國立交通大學

電子工程學系　電子研究所碩士班

碩 士 論 文

利用 K-Best 演算法之軟性里德索羅門解碼器

Soft RS Decoder Based on K-Best Algorithm

學生：鄭晶今

指導教授：張錫嘉教授

中華民國九十八年十月

利用 K-Best 演算法之軟性里德索羅門解碼器

Soft RS Decoder Based on K-Best Algorithm

研 究 生：鄭晶今　　Student：Ching-Chin Cheng

指導教授：張錫嘉教授　　Advisor：Hsie-Chia Chang

國 立 交 通 大 學

電子工程學系 電子研究所 碩士班

碩 士 論 文

A Thesis

Submitted to Department of Electronics Engineering & Institute Electronics

College of Electrical and Computer Engineering

National Chiao Tung University

In Partial Fulfillment of the Requirements

for the Degree of

Master of Science

in

Electronics Engineering

Octorber 2009

Hsinchu, Taiwan, Republic of China

中華民國九十八年十月

利用 K-Best 演算法之軟性里德索羅門解碼器

學生：鄭晶今　　指導教授：張錫嘉 教授

國立交通大學

電子工程學系 電子研究所碩士班

摘　要

本論文提出了利用 K-Best 演算法之軟性里德索羅門解碼器。這個方法主要可以分成三個部分：前置處理、候選人選擇機制、消去解碼。

在前置處理的部分，我們根據接收到的軟性資訊，給予每一個接收到的符號一個可信度。候選人選擇機制的部分，利用可信度的資訊以及獨立的特性去產生出可能的候選人組合。因為可能的組合有很多，所以利用 K-Best 演算法的限制來降低運算量。在第三個部分，消去解碼被用來解可能的候選人組合。為了找到合理的候選人數量，需要考慮到兩個相抗衡的項：性能和複雜度。

模擬的結果顯示，在里德索羅門碼(15,11)的狀況，所提出的演算法在字碼錯誤率(CER)為 10^{-4}時，其效能比硬性的BerleKamp-Messy(HD-BM)演算法好 2.4dB，並且比Kotter-Vardy(KV)好 1.3dB。與KV演算法比較時，運算複雜度至少降低了 41.7%。而與動態可靠度傳播-代數軟性選擇(ABP-ASD)演算法比較，當字碼錯誤率等於 10^{-4}還有 0.3dB的效能差距，但是複雜度降低了至少 75.5%。對於里德索羅門碼(31,25)，提出的方法在字碼錯誤率等於 10^{-4}時，比HD-BM演算法好 1.4dB並且比KV演算法好 0.55dB。複雜度部分，則比KV降了至少 61.6%。但是對於ABP-ASD演算法尚有 1.25dB的效能差距，但是複雜度至少降低了 97%。

Soft RS Decoder Based on K-Best Algorithm

Student：Ching-Chin Cheng Advisor：Dr. Hsie-Chia Chang

Department of Electronics Engineering

Institute of Electronics

National Chiao Tung University

Abstract

In this thesis, the soft Reed-Solomon (RS) decoder based on K-Best algorithm with constraint is proposed. The proposed algorithm consists of three parts, pre-processing, candidate selecting and erasure only decoding.

In pre-processing, the reliability is assigned to the received symbols according to the soft information from the channel. The candidate selecting uses the reliability information and the independent property to generate the possible candidate sets. Since the number of possible candidate sets is large, the K-Best algorithm is utilized to reduce the computation number. In the third part, erasure-only decoding is used to decode possible candidate sets. To provide a reasonable number of candidates, the trade-off between performance and complexity is considered.

Simulation results show that for RS (15,11), the proposed algorithm outperforms the hard-decision Berlekamp-Messy (HD-BM) algorithm by 2.4dB and the Kotter-Vardy(KV) algorithm by 1.3dB at codeword error rate (CER) of 10^{-4}. As compared with the KV algorithm the complexity reduction is at least 41.7%. And comparing with the adaptive belief propagation-algebraic soft decision (ABP-ASD) algorithm, there is 0.3dB performance gap at CER of 10^{-4}. However, the complexity

reduction is at least 75.5%. For RS (31,25), it outperforms the HD-BM algorithm by 1.4dB and the KV algorithm by 0.55dB at CER of 10^{-4}. The complexity reduction is at least 61.6% as compared to the KV algorithm. There is 1.25dB performance gap between ABP-ASD and proposed method at CER of 10^{-4}. However, the complexity reduction is at least 97%.

誌　謝

兩年的時間過得很快，研究所的生活既緊湊又豐碩，在這兩年學到許多做學問的方法與態度。最要感謝張錫嘉教授給我自由的研究風氣以及舒適的研究環境，可以讓我在這兩年的研究中有所成長並得到一些研究的成果。再來要感謝Ocean group裡的學長姊，尤其是建青學長以及彥欽學姊，每每在我遇到困難的時候可以給予建議。感謝Oasis實驗室的學長姊、同學及學弟妹可以適時給予我提供協助。另外要感謝口試委員胡大湘教授，翁詠祿教授，蘇育德教授給予我的研究建議與指導。

最後要我感謝我的家人，爸爸、媽媽、哥哥讓我無後顧之憂的完成我的學業。感謝修齊一路以來的陪伴與指導，每次的討論都讓我有更多的成長。要感謝的人太多了，就謝謝上天吧!

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Reed-Solomon (RS) codes are one of the most widely used error correcting code for wireless communication and storage systems. The current issue about the RS code is to find an efficient way to improve the performance of traditional algebraic hard decision Berlekamp-Messey(HD-BM) [1] algorithm. HD-BM algorithm does not use channel reliability information, this causes significant performance loss. Soft decision decoding algorithm takes advantage of the channel value, and is believed to have $2 \sim 3$ dB performance gain in additive white Gaussian noise (AWGN) channel. Guruswami and Vardy [2] have shown that Maximum-likelihood (ML) decoding of RS code is NP-hard. It remains an open area to find a soft decision code with moderate complexity with near ML performance.

Recently, the algebraic soft decision decoding (ASD) developed by Koetter and Vardy (KV) [3] use a list decoding technique outperforms Guruswami and Sudan (GS) [4] hard decision decoding method. On the other hand, Jiang and Narayanan (JN) [5] use adaptive parity check matrix and belief propagation to execute soft decision decoding based on iterative process. In [6], El-Khamy and McEliece combined JN and KV algorithm. They used belief propagation method to improve the reliability of the symbols and then fed these reliability information into a algebraic soft decision(KV) decoder.

There are several soft decision decoding algorithm based on the reliability-based decoding. The order statistics decoding algorithm [7] sorts the received bits with respect

to their reliability, then the reprocessing step is designed to improve the hard decision codeword until a desired error performance is achieved. Other reliability-based decoding such as the generalized minimum distance (GMD) decoding algorithm [8], Chase algorithm [9] and a hybrid of chase and GMD algorithms [10] use reliability information to enhance HD-BM algorithm. Sequential algorithm, or M-algorithm(K-Best) algorithm has been presented in [11], [12] offer a good deal in complexity reduction at the cost of some loss in performance. However, the previous paper only discuss the case for binary linear block codes or convolutional codes. Non-binary codes such as RS code is still open for sequential algorithm.

In this thesis, we develop a K-Best algorithm based on reliability-based scheme for soft RS decoding. Reliability-based decoding is based on reordering the received symbols according to their reliability measure. Hard decision reliability decoding use the k reliable symbols as information sequence and use the reordered generator matrix to re-encode the possible codeword. However, we did not do the re-encode process in this thesis. We summarize the decoding steps as follows:

1) For k reliable symbols, the possible candidates for each symbol are generated. We do not execute re-encoding process while use the property of the k reliable independent symbols to execute tree-oriented operation.

2) K-Best algorithm is introduced to collect the first k possible reliable symbols of K combination set. A proper constraint is also set up to reduce the computation complexity.

3) An erasure-only decoding is used to generate the rest of N-k unreliable symbols.

4) Combine k reliable symbols with N-k unreliable symbols as combination set. Each combination sets is calculated its distance metric between the received sequence. The one with minimum distance is regarded as the transmitted codeword.

1.2 Thesis organization

The rest of this thesis is organized as follows. Chapter 2 introduces the background of RS codes, and some soft decision decoding techniques. The proposed schemes, K-Best algorithm with constraint are presented in chapter 3. To accelerate the decoding speed, description for the parallel K-Best is in this section as well. The simulation results are

shown in chapter 4. The computation complexities of our K-Best algorithm are compared with popular algebraic soft decision decoding RS code is also in this chapter. Finally, a summary concluding our work is given in chapter 5. The techniques used in this thesis are all in chapter 6, including simplified cost function, re-encode decoding based on reliability ordering, sphere decoding and K-Best algorithm, and erasure-only decoding.

Chapter 2

Soft decoding of RS codes

Reed-Solomon(RS) codes were first introduced by Reed and Solomon in 1960 [13]. RS codes can be viewed as the symbol level cyclic non-binary BCH codes [14] whose symbols are in $GF(2^q)$. The (N,k) RS code is with N symbols consisting of k messages in $GF(2^q)$. It is a maximum distance separable code with minimum distance of (N-k+1). The decoding region is in $2v + e < d_{min}$, where e is the number of erasures. The decoding complexity is usually in the $O(N^2)$. In this chapter, The system model of RS codes is introduced in section 2.1. The RS code is then reviewed in section 2.2. For soft decision RS codes, symbol-level algebraic soft decision decoding such as GMD and Chase algorithm is described in section 2.3 and 2.4. Algebraic soft decision decoding algorithm(KV) is introduced in section 2.5. Iterative decoding of RS codes is shown in section 2.6.

2.1 System model of RS code

We use a systematic (N, k) Reed-Solomon(RS) code. The items in RS code are elements of Galois Field $GF(2^q)$ where $N = 2^q - 1$. The operations in the process must be operated over the Galois Field $GF(2^q)$. Let $\mathbf{B}$ be the information sequence of length k. The information sequence $\mathbf{B}$ multiply with RS generator matrix $\mathbf{G}$ to obtain the codeword sequence $\mathbf{C}$ of length N.

$$\mathbf{C} = \mathbf{B} \times \mathbf{G} \tag{2.1}$$

$\mathbf{C} = (C_0, C_1, C_2, \ldots, C_{N-1})$ is a codeword which its binary extension is $\mathbf{c} = (c_0, c_1, c_2, \ldots, c_{Nq-1})$.

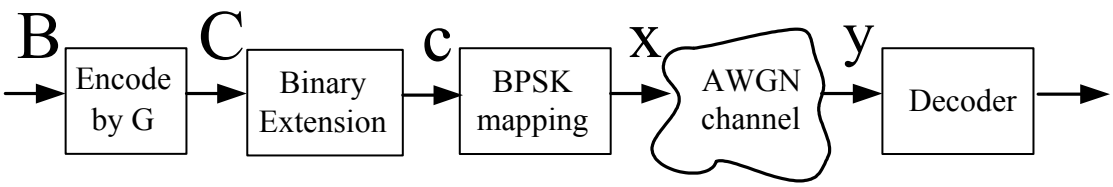


Figure 2.1: Channel model of RS code

Then **c** is BPSK modulated from the code symbols to the constellation symbols.

$$x_j = f(c_j) = (-1)^{c_j}, \mathbf{x} = (x_0, x_1, x_2, \ldots, x_{Nq-1}) \tag{2.2}$$

Then the modulated signal **x** is transmitted across the AWGN channel. The received sequences $\mathbf{y} = (y_0, y_1, y_2, \ldots, y_{Nq-1})$ are soft values.

$$\mathbf{y} = \mathbf{x} + \mathbf{w} \tag{2.3}$$

Where **w** is the independent noise value Gaussian random variables with zero mean and variance $N_0/2$. The system model of RS code is shown in Fig. 2.1.

2.2 Basic properties of Reed Solomon codes

The (N,k) RS code with bases 2 is defined over $GF(2^q)$, where $N = 2^q - 1$. The generator polynomial is used to encode the RS code. The generator polynomial g(x) of RS code which correct t or fewer error can be described by the minimum polynomials of α,α^2,α^3,...,and α^{2t}, and α is a primitive element in $GF(q^m)$. Therefore the generator polynomial has the following form

$$g(x) = \prod_{i=1}^{i=2t} (x - \alpha^i) \tag{2.4}$$

The generator polynomial has degree $2t$, thus an (N,k) RS code satisfies $N - k = 2t$. Notice that g(x) can also be characterized by the minimum polynomials of α^b,α^{b+1},α^{b+2},...,and α^{b+2t-1} and can be generalized to

$$g(x) = \prod_{i=b}^{i=b+2t-1} (x - \alpha^i) \tag{2.5}$$

where b is a random integer number.

2.3 Generalized minimum distance algorithm

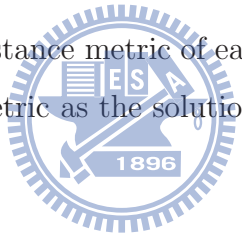
Generalized minimum distance (GMD) decoding is proposed by Forney [8]. The GMD decoding used reliability information of the received symbols to improve algebraic decoding. Based on successively erasing the least reliable symbols, GMD decoding runs the hard decision decoder. It is shown that GMD decoding can be asymptotically optimal. In ref [15] shows that an error-and-erasue method can correct all combinations of v errors and e erasures provided that $2v + e \leq d_{min} - 1$. The GMD decoding consider all cases of erasures $e \leq d_{min} - 1$ in the least reliable position(LRPs) which are the most likely positions to be in error. The decoding method are as follows:

1. GMD decoding generates the hard decision $\mathbf{Z}$ from the received sequence $\mathbf{y}$ and assigns a reliability value to each symbol $\mathbf{Z}$.

2. GMD decoding generates a list of $\lfloor \frac{d_{min}+1}{2} \rfloor$ sequence by modifying the hard decision sequence $\mathbf{Z}$ with erasing the least reliable symbols.

3. GMD decoding decodes each modified $\mathbf{Z}$ into a codeword. Then it feeds $\mathbf{Z}$ into an algebraic decoder.

4. GMD decoding computes the distance metric of each generated symbol and selects the one with the minimum distance metric as the solution.



2.4 Chase decoding algorithm

Chase algorithm [9] is the generalization of the GMD algorithm. Chase-1 algorithm does not consider reliable or unreliable positions. It always generates $C^n_{\frac{d_{min}}{2}}$ candidate codewords by considering all possible combinations of $\lfloor \frac{d_{min}}{2} \rfloor$ in the hard received sequence $\mathbf{Z}$. But the computational complexity of Chase-1 algorithm is too heavy, few people discuss the algorithm.

Chase-3 algorithm does similar operations as the GMD algorithm, except the erasure operation in the GMD is being replaced by flipping the least reliable symbols. For binary codes, Chase-3 algorithm achieves the same error performance as GMD and require same computational complexity.

Chase-2 decoding is another reliability based decoding to assist hard decision decod-

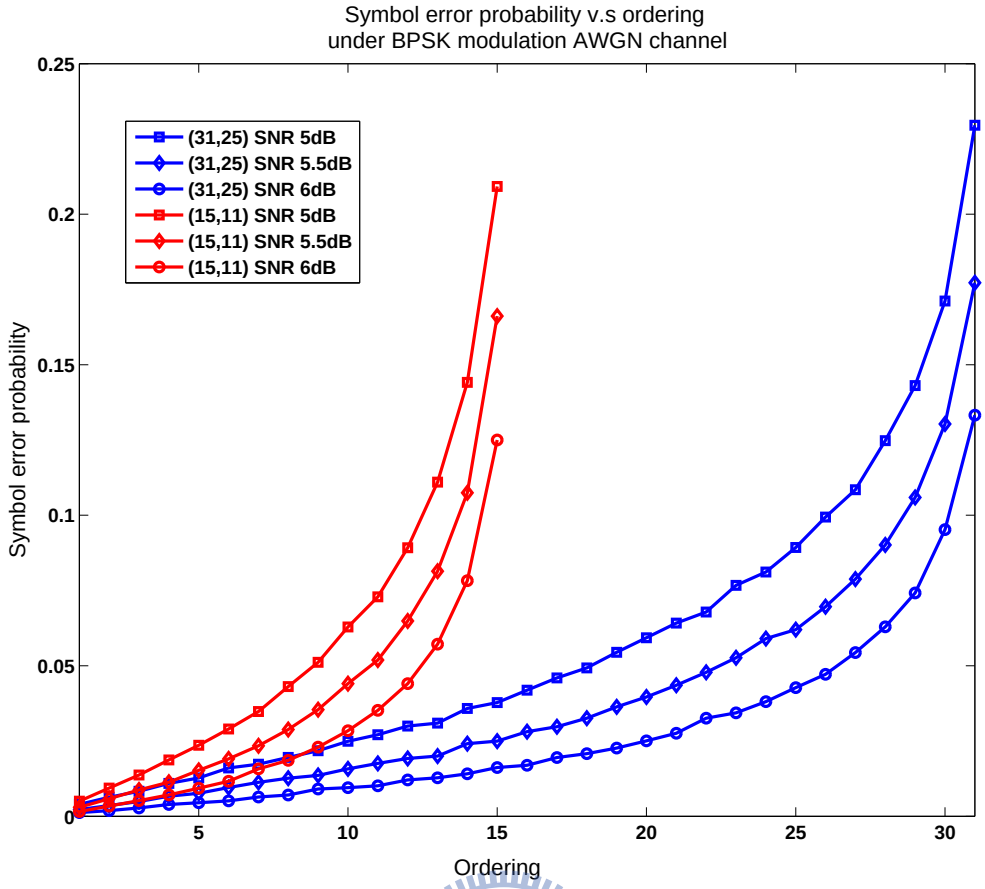


Figure 2.2: Error probability v.s ordered positions RS(15,11) and RS(31,25)

ing. It is an improvement of Chase-3 algorithm. It generates a larger candidate lists. All possible errors in the range of $\lfloor \frac{d_{min}}{2} \rfloor$ of LRPs is used to improve $\mathbf{Z}$. The algorithm exhaustively flip the least reliable symbols and run the hard decision decoder. The candidate list grows exponentially with d_{min}. Since the larger candidate list leads to a greater possibility to correct the errors. Chase-2 algorithm performs a much better performance than Chase-3 algorithm.

Among three algorithm, Chase-2 algorithm is the best algorithm considering the trade-off between complexity and performance.

Fig. 2.2 shows the error probability versus reliability ordering for RS(15,11) and RS(31,25). From the figure, we can observe that the error probability has an exponential like curve. The received vectors are sorted according to their reliabilitiew. The first k positions can be regarded as reliable positions, and the last N-k positions are unreliable positions. The least reliable N-k region has the higher probability to occur errors than

the previous k positions. This tells why the GMD decoding and Chase algorithm work. They generate the candidate codewords by dealing with the last N-k positions, and feed the modified candidate codewords in to hard decision decoder. They hope that the hard decision decoder can help them correct the errors occurred in the previous k position, if exists. However, the drawback of these two algorithm is that if the error does not occur in the last N-k region, and all concentrated in the reliable k regions and the number of errors are larger than the error correction capabilities. Then, the decoder will never find a correct codeword.

2.5 Algebraic soft decision decoding algorithm

Guruswami and Sudan (GS) invented a polynomial-list decoding algorithm [4] for RS codes capable of correcting beyond half the minimum distance of the code. Koetter and Vardy developed an algebraic soft decision(ASD) decoding [3] based on GS algorithm using the reliability information to assign multiplicity for RS codes. the decoding scheme is briefly summarized as follows: The transmitted codeword can be described as $(f(\alpha_1), f(\alpha_2), \ldots, f(\alpha_N))$ the received vector is $(\beta_1, \beta_2, \ldots, \beta_N)$. The basic idea is to find a $f(x)$ which fits as many points in $(f(\alpha_i), \beta_i)$ pairs. The KV algorithm consists of two main steps, interpolation step and factorization step.

Step 1) Interpolation: Construct a bivariate polynomial $Q(x, y)$ of minimum (1,k-1) degree, which has a zero order of v at (α_l, β_l), l=1,. . .,N, i.e: if $Q(x - \alpha_l, y - \beta_l)$ involves no term of degree less than $i + j = v$.

Step 2) Factorization: Generate a list of y-roots,i.e:

$L = f(x) \in F[x] : (y - f(x)|Q(x, y), deg(f(x)) < k)$

Then, pick up the most likely codeword $\hat{f(x)}$ from the list L.

The ultimate gain of algebraic soft decoding (ASD) over AWGN channel is about 1dB. The complexity is scalable but prohibitively large for huge multiplicity.

2.6 Iterative Reed-Solomon decoding algorithm

In [5], Jiang proposed the iterative decoding algorithm based on sum-product algorithm(SPA). The main idea is to adapt the parity-check matrix at each iteration according

to the reliabilities such that the unreliable bits correspond to a sparse matrix, so that the SPA algorithm can continue applying to the adapted parity check matrix. The adaptation prevents iterative decoding from getting stuck at local equilibrium region, thus, the decoding process can continue to a convergence region. The following is the parity-check matrix of an (N,k) RS code over $GF(2^q)$:

$$H_s = \begin{pmatrix} 1 & \beta & \dots & \beta^{(N-1)} \\ 1 & \beta^2 & \dots & \beta^{2(N-1)} \\ & & \dots & \\ 1 & \beta^{(d_{min}-1)} & \dots & \beta^{(d_{min}-1)(N-1)} \end{pmatrix} \tag{2.6}$$

where β is a primitive element in $GF(2^q)$, $d_{min} = N - k + 1$. Let $n = N \times q$ and $\mathbf{k} = k \times q$ be the length of the codeword and information at the bit level, respectively. H_s has an equivalent binary image expansion H_b (One can find in [15]), where H_b is an $(n - \mathbf{k}) \times n$ binary parity-check matrix.

The iterative algorithm is composed of two stages:

1) The matrix updating stage.

2) The bit-reliability updating stage.

Assume $\mathbf{c} = (c_0, c_1, c_2, \dots, c_{Nq-1})$ is the binary representation of RS codeword. By using BPSK modulation and transmitted the codeword over an AWGN channel the received vectors $\mathbf{y} = (y_0, y_1, y_2, \dots, y_{Nq-1})$ can be given by

$$\mathbf{y} = \mathbf{c} + \mathbf{w} \tag{2.7}$$

where $\mathbf{w}$ is the AWGN noise. The initial reliability of each bit can be expressed in terms of the log-likelihood ratios(LLR):

$$L^{(0)}(c_i) = log\frac{P(c_i = 0|y_i)}{P(c_i = 1|y_i)} \tag{2.8}$$

where c_i stands for codeword and y_i stands for received vector in the bit representation. The magnitude of the received LLRs $|L(c_i)|$ are sorted in an descending order $i_N, \dots, i_{N-k}, \dots, i_2, i_1$, the bit c_{i_N} is the most reliable and the bit c_{i_1} is the least reliable. The algorithm begins with the original parity-check matrix H_b, and reduce the i_1 th column of H_b to a form $[10 \dots 0]^T$. the process is continued to reduce $(N - \mathbf{k})$-th columns of H_b to be the identity matrix. The matrix is thus reduced like Fig. 2.3.

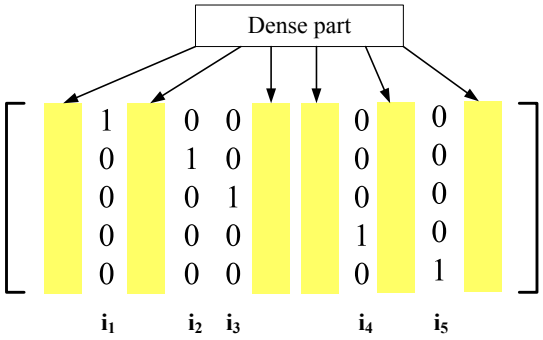


Figure 2.3: Form of the parity check matrix suitable for iterative decoding by row operations

In the matrix updating stage. For the v-th iteration, the vector of LLRs will be

$$L^v = [L^{(v)}(c_1), L^{(v)}(c_2), \ldots, L^{(v)}(c_n)] \tag{2.9}$$

and L_0 is determined from the channel output. Then, the parity-check matrix is reduced to a form based on L^v

$$H_b^v = \phi(H_b, |L^v|) \tag{2.10}$$

In the bit-reliability updating stage, the extrinsic LLR vector L_{ext}^v is generated by L^v using the SPA based on the previous adapted parity-check matrix H_b^v

$$L_{ext}^{(v)} = \psi(H_b^v, L^v) \tag{2.11}$$

For each bit, the extrinsic LLR is updated according to

$$L_{ext}^{(v)}(c_i) = \sum_{j=1, H_{ji}^v=1}^{n-k} 2tanh^{-1}\left(\prod_{p=1, p\neq i, H_{jp}^v=1}^{n} tanh(\frac{L^{(v)}(c_p)}{2} \right) \tag{2.12}$$

The bit-reliability is then updated as

$$L^{(v+1)} = L^{(v)} + \alpha L_{ext}^{(v)} \tag{2.13}$$

where $0 \leq \alpha \leq 1$ is a damping coefficient. Then, take hard decision of the decoded bit codeword $\hat{c}_i$

$$\hat{c}_i = \begin{cases} 0, & L^{(v+1)}(c_i) > 0 \\ 1, & L^{(v+1)}(c_i) < 0 \end{cases} \tag{2.14}$$

The termination criterion is, if all the checks are satisfied, output the estimated bits; else if $v = v_{max}$ iteration, declare a decoding failure; otherwise set v to v+1 and go for another iteration.

The drawback of the iterative decoding by adapting parity-check matrix of RS code is that the algorithm has the "error floor" problem. And the iterative decoding needs different iteration to complete the decoding process.

Chapter 3

Proposed soft RS decoding with K-Best algorithm

Reliability based decoding with ordered statistics [7] has been shown to be efficient to decode binary linear block codes. In this chapter, based on reliability decoding and re-encode process we introduce a new matrix **AG** for our decoding operation. We didn't do the re-encode process as the previous paper. The candidate of each reliable symbols is generated when the received soft value is determined. We take advantage of the more reliable column of matrix **AG**, then transform it into a tree-oriented problem. The K-Best algorithm is introduced to search the possible candidate previously generated. Since the decoding speed is the drawback of the K-Best algorithm, parallel K-Best algorithm is used to improve the speed issue. The K-Best algorithm with constraint is used to reduce the computation complexity, especially the comparison for the parallel K-Best algorithm. Erasure-only decoding is described next to use the property of RS codes, we can regard the N-k unreliable position as erasures. The k-reliable symbols of K-Best combinational sets can be used to decode the other N-k erasures. A brief summary is at the end of this chapter, to describe the decoding flow of the proposed soft RS decoding algorithm.

∗Note: The method including cost function, re-encode, sphere decoding, K-Best algorithm and erasure-only decoding in this chapter can be found in chapter 6, appendix.

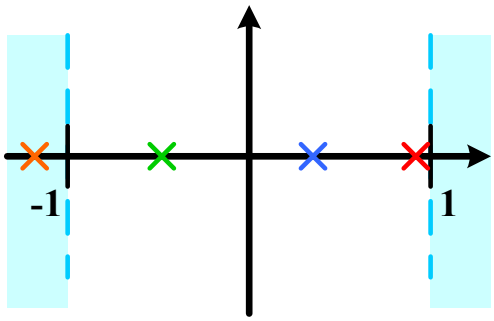


Figure 3.1: BPSK mapping and the distribution of unreliable bits

3.1 Pre-processing

In Pre-processing, we first generate the symbol candidates, and reorder the symbol based on reliability. The generator matrix is also permuted according to the reliability. Then, Gauss elimination is used to avoid the correlation between reliable symbols. At last, re-encode is used to generate the possible codeword sets. It is observed in the pre-processing process that reliable symbols do not need to execute Gauss elimination, since they are already independent.

3.1.1 Candidate generation

Decoding based on most reliable independent position (MRIP) requires the ordering of the received sequence according to their reliability. At first, we use hard decision symbols to be the foundation to choose the reliable symbols and produce the candidates. After DE-BPSK, we declare the hard decision of $\mathbf{y}$ as $\mathbf{z}$, where $\mathbf{z} = (z_0, z_1, z_2, \ldots, z_{Nq-1})$. The vector $\mathbf{z}$ is transmitted from the binary sequence into symbol sequence $\mathbf{Z}$ where $\mathbf{Z} = (Z_0, Z-1, \ldots, Z_{N-1})$. Every symbol Z_j where j is form 0 to $N-1$ is composed by $(z_{jq}, z_{jq+1}, \ldots, z_{jq+(q-1)})$.

Based on the re-encoding schemes, the selection of candidate is as follows. We decide the unreliable bits number of symbols and generate the candidates correspond to those symbols. Because those bits are not reliable, we consider all possible combination of their position with 1 and 0.

Fig. 3.1 shows the example of unreliable bits. We can set up criterion to decide how many number of unreliable bits need to be considered. The criterion is as follows:

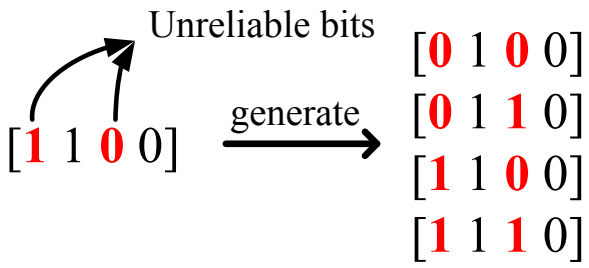


Figure 3.2: Candidates generation based on unreliable bits

Design criterion

1) The received soft values lie between 1 and -1.

2) The number of unreliable bits could be decided off-line according to the absolute soft values.

After the unreliable bits are found, DE-BPSK is done in the next. According to Fig. 3.1, the corresponding hard decision bits of jth symbol $(z_{0,j}, z_{1,j}, z_{2,j}, z_{3,j})$ are $(1, 0, 0, 1)$. Then, for each symbol we could use the unreliable bits to produce its candidates. Fig. 3.2 shows the example of candidate generation.

To produce all candidates of the symbol, there are three steps as follows.

1) We decide the unreliable bits number of the symbol and mark the position of these unreliable bits.

2) We use 1 and 0 to produce 2^v combination sets, where v is the number of unreliable bits.

3) There are 2^v sets of v unreliable bits, they would be placed into the marked position of unreliable bits.

After the above three steps, all candidates would be generated. We use the above three steps and continue to demonstrate the previous example. There are two unreliable bits, $z_{0,j}$ and $z_{2,j}$, corresponding to $y_{0,j}$ and $y_{2,j}$. The positions of these bits are 0 and 2. Then, all 2^2 combination sets of 1 and 0 are generated. They are $(0, 0), (0, 1), (1, 0)$ and $(1, 1)$. Next, we replace the origin hard decision symbol bits position with these bit sets. The origin hard decision bits of the symbol are $(1, 0, 0, 1)$. Therefore, the candidates are $(0, 0, 0, 1), (0, 0, 1, 1), (1, 0, 0, 1)$ and $(1, 0, 1, 1)$.

The candidates of symbol are produced by the order of reliability. Table 3.1 shows

Table 3.1: Example of cost table

	−0.5	−1.2	0.3	0.9	cost
4	1	−1	1	1	0.5
6	1	−1	−1	1	0.8
12	−1	−1	1	1	0
14	−1	−1	−1	1	0.3

the example of different cost between candidates and received soft values. We order the candidates of the symbol by their cost compared to the received value. Every symbol Z_j would have its own candidates $(\hat{Z}_{j,0}, \hat{Z}_{j,1}, \ldots, \hat{Z}_{j,m})$. The selection of m which is between 1 to $N+1$ is obtained by the number of unreliable bits. The candidates of jth symbol can be denoted as a vector $\hat{Z}_j$. We calculate the cost between candidates $\hat{Z}_{j,i}$ and the received soft values y, then arrange $\hat{Z}_{j,i}$ in a ascending order denoted as $S_j = (S_{j,0}, S_{j,1}, \ldots, S_{j,m})$.Where the permutation is denoted as λ_1.

$$S_j = \lambda_1(\hat{Z}_j) \tag{3.1}$$

Obviously, $S_{j,0}$ is the hard decision symbol. The candidate with the smallest cost value is the most possible transmission symbol. First k reliable candidate sets are used in this algorithm through the process, and the last $N-k$ candidate sets would be used in the erasure only decoding.

After producing the candidates of all symbols, we compose all possible candidate sets. Fixed number (m) of candidate symbols is used in re-encode method. There are m^k possible combination sets to operate re-encode process. These possible candidate sets should execute the same operation as the columns of G.

3.1.2 Reliability ordering

For each hard decision symbol $S_{j,0}$, reliability is calculated by adding the corresponding absolute value of received soft values y_j.

$$\mathbf{R}_j = \Sigma_{i=0}^{q-1} \|y_{jq+i}\| \tag{3.2}$$

The symbols are rearranged according to the ordering of the largest absolute value.

Then $\mathbf{S}$ is rearranged to be $\mathbf{S}' = (S'_0, S'_1, S'_2, \ldots, S'_{N-1})$.

The symbols are re-arranged as $\mathbf{S}'$. We denote the permutation as λ_2. We selected first k most reliable symbols $(S'_0, S'_1, \ldots, S'_{k-1})$ to perform the pre-processing. Since the columns of $\mathbf{G}$ correspond to the position of symbols, thus we permute the columns of generator matrix based on the ordering of reliability. After the permutation of generator matrix $\mathbf{G}$, the new matrix can be marked as $\mathbf{G}'$.

$$S' = \lambda_2(S) \tag{3.3}$$

$$G' = \lambda_2(G) \tag{3.4}$$

3.1.3 Re-order generator matrix

The first k columns of G' are not necessarily independent while in re-encoding scheme, thus such k columns can not represent the information set. We are going to rearrange the first k columns of $\mathbf{G}'$ to make each k column independent. The rearranged matrix is $\mathbf{G}''$ and its first k columns are linearly independent. After the rearrange operation λ_3, the columns of $\mathbf{G}''$ also maintain the descending order of the reliability values. The operation λ_3 could be regarded as Gaussian eliminations which let $\mathbf{G}''$ in the reduce echelon form that contain only one 1. We use matrix $\mathbf{A}$ to represent the Gaussian elimination operation.

$$\mathbf{G}'' = \lambda_3(\mathbf{G}') = \mathbf{G}' \times \mathbf{A} \tag{3.5}$$

The generator matrix becomes $\mathbf{G}''$ after the Gaussian elimination which is shown as Fig. 3.3. The corresponding symbols should execute the same operation, then the resultant is denoted as $\mathbf{S}''$.

$$\mathbf{S}'' = \lambda_3(\mathbf{S}') = \mathbf{S}' \times \mathbf{A} \tag{3.6}$$

Every possible set of $\mathbf{S}''$ should do the same operations to produce the possible codeword in re-encode process. Because the $\mathbf{G}''$ is systematic matrix, it means the corresponding $\mathbf{S}''$ is the possible independent message sets. The possible message sets multiply generator matrix $\mathbf{G}$ to produce possible codeword sets. In re-encode method, we calculate the cost between the received soft value and possible codeword and decide the codeword with minimum cost as the decoded codeword.

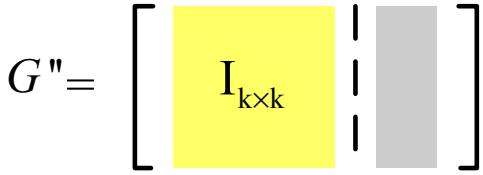


Figure 3.3: Gauss illustration

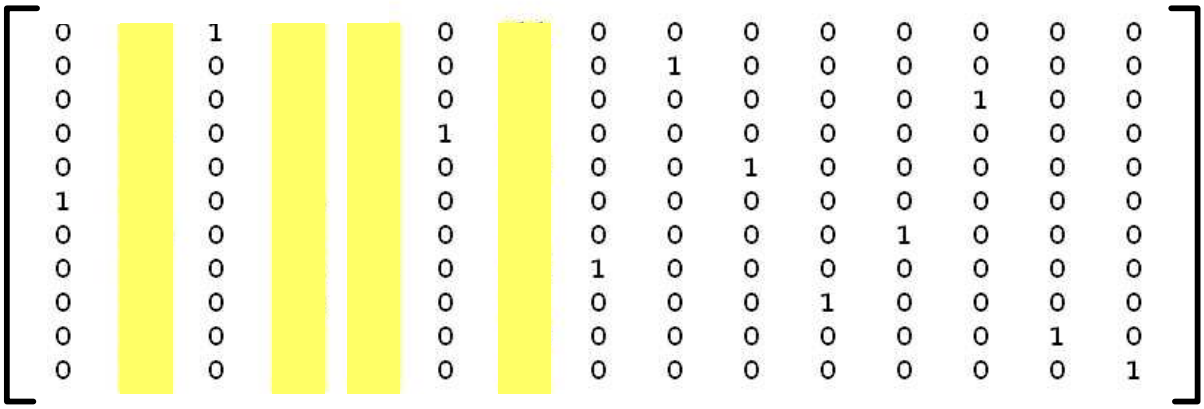


Figure 3.4: Example of AG matrix

3.1.4 Re-encoding process

In re-encode method, every $\mathbf{S}'$ set must multiply matrix $\mathbf{A}$ and then multiply matrix $\mathbf{G}$. Because there are m^k sets, re-encode method needs a lot of multiply operations. We want to reduce the operation number, so we multiply $\mathbf{A}$ and $\mathbf{G}$ together to be matrix $\mathbf{AG}$. Fig. 3.4 shows an example of the matrix $\mathbf{AG}$. Then every $\mathbf{S}'$ set only needs to multiply matrix $\mathbf{AG}$ once. The re-encode process speeds up 1.5 times. It reduces the operation number and produces the same codeword sets.

After we produce matrix $\mathbf{AG}$, we discover that there is a special property of matrix $\mathbf{AG}$. The property of matrix $\mathbf{AG}$ can be divided in to two parts. First, the columns with only one 1 correspond to the position of reliable symbols can be regarded as the permutation of $k \times k$ identity matrix. Second, the columns correspond to the position of $N - k$ unreliable symbols can be regarded as the combination of the other k reliable symbols. Observing the matrix $\mathbf{AG}$, we find out that the k reliable symbols are independent.

1) We marked the position of 1 in the columns of matrix $\mathbf{AG}$ which has only one 1 as index $\mathbf{L}$. $\mathbf{L} = (l_0, l_1, l_2, \ldots, l_{k-1})$ are indexes corresponding to the most k reliable symbols.

2) According to the index vector $\mathbf{L}$, the candidates of the most k reliable symbols would

be selected in K-Best scheme.

3) The order of k column with only one 1 stands for the position of the corresponding reliable symbol.

4) The value of index $l_i (i = 0 \sim k-1)$ represents for the l_ith transmitted symbol.

The index would point out the most k reliable symbol corresponding to the l_jth symbol of the received codeword, respectively.

3.2 Candidate selecting

In section 3.1.4, it has mentioned that the possible candidate sets will be m^k. This is a large number when k increases, and it is not feasible to use these candidate sets to do the re-encoding process. In the following section, we proposed K-Best algorithm, parallel K-Best algorithm and parallel K-Best algorithm with constraint to reduce the number of candidate sets.

3.2.1 K-Best algorithm

There are m^k combination sets of $\mathbf{S}'$ needed to multiply matrix $\mathbf{AG}$, which still leads to a large number of operations. The independent property of matrix $\mathbf{AG}$ can be used to reduce the re-encode operation. Thus, we can use the independent property of matrix AG to execute K-Best operation. K-Best operation needs k layers to select the candidate sets. Every candidate set of symbol would be a layer in K-Best. A tree Fig. 3.5 is used to represent the whole process.

We assume an origin point in layer 0 and use the dotted line to mark it as a pseudo node. A pseudo node is set up to start the tree-like algorithm. The path between layer 0 and layer 1 would be marked as dotted line too. Then, the first candidate sets would spread up at layer 1. In the Fig. 3.5, the node represents the accumulated cost. There are m candidates of jth symbol in each layer where $\mathbf{S}'_j = (S'_{j,0}, S'_{j,1}, \ldots, S'_{j,m})$.

The value in the line stands for the cost of that candidate compared to the received soft value. The value of node stands for the accumulated cost. Each layer accumulates the cost from the previous layer, and then chooses the combination sets of K minimum

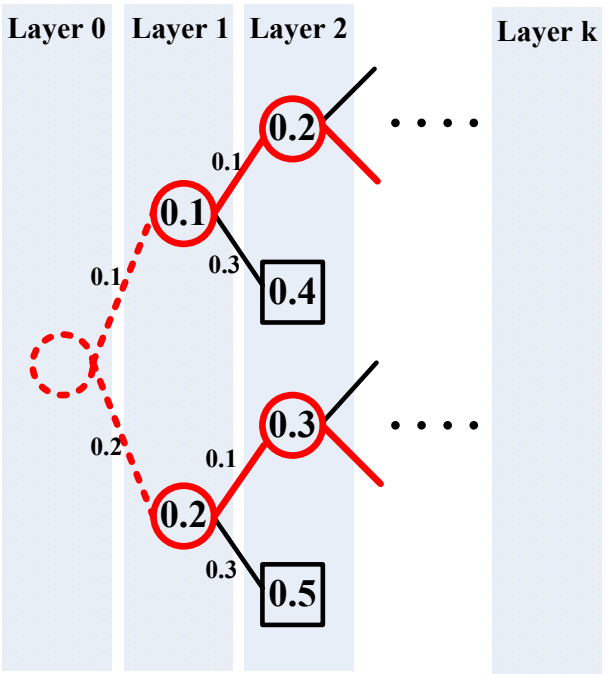


Figure 3.5: Example of K-Best candidate selection, K=2

cost candidates. The chosen path and node would be marked as line and circle in solid line. The next layer would start from the chosen node. The stop criterion is that the candidates of k reliable symbols are all considered.

We can summary the process in four steps as follows until the stop criterion is achieved.

1) For each parent node, it spreads out the possible candidates as child node in that layer

2) Each child node accumulates the cost

3) We choose K minimum cost sets in the decoding layer.

4) Check the stopping criterion. If the stopping criterion is not achieved, start the next layer from chosen K nodes in last layer

Therefore, we have less candidate sets chosen by K-Best method than the re-encode method.

3.2.2 Parallel K-Best algorithm

Although the K-Best method reduces the operation, the speed of execution is still not good enough. In this thesis a parallel K- Best algorithm is proposed to improve the speed issue, it separates most k reliable candidate sets into several groups. There are q layers in Galois Field $GF(2^q)$. Fig. 3.6 shows an example of RS(15,11) with 4 layers.

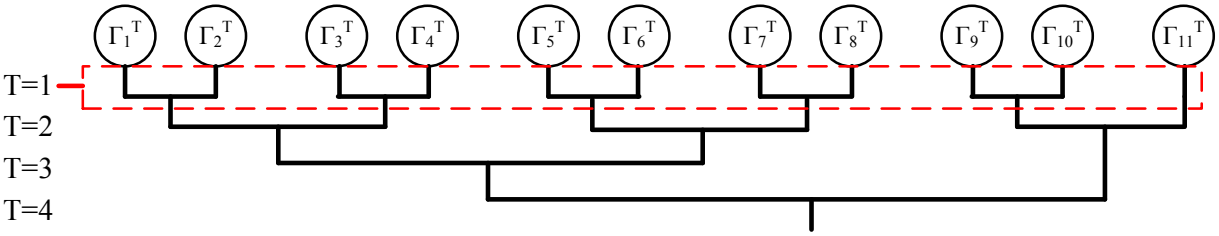


Figure 3.6: Example of parallel K-Best candidate selection with 4 layer for RS(15,11), T=1,. . .,4.

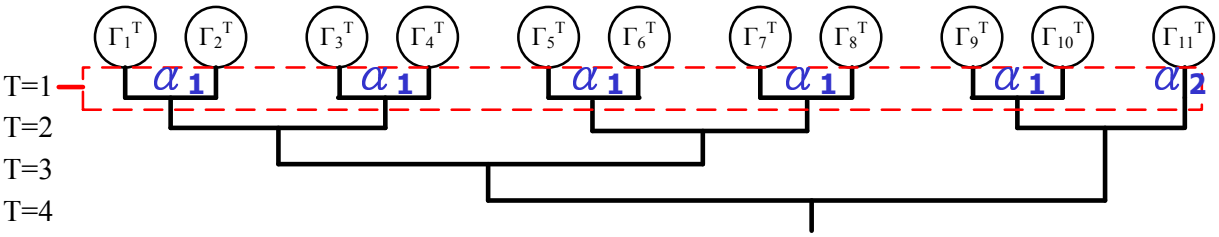
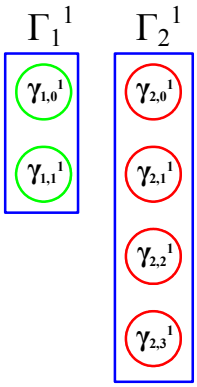


Figure 3.7: Example of subgroup selection in parallel K-Best scheme

Here are the steps for parallel K-Best algorithm.

1) For the selection of the candidate, we define some parameters as following. Here, T stands for the number of layer from 1 to q, α_1 is the number of sub-groups with two candidate sets in that layer and α_2 is the number that a sub-group has only one candidate sets in that layer. α_1 and α_2 are integers, and α_2 would always be 0 or 1. Fig. 3.7 is the example of subgroup selection in parallel K-Best scheme. In an (N, k) RS code, α_1 would be initialized as $k/2$ and α_2 would be initialized as $k\%2$ at first layer. % stands for module.

2) We use a sequential tree-like method to help us choose candidate sets in K-Best scheme. If there are two input candidate sets $(\mathbf{\Gamma_1^T}, \mathbf{\Gamma_2^T})$ with (m_1^T, m_2^T) candidates in layer T where $\mathbf{\Gamma_1^T} = (\gamma_{1,0}^T, \gamma_{1,1}^T, \ldots, \gamma_{1,m_1-1}^T)$ and $\mathbf{\Gamma_2^T} = (\gamma_{2,0}^T, \gamma_{2,1}^T, \ldots, \gamma_{2,m_2-1}^T)$, and $\gamma_{j,i}^T$ means the i-th candidate of input candidate set $\mathbf{\Gamma_j^T}$ in layer T. There are total $M^T = m_1^T \times m_2^T$ combinational candidate sets. Fig. 3.8 shows an example of the subgroup. We must use K-Best method to choose K combined candidate sets with smaller cost value. As we choosing these candidate sets, we must denote the cost value and the index of each candidate combination sets. The index here means that the candidates of combined sets which its original position is in input candidate set?



Cost	0	1	2	3
Γ_1^1	0.1	0.2		
Γ_2^1	0.1	0.3	0.4	0.5

Figure 3.8: Example of subgroup in parallel K-Best scheme.

Table 3.2: Example of subgroup cost table

Accumulated Cost	order of Γ_1^1	order of Γ_2^1
0.2	0	0
0.3	1	0
0.4	0	1
0.5	0	2
0.5	1	2
0.6	0	3
0.6	1	2
0.7	1	3

Obviously, each input candidates sets needs M^T indexes. The cost information is saved in another M^T register. Thus, total $3M^T$ registers are needed to be saved for K-Best scheme. Table. 3.2 shows an example of $3M^T$ registers to save the index and cost information.

Now we use a scheme which needs only $2M$ register to execute K-Best algorithm.

1) We produce the combination candidate sets in sequence as following. We fix the candidate of $\mathbf{\Gamma_1^T}$ and displace the candidates in $\mathbf{\Gamma_2^T}$ until all m_2^T candidates are changed. Then we move on to the next candidate of $\mathbf{\Gamma_1^T}$, and do the same process as above. Fig. 3.9 shows an example of producing path index and candidate sets for parallel K-Best algorithm.

2) During producing the combinational candidate sets, we give every combination set an index number Ω from 0 to $M^T - 1$ sequentially, and denote the accumulated cost of every combined candidate sets.

3) As all M^T combination candidate sets are produced, all combination sets are sorted by the cost and also the index is permuted according the new order.

4) K-Best scheme choose K candidate combination sets which are the input sets of next layer. The elements of chosen combination sets are traced by the index Ω. The element from $\mathbf{\Gamma_1^T}$ is the γ_1^T th term of the input candidate set where $\gamma_1^T = \Omega / m_2^T$. The element from $\mathbf{\Gamma_2^T}$ is the γ_2^T th term of the input candidate set where $\gamma_2^T = \Omega \% m_2^T$. Then we could use only M registers to memorize the index of elements in combination sets. Fig. 3.10 shows an example of reduced index for parallel K-Best algorithm

5) Then, K-Best scheme is executed in every sub-group independently.

6) After the K-best algorithm, every sub-group chooses its own candidate sets. Fig. 3.11 shows the chosen candidate sets for parallel K-Best algorithm.

7) We repeat the steps above until the output candidate sets have k elements.

3.2.3 Parallel K-Best algorithm with constraint

Parallel K-Best method reduces the total comparing number for selecting the candidate sets. As the k becomes bigger, we need a large number of sorting. To prevent the sorting complexity becomes too heavy, we decide to use a bound to consistent the number of sorting.

Path	Accumulated Cost
0	0.2
1	0.4
2	0.5
3	0.6
4	0.3
5	0.5
6	0.6
7	0.7

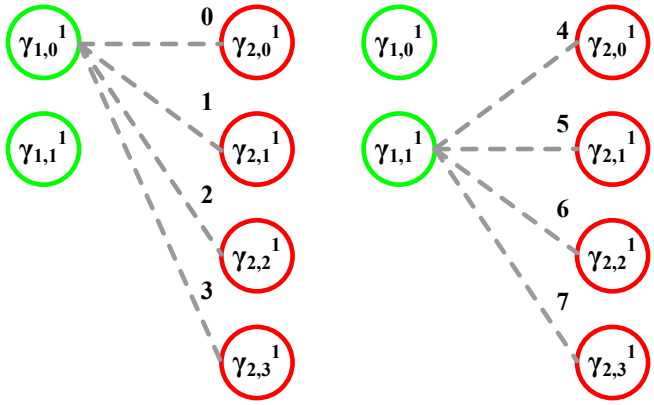


Figure 3.9: Example of path index and candidate sets for parallel K-Best algorithm

Re-ordered Path	Accumulated Cost
0	0.2
4	0.3
1	0.4
2	0.5
5	0.5
3	0.6
6	0.6
7	0.7

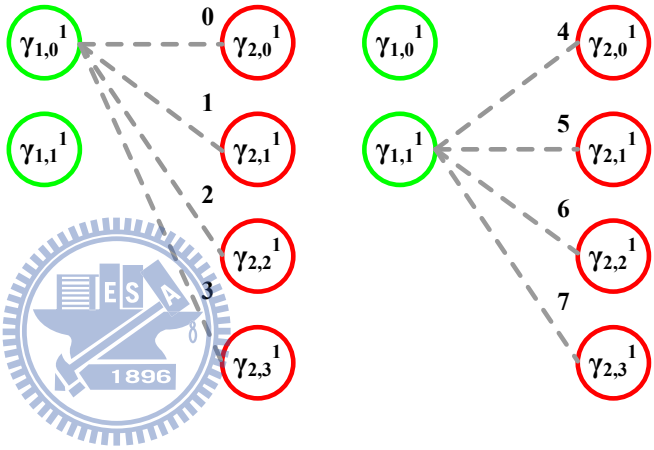


Figure 3.10: Example reduced index for parallel K-Best algorithm

Re-ordered Path	Accumulated Cost
0	0.2
4	0.3
1	0.4
2	0.5
5	0.5
3	0.6
6	0.6
7	0.7

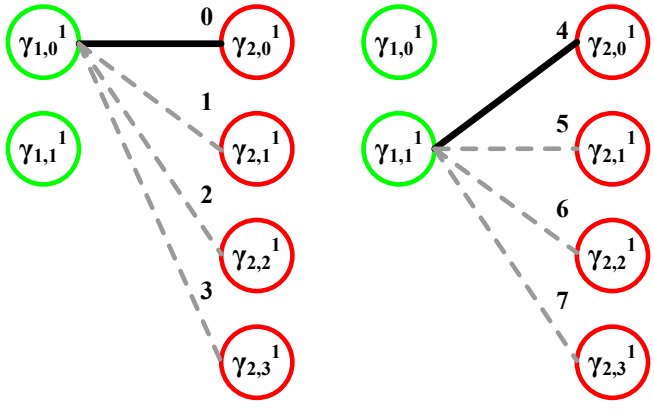


Figure 3.11: Example of chosen K candidate sets for parallel K-Best algorithm

Bound = 0.6

Path	Accumulated Cost
0	0.2
1	0.4
2	0.5
3	0.6
4	0.3
5	0.5
6	0.6
7	0.7

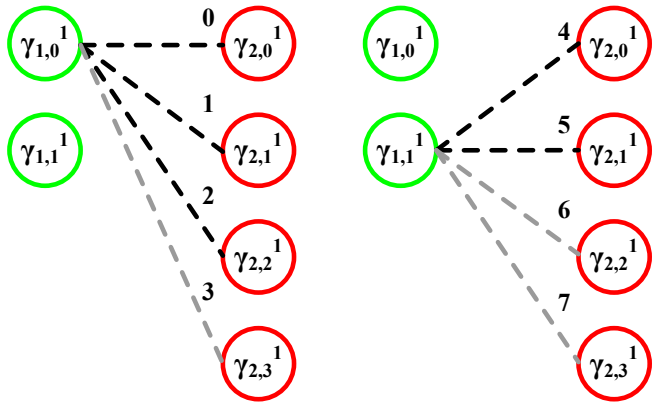


Figure 3.12: Parallel K-Best using constraint

Bound = 0.6

Path	Accumulated Cost
0	0.2
4	0.3
1	0.4
2	0.5
5	0.5

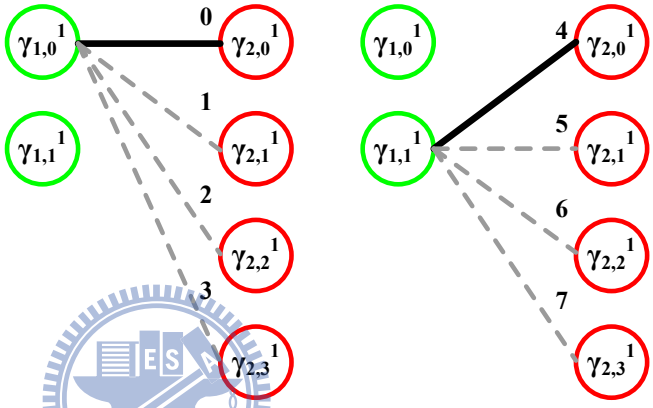


Figure 3.13: The chosen sets of parallel K-Best using constraint

In the process of parallel K-Best method, we define different bound for each layer. The bound of each layer is defined by observing the cost distribution. The accumulated cost of candidate sets smaller than the bound are chosen. The chosen combination sets would be re-arranged according the order of cost.

In the Fig. 3.12, the bound is defined as 0.6, and K is set to equal 2. Thus, the combination sets 0, 1, 2, 4 and 5 marked as black dotted line in order which would be chosen to be the possible K candidates.

By using the constraint and combined with the K-Best algorithm, the number of sorting sets are decreased. Fig. 3.13 shows the example of constraint, in this example There are total five candidates fall in the bound. If K is equal to 2, then the path 0 and path 4 are the selected combinational sets. The complexity of operation is reduced and the speed of process is faster almost two times than before.

3.3 Erasure-only decoding

We use K-Best method to choose K minimum cost sets correspond to K minimum candidate sets. Every possible unreliable sets and the chosen K-Best sets are combined together as possible codeword sets. Since re-encode process needs to deal with the ordering of generator matrix. In C program, this means the process needs a lot of temporary memory to store related data. Besides, the criterion for selecting codeword is only based on K-Best cost, it is not sufficient to obtain good performance. From AG matrix, we notice that the unreliable symbols can be composed by the reliable symbols. Thus, the erasure-only decoding is suitable for this task. And it adds another condition to see whether the codeword is in the generated candidate sets. Based on the above improvement of the decoding process, the re-encode process is replaced by erasure only decoding. Every possible codeword are put into syndrome equation to find out whether the possible codeword is in the codeword set of this system. If the syndrome equations are all zeros, it means that the possible codeword is in the codeword set. We choose the codeword under the following constraint

(1) Syndrome equations are all zeros

(2) The codeword with the minimum accumulated cost.

The codeword with the above two condition should be the transmission codeword.

The method mentioned before this section needs $\mathbf{S}''$ multiply matrix $\mathbf{AG}$ to obtain possible codeword sets. There are just one constraint, to find minimum cost. The possible codeword set is judged by which codeword set has minimum cost. In this section, we do not execute re-encode method because we do not use $\mathbf{S}'$ to multiply matrix $\mathbf{AG}$. Here, we add another constraint to consider the possible codeword. Is it in the codeword sets or not? This will help us choosing the right possible codeword. Using the minimum cost and syndrome equation constraints make us reach better performance then re-encode. But we discover that there are three kinds of errors as follows.

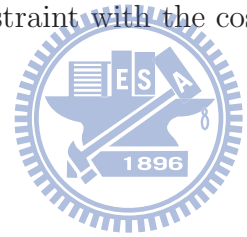
1) **The transmitted codeword is in the K-Best sets, but the syndrome equations of it (the transmitted set) are not all zeros.** This kind of error is produced because the candidates of unreliable symbols are not chosen properly. We could choose more unreliable bits to extend the possible candidates of unreliable symbols.

2) **The transmitted codeword is not in the K-Best sets.** Because the cost of

transmitted codeword is larger than the Kth Best sets, the transmitted codeword is not chosen. We could solve this error by choosing the larger K to contain the transmitted codeword.

3) **The transmission codeword is in the K-Best sets and the syndrome equations are all zeros but we do not choose the transmitted set.** The reason is that there is another possible codeword set which cost is smaller than the transmitted set and the syndrome equations are all zero, too. Under these conditions, we choose the wrong codeword which satisfy both constraints better than the transmitted one. This kind of error is the maximum-likelihood error, since the noise interference caused the right answer jump to another, thus we could not solve this problem

Though we use constraint for accumulated cost and syndrome equation to improve the performance, there still needs a better algorithm. We realize that the accumulated cost of MRIP k symbols being calculated by K-Best should be replaced by cost of all N candidates. Because the codeword sets produced by syndrome equation are possible transmitted codeword, the most possible codeword set should consider the cost of all N symbols. After we change the constraint with the cost of N symbols, the performance is improved.



3.4 Summary

The proposed algorithm consists of three parts, pre-processing, candidate selecting and erasure only decoding. It should be mentioned that in pre-processing process, the matrix AG shows that after reliability ordering from the received value, the most k reliable symbols are already independent. It implies we can choose the candidate sets after the symbols being reordered by reliability. Thus, it is needless to do any operation for generator matrix. For candidate selecting, the K-Best algorithm is introduced to reduce the computation complexity for possible candidate sets. Since the decoding speed is too slow for the layer by layer property of the K-Best algorithm, the parallel K-Best algorithm is introduced. The parallel K-Best algorithm divides the candidates sets into several groups, each subgroup chooses local K-Best candidate. The decoding process is continued until the candidate sets is combined with k symbols. The parallel K-Best algorithm reduces

the computation layer from k to q, thus increases the decoding speed and decrease the computation complexity. Finally, the re-encoding process is replaced by the erasure-only decoding.

Chapter 4

Simulation results and complexity comparison

In this section, RS(15,11) and RS(31,25) are simulated for comparing the proposed algorithm and the other RS code such as hard decision Berlekamp-Massy algorithm, KV algorithm, adaptive belief propagation algorithm(JN), adaptive belief propagation-algebraic soft decoding(ABP-ASD), and sphere decoding algorithm. The signal is modulated by BPSK and transmitted through the AWGN channel. For RS(15,11) when the SNR is below 5dB, 10^6 bits are simulated, and over 10^8 bits are simulated for SNR $\geq$ 6dB. For RS(31,25), 10^4 bits are simulated when the SNR is below 5dB, and 10^8 bits are simulated for SNR $\geq$ 6dB .

The complexity comparison includes the number of sorting and additions for K-Best algorithm, parallel K-Best, and parallel K-Best with sphere constraint. The HD-BM algorithm is used as an performance baseline.

4.1 Simulation results

In section 3.3, we discuss that we change the cost function, accumulated the metric from k to N, Fig. 4.1 shows the results. For RS(15,11), there is 1dB gap between the k cost and N cost for total K= 30 at codeword error rate(**CER**) 10^{-4}. Fig. 4.2 shows the performance comparison between different selection of K for RS(15,11). It can be observed that for the full selection of candidate symbol. K= 300 has the best

performance within 0.3dB gap of ML performance and the ABP-ASD [6]. There is 0.3dB loss between K= 100 and K= 300, another 0.7dB loss when we take K= 30. To reduce the computation complexity, parallel K-Best is proposed in section 3.2.2. Fig. 4.3 shows the performance comparison. All the case consider K= 300 in a parallel scheme. The difference is only the selection number of the candidate. From the figure we can observe that when candidate number is up to 5, the performance loss compared to candidate= 16 is smaller than 0.1dB. And the gap to the ML and ABP-ASD performance is also 0.3dB. In section 3.2.3, the parallel K-Best algorithm with constraint is proposed. Fig. 4.4 shows the simulation results. We can observe that setting up constraint for the parallel K-Best algorithm will degrade performance within 0.05dB, but we have mentioned in section 3.1.1 that the computation reduction, especially the sorting complexity is over 70% reduction. Fig. 4.5 compares the performance of different soft decision decoding scheme. It can be observed that the proposed algorithm outperforms the KV algorithm for 1.3dB at **CER** 10^{-4}, and JN belief propagation with BM about 1.5dB at **CER** 10^{-4}. The performance difference between our proposed algorithm, K-Best, parallel K-Best and parallel K-Best with constraint is fairly small. There is still a performance gap between ABP-ASD and ML. The reason is the nature of the sequential algorithm or (K-Best) algorithm. In the decoding process, once the size of K is not big enough to collect the true transmitted symbols, the decoded codeword may have errors. The larger K will cause more computation effort. Thus, it is a tradeoff between performance and computation complexity.

Fig. 4.6 is the simulation results for RS(31,25), at **CER** 10^{-3}, the performance gain over HD-BM is about 1.4dB. The proposed algorithm also performs better than KV and sphere decoding algorithm [16]. But there is a performance gap 1.25dB between the proposed algorithm and ML or ABP-ASD. The reason is that as the N increases, the candidate symbol selection for parallel K needs to increase, too. In this thesis, the K-Best combinational sets of the last layer is up to 3000. This number becomes infeasible when it comes to consider hardware implementation. Thus, the proposed method need to be improved when N is increased.

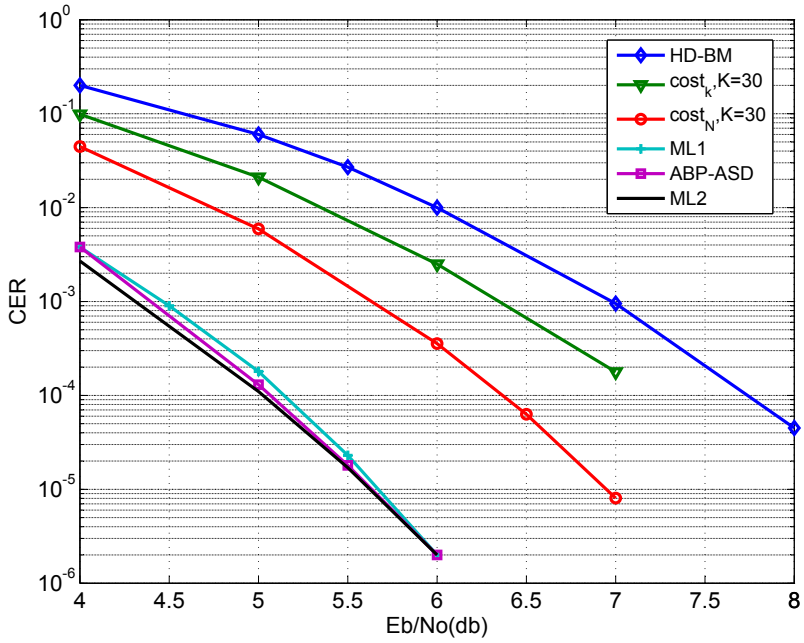


Figure 4.1: Performance comparison for K-Best of RS (15,11) BPSK mapping AWGN channel with different size of distance metric.

∗Note1: cost_N, cost_k represents the distance metric of N symbols and k symbols respectively.

∗Note2: ML1 and ML2 represents different estimation of ML performance.

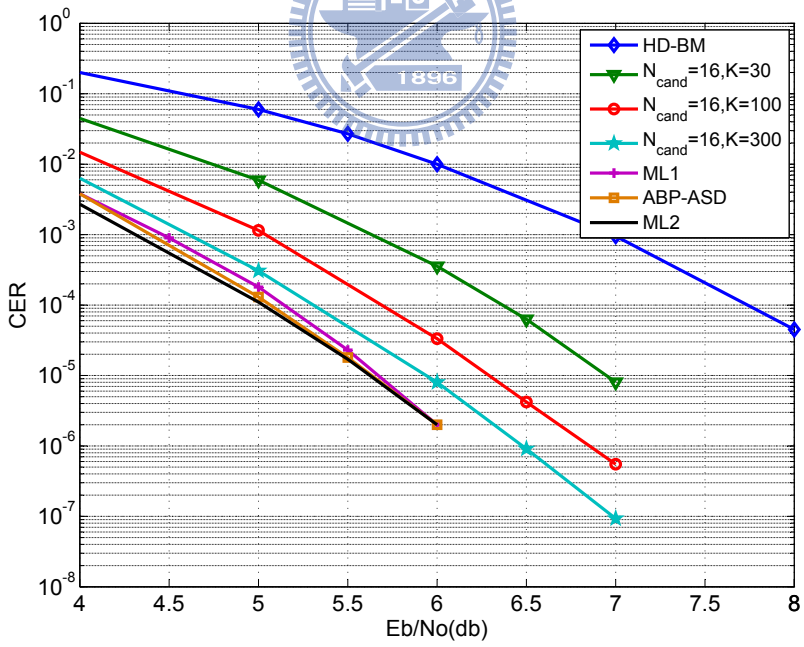


Figure 4.2: Performance comparison for K-Best of RS (15,11) BPSK mapping AWGN channel with different selection of K.

∗Note: N_{cand} represents the number of candidates is N.

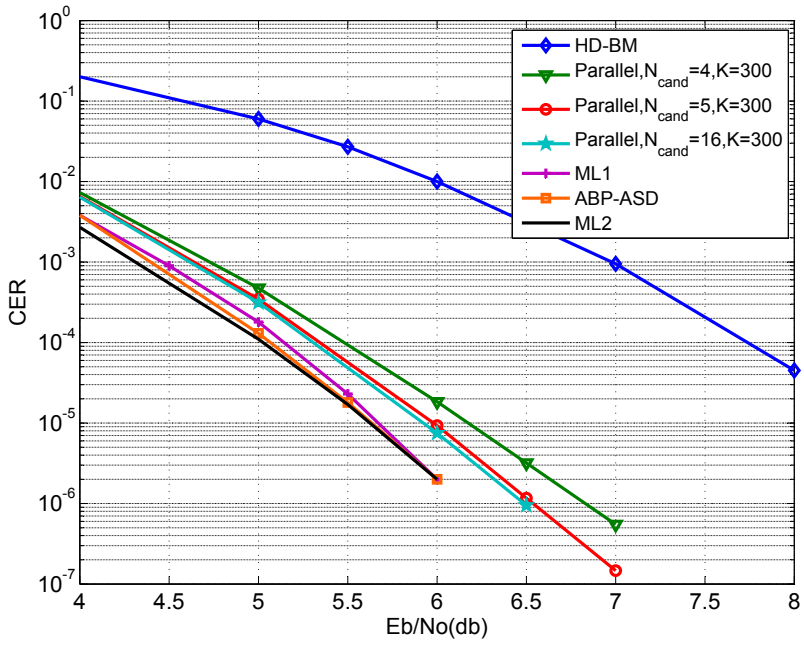


Figure 4.3: Performance comparison for parallel K-Best (K=300) of RS (15,11) BPSK mapping AWGN channel with different candidate number

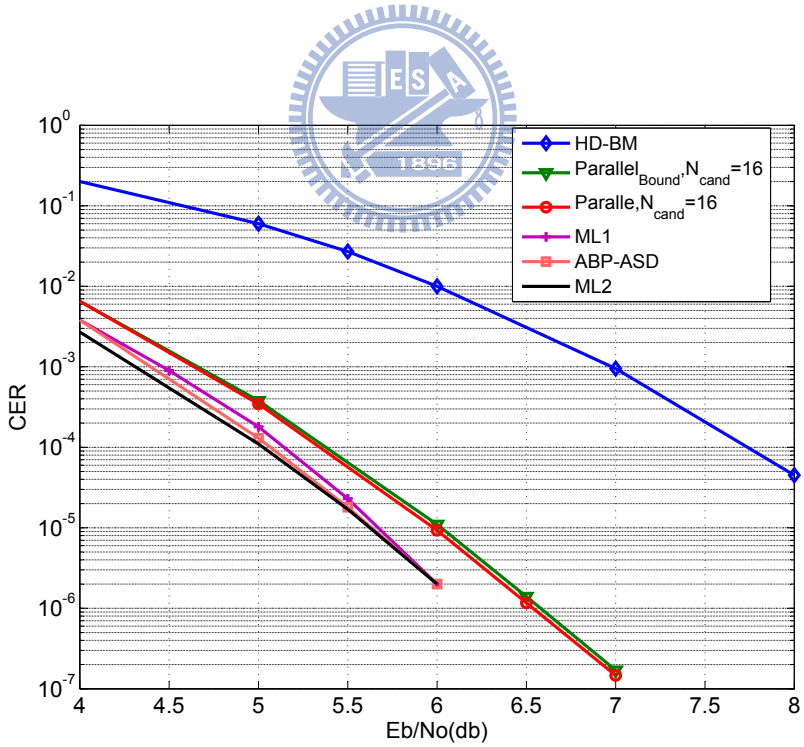


Figure 4.4: Performance comparison for parallel K-Best of RS (15,11) BPSK mapping AWGN channel with constraint

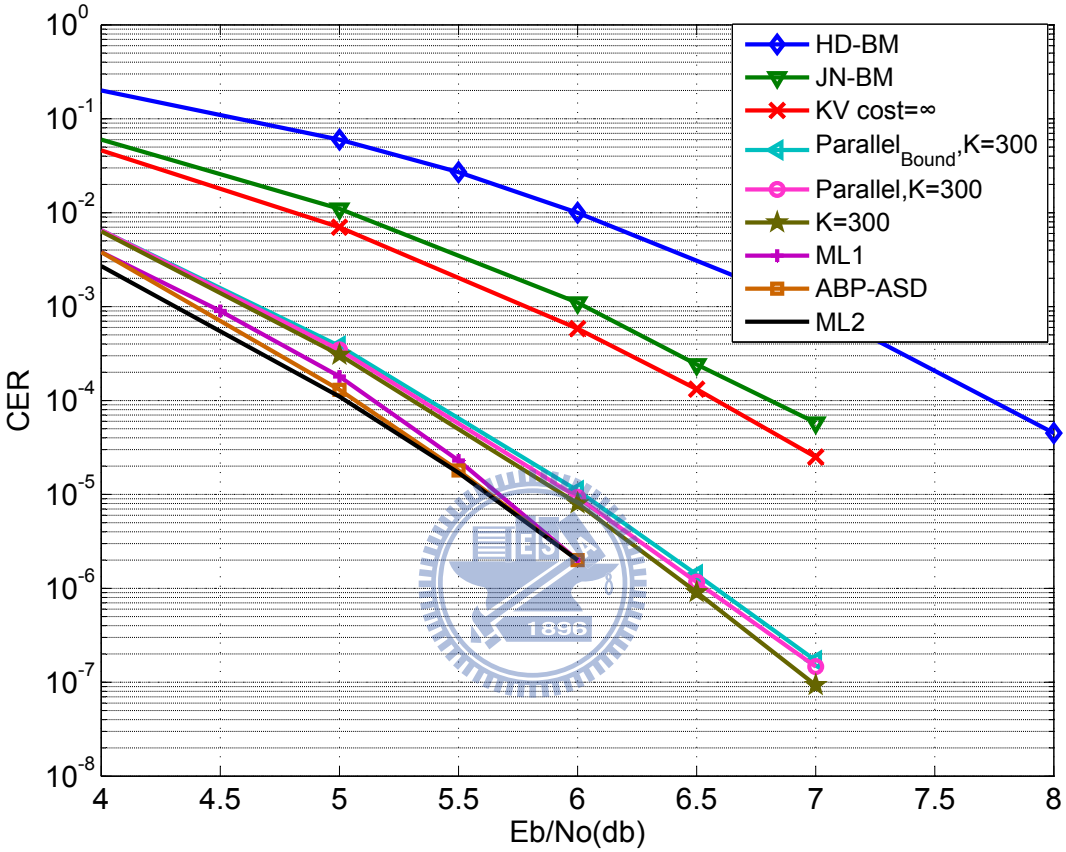


Figure 4.5: Performance of different soft decision decoding algorithm for RS (15,11) BPSK mapping AWGN channel

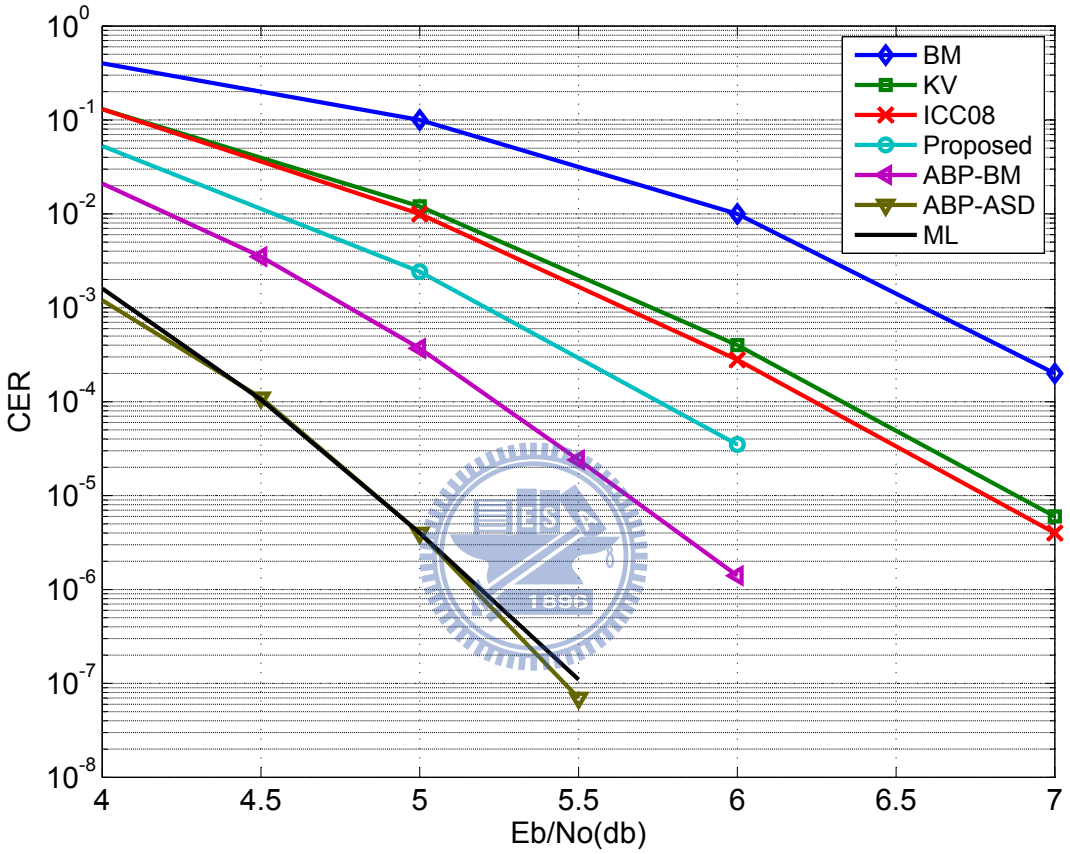


Figure 4.6: Performance for parallel K-Best of RS(31,25) BPSK mapping AWGN channel