Nm(m+1)(m+2)(K_1-1)(K_2-1)}{\delta_1^3}-\frac{1}{3}}\right) \quad (5.34)$$

As we mentioned before, in order for $G_2(X, f^1(X), f^2(X)) = 0$, the weighted degree of G_2 should satisfy $\delta_2 - 1 \leq m(N-t)$. Combining this expression with the bound in Equation (5.34), expression in (5.33) is concluded. ■

We have checked t_{max2} and realized that the value of t_{max2} for different cases is almost the same as the value of t_{max} especially as the code rate increases. As a result, using $G_2(X, Y^1, Y^2)$ for the decoding does not change the error correction capability of the collaborative decoding.

We should mention that when G_1 and G_2 have common factors with positive degrees in Y^1 and Y^2, their resultants become zero [78]. In order to deal with situations like this, the factorization process should be modified as follows [77]:

1. Initializations: $H_1 = G_1$, $H_2 = G_2$, $j = 3$.
2. $H_1 = gcd(H_1, H_2) \times F_1$, $H_2 = gcd(H_1, H_2) \times F_2$.
3. If $gcd(H_1, H_2)$ is only in X, use Roth-Ruckenstein algorithm [58] to factor $R_1 = Res(H_1, H_2, Y^2)$ and $R_2 = Res(H_1, H_2, Y^1)$ in order to find all $f^1(X)$ and $f^2(X)$ with degrees less than K_1 and K_2 respectively.
4. If $gcd(H_1, H_2)$ has positive degrees in Y^1 or Y^2, use Roth-Ruckenstein algorithm [58] to factor $R_1 = Res(F_1, F_2, Y^2)$ and $R_2 = Res(F_1, F_2, Y^1)$ in order to find all $f^1(X)$ and $f^2(X)$ with degrees less than K_1 and K_2 respectively.
5. $H_1 = gcd(H_1, H_2)$, $H_2 = G_j$, $j = j + 1$ and go to step 2.

Here, 'gcd' represents the greatest common divisor. As it can be seen from the above factorization process, we might need to use G_i's, $i > 2$ in order to find all the possible $f^1(X)$ and $f^2(X)$ with degrees less than K_1 and K_2 respectively. Therefore, the error correction capability might be less than t_{max2}. However, we have verified using simulation that in most of the cases up to t_{max2} simultaneous errors can be corrected. Therefore, the performance using simulation is almost the same as the bound obtained from t_{max2}. As an example, we have considered the decoding of IRS(15,9,7) over a 256-ary symmetric channel. Each codeword is sent column by column where each column can be seen as a symbol over $GF(256)$. As we know, in a q-ary symmetric channel, a symbol is either received unchanged with probability $1-p$ or it is received as one of the other $q-1$ symbols

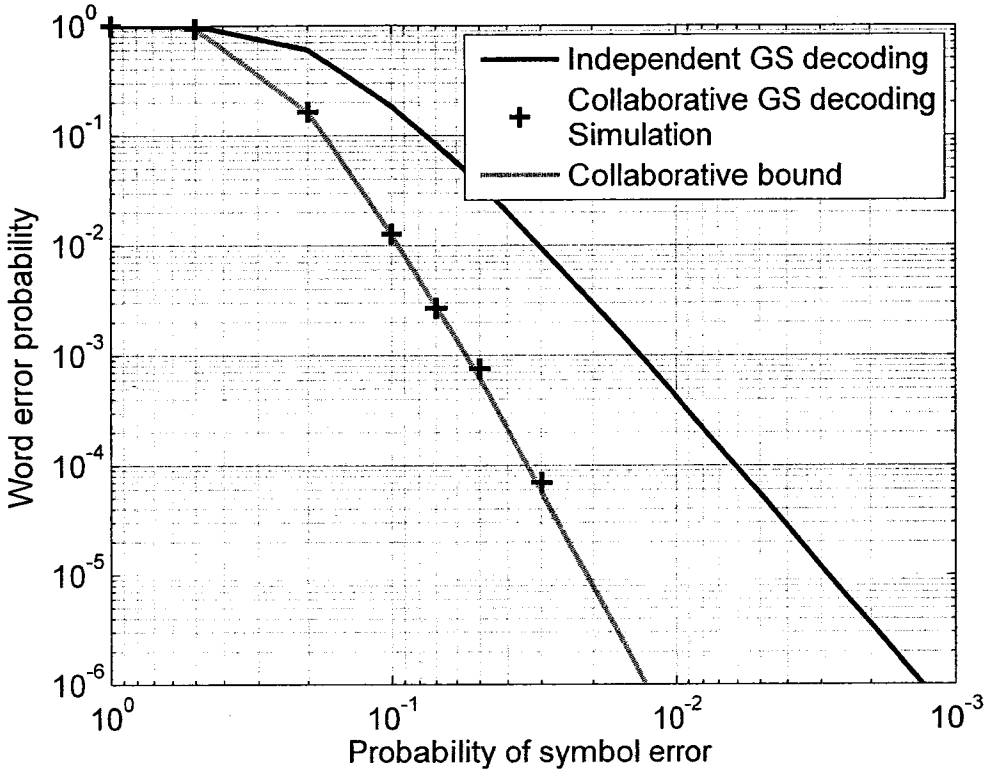

Figure 5.2: Performance of heterogeneous IRS(15,9,7) over a 256-ary symmetric channel. Collaborative and also independent decoding have been considered.

with probability $\frac{p}{q-1}$. With this scenario, we can be sure that errors happen simultaneously for codewords of the IRS scheme. From Figure , there is a slight difference between the performance from simulation and the bound using t_{max2}.

5.3 IRS Codes in Concatenated Code Design

Interleaved RS codes are usually used as outer codes in concatenated code designs. If the decoding of each RS codeword is performed independently, the special interleaved structure is not taken into account by the decoder. In this section, we investigate collaborative interpolation decoding of IRS codes in concatenated code designs. Simple concatenated schemes with outer IRS codes and inner binary linear block codes are considered where

columns of IRS codewords are protected by the inner code. In this scenario, a decoding error at the output of the inner decoder will affect one whole column of the IRS codeword (similar to burst errors) and therefore we will have synchronous errors at the input of the outer decoder.

We consider an IRS codeword c over $GF(2^p)$. Each column a_j, $j = 0, 1, ..., N-1$ of c is transformed into a binary vector of length Mp and encoded using a binary linear block inner code of length N_{in} and dimension $K_{in} = Mp$. In the end we will end up with an $N_{in} \times N$ binary matrix C^c which is the codeword of our concatenated code and is shown as

$$C^c = [c^c_{i,j}]_{N_{in} \times N} \tag{5.35}$$

where $c^c_{i,j} \in GF(2)$. We assume that the concatenated code is transmitted over an AWGN channel with BPSK modulation. The transmitted matrix is

$$X = \left[x_{i,j} | x_{i,j} = -2c^c_{i,j} + 1\right]_{N_{in} \times N}. \tag{5.36}$$

We denote the received matrix by Y with $y_{i,j} = x_{i,j} + n_{i,j}$ where $n_{i,j}$'s are independent Gaussian random variables with zero mean and variance σ^2. The rate of the concatenated code is

$$R_c = \frac{(K_1 + K_2 + ... + K_M)p}{N_{in}N} = \frac{K_{in}}{N_{in}} \frac{K_1 + K_2 + ... + K_M}{NM} \tag{5.37}$$

Using the code rate, we are able to characterize the AWGN channel by its signal to noise ratio:

$$\frac{E_b}{N_0} = \frac{1}{2R_c\sigma^2}. \tag{5.38}$$

At the receiver, first we need to decode Y with respect to the inner code for which we use a maximum likelihood (ML) decoder. The inner decoder response can be shown by an $M \times N$ matrix $\widehat{c}$ over $GF(2^p)$. Any row of $\widehat{c}$ can be decoded independently with respect to the corresponding RS code. However, since an erroneous decision of the inner ML decoder may affect a complete column of the matrix $\widehat{c}$, it might be more efficient to apply the collaborative decoding strategy. In this case, the word error probability P^{ML} at the output of the inner decoder will be the column burst error probability at the input of the collaborative decoder for the outer IRS code.

Unfortunately, the exact analytical calculation of ML performance is generally not possible. However, there are several known upper and lower bounds. Here, we use Poltyrev's Tangential Sphere Bound (TSB) [80] for the upper bound. To bound the ML performance from below, both Shannon's sphere packing Bound (SPB) [81] and Seguin's L_2-Bound [82] are used. The latter is tight for high signal to noise ratios. Therefore,

$$P_L = max(P_{SPB}, P_{L_2}) \tag{5.39}$$

$$P_U = P_{TSB} \tag{5.40}$$

$$0 \leq P_L \leq P^{ML} \leq P_U \leq 1. \tag{5.41}$$

5.3.1 Bounds on the Word Error Probability of Concatenated Codes

In this section, we consider the performance of the collaborative decoding of IRS codes in a concatenated design. We focus on the asymptotic case where the multiplicity m tends to infinity. From Equation (5.19), the maximum number of errors t_{max} that can be corrected using collaborative interpolation is calculated. The word error probability at the end of the outer decoder can be expressed as

$$P^c \cong \sum_{t=t_{max}+1}^{N} \binom{N}{t} p^t (1-p)^{N-t} \tag{5.42}$$

where p is the column burst error probability at the input of the outer decoder which is equal to P^{ML}, the codeword error probability of the inner decoder. From Equation (5.41), we have

$$P_L^t (1-P_U)^{N-t} \leq p^t (1-p)^{N-t} \leq P_U^t (1-P_L)^{N-t}, \tag{5.43}$$

which results in the following lower and upper bounds for the probability of word error under collaborative decoding of the outer code:

$$\sum_{t=t_{max}+1}^{N} \binom{N}{t} P_L^t (1-P_U)^{N-t} \leq P^c \leq \sum_{t=t_{max}+1}^{N} \binom{N}{t} P_U^t (1-P_L)^{N-t}. \tag{5.44}$$

As we mentioned before, the asymptotic error correction capability of independent decoding of IRS codes is shown by t_g form Equation (5.4). Replacing t_{max} with t_g in

Equation (5.44), we can estimate the lower and upper bounds for the probability of word error under independent decoding of the outer code.

5.4 Numerical Results and Discussions

In this section, we consider some examples of concatenated codes with different inner and outer codes. For IRS codes, we compare the error correction capability of collaborative decoding with that of independent decoding and we show that if errors happen simultaneously for all RS codewords, more errors can be corrected using collaborative decoding. Using the results in [76], we show that the correction capability of our method based on GS algorithm is better than that of collaborative decoding based on BM algorithm. We also compare the performance of GC codes under collaborative and also independent decoding of their outer IRS codes. For all examples, we assume GC codes are BPSK modulated and sent over the AWGN channel.

Example (1): The inner code is a binary $(23, 12)$ Golay code [6]. Weight distribution of the Golay code is known. Therefore, we can estimate its ML performance using Equation (5.41). For the outer IRS code, we consider two cases: a homogeneous IRS code, IRS(63,43,43) over $GF(64)$ and a heterogeneous IRS code, IRS(63,54,43) over $GF(64)$. For the IRS codes, the error correction capability of each RS code under GS decoding along with asymptotic t_g and t_{max} are shown in table 5.1. For comparison, the error correction capabilities of collaborative (t_{max-BM}) and independent (t_{g-BM}) decoding based on BM algorithm [76] are also shown. The expressions for these correction capabilities are given as

$$t_{g-BM} = \frac{N - \overline{K}}{2} \tag{5.45}$$

$$t_{max-BM} = \frac{M}{M+1}\left(N - \widetilde{K}\right) \tag{5.46}$$

$$\widetilde{K} = \frac{1}{M}\sum_{i=1}^{M} K_i \tag{5.47}$$

From table 5.1, GS collaborative decoding of the homogeneous IRS(63,43,43) can

correct three more errors compared to the GS independent decoding. For the heterogeneous code, 6 extra errors can be corrected. Also, GS collaborative decoding can correct more than 2 extra errors compared to BM collaborative decoder for the case of homogeneous IRS code. For the heterogeneous code, more than 1 extra error can be corrected.

Now, we use Equation (5.44) to estimate lower and upper bounds for the probability of word error under GS collaborative and also independent decoding of the outer IRS code. The results are shown in Figures 5.3 and 5.4. From these figures, using collaborative GS decoder in the case of homogeneous IRS code more than 0.5 dB coding gain can be expected compared to independent GS decoding while 1.5 dB coding gain might be possible in the case of the heterogeneous IRS code. For comparison, the upper bounds for BM collaborative and independent decoding from [76] are also given. As expected, the upper bound of our collaborative method based on GS algorithm is lower then that of the collaborative method based on BM algorithm.

Table 5.1: Error correction capability of GS and BM decoding of RS codes and also collaborative and independent GS and BM decoding of IRS codes over $GF(64)$

RS(63,54)	RS(63,43)	IRS(63,54,43)	IRS(63,54,43)	IRS(63,43,43)	IRS(63,43,43)
GS Decoding	GS Decoding	GS Independent	GS Collaborative	GS Independent	GS Collaborative
5	11	5	11	11	14
BM Decoding	BM Decoding	BM Independent	BM Collaborative	BM Independent	BM Collaborative
4	10	4	≤ 9	10	≤ 13

Example (2): Here, for the inner code, we use a $(31, 16)$ binary BCH code whose weight distribution is available. For the outer code, we use two different IRS codes including heterogeneous IRS(255,239,191) and homogeneous IRS(255,191,191) both over $GF(256)$. For the IRS codes, the error correction capability of each RS code under GS decoding along with asymptotic t_g and t_{max} for each case are shown in table 5.2. For homogeneous IRS(255,191,191) collaborative interpolation decoding can correct 11 more errors compared to the independent GS decoding. For the heterogeneous case, 21 extra errors can be corrected using the collaborative decoder. The error correction capabilities of

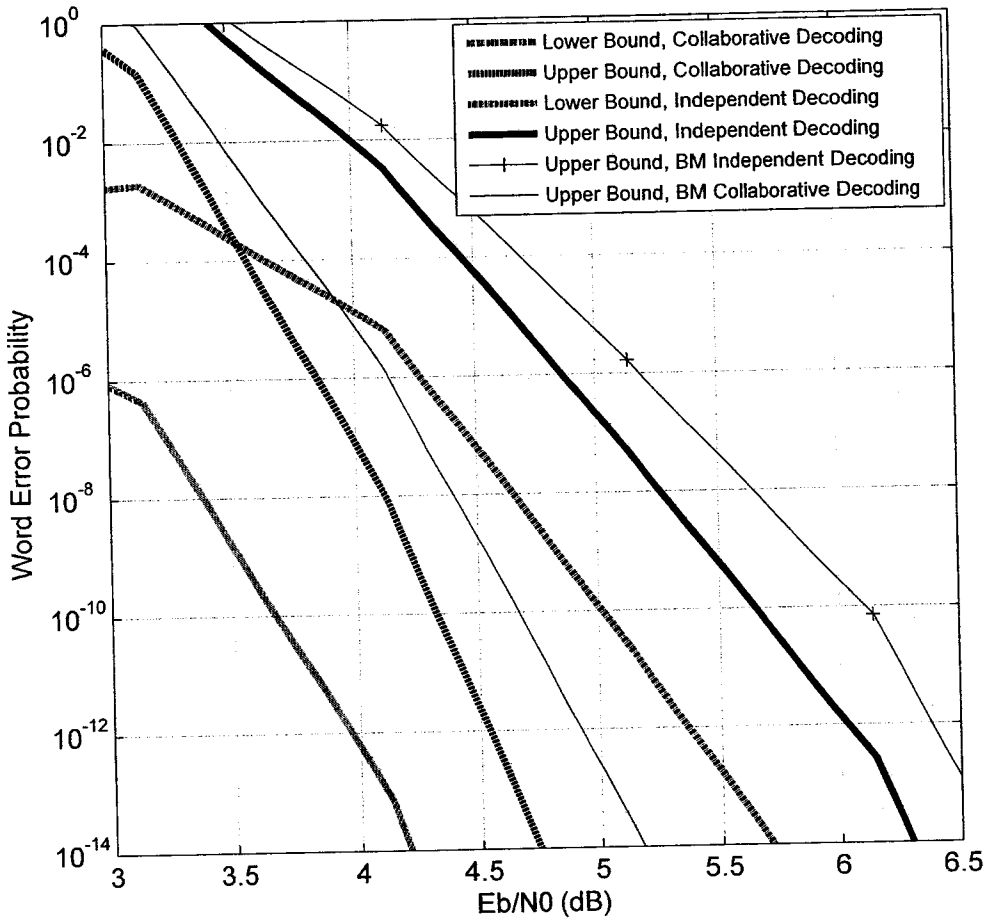

Figure 5.3: Lower and upper bounds for the probability of word error of a concatenated code (The inner code: $(23, 12)$ binary Golay code. The outer code: heterogeneous IRS(63,54,43) over $GF(64)$), under collaborative and also independent decoding of the outer IRS codes.

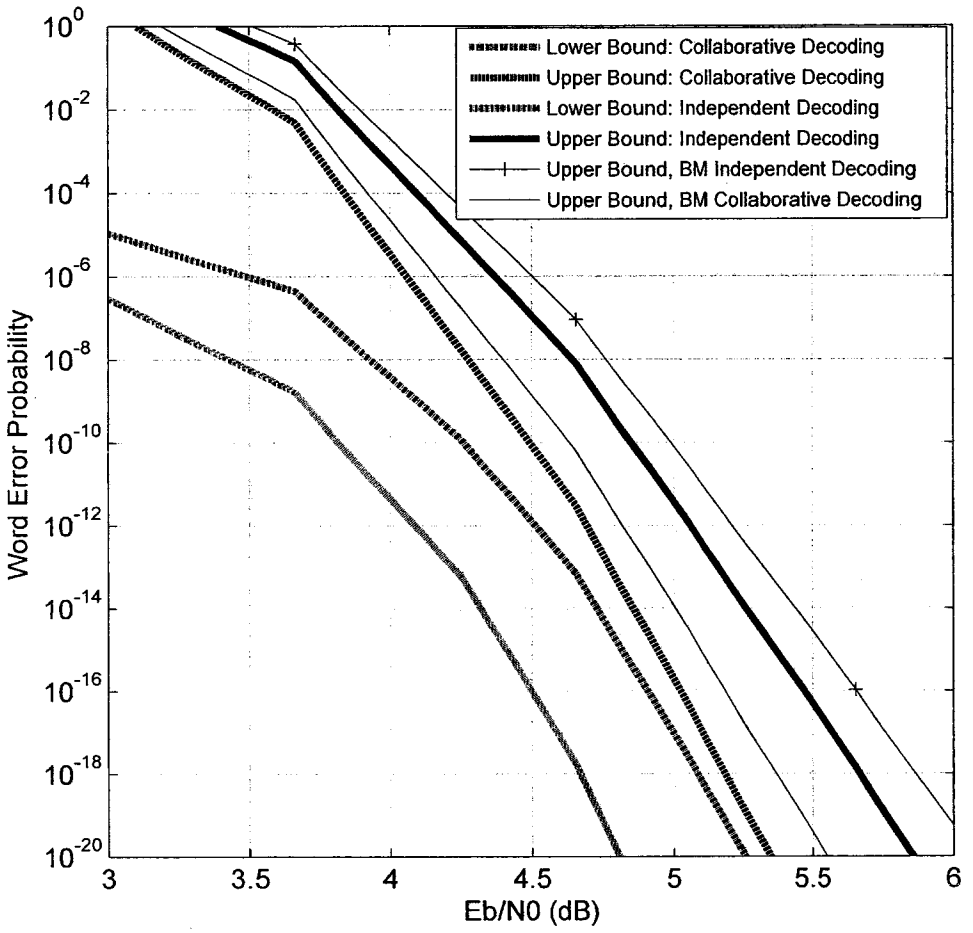

Figure 5.4: Lower and upper bounds for the probability of word error of a concatenated code (The inner code: $(23, 12)$ binary Golay code. The outer code: homogeneous IRS(63,43,43) over $GF(64)$), under collaborative and also independent decoding of the outer IRS codes.

collaborative (t_{max-BM}) and independent (t_{g-BM}) decoding based on BM algorithm [76] are also shown in table 5.2. For both homogeneous and heterogeneous cases, collaborative GS decoder can correct more than 3 extra errors compared to the collaborative BM decoder.

The lower and upper bounds for the probability of word error under collaborative and also independent decoding of the outer IRS codes are shown in Figures 5.5 and 5.6. For the case with heterogeneous IRS outer code, large coding gains are achieved using GS collaborative decoding compared to independent GS decoding. For the homogeneous case, the coding gain of less than 1 dB might be possible.

Table 5.2: Error correction capability of GS and BM decoding of RS codes and also collaborative and independent GS and BM decoding of IRS codes over $GF(256)$

RS (255,239)	RS (255,191)	IRS (255,239,191)	IRS (255,239,191)	IRS (255,191,191)	IRS (255,191,191)
GS Decoding	GS Decoding	GS Independent	GS Collaborative	GS Independent	GS Collaborative
8	34	8	29	34	45
BM Decoding	BM Decoding	BM Independent	BM Collaborative	BM Independent	BM Collaborative
8	32	8	≤ 26	32	≤ 42

5.5 Conclusion

In this chapter, based on the GS decoding method, an algorithm for collaborative decoding of heterogeneous IRS codes in the presence of simultaneous errors has been derived and analyzed. We have shown that when errors happen simultaneously for all the codewords of an IRS code, decoding all the RS codewords collaboratively can provide an error correction capability larger than the decoding radius of the GS algorithm for the RS code with the largest dimension. As an example, we have analyzed the performance of concatenated codes using IRS codes as their outer codes where a decoding error at the output of the inner decoder would affect one whole column of the IRS codeword (similar to burst errors). We have derived upper and lower bounds for the word error probability of concatenated codes

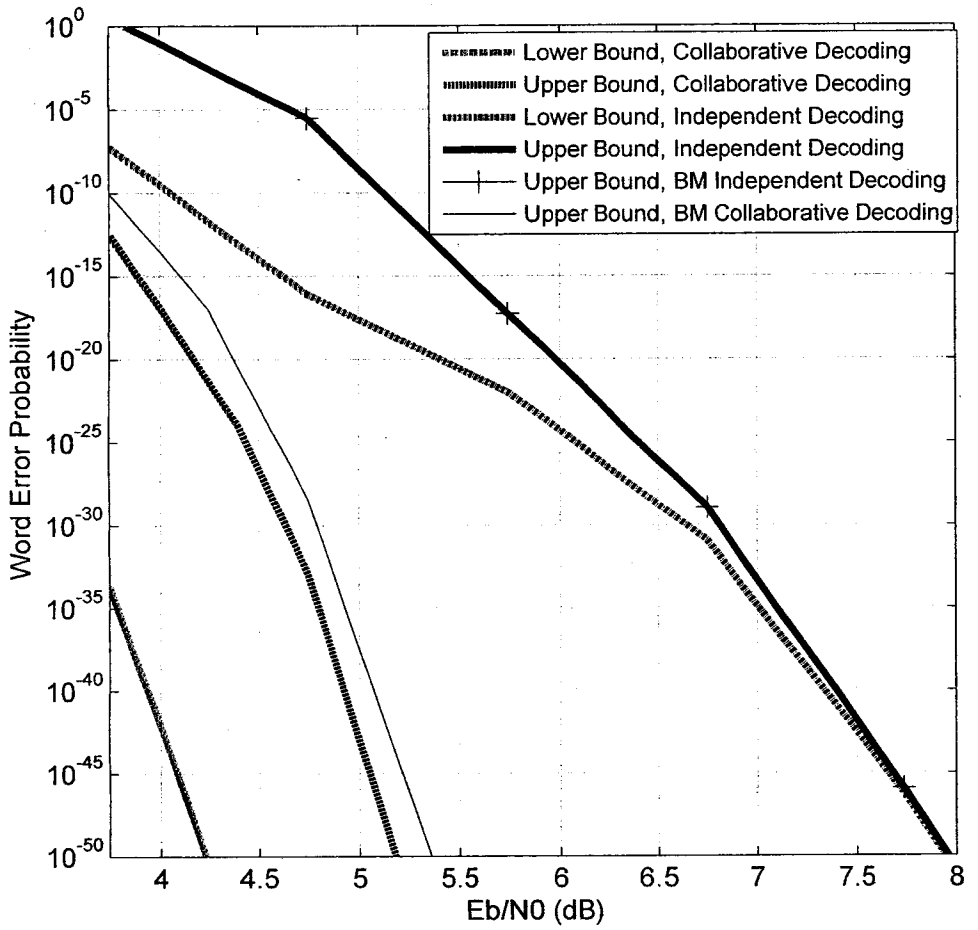

Figure 5.5: Lower and upper bounds for the probability of word error of a concatenated code (The inner code: $(31, 16)$ binary BCH code. The outer code: heterogeneous IRS(255,239,191) over $GF(256)$), under collaborative and also independent decoding of the outer IRS codes.

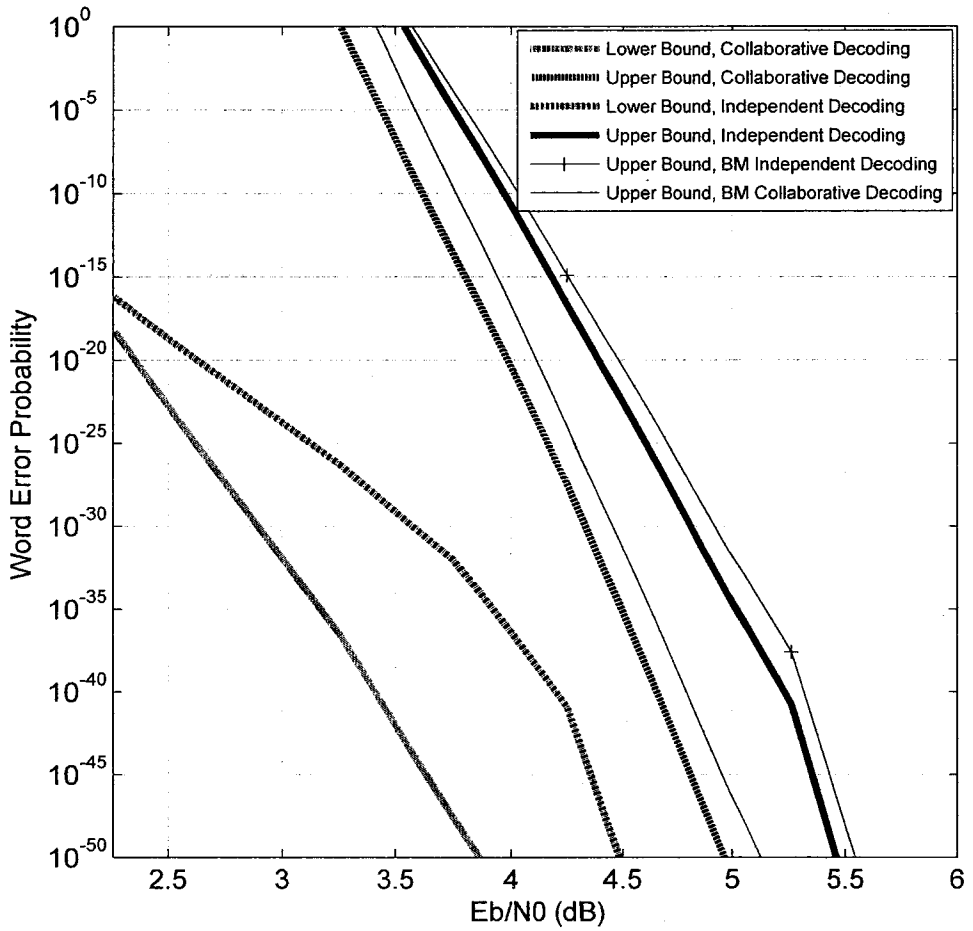

Figure 5.6: Lower and upper bounds for the probability of word error of a concatenated code (The inner code: $(31, 16)$ binary BCH code. The outer code: homogeneous IRS(255,191,191) over $GF(256)$), under collaborative and also independent decoding of the outer IRS codes.

over AWGN channel with BPSK modulation for both cases of independent and collaborative decoding of the outer IRS codes. We have shown through numerical results that using collaborative decoding, the word error probability is lower compared to the case of independent decoding. We have also compared our algorithm with the collaborative decoder based on the BM method and realized our method can correct more synchronous errors.

Chapter 6

Conclusions and Future Works

In this thesis, we have investigated soft decision decoding of Reed-Solomon codes. Despite their wide areas of applications, efficient soft decoding of RS codes still remains open. In recent years lots of improvement has been made in decoding RS codes. For example, Guruswami and Sudan (GS) [11] have proposed a new hard decision algorithm with higher decoding capability compared to the traditional error-correction capability, half the minimum distance. Koetter and Vardy (KV) [29] have further improved the GS algorithm by utilizing the soft information from the channel. Also, several iterative decoders based on belief propagation have been introduced for RS codes [12] [13] [64].

In order to find efficient decoding techniques for RS codes, we have investigated three classes of complexity-reducing methods including sphere decoding (SD), belief propagation (BP) decoding and interpolation-based decoding. In Chapter 3, the concept of SD has been used to implement a soft decision decoder for RS codes. In Chapter 4, we have introduced two BP based algorithms for efficient iterative decoding of RS codes. Finally, in Chapter 5, the basis of the GS interpolation decoding algorithm has been used for collaborative decoding of interleaved Reed-Solomon (IRS) codes.

Here, we present the major contributions of this thesis and suggestions for future research.

6.1 Major Contributions

First, we have presented a novel soft decision decoding method for RS codes based on sphere decoding. The main idea of this method is to look for lattice points inside a sphere centered at the received signal that are also valid codewords of an (N, K) RS code. The sphere decoder algorithm first selects the K most reliable code symbols that fall inside the sphere. The acceptable values for each of these K code symbols are determined based on the ordered set of most probable transmitted symbols. Each time K code symbols are selected using the sphere decoder, they are used to find the rest of RS symbols. If the resulting codeword is within the search radius, it is saved as a possible transmitted codeword. Two kinds of ordering is used to improve the speed of sphere decoding. Firstly, we try to find the K most reliable code symbols and secondly, for each of them we use an ordered set of most probable transmitted symbols. This method provides considerable coding gain compared to the hard decision decoding with a moderate increase in complexity.

Then, based on BP decoding, we have proposed two new iterative soft decision decoding methods for RS codes. A low density extended binary parity check matrix has been used for BP decoding. The first proposed method uses the cyclic structure of RS codes. We apply the BP algorithm on cyclically shifted versions of the received symbols. For each of them, the geometry of the factor graph will change and deterministic errors can be avoided. The performance of this method is considerably better than hard decision decoding. The performance is also superior to the KV method and the ADP method. The second method is based on information correction. We determine least reliable bits and by changing their channel information, the convergence of the decoder is improved. Compared to the first method, this method is less complex but its performance is not as good.

Finally, based on GS decoding, an algorithm for collaborative interpolation decoding of heterogeneous interleaved Reed-Solomon (IRS) codes has been derived. All the codewords of the interleaved scheme are decoded at the same time using multivariate interpolation. In the presence of burst errors, the error correction capability of this algorithm is larger than that of independent decoding of each codeword using the standard GS method.

We have used IRS codes as the outer codes in generalized concatenated (GC) codes. We have found upper and lower bounds for the word error probability of GC codes over AWGN channel with BPSK modulation for both cases of independent and collaborative decoding of the outer IRS codes. Collaborative decoding provides considerable coding gain compared to independent decoding.

6.2 Future Works

In this section, based on the research completed in the thesis, we present potential future goals:

- In the proposed soft decoding technique based on sphere decoding, two types of ordering have been used to reduce the complexity. However, the main complexity issue is the Gaussian elimination of the generator matrix. One may look for possible alternative methods to perform sphere decoding of RS codes without the need for Gaussian elimination.

- For BP based decoding algorithms, at the end of each iteration, a BM algorithm has been used to perform hard decision decoding on the updated reliabilities. One can consider other choices of decoding algorithms such as the GS hard decision decoding, the KV soft decision decoding and so on. As another future goal, one may look for more suitable binary parity check matrices with lower number of short cycles.

- Our proposed method for collaborative interpolation decoding of IRS codes is based on the GS hard decision decoding algorithm. As a future goal, it is of great value to extend the collaborative method to soft decision decoding using the approach of KV. This way, it might be possible to outperform the standard KV method in the presence of burst errors. One may also look into efficient implementation of multivariate polynomial interpolation and factorization. The factorization process has only been

explained for the case of IRS codes with two codewords. One may investigate the problem for larger IRS codes with more than two RS codewords.

In order to find upper and lower bounds on the performance of collaborative decoding of outer IRS codes in a concatenated design, we have used ML lower and upper bounds on the performance of the inner code. For the ML lower bound, we have used Shannon's sphere packing bound (SPB) [81] and Seguin's L_2 bound [82]. The first one is tight at low SNR's and the second one is tight at high SNR's. In [83], an efficient algorithmic ML lower bound has been proposed which is tighter than L_2 bound and its computation time is shorter. One may use this bound in order to find tighter upper and lower bounds for the performance of the concatenated code.

- RS codes have been used in cooperative communications. One may replace the standard BM decoding method with the proposed decoding techniques and investigate the performance gains that are provided. Since the proposed methods have higher complexity compared to the BM method, the trade-off between the performance and the complexity should be taken into account.

Bibliography

[1] I. S. Reed and G. Solomon, "Polynomial codes over certain finite fields," *Journal of the Society for Industrial and Applied Mathematics*, vol. 8, no. 2, pp. 300–304, 1960.

[2] S. Arimoto, "Encoding and decoding of p-ary group codes and the correction system," *Information Processing in Japan*, vol. 2, pp. 321–325, 1961.

[3] S. Cho, G. Lee, B. Bae, K. Yang, C. Ahn, S. Lee, and C. Ahn, "System and services of terrestrial digital multimedia broadcasting (T-DMB)," *IEEE Transactions on Broadcasting*, vol. 53, no. 1 Part 2, pp. 171–178, 2007.

[4] M. Kornfeld, "DVB-H-the emerging standard for mobile data communication," in *IEEE International Symposium on Consumer Electronics*, Reading, United Kingdom, September 2004, pp. 193–198.

[5] G. Forney, *Concatenated Codes*. MIT Press, Cambridge, USA, 1966.

[6] S. Lin and D. Costello, *Error Control Coding: Fundamentals and Applications*. Prentice-Hall, Englewood Cliffs, NJ, 1983.

[7] D. Knisely, S. Kumar, S. Laha, and S. Nanda, "Evolution of wireless data services: IS-95 to CDMA 2000," *IEEE Communications Magazine*, vol. 36, no. 10, pp. 140–149, 1998.

[8] J. Ulvr and A. Kho, "Mail piece bar code having a data content identifier," US Patent, 11, 1997.

[9] R. Gallager, "Low-density parity-check codes," *IEEE Transactions on Information Theory*, vol. 8, no. 1, pp. 21–28, 1962.

[10] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: turbo-codes," *IEEE Transactions on Communications*, vol. 44, no. 10, pp. 1261–1271, 1996.

[11] V. Guruswami and M. Sudan, "Improved decoding of Reed-Solomon and algebraic-geometry codes," *IEEE Transactions on Information Theory*, vol. 45, no. 6, pp. 1757–1767, 1999.

[12] J. Jiang and K. Narayanan, "Iterative soft input soft output decoding of Reed-Solomon codes by adapting the parity check matrix," *IEEE Trans. Inform. Theory*, vol. 52, no. 8, pp. 3746–3756, 2006.

[13] M. El-Khamy and R. McEliece, "Iterative algebraic soft-decision list decoding of Reed-Solomon codes," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 3, pp. 481–490, March 2006.

[14] U. Fincke and M. Pohst, "Improved methods for calculating vectors of short length in a lattice, including a complexity analysis," *Mathematics of Computation*, vol. 44, no. 170, pp. 463–471, 1985.

[15] R. Singleton, "Maximum distance q-nary codes," *IEEE Transactions on Information Theory*, vol. 10, no. 2, pp. 116–118, 1964.

[16] E. Berlekamp, *Algebraic coding theory*. McGraw-Hill, New York, NY, 1968.

[17] J. Massey, "Shift-register synthesis and BCH decoding," *IEEE Transactions on Information Theory*, vol. 15, no. 1, pp. 122–127, 1969.

[18] V. Guruswami and A. Vardy, "Maximum-likelihood decoding of Reed-Solomon codes is NP-hard," in *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, Vancouver, Canada, January 2005, pp. 470–478.

[19] Y. Sugiyama, M. Kasahara, S. Hirasawa, and T. Namekawa, "A method for solving key equation for decoding Goppa codes," *Information and Control*, vol. 27, no. 1, pp. 87–99, 1975.

[20] L. Welch and E. Berlekamp, "Error correction for algebraic block codes," US Patent 4,633,470, December 30, 1986.

[21] G. Forney Jr, "Generalized minimum distance decoding," *IEEE Transactions on Information Theory*, vol. 12, no. 2, pp. 125–131, 1966.

[22] D. Chase, "A Class of algorithms for decoding block codes with channel measurement information," *IEEE Transactions on Information Theory*, vol. 18, no. 1, pp. 170–182, 1972.

[23] F. Alajaji, N. Phamdo, and T. Fuja, "Channel codes that exploit the residual redundancy in CELP-encoded speech," *IEEE Transactions on Speech and Audio Processing*, vol. 4, no. 5, pp. 325–336, 1996.

[24] H. Tang, Y. Liu, M. Fossorier, and S. Lin, "On combining Chase-2 and GMD decoding algorithms for nonbinaryblock codes," *IEEE Communications Letters*, vol. 5, no. 5, pp. 209–211, 2001.

[25] A. Vardy and Y. Beery, "Bit-level soft-decision decoding of Reed-Solomon codes," *IEEE Transactions on Communications*, vol. 39, no. 3, 1991.

[26] M. Fossorier and S. Lin, "Soft-decision decoding of linear block codes based on ordered statistics," *IEEE Transactions on Information Theory*, vol. 41, no. 5, pp. 1379–1396, 1995.

[27] Y. Wu, R. Koetter, and C. Hadjicostis, "Soft-decision decoding of linear block codes using preprocessing," in *Proceedings of International Symposium on Information Theory (ISIT)*, Chicago, USA, 2004, p. 259.

[28] M. Fossorier and A. Valembois, "Reliability-based decoding of Reed-Solomon codes using their binary image," *IEEE Communications Letters*, vol. 8, no. 7, pp. 452–454, 2004.

[29] R. Koetter and A. Vardy, "Algebraic soft-decision decoding of Reed-Solomon codes," *IEEE Transactions on Information Theory*, vol. 49, no. 11, pp. 2809–2825, 2003.

[30] F. Parvaresh and A. Vardy, "Multiplicity assignments for algebraic soft-decoding of reed-solomon codes," in *Proceedings. IEEE International Symposium on Information Theory (ISIT)*, Yokohama, Japan, 2003, pp. 205–205.

[31] M. El-Khamy and R. McEliece, "Interpolation multiplicity assignment algorithms for algebraic soft-decision decoding of Reed-Solomon codes," *AMS-DIMACS volume on Algebraic Coding Theory and Information Theory*, vol. 68, pp. 99–120, 2005.

[32] B. Hassibi and H. Vikalo, "On the sphere-decoding algorithm I. Expected complexity," *IEEE Transactions on Signal Processing*, vol. 53, no. 8, pp. 2806–2818, 2005.

[33] H. Vikalo and B. Hassibi, "On joint detection and decoding of linear block codes on Gaussian vector channels," *IEEE Transactions on Signal Processing*, vol. 54, no. 9, p. 3330, 2006.

[34] M. El-Khamy, H. Vikalo, and B. Hassibi, "Bounds on the performance of sphere decoding of linear block codes," in *IEEE Information Theory Workshop on Coding and Complexity (ITW2005)*, Rotorua, New Zealand, 2005.

[35] B. Kamali and A. Aghvami, "Belief propagation decoding of Reed-Solomon codes; a bit-level soft decision decoding algorithm," *IEEE Transactions on Broadcasting*, vol. 51, no. 1, pp. 106–113, 2005.

[36] S. Sankaranarayanan and B. Vasic, "Iterative decoding of linear block codes: A parity-check orthogonalization approach," *IEEE Transactions on Information Theory*, vol. 51, no. 9, pp. 3347–3353, 2005.

[37] J. Jiang and K. Narayanan, "Iterative soft decoding of Reed-Solomon codes," *IEEE Communications Letters*, vol. 8, no. 4, pp. 244–246, 2004.

[38] F. Kschischang, B. Frey, and H. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Transactions on information theory*, vol. 47, no. 2, pp. 498–519, 2001.

[39] T. Halford and K. Chugg, "Random redundant iterative soft-in soft-out decoding," *IEEE Transactions on Communications*, vol. 56, no. 4, p. 513, 2008.

[40] I. Dimnik and Y. Beery, "Improved random redundant iterative HDPC decoding," *IEEE Transactions on Communications*, vol. 57, no. 7, pp. 1982–1985, 2009.

[41] J. Knudsen, C. Riera, L. Danielsen, M. Parker, and E. Rosnes, "Iterative Decoding on Multiple Tanner Graphs Using Random Edge Local Complementation," in *Proceedings of International Symposium on Information Theory (ISIT)*, Seoul, Korea, June 2009.

[42] A. Kothiyal, O. Takeshita, W. Jin, and M. Fossorier, "Iterative reliability-based decoding of linear block codes with adaptive belief propagation," *IEEE communications letters*, vol. 9, no. 12, pp. 1067–1069, 2005.

[43] C. Wang, Y. Hsieh, and H. Kuo, "Bilateral exchange of soft-information for iterative reliability-based decoding with adaptive belief propagation," *IEEE Communications Letters*, vol. 13, no. 9, pp. 682–684, 2009.

[44] T. Hehn, J. Huber, O. Milenkovic, and S. Laendner, "Multiple-bases belief-propagation decoding of high-density cyclic codes," *IEEE Transactions on Communications*, vol. 58, no. 1, pp. 1–8, 2010.

[45] H. Vikalo, B. Hassibi, and T. Kailath, "Iterative decoding for MIMO channels via modified sphere decoding," *IEEE Transactions on Wireless Communications*, vol. 3, no. 6, pp. 2299–2311, 2004.

[46] K. Wong and A. Paulraj, "On the decoding order of MIMO maximum-likelihood sphere decoder: linear and non-linear receivers," in *Proceedings of IEEE 59th Vehicular Technology Conference (VTC04*, vol. 2, Milan, Italy, May 2004.

[47] L. L. S. A. Grötschel, M., *Geometric algorithms and combinatorial optimization, Second edition.* Springer-Verlag, Berlin, 1993.

[48] G. Golub and W. Kahan, "Calculating the singular values and pseudo-inverse of a matrix," *Journal of the Society for Industrial and Applied Mathematics: Series B, Numerical Analysis*, vol. 2, no. 2, pp. 205–224, 1965.

[49] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE transactions on Information Theory*, vol. 13, no. 2, pp. 260–269, 1967.

[50] H. Vikalo and B. Hassibi, "On the sphere-decoding algorithm II. Generalizations, second-order statistics, and applications to communications," *IEEE transactions on signal processing*, vol. 53, no. 8, p. 2819, 2005.

[51] F. Jensen, *An Introduction to Bayesian Networks.* Springer-Verlag, Secaucus, NJ, USA, 1996.

[52] L. Baum and T. Petrie, "Statistical inference for probabilistic functions of finite state Markov chains," *The Annals of Mathematical Statistics*, pp. 1554–1563, 1966.

[53] R. Tanner, "A recursive approach to low complexity codes," *IEEE Transactions on Information Theory*, vol. 27, no. 5, pp. 533–547, 1981.

[54] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Transactions on Information Theory*, vol. 20, no. 2, pp. 284–287, 1974.

[55] J. Pearl and G. Shafer, *Probabilistic reasoning in intelligent systems: networks of plausible inference.* Morgan Kaufmann, San Mateo, CA, 1988.

[56] R. Horn and C. Johnson, *Matrix Analysis.* New York: Cambridge University Press, 1985.

[57] R. Koetter, *On algebraic decoding of algebraic-geometric and cyclic codes.* Ph.D. Thesis, Department of Electrical Engineering, Linkoping University, Sweden, 1996.

[58] R. Roth and G. Ruckenstein, "Efficient decoding of Reed-Solomon codes beyond half the minimum distance," *IEEE Transactions on Information Theory*, vol. 46, no. 1, pp. 246–257, 2000.

[59] R. Sedgewick, *Algorithms in C.* New York: Addison-Wesley, 1990.

[60] F. MacWilliams and N. Sloane, *The theory of error-correcting codes.* North-Holland Amsterdam, 1988.

[61] W. Gross, F. Kschischang, R. Koetter, and P. Gulak, "Simulation results for algebraic soft-decision decoding of Reed-Solomon codes," in *Proceedings of the 21st Biennial Symposium on Communications*, Kingston, Canada, 2002, pp. 356–360.

[62] R. McEliece, "The Guruswami-Sudan decoding algorithm for Reed-Solomon codes," *IPN Progress Report*, pp. 42–153, 2003.

[63] C. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, pp. 379–423, July 1948.

[64] J. Bellorado, *Low-complexity soft decoding algorithms for Reed-Solomon codes.* Ph.D. Thesis, Department of Electrical Engineering, Harvard University, USA, 2006.

[65] N. Varnica and M. Fossorier, "Belief-propagation with information correction: improved near maximum-likelihood decoding of low-density parity-check codes," in *Proceedings. International Symposium on Information Theory (ISIT)*, Chicago, USA, 2004, pp. 343–343.

[66] A. Shokrollahi, "LDPC codes: An introduction," *Coding, cryptography and combinatorics*, vol. 23, pp. 85–110, 2004.

[67] T. Richardson and R. Urbanke, *Modern coding theory*. Cambridge University Press, 2008.

[68] R. Lucas, M. Bossert, and M. Breitbach, "On iterative soft-decision decoding of linear binary block codes and product codes," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 2, pp. 276–296, 1998.

[69] C. Retter, "An average weight-distance enumerator for binary expansions of Reed-Solomon codes," *IEEE Transactions on Information Theory*, vol. 48, no. 5, pp. 1195–1200, 2002.

[70] D. Divsalar, "A simple tight bound on error probability of block codes with application to turbo codes," *TMO Progress Report*, vol. 19, pp. 42–139, 1999.

[71] V. Krachkovsky and Y. Lee, "Decoding of parallel Reed-Solomon codes with applications to product and concatenated codes," in *Proceedings. IEEE International Symposium on Information Theory (ISIT)*, Cambridge, USA, 1998.

[72] D. Bleichenbacher, A. Kiayias, and M. Yung, "Decoding interleaved Reed–Solomon codes over noisy channels," *Theoretical Computer Science*, vol. 379, no. 3, pp. 348–360, 2007.

[73] A. Brown, L. Minder, A. Shokrollahi, S. de Math, and E. de Lausanne, "Probabilistic decoding of interleaved RS-codes on the q-ary symmetric channel," in *Proceedings. International Symposium on Information Theory (ISIT)*, Chicago, USA, 2004, p. 326.

[74] J. Justesen, C. Thommesen, and T. Hoholdt, "Decoding of concatenated codes with interleaved outer codes," in *Proceedings. International Symposium on Information Theory (ISIT)*, Chicago, USA, 2004, p. 328.

[75] È. Blokh and V. Zyablov, "Coding of generalized concatenated codes," *Problems in Information Transmission (Problemy Peredachi Informatsii)*, vol. 10, no. 3, pp. 45–50, 1974.

[76] G. Schmidt, V. Sidorenko, and M. Bossert, "Collaborative decoding of interleaved Reed–Solomon codes and concatenated code designs," *IEEE Transactions on Information Theory*, vol. 55, no. 7, pp. 2991–3012, 2009.

[77] F. Parvaresh and A. Vardy, "Multivariate interpolation decoding beyond the Guruswami-Sudan radius," in *Proceedings of the 42nd Allerton Conference on Communication, Control and Computing*, Monticello, Illinois, USA, 2004.

[78] D. Cox, J. Little, and D. O'Shea, *Ideals, varieties, and algorithms: an introduction to computational algebraic geometry and commutative algebra.* Springer Verlag, New York, USA, 1997.

[79] J. Ma, P. Trifonov, and A. Vardy, "Divide-and-conquer interpolation for list decoding of Reed-Solomon codes," in *Proceedings. IEEE International Symposium on Information Theory (ISIT)*, Chicago, USA, 2004, p. 387.

[80] G. Poltyrev, "Bounds on the decoding error probability of binary linear codes viatheir spectra," *IEEE Transactions on Information Theory*, vol. 40, no. 4, pp. 1284–1292, 1994.

[81] C. Shannon, "Probability of error for optimal codes in a Gaussian channel," *The Bell System Technical Journal*, vol. 38, no. 3, pp. 611–656, May 1959.

[82] G. Seguin, "A lower bound on the error probability for signals in white Gaussian noise," *IEEE Transactions on Information Theory*, vol. 44, no. 7, pp. 3168–3175, 1998.

[83] F. Behnamfar, F. Alajaji, and T. Linder, "An efficient algorithmic lower bound for the error rate of linear block codes," *IEEE Transactions on Communications*, vol. 55, no. 6, pp. 1093–1098, 2007.