

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



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

October 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

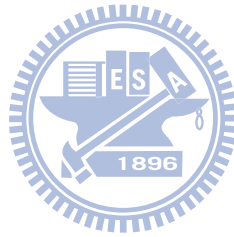
Abstract

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

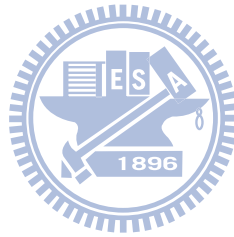
reduction is at least 75.5%. For RS (31,25), it outperforms the HD-BM algorithm by 1.4dB and the KV algorithm by 0.55dB at CER of 10^{-4} . The complexity reduction is at least 61.6% as compared to the KV algorithm. There is 1.25dB performance gap between ABP-ASD and proposed method at CER of 10^{-4}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

1	Introduction	1
1.1	Motivation	1
1.2	Thesis organization	2
2	Soft decoding of RS codes	4
2.1	System model of RS code	4
2.2	Basic properties of Reed Solomon codes	5
2.3	Generalized minimum distance algorithm	6
2.4	Chase decoding algorithm	6
2.5	Algebraic soft decision decoding algorithm	8
2.6	Iterative Reed-Solomon decoding algorithm	8
3	Proposed soft RS decoding with K-Best algorithm	12
3.1	Pre-processing	13
3.1.1	Candidate generation	13
3.1.2	Reliability ordering	15
3.1.3	Re-order generator matrix	16
3.1.4	Re-encoding process	17
3.2	Candidate selecting	18
3.2.1	K-Best algorithm	18
3.2.2	Parallel K-Best algorithm	19
3.2.3	Parallel K-Best algorithm with constraint	22
3.3	Erasure-only decoding	25
3.4	Summary	26

4	Simulation results and complexity comparison	28
4.1	Simulation results	28
4.2	Complexity comparison	34
5	Conclusion and Future Work	39
5.1	Conclusion	39
5.2	Future work	40
6	Appendix: several techniques used for the proposed soft RS decoding	41
6.1	Simplified cost function	41
6.2	Re-encode decoding based on reliability ordering	42
6.3	Sphere decoding and K-Best algorithm	43
6.4	Erasur-Only Decoding	45



List of Figures

2.1	Channel model of RS code	5
2.2	Error probability v.s ordered positions RS(15,11) and RS(31,25)	7
2.3	Form of the parity check matrix suitable for iterative decoding by row operations	10
3.1	BPSK mapping and the distribution of unreliable bits	13
3.2	Candidates generation based on unreliable bits	14
3.3	Gauss illustration	17
3.4	Example of AG matrix	17
3.5	Example of K-Best candidate selection, $K=2$	19
3.6	Example of parallel K-Best candidate selection with 4 layer for RS(15,11), $T=1, \dots, 4$	20
3.7	Example of subgroup selection in parallel K-Best scheme	20
3.8	Example of subgroup in parallel K-Best scheme.	21
3.9	Example of path index and candidate sets for parallel K-Best algorithm	23
3.10	Example reduced index for parallel K-Best algorithm	23
3.11	Example of chosen K candidate sets for parallel K-Best algorithm	23
3.12	Parallel K-Best using constraint	24
3.13	The chosen sets of parallel K-Best using constraint	24
4.1	Performance comparison for K-Best of RS (15,11) BPSK mapping AWGN channel with different size of distance metric.	30
4.2	Performance comparison for K-Best of RS (15,11) BPSK mapping AWGN channel with different selection of K.	30

4.3	Performance comparison for parallel K-Best (K=300) of RS (15,11) BPSK mapping AWGN channel with different candidate number	31
4.4	Performance comparison for parallel K-Best of RS (15,11) BPSK mapping AWGN channel with constraint	31
4.5	Performance of different soft decision decoding algorithm for RS (15,11) BPSK mapping AWGN channel	32
4.6	Performance for parallel K-Best of RS(31,25) BPSK mapping AWGN channel	33
6.1	Illustration of sphere decoding algorithm	44
6.2	Traditional K-Best algorithm	44



List of Tables

3.1	Example of cost table	15
3.2	Example of subgroup cost table	21
4.1	Comparison between K-Best, parallel K-Best and parallel K-Best with constraint for RS (15,11).	34
4.2	Different candidate number of parallel K-Best algorithm for RS (15,11) . .	35
4.3	Computation comparison of ABP-ASD and proposed method	35
4.4	Numerical comparison of ABP-ASD and proposed method for RS (15,11) .	35
4.5	Numerical comparison of ABP-ASD and proposed method for RS (31,25) .	36
4.6	Numerical complexity of proposed method of RS (15,11)	36
4.7	Numerical complexity of proposed method of RS (31,25)	36



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 ~ 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 .

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

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

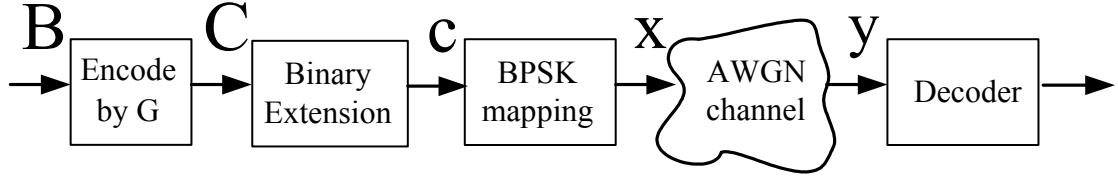


Figure 2.1: Channel model of RS code

Then \mathbf{c} is BPSK modulated from the code symbols to the constellation symbols.

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

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

$$\mathbf{y} = \mathbf{x} + \mathbf{w} \quad (2.3)$$

Where \mathbf{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 $\alpha, \alpha^2, \alpha^3, \dots, \text{and } \alpha^{2t}$, and α is a primitive element in $GF(q^m)$. Therefore the generator polynomial has the following form

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

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 $\alpha^b, \alpha^{b+1}, \alpha^{b+2}, \dots, \text{and } \alpha^{b+2t-1}$ and can be generalized to

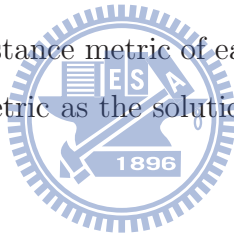
$$g(x) = \prod_{i=b}^{i=b+2t-1} (x - \alpha^i) \quad (2.5)$$

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 $C_{\frac{d_{min}}{2}}^n$ 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-

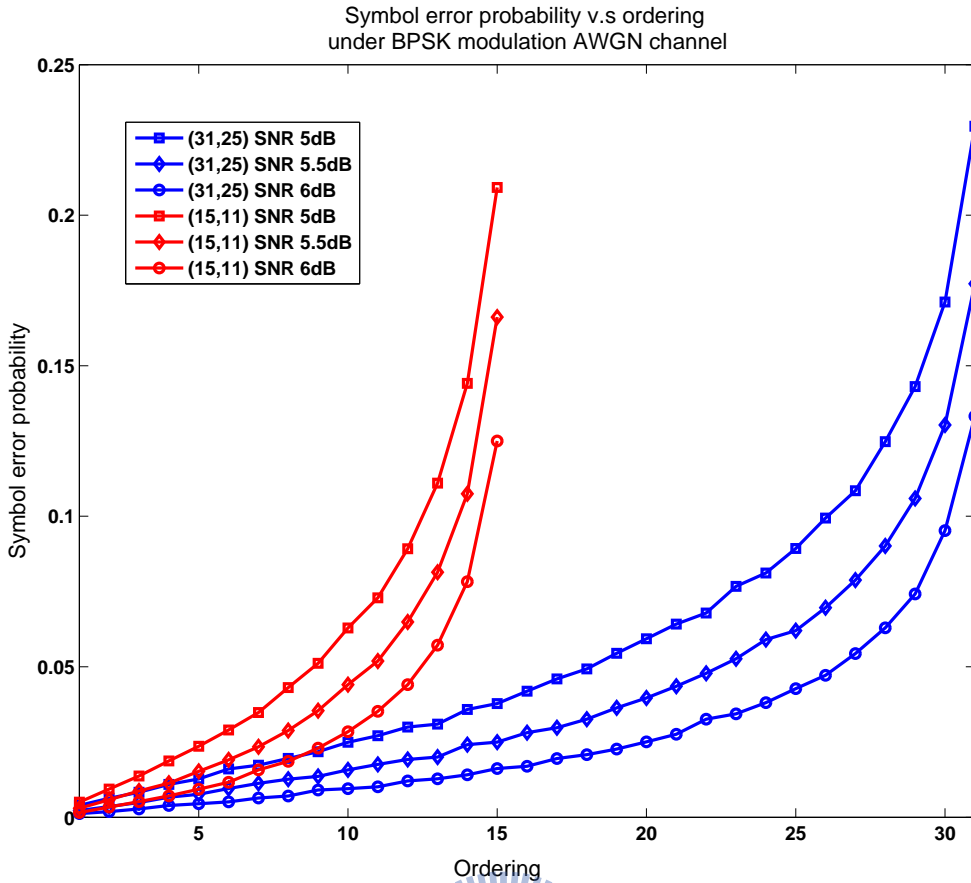


Figure 2.2: Error probability v.s ordered positions RS(15,11) and RS(31,25)

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 reliability. 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 $(f(\alpha_1), f(\alpha_2), \dots, f(\alpha_N))$ the received vector is $(\beta_1, \beta_2, \dots, \beta_N)$. 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) , $l=1, \dots, N$, 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 $f(\hat{x})$ 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 $GF(2^q)$:

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

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

$$\mathbf{y} = \mathbf{c} + \mathbf{w} \quad (2.7)$$

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

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

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.

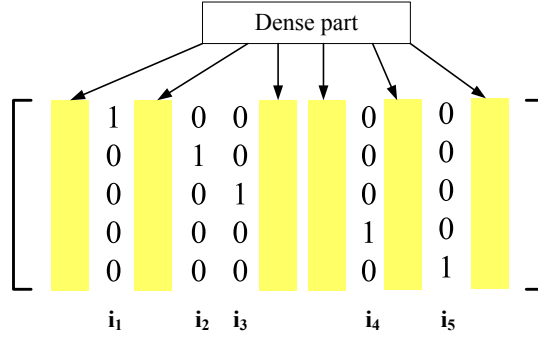


Figure 2.3: Form of the parity check matrix suitable for iterative decoding by row operations

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

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

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

$$H_b^v = \phi(H_b, |L^v|) \quad (2.10)$$

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

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

For each bit, the extrinsic LLR is updated according to

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

The bit-reliability is then updated as

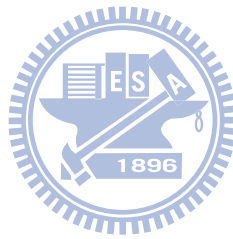
$$L^{(v+1)} = L^{(v)} + \alpha L_{ext}^{(v)} \quad (2.13)$$

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

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

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 \mathbf{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 \mathbf{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.

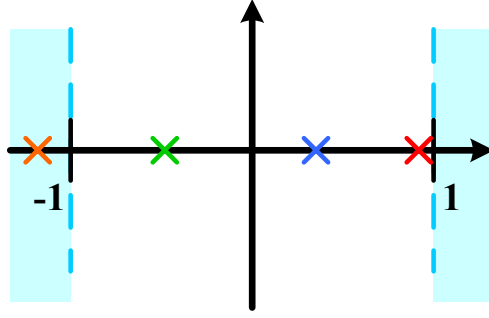


Figure 3.1: BPSK mapping and the distribution of unreliable bits

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 $\mathbf{z} = (z_0, z_1, z_2, \dots, z_{Nq-1})$. The vector \mathbf{z} is transmitted from the binary sequence into symbol sequence \mathbf{Z} where $\mathbf{Z} = (Z_0, Z_1, \dots, Z_{N-1})$. Every symbol Z_j where j is form 0 to $N - 1$ is composed by $(z_{jq}, z_{jq+1}, \dots, z_{jq+(q-1)})$.

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:

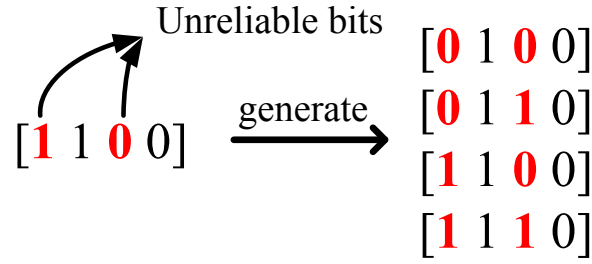


Figure 3.2: Candidates generation based on unreliable bits

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 j th 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 $(\hat{Z}_{j,0}, \hat{Z}_{j,1}, \dots, \hat{Z}_{j,m})$. The selection of m which is between 1 to $N+1$ is obtained by the number of unreliable bits. The candidates of j th 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 $S_j = (S_{j,0}, S_{j,1}, \dots, S_{j,m})$. Where the permutation is denoted as λ_1 .

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

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 .

$$\mathbf{R}_j = \sum_{i=0}^{q-1} \|y_{jq+i}\| \quad (3.2)$$

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

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

The symbols are re-arranged as \mathbf{S}' . We denote the permutation as λ_2 . We selected first k most reliable symbols $(S'_0, S'_1, \dots, S'_{k-1})$ 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}' .

$$S' = \lambda_2(S) \quad (3.3)$$

$$G' = \lambda_2(G) \quad (3.4)$$

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.

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

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}'' .

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

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.

$$G'' = \left[\begin{array}{c|c} I_{k \times k} & \text{---} \end{array} \right]$$

Figure 3.3: Gauss illustration

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Figure 3.4: Example of AG matrix

3.1.4 Re-encoding process

In re-encode method, every S' set must multiply matrix A and then multiply matrix 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 A and G together to be matrix AG . Fig. 3.4 shows an example of the matrix AG . Then every S' set only needs to multiply matrix 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 AG , we discover that there is a special property of matrix AG . The property of matrix 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 AG , we find out that the k reliable symbols are independent.

- 1) We marked the position of 1 in the columns of matrix AG which has only one 1 as index L . $L = (l_0, l_1, l_2, \dots, l_{k-1})$ are indexes corresponding to the most k reliable symbols.
- 2) According to the index vector 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_i th transmitted symbol.

The index would point out the most k reliable symbol corresponding to the l_j th 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 \mathbf{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 j th symbol in each layer where $\mathbf{S}'_j = (S'_{j,0}, S'_{j,1}, \dots, S'_{j,m})$.

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



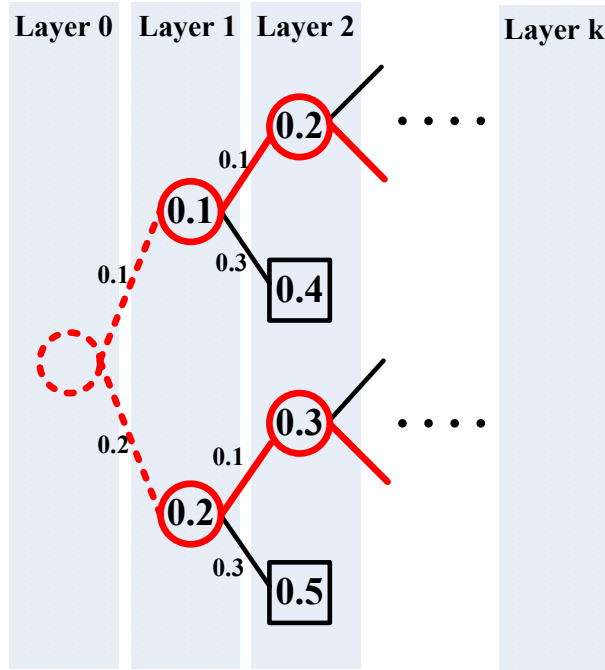


Figure 3.5: Example of K-Best candidate selection, $K=2$

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 summarize 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.

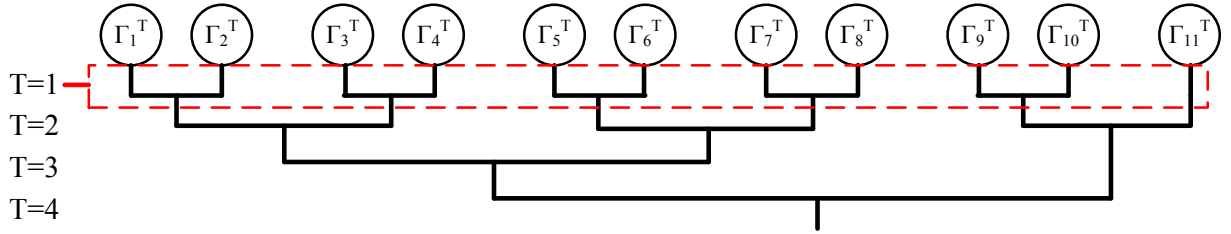


Figure 3.6: Example of parallel K-Best candidate selection with 4 layer for RS(15,11), $T=1, \dots, 4$.

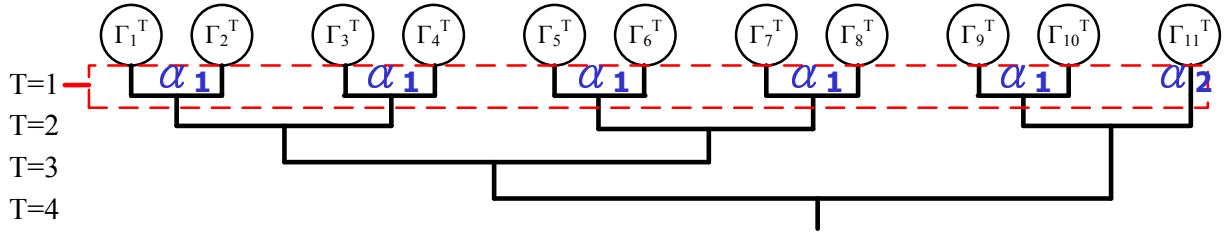


Figure 3.7: Example of subgroup selection in parallel K-Best scheme

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 $\mathbf{\Gamma}_1^T = (\gamma_{1,0}^T, \gamma_{1,1}^T, \dots, \gamma_{1,m_1-1}^T)$ and $\mathbf{\Gamma}_2^T = (\gamma_{2,0}^T, \gamma_{2,1}^T, \dots, \gamma_{2,m_2-1}^T)$, 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?

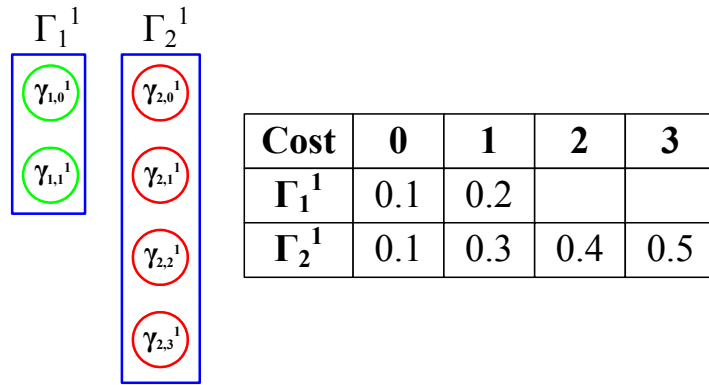


Figure 3.8: Example of subgroup in parallel K-Best scheme.

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 Γ_1^T and displace the candidates in Γ_2^T until all m_2^T candidates are changed. Then we move on to the next candidate of Γ_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 Γ_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 Γ_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

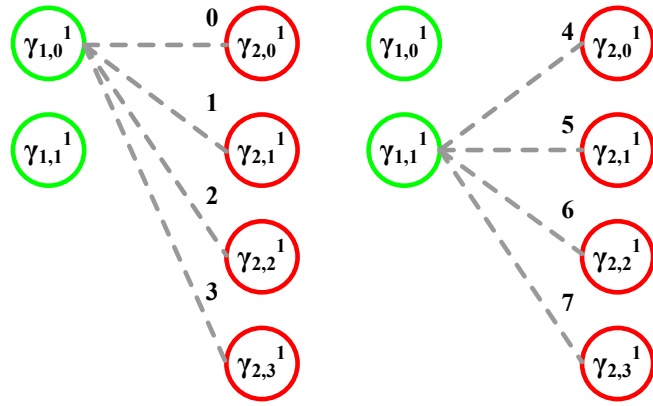


Figure 3.9: Example of path index and candidate sets for parallel K-Best algorithm

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

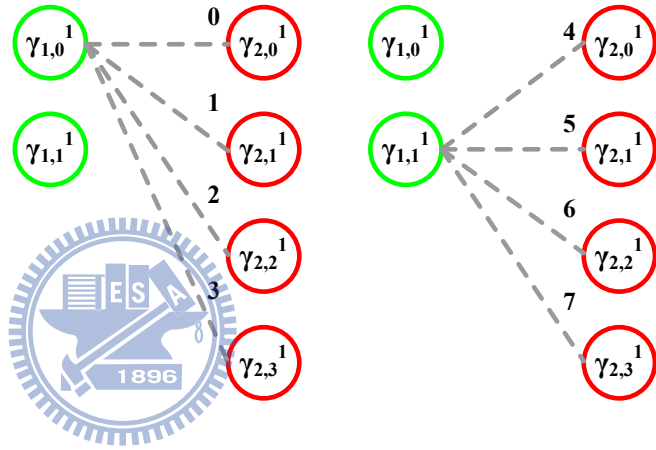


Figure 3.10: Example reduced index for parallel K-Best algorithm

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

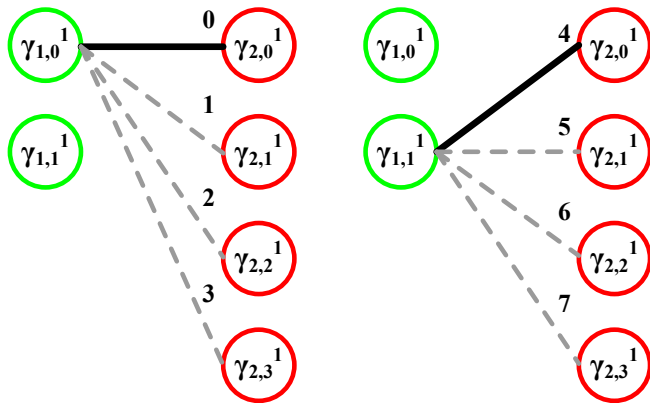


Figure 3.11: Example of chosen K candidate sets for parallel K-Best algorithm

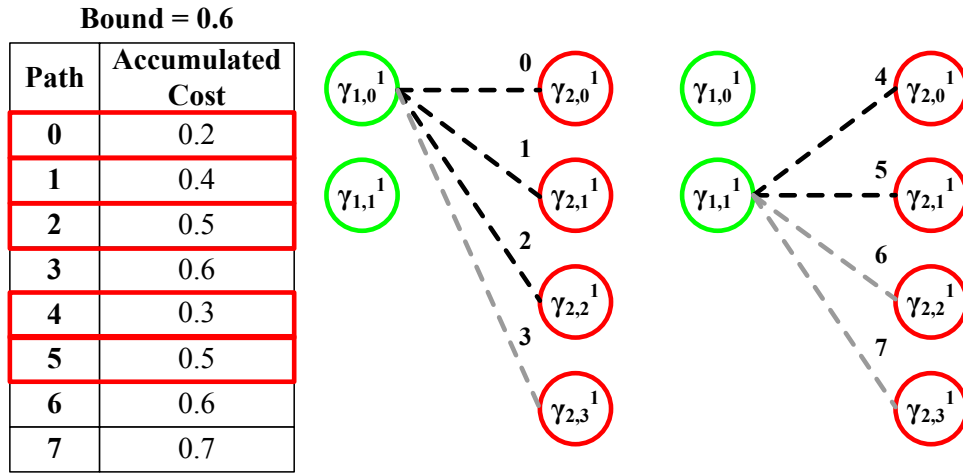


Figure 3.12: Parallel K-Best using constraint

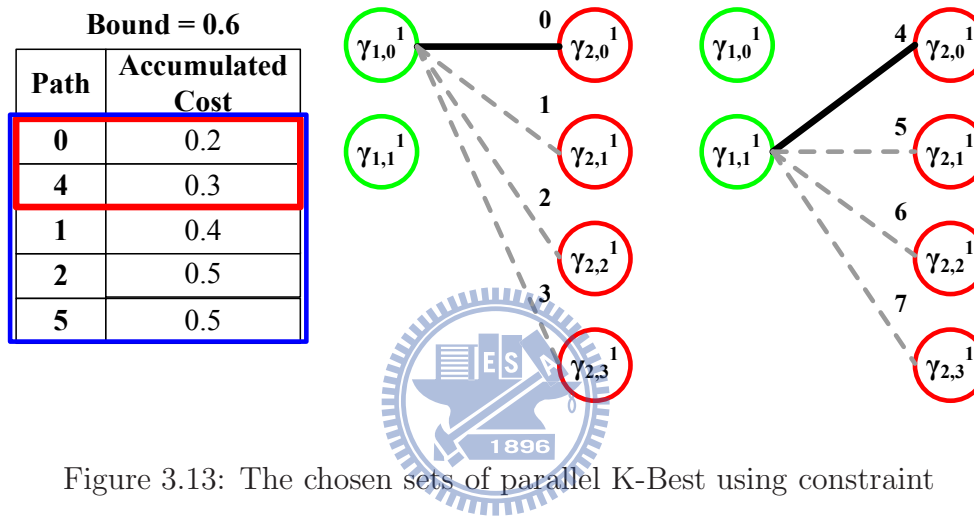


Figure 3.13: The chosen sets of parallel K-Best using constraint

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.

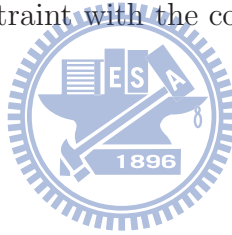
The method mentioned before this section needs \mathbf{S}'' multiply matrix \mathbf{AG} to obtain possible codeword sets. There are just one constraint, to find minimum cost. The possible codeword set is judged by which codeword set has minimum cost. In this section, we do not execute re-encode method 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 than 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 K th 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

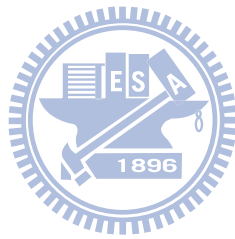
Though we use constraint for accumulated cost and syndrome equation to improve the performance, there still needs a better algorithm. We realize that the accumulated cost of MRIP k symbols being calculated by K -Best should be replaced by cost of all N candidates. Because the codeword sets produced by 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-Massey 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 $\text{SNR} \geq 6\text{dB}$. For RS(31,25), 10^4 bits are simulated when the SNR is below 5dB, and 10^8 bits are simulated for $\text{SNR} \geq 6\text{dB}$.

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.

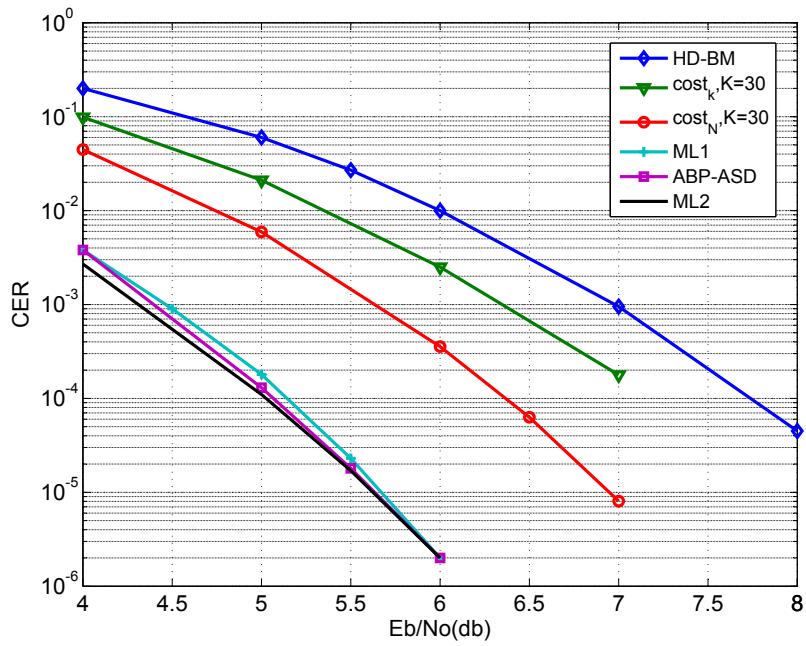


Figure 4.1: Performance comparison for K-Best of RS (15,11) BPSK mapping AWGN channel with different size of distance metric.

*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.

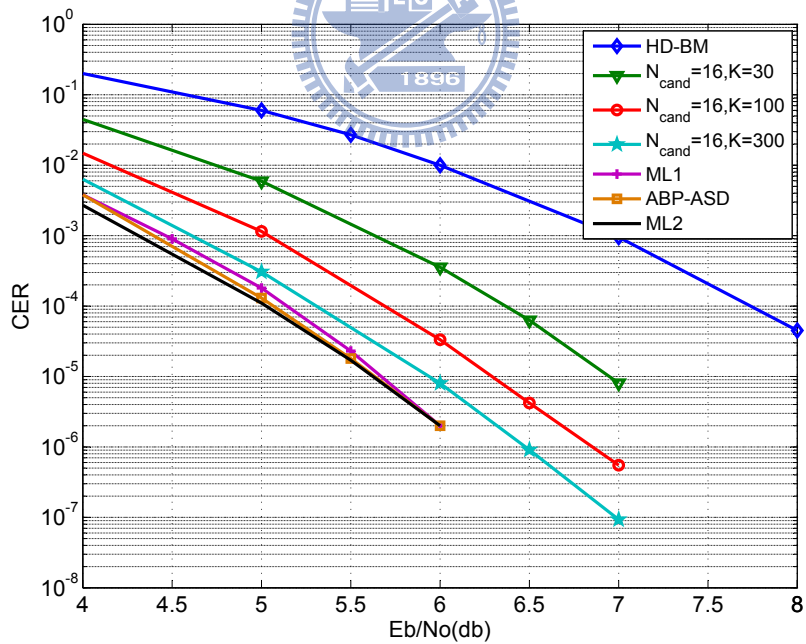


Figure 4.2: Performance comparison for K-Best of RS (15,11) BPSK mapping AWGN channel with different selection of K.

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

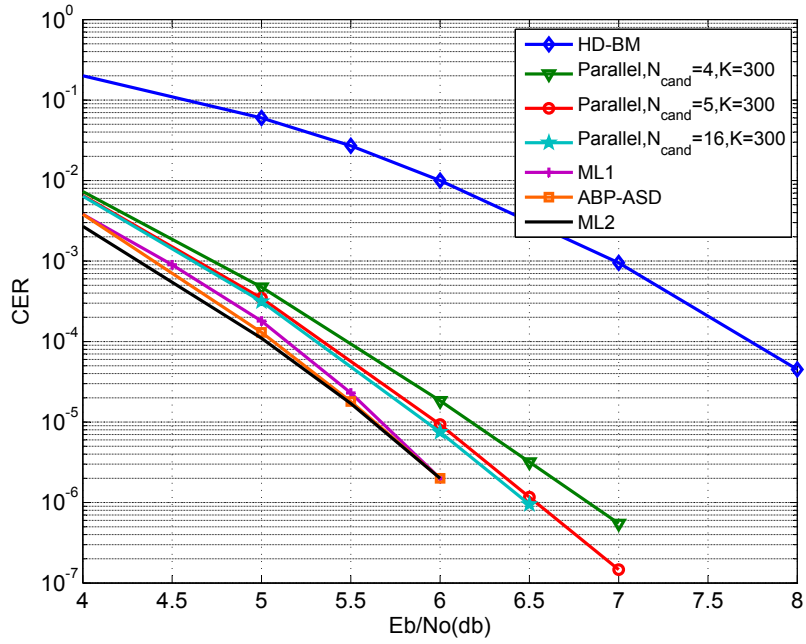


Figure 4.3: Performance comparison for parallel K-Best ($K=300$) of RS (15,11) BPSK mapping AWGN channel with different candidate number

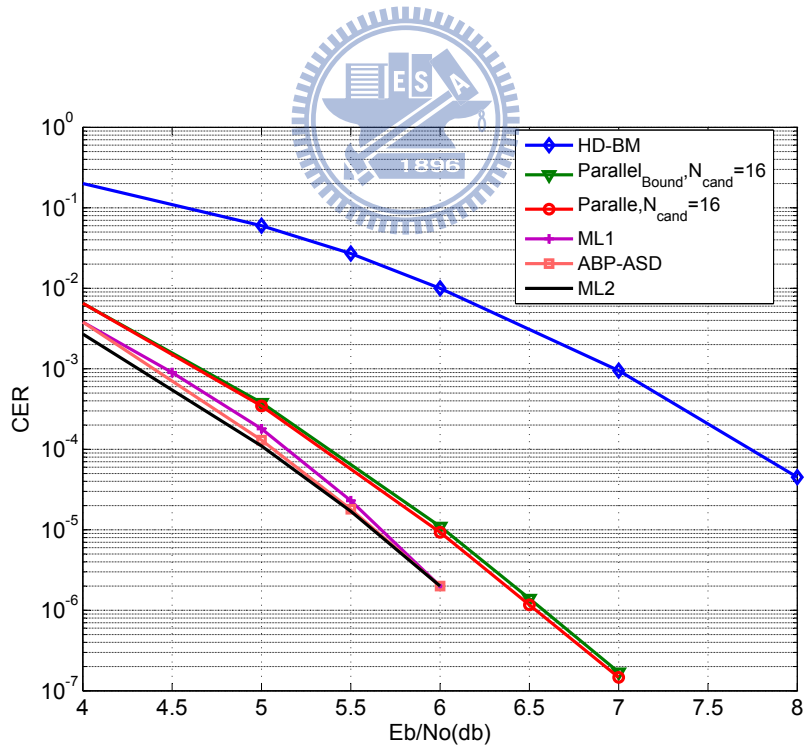


Figure 4.4: Performance comparison for parallel K-Best of RS (15,11) BPSK mapping AWGN channel with constraint

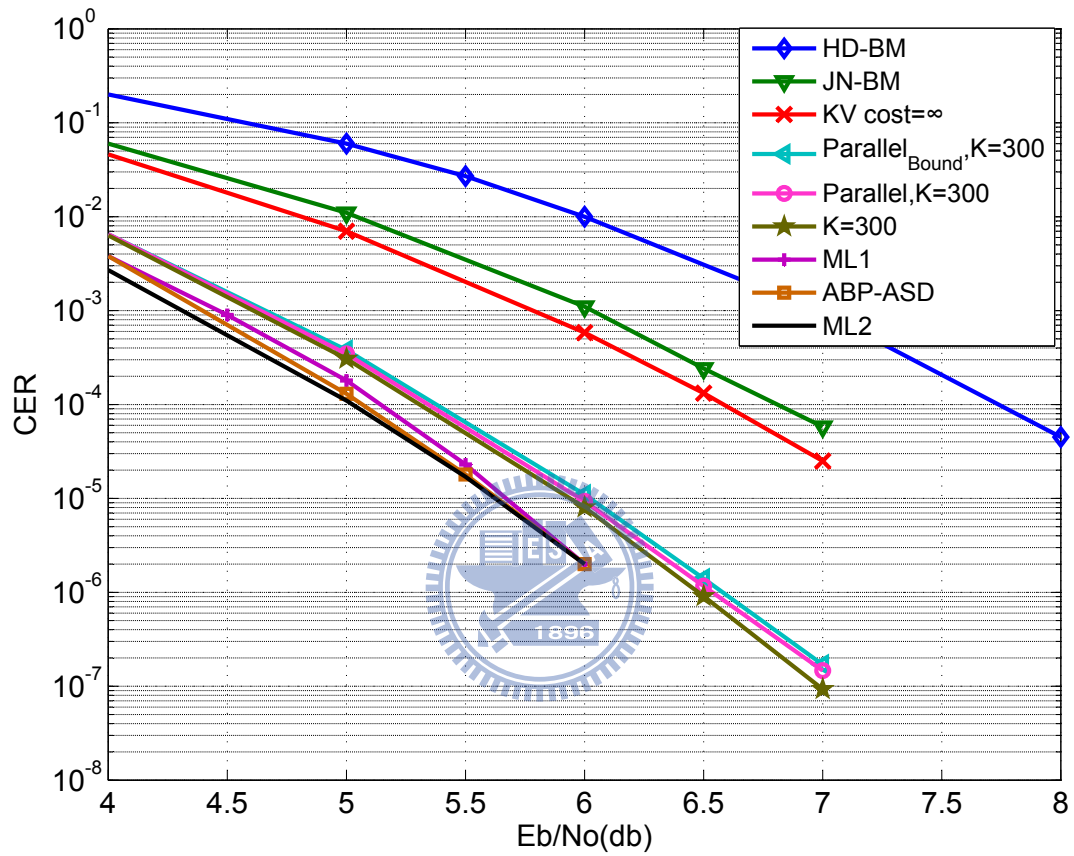


Figure 4.5: Performance of different soft decision decoding algorithm for RS (15,11) BPSK mapping AWGN channel

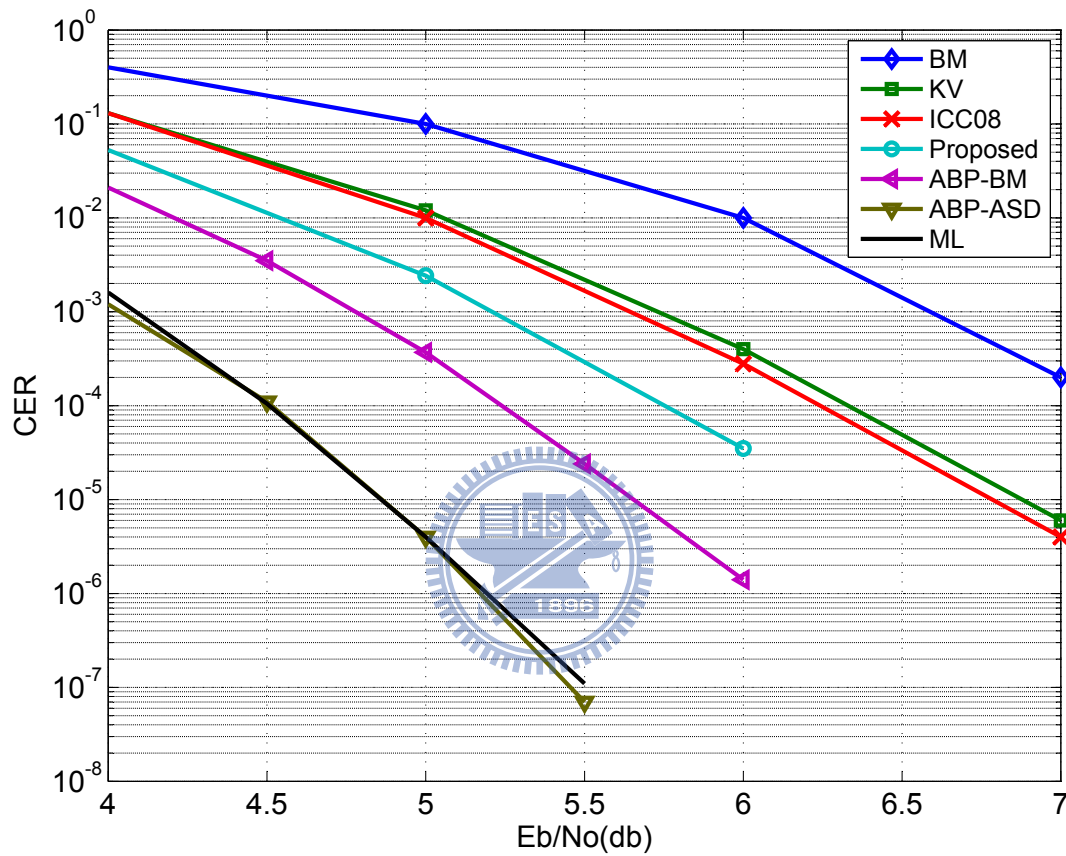


Figure 4.6: Performance for parallel K-Best of RS(31,25) BPSK mapping AWGN channel

Table 4.1: Comparison between K-Best, parallel K-Best and parallel K-Best with constraint for RS (15,11).

	K-Best		Parallel K-Best	
	K=300	K=100	K=300	K=300 with constraint
Normalized comparison	100%	29.83%	27.76%	8.39%
Normalized addition	100%	34.27%	28.99%	28.99%
Performance gain over HD	2.45dB	2.1dB	2.43dB	2.4dB

* The constraint is derived from simulation results.

* The candidate number N_{cand} is 16.

* "Normalized" means we use K-Best=300 as the comparison baseline.

4.2 Complexity comparison

The computation comparison of proposed K-Best algorithm and parallel K-Best algorithm is shown in Table 4.1. HD-BM is the performance baseline. K-Best with $K=300$ is the baseline of normalized comparison and normalized addition, the complexity reduction is about 70% when $K=100$ is chosen, but the performance is 0.35dB loss from $K=300$. For parallel K-Best scheme, $K=300$ is an sufficient number to maintain near same performance as the original K-Best algorithm with $K=300$. The difference is only 0.02dB performance loss. However, there is over 70% comparison and addition computation reduction in this case. We can see that performance gap between K-Best algorithm and parallel K-Best with constraint is only 0.05dB, but the comparison reduction is 91.61%. Since, parallel K-Best with constraint still needs to calculate all the distance metric of the possible candidate, thus there is no computation reduction for addition compare with parallel K-Best.

Table 4.2 shows the comparison of different candidate size for parallel K-Best algorithm. The comparison reduction for candidate=4 and candidate=5 is about 10% from candidate=16, and the addition reduction is 14% and 11%, respectively. The performance loss is 0.13dB and 0.03dB for each case.

Arithmetic complexity is generally written in a form known as Big- O notation. $O(n)$ is the general notation for linear complexity. Where O represents the complexity of the

Table 4.2: Different candidate number of parallel K-Best algorithm for RS (15,11)

	Parallel K-Best		
	$N_{cand}=16$	$N_{cand}=5$	$N_{cand}=4$
Normalized comparison	100%	91.94%	89.61%
Normalized addition	100%	88.9%	85.9%
Performance gain over HD	2.43dB	2.4dB	2.3dB

Table 4.3: Computation comparison of ABP-ASD and proposed method

ABP-ASD			Proposed		
ABP	Floating operation	$O(N'^2)$	Pre-processing	Reliability ordering	$O(N \log N)$
	$i \times \text{GE}$	$i \times O(\min(k', (N' - k')^2) N')$		Candidate generation	$O(N N_{cand} \log N_{cand})$
ASD	Matrix assignment	$O(N^2)$	Candidate selecting	Parallel K-best with constraint	$O(K_{q-1} K_{q-2} \log(K_{q-1} K_{q-2}))$
	Interpolation	$O(N^2 \lambda^4)$		Erasure-only decoding	
	Factorization	$O(l \log^2 l) k (N + l \log q)$			

* Note1: $N' = N \times q$

* Note2: $k' = k \times q$

* Note3: K_{q-1}, K_{q-2} stand for the size of K in layer q-1 and q-2.

* ASD represents the KV algorithm.

Table 4.4: Numerical comparison of ABP-ASD and proposed method for RS (15,11)

ABP-ASD				Proposed			
ABP	Floating operation	3.6 K			Pre-processing	Reliability ordering	18
	$5 \times \text{GE}$	76.9 K				Candidate generation	289
ASD	Matrix assignment	225			Candidate selecting	Parallel K-Best with constraint	29 K
	Interpolation	$\lambda = 4$	$\lambda = 5$	$\lambda = 10$			
			57.6 K	140.6 K	2250 K		
	Factorization	278	743	3417	Erasure-only decoding		4.5 K
Total		138.6 K	222.1 K	2334.1 K	Total		33.9 K

* Note: In the table, K represents 10^3 .

Table 4.5: Numerical comparison of ABP-ASD and proposed method for RS (31,25)

ABP-ASD					Proposed		
ABP	Floating operation	24 K			Pre-processing	Reliability ordering	46
	5×GE	697.5 K				Candidate generation	310
10×ASD	Matrix assignment	9.6 K			Candidate selecting		Parallel K-Best
	Interpolation	$\lambda = 4$	$\lambda = 5$	$\lambda = 10$			
		2460 K	6006 K	96100 K			
	Factorization	12 K	21 K	115 K	Erasure-only decoding		9.3 K
Total		3203 K	6758 K	96946 K	Total		95.4 K

Table 4.6: Numerical complexity of proposed method of RS (15,11)

		Fixed add	abs.	compare	xor	and
Proposed	Re-arrange	45	60	121	0	0
	Candidate generation	0	0	60	120	0
	Parallel K-Best	0	0	2.1K	44.1K	0
	Erasure-only decoding	0	0	0	2K	2.2K
Total		45	60	2.3K	46.2K	2.2K

- * The bits number of fixed addition is defined by the bit number of the channel noise.
- * The bits number of absolute is the same as fixed addition.
- * The compare component with 2 inputs.
- * The xor component with 2 inputs.
- * The and component with 2 inputs.

Table 4.7: Numerical complexity of proposed method of RS (31,25)

		Fixed add	abs.	compare	xor	and
Proposed	Re-arrange	124	155	625	0	0
	Candidate generation	0	0	155	248	0
	Parallel K-Best	0	0	1059K	6155K	0
	Erasure-only decoding	0	0	0	9.9K	10.9K
Total		124	155	1059.8K	6165.1K	10.9K

algorithm and a value n represents the size of the set the algorithm is run against. In this thesis, the sorting complexity can be regarded as $O(N\log N)$ [17], which includes the heap, merge and quick sorts.

In this thesis, we compare the computation complexity with ABP-ASD algorithm [6], since it is known as the best soft RS decoding performance, recently. We divide the ABP-ASD algorithm into two parts, the pre-processing part and decoding part. Table 4.3 compare each process with our proposed parallel K-Best algorithm, respectively. The ABP step involves $O(N'^2)$ floating operations, for sorting and BP, and $O(\min(k', (N' - k')^2)N')$ binary operations for Gauss elimination [5]. For ASD, matrix assignment requires $O(N^2)$ time complexity, interpolation needs time complexity with $O(N^2\lambda^4)$. An efficient factorization algorithm with a time complexity $O((l\log^2 l)k)(N + l\log l)$, where l is an upper bound on the ASD's list size and is determined by λ .

Table. 4.4 shows the numerical comparison between ABP-ASD and proposed method for RS(15,11). It is observed that the dominate terms of ABP-ASD are Gauss elimination and interpolation. And for ASD, it needs 5 iteration of Gauss elimination corresponding to Fig. 4.5. The dominate term in proposed method is parallel K-Best operation. We substitute the number into the parameters and compare the complexity. Comparing with ABP-ASD, the proposed scheme has 75.5% complexity reduction of $\lambda = 4$. And there is 84.7% complexity reduction of $\lambda = 5$. As for $\lambda = 10$, there is 98.5% complexity reduction. ABP-ASD has 0.3dB performance gain with proposed method at CER of 10^{-4} .

Table. 4.5 shows the numerical comparison between ABP-ASD and proposed method for RS(31,25). We substitute the number into the parameters and compare the complexity. Comparing with ABP-ASD, the proposed scheme has 97% complexity reduction of $\lambda = 4$. And there is 98.6% complexity reduction of $\lambda = 5$. As for $\lambda = 10$, there is 99.9% complexity reduction. ABP-ASD has 1.25dB performance gain with proposed method at CER of 10^{-4} .

The difference between KV and ABP-ASD is that KV did not do the ABP process. In fact, KV operation is the same as ASD.

In Table 4.4, the numerical comparison between KV and proposed scheme for RS(15,11) is shown. The dominate term in KV algorithm is interpolation. Comparing with KV algorithm, the proposed method has 41.7% complexity reduction and at least 1.3dB per-

formance gain. Since, the performance of KV algorithm shown in Fig. 4.5 with infinite complexity cost.

In Table 4.5, the numerical comparison between KV and proposed scheme for RS(31,25) is shown. The dominate term in KV algorithm is interpolation. Comparing with KV algorithm, the proposed method has 61.6% complexity reduction and at least 0.6dB performance gain at CER of 10^{-4} . Since, the performance of KV algorithm shown in Fig. 4.6 with infinite complexity cost.

Next, we shows the complexity of our proposed algorithm, parallel K-Best with sphere constraint. Unlike the ABP-ASD algorithm decompose the code and parity-check matrix into binary image, we focus on the symbol level decoding. We also divide the decoding process into two part, pre-processing includes re-arrange received soft values and the candidate generation. Each have a time complexity, $O(N\log N)$ and $O(NN_{cand}\log N_{cand})$. For the decoding part, parallel K-Best algorithm with sphere constraint needs $O(K_{q-1}K_{q-2}\log(K_{q-1}K_{q-2}))$, and erasure only decoding have a time complexity with $O(KN)$.

Table. 4.6, 4.7 show the computation complexity of different decoding steps. The re-arrange steps includes the floating addition, absolute operation, and re-ordering the received value. Since we operate the order decoding steps from re-order G to erasure only decoding in symbol level. The computation calculation is slightly different from binary case. We consider the worst case for symbol level multiplication. There are $q \times q$ **and** operation, and $(q-1) \times q$ **xor** operation while our symbol is considered in $GF(2^q)$.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

In this thesis, the soft RS decoder based on K-Best algorithm with constraint is proposed and divided into three parts, pre-processing, candidate selecting, and erasure only decoding. In pre-processing, the decoder uses the soft information to derive the reliability for each symbol. It is observed from the re-encoding process that k reliable symbols are independent. In candidate selecting, the K-Best algorithm is introduced to reduce the computation number of the possible candidate sets. In order to accelerate the decoding speed and reduce the computation complexity of the K-Best algorithm, the parallel K-Best algorithm is introduced. In parallel K-Best algorithm, we divided the candidate sets into several groups. Each subgroup chooses local K-Best candidates, and the process is kept running until the q layers are completed. Then, a constraint is introduced to reduce the computation complexity. Only the sets with distance metrics smaller than the constraint are needed to be computed. Erasure only decoding uses the property of the RS code. By using the syndrome equation, the first k reliable symbols can be used to decode the other $N - k$ unreliable symbols. The candidate codeword of the N symbols with the smallest distance metric will be determined as the transmitted codeword.

In conclusion, the computation complexity of the proposed parallel K-Best algorithm with constraint for RS(15,11) outperforms the HD-BM algorithm by 2.4dB and the KV algorithm by 1.3dB at a codeword error rate (CER) of 10^{-4} . Comparing with the ABP-ASD algorithm, there is a 0.3 dB performance gap at CER 10^{-4} . The complexity reduction is at

least 41.7% and 75.5% for KV and ABP-ASD algorithm, respectively. 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 performance gap between ABP-ASD and proposed method is 1.25 dB at CER of 10^{-4} . The complexity reduction is at least 61.6% and 97% for KV and ABP-ASD algorithm, respectively.

5.2 Future work

There are still some issues to be concerned in our algorithm. First, the algorithm is suited for short RS code such as (15,11) and (31,25). For longer code, a large K is required for the parallel K-Best method to maintain the performance. Second, the complexity of the parallel K-Best with sphere is still needed to be improved. One can generate all k reliable symbols at once and decode the other $N-k$ erasures. Then, the decoded codeword can derive N distance metric. Based on these N distance metric, K-Best algorithm can be used to choose K possible codewords. The codeword with minimum distance metric will be chosen as the output codeword. Further, our algorithm can also be concatenated with other techniques such as interleaver to improve the error control capability.



Chapter 6

Appendix: several techniques used for the proposed soft RS decoding

Soft decision RS decoding have recently drawn significant attention. Soft decoding takes advantage of the information from channel ,and it is believed to have 2-3dB performance gain over HD-BM algorithm. The following introduces the simplified cost function ,and it briefly explains reliability based decoding. The re-encode process is also described in this section. Further, the sphere decoding algorithm and K-Best algorithm are mentioned for the reduction of candidate selecting. At last, erasure-only decoding is discussed to take advantage of RS code.

6.1 Simplified cost function

The traditional cost function is calculated by the Euclidian distance or Hamming distance. The conditional probability is that given x is transmitted, the received value is y . The conditional probability is the bigger the better.

$$p(y | x) = \frac{1}{(\pi N_0)^{-n/2}} \exp \left\{ - \sum_{i=0}^{n-1} \frac{(y_i - c_i)^2}{N_0} \right\} \quad (6.1)$$

Here, c is the bit of BPSK mapping and x is the transmitted codeword bit.

We observe eq.(6.1) that the exponential term without N_0 is Euclidian distance. As the Euclidian distance being smaller the conditional probability is bigger. Next, we only consider the Euclidian distance as cost function. The Euclidian distance formula could

be rewritten as following [15]

$$d_E^2(y, c) = \sum_{i=0}^{N-1} (y_i - c_i)^2 \quad (6.2)$$

$$= \sum_{i=0}^{N-1} y_i^2 + n - 2 \sum_{i=0}^{N-1} y_i c_i \quad (6.3)$$

where n is a constant, and $\sum_{i=0}^{N-1} y_i c_i$ is a common term. Because the received values are fixed for the same codeword, the constant and the common term can be neglected. Minimizing Euclidian distance is equivalent to maximizing the variable term.

$$m(y, c) \triangleq \sum_{i=0}^{N-1} y_i^2 \quad (6.4)$$

We rewrite the variable term as following

$$m(y, c) = \sum_{i=0}^{N-1} |y_i| - 2 \sum_{i: y_i c_i < 0} |y_i|, \text{ Let } \lambda(y, c) \triangleq \sum_{i: y_i c_i < 0} |y_i| \quad (6.5)$$

$$m(y, c) = \sum_{i=0}^{N-1} |y_i| - 2\lambda(y, c) \quad (6.6)$$

It shows that maximizing the variable term is equivalent to minimizing $\lambda(y, c)$. The condition probability is only affected by $\lambda(y, c)$. The cost function can be replaced by $\lambda(y, c)$.

6.2 Re-encode decoding based on reliability ordering

Reliability based decoding and re-encode process can be referred in [7]. Suppose an (N, k) Reed Solomon (RS) code with generator matrix \mathbf{G} is used for error control over AWGN channel. The items in RS code are elements of Galois Field $GF(2^q)$ where $N = 2^q - 1$. Let $\mathbf{C} = (C_1, C_2, \dots, C_N)$ be a codeword. For transmission, the binary extension of codeword, $\mathbf{c} = (c_1, c_2, \dots, c_{Nq})$, is mapped into the bipolar sequence $\mathbf{x} = (x_1, x_2, \dots, x_{Nq})$ with $x_i = (-1)^{c_i} \in \pm 1$. After the transmission, the received sequence in the demodulator is $\mathbf{y} = (y_1, y_2, \dots, y_{Nq})$ with $y_i = x_i + w_i$, where $1 \leq i \leq Nq$, w_i 's are statistically independent Gaussian random variables with zero mean and variance $N_0/2$.

The hard decision symbol of received sequences is $\mathbf{Z} = (Z_1, Z_2, \dots, Z_N)$. The decoding begins with reordering the elements of received symbols according to the reliability values.

The reordered sequence is denoted as $\mathbf{S} = (S_1, S_2, \dots, S_N)$. The reliability of sequence \mathbf{S} is in decreasing order by degrees. This reordering defines a permutation function λ_1 for which

$$\mathbf{S} = \lambda_1(\mathbf{Z}). \quad (6.7)$$

Then the columns of the generator matrix \mathbf{G} are permuted based on λ_1 . The result matrix is denoted

$$\mathbf{G}' = \lambda_1(\mathbf{G}) = [g'_1, g'_2, \dots, g'_N] \quad (6.8)$$

where the g'_i means the i th column of \mathbf{G}' . The column g'_i of \mathbf{G}' corresponds to the S_i of \mathbf{S} with i th reliability value.

The first k components of \mathbf{S} are the most reliable symbols. Use the first k columns to perform the Gaussian elimination to remove the dependency of each most k reliable symbols. This Gaussian elimination defines a permutation function λ_2 for which

$$\mathbf{G}'' = \lambda_2(\mathbf{G}') = [I_k P] \quad (6.9)$$

Here, I_k is the $k \times k$ identity matrix, and P is the $k \times (N - k)$ parity check matrix. The corresponding most k reliable symbols perform the same operation based on λ_2 . The obtained sequence is denoted as

$$\mathbf{S}' = \lambda_2(\mathbf{S}) \quad (6.10)$$

After the permutation λ_2 , the first k components of sequence \mathbf{S}' are the possible transmission messages. As implied from the name, re-encode algorithm uses possible message sets to multiply generator matrix \mathbf{G} . After re-encoding, the cost values between possible codeword sets and the received values are calculated. Then, the codeword with minimum cost is considered as the transmitted codeword.

6.3 Sphere decoding and K-Best algorithm

Sphere decoding (SD) algorithm has been proposed, and it's recognized as a powerful means to solve the maximum likelihood (ML) detection problem Ref [18]. SD algorithm reduces the computation by restricting the search range. We define all the candidates are in Θ , then SD algorithm constrains a much smaller search range $\Theta_{SD} = \{s : \lambda(r, s) < bound\}$;

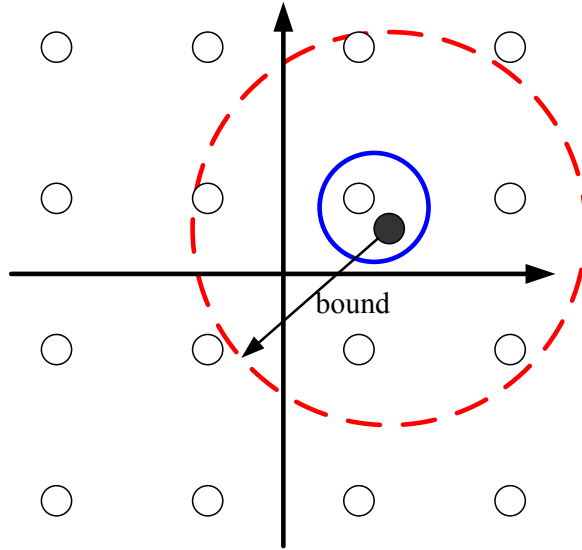


Figure 6.1: Illustration of sphere decoding algorithm

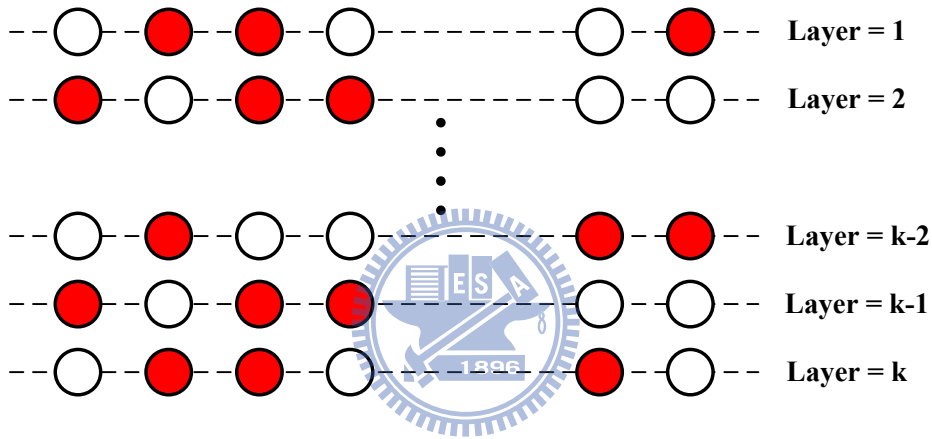


Figure 6.2: Traditional K-Best algorithm

only the candidates fall in Θ_{SD} , and their cost is smaller than the bound will be compared. By the previous procedure, the chosen candidate is always the ML solution as long as d is properly defined. The problem can be illustrated as a two-dimensional problem in Fig. 3.1, the solution can be obtained by drawing a circle around the received signal, and choose the proper radius to discard the point outside of the circle. In [16], the author suggests to use sphere decoding to decode the soft RS code. For AWGN channel, a sphere constraint is introduced to reduce the candidate search in decoding process.

K-Best algorithm is an alternative method that improves the decoding throughput [19]. It simplified the original SD algorithm and maintain a constant throughput by keeping only the smallest accumulated cost value at each layer. However, K-Best algorithm can

not guarantee the performance of ML detection since the ML solution may be eliminated when it is not in the K-Best accumulated cost values. Thus, the larger K is required and the value K becomes a tradeoff between complexity and error performance. Fig. 3.2 illustrates that there are chosen K paths (combination sets) in each layer .

6.4 Erasure-Only Decoding

It is known that any k symbols in a RS codeword can be used to determine the remaining $N - k$ symbols. Thus, it is important to decide k reliable messages or $N - k$ unreliable (error) symbols of codeword. Let $\mathbf{S}_{\mathbf{K-Best}}$ be the K chosen combination sets composed with the candidates of most k reliable symbols.

$$\mathbf{S}_{\mathbf{K-Best}} = \{S = (s_0, s_1, \dots, s_{k-1}) | s_i \in Z_i\} \quad (6.11)$$

where s_i is the candidate of i th symbol Z_i .

In this thesis, the traditional erasure-only decoding is replaced with algebraic term solution. The unreliable symbols can be obtain by brought the reliable symbols into the syndrome equations $Z(x)$. There are total $2t$ syndrome equations $Z(\alpha), Z(\alpha^2), Z(\alpha^3) \dots Z(\alpha^{2t})$, and thus there are $2t$ erasures.

For example, in a (15,11) RS code the received sequence is $Z = (9, 3, 14, 4, *, 12, 6, 14, 2, *, 9, *, 11, *, 11)$ where * represents the erasures.

$$Z(x) = z_0 + z_1x + z_2x^2 + \dots + z_{14}x^{14} = 0 \quad (6.12)$$

$$Z(\alpha) : z_4x^4 + z_9x^9 + z_{11}x^{11} + z_{13}x^{13} = \alpha^4 \quad (6.13)$$

$$Z(\alpha^2) : z_4x^8 + z_9x^3 + z_{11}x^7 + z_{13}x^{11} = \alpha^4 \quad (6.14)$$

$$Z(\alpha^3) : z_4x^{12} + z_9x^{12} + z_{11}x^3 + z_{13}x^9 = \alpha^8 \quad (6.15)$$

$$Z(\alpha^4) : z_4x^1 + z_9x^6 + z_{11}x^{14} + z_{13}x^7 = \alpha^5 \quad (6.16)$$

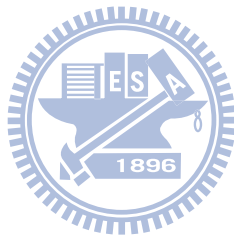
The variable x in different equations presents a exponent relation. Therefore the erasures can be solved with Vandermonde matrix.

Then, the erasure only decoding schedule is given as following:

- 1) For all $S \in S_{\mathbf{K-Best}}$, erase the corresponding $N - k$ unreliable symbols of chosen set S. Then, decode the erasures by syndrome equations and solve the equation by Vandermonde

matrix. The decoding codeword set is called C_E .

2) Choose the codeword from C_E with the smallest accumulated cost by K-Best algorithm as the decoding output.



Bibliography

- [1] E. R. Berlekamp, G. Seroussi, and P. Tong, *Reed-Solomon Codes and Their Application*. Piscataway, NJ: IEEE Press, 1968.
- [2] V. Guruswami and M. Sudan, “Maximum-likelihood decoding of Reed-Solomon codes is NP-hard,” *IEEE Trans. Inf. Theory*, vol. 51, pp. 2249–2256, July 2005.
- [3] R. Kotter and A. Vardy, “Algebraic soft-decision decoding of Reed-Solomon codes,” *IEEE Trans. Inf. Theory*, vol. 49, no. 11, pp. 2809–2825, Nov. 2003.
- [4] V. Guruswami and M. Sudan, “Improved decoding of Reed-Solomon codes and algebraic geometry codes,” *IEEE Trans. Inf. Theory*, vol. 45, pp. 1757–1767, Sep. 1999.
- [5] J. Jiang and K. R. Narayanan, “Iterative soft-input soft-output decoding of Reed-Solomon codes by adapting the parity-check matrix,” *IEEE Trans. Inf. Theory*, vol. 52, no. 8, pp. 3746–3756, Aug. 2006.
- [6] M. E. Khamy and R. McEliece, “Iterative algebraic soft decision list decoding of Reed-Solomon codes,” *IEEE Journal on selected areas in communications*, vol. 24, pp. 481–489, Mar. 2006.
- [7] M. Fossorier and S. Lin, “Soft decision decoding of linear block codes based on ordered statistics,” *IEEE Trans. Inf. Theory*, vol. 41, pp. 1379–1396, Sep. 1995.
- [8] G. Forney, “Generalized minimum distance decoding,” *IEEE Trans. Inf. Theory*, vol. IT-12, no. 2, pp. 273–276, 2002.
- [9] D. Chase, “A class of algorithms for decoding block codes with channel measurement information,” *IEEE Trans. Inf. Theory*, vol. IT-18, no. 1, pp. 170–182, Jan. 1972.

- [10] S. L. H. Tang, "On combining chase-2 and GMD decoding algorithms for nonbinary block codes," *IEEE Commun. Lett.*, vol. 5, no. 5, pp. 209–211, May 2001.
- [11] L. E. Aguado and P. G. Farrel, "On hybrid stack decoding algorithms for block codes," *IEEE Trans. Inf. Theory*, vol. 44, no. 1, pp. 398–409, Jan. 1998.
- [12] J. Belzile and D. Haccoun, "Bidirectional breadth-first algorithms for the decoding of convolutional codes," *IEEE Trans. Inf. Theory*, vol. 41, no. 2, pp. 370–380, Feb. 1993.
- [13] E. R. Berlekamp, *Algebraic coding theory*. New York: McGraw-Hill, 1968.
- [14] R. Bose and D. K. Ray-Chaudhuri, "On a class of error correcting binary group codes," *Inform. Control*, Mar. 1960.
- [15] S. Lin and D. Costello, *Error Control Coding-Fundamentals and Applications*. 2nd ed. Pearson Prentice Hall, 2004.
- [16] F. Shayegh and M. Soleymani, "Soft decision decoding of Reed-Solomon codes using sphere decoding," *IEEE International Conference on Communication(ICC) 2008*, May 2008.
- [17] T. Cormen, C. Leiserson, R. Rivest, and C. Stein, *Introduction to algorithms*. 2nd ed. McGraw-Hill Book Company, 2003.
- [18] B. Hassibi and H. Vikalo, "On the sphere-decoding algorithm I. Expected complexity," *IEEE Trans. Signal Processing*, vol. 53, no. 8, pp. 2806–2818, Aug. 2005.
- [19] H. Chang, Y. Liao, and H. Chang, "Low-complexity prediction techniques of K-Best sphere decoding for mimo systems," *IEEE Workshop on Signal Processing Systems(SiPS) 2007*, Oct. 2007.

作者簡歷

姓名：鄭晶今

出生地：台灣 宜蘭縣

出生日期：1984.11.3

學歷：2000.9~2003.6 國立羅東高級中學

2003.9~2007.6 國立清華大學 工程與系統科學系 學士

2007.9~2009.10 國立交通大學 電子研究所 系統組 碩士

