CONTRIBUTIONS TO FOLDED REED-SOLOMON CODES FOR BURST ERROR CORRECTION

ZHANG JIANWEN

NATIONAL UNIVERSITY OF SINGAPORE

2008

CONTRIBUTIONS TO FOLDED REED-SOLOMON CODES FOR BURST ERROR CORRECTION

ZHANG JIANWEN

(*B. Eng., M. Eng., HUST*)

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2008

Acknowledgment

Thanks to my supervisor, Dr. Armand, for his patience and guidance in these four years. I learned a lot from him in shaping ideas and writing technical papers. Thanks to Dr. Xin Yan. His encouragement is greatly appreciated. I thank him for his time and kindness. Thanks to Professor P. Y. Kam and Professor C. S. Ng. Their lectures helped me understand concepts in digital communications and random processes.

Thanks to my friends in ECE-I^2R-CWC lab and Communications lab. The discussion with Jiang Jinhua, Zhang Lan, Gao Feifei, Khaisheng, Anwar, Zhang Qi and Chong Hon Fah are helpful. Talking with Cao Wei, He Jun, Lu Yang, Li Yan, Li Rong, Li Mi, Zhu Yonglan, Lokesh and Cao Le is joyful. The days traveling with Kim Cheewee in Helsinki was relaxing and the discussion with him was very interesting. Thanks to Eric in ECE-I^2R-CWC. Thanks to Sun Zhenyu, Bao Qingming, Gao Xiang and Dai Zhenning for providing me a lot of help during these four years.

Thanks to my parents. I am indebted to them for their love, tenderness and patience over these four years. I could not go through my education without their understanding and support. Thanks to my wife Liu Jing. I thank her for her love, support and understanding. Thanks to my younger sister. She has brought us a lot precious memories.

Contents

Summary

We show that Folded Reed-Solomon (FRS) codes can be constructed from any Reed-Solomon (RS) code with codelength a composite number. The zeros of the row codes of the resulting code array are shown to be a distribution of the zeros of the original RS code. FRS codes can be used to correct burst errors when the code array is transmitted column by column in burst error channels. To detect burst errors effectively, Transformed Folded Shortened RS codes and a corresponding decoding algorithm based on the Guruswami-Sudan Algorithm (GSA) are proposed. Estimates of the probability of successful decoding, decoder error and decoding failure for this algorithm are derived.

A RS code is often encoded by its generator polynomial. The output of the GSA on this code is a coset of the candidate messages. How to recover the candidate messages from this coset is studied in this thesis. A relation between the codeword resulting from the generator-matrix-based encoding and the codeword obtained via the evaluation map is established. Based on this relation, a transform for retrieving the generator-polynomial-based coded message data under the interpolation-based list decoding is derived. To retrieve the message data, an average computational overhead of $\mathcal{O}(k^2)$ is required for an (n, k) RS code.

It is also shown that folded codes can be constructed from Generalized RS (GRS) codes with codelengths being composite numbers. The resulting arrays are codewords of a Folded GRS (FGRS) code. The rows in the resulting array can be modified as GRS codes with zeros from the same support set. However,

the syndromes of this row codes may not be consecutive. Also, a method for the synthesis of multisequences with unknown elements in the middle is derived. Based on this method, a decoding algorithm for decoding these FGRS code is proposed.

A search-type list decoding algorithm is proposed for an (n, k) RS code. This algorithm can correct up to $n - k - 1$ errors in the list decoding sense. We show that for short, high rate codes, it is possible that the average complexity of the proposed search procedure is less than n^2 at Word Error Rates (WER's) of practical interests. This algorithm can be applied to decode FRS codes. An appropriate choice of dimension for the code array will thus permit the proposed algorithm to be applied with reasonable complexity at practical WER's.

Finally, a list decoding algorithm based on Gröbner Bases (GB) and Generalized Newton's Identities (GNI) is studied. The GB are from the relation $x^{\deg(\sigma(x))}\sigma(x^{-1})h(x) = x^n - 1$, where $\sigma(x)$ is the error locator polynomial and $h(x) = \frac{x^n-1}{x^{\deg(\sigma(x))}\sigma(x^{-1})}$. The group of linear equations from GNI for a received vector are combined with the GB. The solutions are the possible error locator polynomials for the received vector. We also apply this method to decode some cyclic codes over GF(8) with restricted error values.

List of Tables

List of Figures

Abbreviations

ARQ	automatic repeat-request.
AWGN	additive white Gaussian noise.
BCH	Bose-Chaudhuri-Hocquenghem.
BER	bit-error-rate.
BMA	Berlekamp-Massey algorithm.
BPSK	binary phase shift keying.
BSC	binary symmetric channel.
CD	Compact disk.
EA	Euclid algorithm.
FGRS	folded generalized Reed-Solomon.
FIA	fundamental iterative algorithm.
FRS	folded Reed Solomon.
GB	Gröbner Bases.
GFFT	Galois field Fourier transform.
GIAMS	Generalized Iterative Algorithm for Multiple Sequences.
GRS	generalized Reed-Solomon.
GSA	Guruswami-Sudan algorithm.
GMD	generalized minimum distance.
GNI	generalized Newton's identities.

IRS	interleaved Reed-Solomon.
KVA	Koetter-Vardy algorithm.
LHS	left hand side.
LLR	log-likelihood ratio.
LDPC	low density parity check.
MDS	maximum distance separable.
ML	maximum Likelihood.
NMDS	nearly maximum distance separable.
PGZA	Perteson-Gorenstein-Zierler Algorithm.
RS	Reed-Solomon.
RSC	recursive systematic convolutional.
SNR	signal-to-noise-ratio.
TFSRS	transformed folded shortened Reed-Solomon.
WER	word-error-rate.

Notations

In this thesis, scalar variables are written as plain letters, row vectors as bold-face lower-case letters, and matrices or arrays as bold-face upper-case letters. Some further used notations and commonly used acronyms are listed in the following:

$\mathcal{C}$	linear block code.
$d(\mathbf{c}_1, \mathbf{c}_2)$	Hamming distance of vectors $\mathbf{c}_1$ and $\mathbf{c}_2$.
$\deg(f(x))$	degree of polynomial $f(x)$.
$\mathrm{Diag}(\mathbf{a})$	diagonal matrix with $\mathbf{a}$ being the vector of elements in the main diagonal.
$\mathrm{GF}(q)$	the finite field with q elements.
$\mathrm{GF}(q)[x]$	the polynomial ring over $\mathrm{GF}(q)$.
$\mathrm{GF}(q)[x]_k$	the polynomial ring over $\mathrm{GF}(q)$ and $\deg(f(x)) < k$, $\forall f(x) \in \mathrm{GF}(q)[x]_k$.
$\mathrm{ord}(\alpha)$	order of a field element α.
$w(\mathbf{c})$	Hamming weight of vector $\mathbf{c}$.

Chapter 1

Introduction

In this chapter, the background of error-correcting codes for point-to-point communications and an overview of this thesis are given. Section 1.1 introduces briefly the point-to-point communication model, how error-correcting codes help achieve reliable communications in the presence of ambience noise in this model, and the development of error-correcting codes since the 1950's. Section 1.2 goes through some current research topics and the challenges in the field of error-correcting codes. Section 1.3 describes the motivation and objective of the work on Folded Reed-Solomon (FRS) codes presented in this thesis as well as the contributions contained therein. Section 1.4 outlines the organization of this thesis.

1.1 Background

A typical point-to-point communication scenario is shown in Fig. 1.1. During a communication session, the source tries to send messages to the destination. These messages are mapped to signals by the transmitter and which then transverse the physical channel. The physical channel can be some medium such as a cable in wired communications, free space in wireless communications

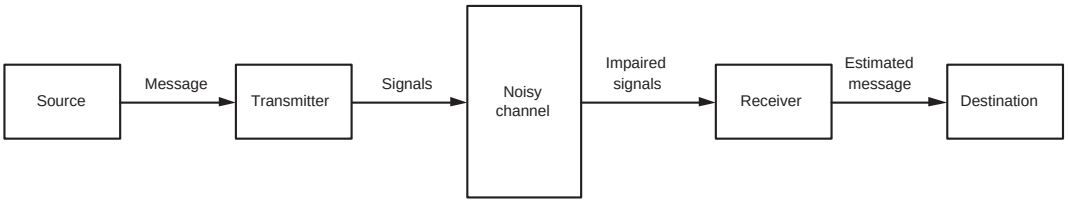

Figure 1.1: A typical point-to-point communication scenario.

and physical materials in storage. The receiver maps the received signals back to messages and passes them to the destination. These messages are expected to be sent and correctly received as fast as possible for the sake of efficiency. However, the transmission of signals is a physical process and thus is subject to the ubiquitous ambient noise, attenuation and imperfection of the physical signaling itself. For instance, random noise, burst noise and fading severely impair both the amplitude and phase of signals in a wireless channel. Moreover, since the bandwidth resource allocated for a communication session is often limited, a signal may interfere successive signals when they are transmitted too fast in a bandwidth-limited channel, a disturbance known as the inter-symbol interference. Due to these noise and disturbance, the real setting is as follows:

- Messages from the source are transmitted as signals through a channel,
- Noise and disturbance in the channel impair the signals,
- Messages are recovered from the noisy signals and passed to the destination.

The receiver may fail to recover the transmitted messages correctly due to high level noise and disturbance. This problem may be solved by increasing the power of the transmission signals. But the power supply for the source is almost always stringent because of weight limitation in different situations such as space communications, mobile communications and sensor networks. Therefore people

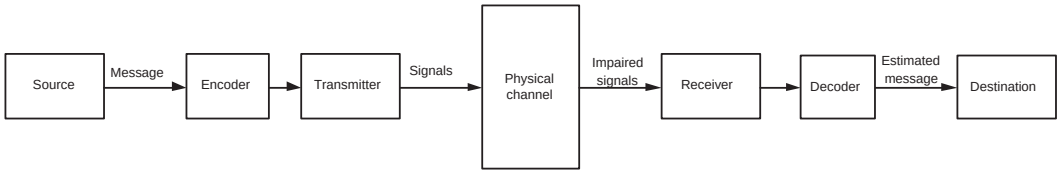

Figure 1.2: A point-to-point communication scenario with error-correcting coding.

have to look for other methods to achieve effective communication when noise and disturbance are unfavorable.

So far, error-correcting codes have provided the most successful method to resolve this problem. The scenario combined with error-correcting codes is as shown in Fig. 1.2. The idea of error-correcting codes is to introduce structured redundancy into the messages to combat noise and disturbance in the channel. Specifically, the messages from the source are described by a data stream with the data symbols from a certain finite field. This stream is then encoded as codewords of a error-correcting code by inserting structured redundant symbols. The redundancy introduced may reduce the average signal transmission power if the raw data (information) rate and the power fed to the transmitter are fixed. But as long as the performance gain due to the error-correcting codes is more dominant than the performance loss due to the reduction in the average signal power, the communication system can benefit from using error-correcting codes. Research results have shown that the gain due to error-correcting codes can be significant if they are properly designed. Hence, by exploiting structured redundancy, error-correcting codes can help the recovery of the transmitted messages from noise and disturbance presenting in the channel. Consequently, reliable communication can be achieved even when high level noise and disturbance are presented. This fact was discovered by Shannon [78]

about sixty years ago. He proved the noisy channel coding theorem [91] stating the existence of the maximal reliable communication rate for a noisy channel. This rate is defined as the capacity of this channel. This capacity was shown to be achievable by random codes of large length. However, how to design error-correcting codes to approach the capacity in a real application was still unknown and the research on error-correcting codes started since then.

Two substantially different classes of error-correcting codes, block codes and convolutional codes, have been well-developed so far. An (n, k) block code $\mathcal{C}$ is obtained by dividing the data stream into segments of k symbols and encoding each of these segments into a codeword of n symbols. The block codes are developed and analyzed using algebraic and combinatorial techniques. The use of these branches of mathematics in coding theory can be found in [58] and [36] respectively. In the study of block codes, three parameters, code rate, Hamming distance and minimum distance are important.

Definition 1.1 – ***The Rate of a Block Codes***

For a block code $\mathcal{C}$ over GF(q)*, the finite field of cardinality q, the code rate R of $\mathcal{C}$ is defined as*

$$R = \frac{\log_q |\mathcal{C}|}{n},$$

where $|\mathcal{C}|$ is the cardinality of $\mathcal{C}$ and n is the codelength of $\mathcal{C}$.

The code rate indicates the average amount of information carried by a code symbol. The rate of redundancy in $\mathcal{C}$ is then $n(1 - R)$. For the sake of efficiency, it is desirable that $\mathcal{C}$ has a high rate while having reasonable error-correction capability. Hence, most of the block codes which are of practical interest in applications such as storage and wireless communications are high rate codes.

Definition 1.2 – ***Hamming Distance of Two Vectors***

Let $\mathbf{c}_1 = (c_{1,0}, c_{1,1}, \ldots, c_{1,n-1})$ and $\mathbf{c}_2 = (c_{2,0}, c_{2,1}, \ldots, c_{2,n-1})$ be two vectors over

GF(q) *of length* n*, the Hamming distance* $d(\mathbf{c}_1, \mathbf{c}_2)$ *of these two vectors is defined as*

$$d(\mathbf{c}_1, \mathbf{c}_2) = \sum_{i=0}^{n-1} \sum_{c_{1,i} \neq c_{2,i}} 1.$$

The minimum distance of a block code $\mathcal{C}$ is defined as follows.

Definition 1.3 – ***Minimum Distance of a Block Code***

Let $\mathbf{c}_1, \mathbf{c}_2$ *be any two codewords in* $\mathcal{C}$*. The minimum distance of* $\mathcal{C}$ *is defined as*

$$d_{min} = \min_{\substack{\mathbf{c}_1, \mathbf{c}_2 \in \mathcal{C}, \\ \mathbf{c}_1 \neq \mathbf{c}_2}} (d(\mathbf{c}_1, \mathbf{c}_2)).$$

A block code with minimum distance d_{min} can correct any received vector with up to $\lfloor \frac{d_{min}-1}{2} \rfloor$ errors successfully without any ambiguity[1], where $\lfloor x \rfloor$ is the maximum integer not larger than x. Hence, minimum distance is an important metric for block codes. We can now show the advantage of error-correcting codes in more detail. Consider binary phase shift keying transmission in an Additive White Gaussian Noise (AWGN) channel and soft decision decoding. The asymptotic coding gain is $10 \log_{10}(Rd_{min})$ dB [42], where $0 < R < 1$ and d_{min} is a positive integer. With a well-designed code such that $Rd_{min} > 1$, the coded system has an advantage over the uncoded one. If the receiver makes a hard decision on the received bits, the channel between the encoder and decoder is a Binary Symmetric Channel (BSC) as in Fig. 1.3. Let the Signal-to-Noise-Ratio (SNR) be τ dB. The crossover probability is computed as

$$p = Q(\sqrt{2 \times 10^{\tau/10}}),$$

where

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{+\infty} e^{-\frac{x^2}{2}} dx,$$

[1]Viewing a brute force method as a method in the worst case, a decoding algorithm always exists.

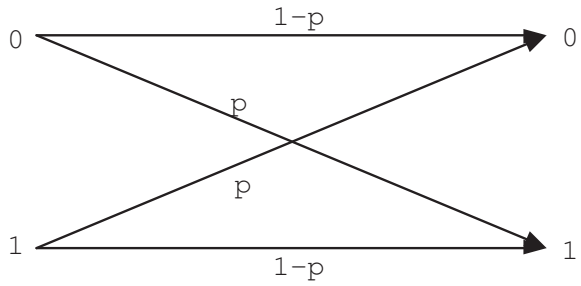

Figure 1.3: Binary symmetric channel with crossover probability p.

according to [67]. Consider a $(31, 21)$ binary code with $d_{min} = 5$. It can correct up to 2 errors. We keep the information rate and transmission power the same. The crossover probability for the uncoded and coded systems are $p_1 = Q(\sqrt{2 \times 10^{\tau/10}})$ and $p_2 = Q(\sqrt{2 \times \frac{21}{31} \times 10^{\tau/10}})$, respectively. The Bit Error Rate (BER) of the uncoded system is p_1 and the BER of the coded system is upper bounded by $1 - \sum_{i=0}^{2} \binom{31}{i} (1 - p_2)^{31-i} p_2^i$. The advantage of coded system in the moderate to high SNR region can be observed from Fig. 1.4. More powerful error-correcting codes can bring this advantage further and about 6 to 9 dB coding gain can be readily obtained in real applications.

Block codes can be linear or nonlinear. Linear block codes receive more interest than nonlinear codes in applications because of the availability of effective decoding algorithms. Nonlinear codes are more for theoretical study. Let $\mathcal{C}$ be an (n, k) block code over GF(q). If all the codewords in $\mathcal{C}$ form a vector subspace of GF$(q)^n$, $\mathcal{C}$ is a linear block code. It is easy to see that the all-zero codeword, denoted by $\mathbf{0}$, is in $\mathcal{C}$. Further, assume $d(\mathbf{c}_1, \mathbf{c}_2) = d_{min}$ for $\mathbf{c}_1, \mathbf{c}_2 \in \mathcal{C}$. By Definition 1.2, $d(\mathbf{c}_1, \mathbf{c}_2) = d(\mathbf{c}_1 - \mathbf{c}_2, \mathbf{0})$. Since $\mathbf{c}_1 - \mathbf{c}_2 \in \mathcal{C}$, by Definition 1.3,

$$d_{min} = \min_{\substack{\mathbf{c}_1, \mathbf{c}_2 \in \mathcal{C}, \\ \mathbf{c}_1 \neq \mathbf{c}_2}} d(\mathbf{c}_1, \mathbf{c}_2) = \min_{\mathbf{c} \in \mathcal{C} \setminus \{\mathbf{0}\}} w(\mathbf{c}),$$

where $w(\mathbf{c})$ denotes the Hamming weight of $\mathbf{c}$. Thus, the codewords of minimum Hamming weight determine the error-correction capability of $\mathcal{C}$. The

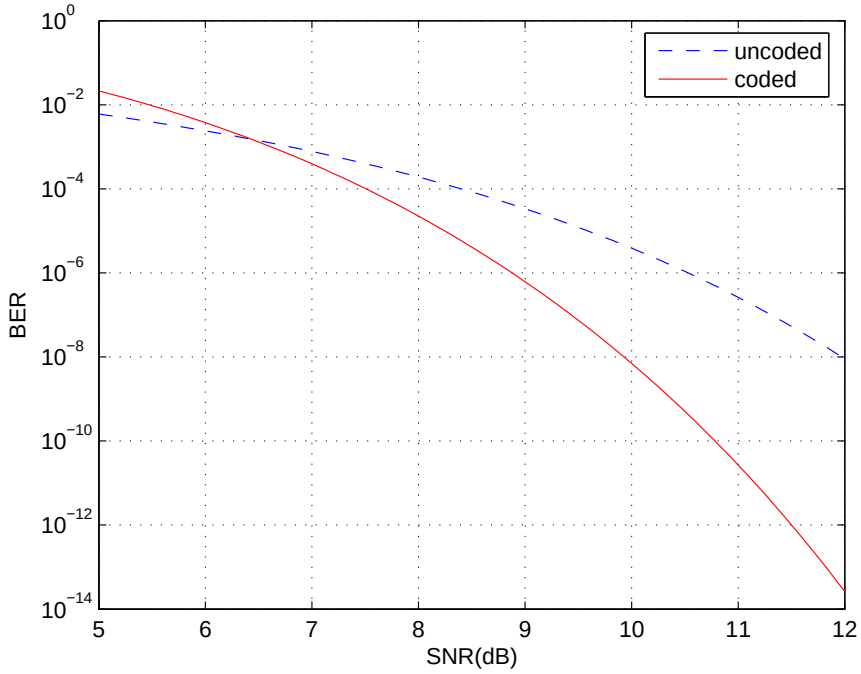

Figure 1.4: Performance comparison between uncoded and coded systems. The code used is a $(31, 21)$ binary code with $d_{min} = 5$.

SNR vs. BER curve of $\mathcal{C}$ on an AWGN channel has an error floor at some BER value. This error floor is determined by the minimum Hamming weight of $\mathcal{C}$ and the proportion of codewords with minimum Hamming weight in $\mathcal{C}$.

Binary linear block codes were first studied. These codes include the well-known Hamming code [37] and some other binary parity check codes. A standard array decoder [91] is applied to decode these simple binary linear block codes. However, these codes can only detect one or two bit error or correct only one erroneous bit and thus they are not very powerful error-correcting codes.

Subsequently, nonbinary linear block codes were studied. These codes with code symbols from larger finite fields are more interesting because they are thought more powerful than binary codes in combating both random and burst noise. Research is focused on designing nonbinary linear block codes

with abundant algebraic and combinatorial structure such that efficient decoding is possible. The well-known nonbinary block codes explored in this stage are nonbinary Reed-Muller codes, nonbinary Bose-Chaudhuri-Hocquenghem (BCH) codes and Reed-Solomon (RS) codes. The discovery of Reed-Muller codes is a significant step beyond binary linear block codes. It leads to the invention of some other interesting codes. BCH and RS codes are rich in algebraic structure due to their cyclic nature. Moreover, efficient decoding algorithms were also developed for these codes. For example, the Berlekamp-Massey Algorithm (BMA) and Euclid's Algorithm (EA) [91] for BCH and RS codes were proposed. In addition, to approach the capacity of a channel, it is desirable that an error-correcting code has a large d_{min} as well as a high R. However, for an (n, k) linear block code, these two metrics cannot be arbitrarily large because the Singleton bound [91] shows that d_{min} is bounded by $d_{min} \leq n - k + 1 = n(1 - R) + 1$. Error-correcting codes that satisfy this bound with equality are said to be Maximum Distance Separable (MDS) codes and they are thought to be *optimal* in the sense that they achieve the best tradeoff between code rate and minimum distance. RS codes and Generalized RS (GRS) codes are two important families of MDS codes and have been adopted in a wide range of applications.

A linear block code can be characterized by the generator matrix and the parity check matrix. Since an (n, k) linear block code $\mathcal{C}$ over GF(q) is a subspace of GF$(q)^n$, a set of k linearly independent codewords can serve as a basis of this subspace and any codeword can be a linear combination of this basis. A $k \times n$ generator matrix $\mathbf{G}$ is obtained by arranging these k codewords as rows. A codeword $\mathbf{c} \in \mathcal{C}$ corresponding to a message vector $\mathbf{m} = (m_0, m_1, \ldots, m_{k-1})$ is then encoded as

$$\mathbf{c} = \mathbf{m} \times \mathbf{G}.$$

An $(n-k) \times n$ matrix $\mathbf{H}$ is the parity check matrix of $\mathcal{C}$ if

$$\mathbf{G} \times \mathbf{H}^T = \mathbf{0}_{k \times (n-k)},$$

where $\mathbf{0}_{i \times j}$ is an $i \times j$ all zero matrix. Hence, if a received vector is $\mathbf{r} = \mathbf{c} + \mathbf{e}$, where $\mathbf{e} = (e_0, e_1, \ldots, e_{n-1})$ is the error vector,

$$\mathbf{r} \times \mathbf{H}^T = (\mathbf{c} + \mathbf{e}) \times \mathbf{H}^T = \mathbf{m} \times \mathbf{G} \times \mathbf{H}^T + \mathbf{e} \times \mathbf{H}^T = \mathbf{e} \times \mathbf{H}^T.$$

The vector $\mathbf{e} \times \mathbf{H}^T$ is known as the syndrome sequence for the received vector $\mathbf{r}$.

Unlike linear block codes, convolutional codes introduce redundancy into a data stream through a linear shift register without dividing the data stream into segments. The construction of convolutional codes is based on heuristic techniques [20]. They are closely related to Shannon's random codes used in the proof of the noisy channel coding theorem. There were no practical decoding algorithms until Wozencraft and Reiffen presented the "sequential algorithms" [93] in 1961. It is the first fast but suboptimal decoding algorithm for convolutional codes. The optimal Viterbi algorithm was proposed by Viterbi in 1967 [90] and was later shown to be a maximum-likelihood decoding algorithm by Forney [27] for convolutional codes.

Although it was still far from discovering error-correcting codes that achieve the Shannon limit as predicted by the noisy channel coding theorem, research on error-correcting codes was relatively quiet after 1970. More recently, the invention of turbo codes [6] and the rediscovery of Low Density Parity Check (LDPC) codes are two important breakthroughs on error-correcting codes. A turbo code is generally in systematic form where the data symbols are followed by the parity check symbols computed by two recursive systematic convolutional (RSC) code encoders. The data stream is fed into one RSC code encoder directly and fed into the other RSC code encoder after interleaving. Some parity check symbols may

be punctured to increase the code rate. Turbo codes are shown to be capacity approaching error-correcting codes, a significant result in both information theory and coding theory. The power of turbo codes is due to their code construction as well as the iterative soft decision decoding algorithm used. This algorithm iterated between two soft decision decoders corresponding to the two RSC codes. They made use of the information obtained from the channel output instead of the hard decision of it. In an AWGN channel, the channel output could be converted to Log-Likelihood Ratios (LLR's) [35] and the iterative algorithm could be easily performed by passing the extrinsic information obtained by one decoder to the other. Interestingly, research on turbo codes led to the rediscovery of LDPC codes.

LDPC codes (also known as Gallager codes) were first constructed using sparse random parity check matrices by Gallager [29]. Gallager showed that codes obtained from this construction had promising distance properties. However, they were largely unnoticed due to lack of computing techniques until they were rediscovered by MacKay and Neal in [51] and [52]. It was shown that their performance was very close to turbo codes in [52]. An LDPC code is characterized by its sparse parity check matrix which can be depicted by a bipartite graph. If all the column weight and row weight are the same, respectively, such codes are termed regular LDPC codes. Otherwise, they are called irregular LDPC codes. Methods for designing sparse parity check matrices from Euclidean geometry, projective geometry [46] and partial geometry [41] were developed. An interesting construction of LDPC codes from RS codes with two information symbols was also presented in [17]. Nonbinary LDPC codes were studied in [16]. In addition, based on the study of LDPC codes, some other codes from very sparse matrices were reported in [49] and [50]. LDPC codes can be decoded by hard decision decoding algorithms, such as majority-logic decoding and bit-flip decoding, as well

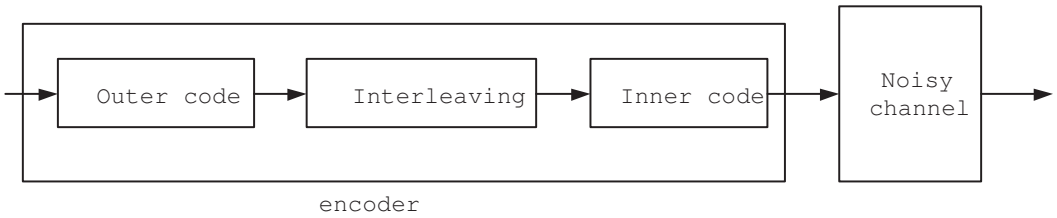

Figure 1.5: A serially concatenated code.

as soft decision decoding algorithms such as the sum-product algorithm [46]. The sum-product algorithm is closely related to the iterative soft decision decoding algorithm of turbo codes. Actually, the sum-product algorithm originates from the message-passing algorithm first proposed by Pearl in [63, 64, 65]. And turbo decoding was shown to be a special case of the message-passing algorithm in [57]. Sum-product algorithm also iteratively refine the LLRs of the received bits and make hard decisions for the received bits according to the signs of the refined LLRs.

It has been shown that well-designed irregular LDPC codes had better performance than regular LDPC codes [72]. The best results for irregular LDPC codes could approach the Shannon limit [67] within 0.0045dB [15] which was even better than the best result of turbo codes. To analyze and design LDPC codes, density evolution [72, 71] and extrinsic information transfer chart [85] had been developed.

Error-correcting codes are also serially concatenated to protect data in some extremely noisy channels. A serially concatenated code consists of an inner code and an outer code as in Fig. 1.5. The inner code is over a small alphabet whereas the outer code is over a large alphabet. Such a scheme has been successfully applied in deep space communications with the inner code being a binary convolutional code and the outer code being an RS code. The serially

concatenated code can be decoded by first decoding the inner code and then the outer code by their respective hard decision decoder. By exploiting the channel output values, an iterative soft decision decoder can achieve better performance than the former approach. Another compound coding scheme is the product code (also known as block turbo code). The resulting code is in array form where rows and columns are codewords of two block codes, respectively. When this code array is transmitted in an AWGN channel, the performance of the iterative soft decision decoding algorithm proposed by Pyndiah [68] is impressive.

Furthermore, the error control strategy widely used in network communications [91] is the Automatic Repeat-reQuest (ARQ). This strategy needs to be combined with error-correcting codes in real applications because otherwise frequent ARQ due to transmission errors will cause congestion and consequently reduce the network throughput. So error-correcting codes also play an important role in such a strategy.

1.2 Current Research and Challenges

Among all these error-correcting codes, RS codes are an important family adopted in many applications such as deep space communications, storage systems, digital video broadcasting and high definition TV. Unlike other linear block codes, RS codes were first defined by evaluation of polynomials over finite fields [70]. To describe this definition, we may start from the definition of GRS codes as follows.

Definition 1.4 – ***Generalized Reed-Solomon Codes***

Let $a_0, a_1, \ldots, a_{n-1}$ *be distinct elements in* GF(q) *and* $v_0, v_1, \ldots, v_{n-1}$ *be nonzero elements in* GF(q). *Denote* $\mathbf{a} = (a_0, a_1, \ldots, a_{n-1})$ *and* $\mathbf{v} = (v_0, v_1, \ldots, v_{n-1})$. *A GRS code is defined as the set of* n*-tuples*

$$GRS_{\mathbf{a},\mathbf{v}}(n,k) = \{(v_0 f(a_0), v_1 f(a_1), \ldots, v_{n-1} f(a_{n-1})) | f(x) \in \mathrm{GF}(q)[x]_k\},$$

where $\mathrm{GF}(q)[x]_k \subset \mathrm{GF}(q)[x]$ *and* $\deg(f(x)) < k$.

An (n, k) RS code over $\mathrm{GF}(q)$ is defined as a special case of GRS codes with $\mathbf{v}$ being the all one vector and $\mathbf{a}$ being consecutive powers of α, where $\alpha \in \mathrm{GF}(q)$ and $\mathrm{ord}(\alpha) = n$. This definition implies an encoding method for RS codes. We refer to this encoding method as the evaluation map. This encoding method can also be interpreted as the Galois Field Fourier Transform (GFFT) [7].

Definition 1.5 – ***Galois Field Fourier Transform***

Let $\alpha \in \mathrm{GF}(q)$ *and* $\mathrm{ord}(\alpha) = n|(q-1)$. *The GFFT of an n-tuple* $\mathbf{v} = (v_0, v_1, \ldots, v_n) \in \mathrm{GF}(q)^n$ *is* $\mathbf{V} = (V_0, V_1, \ldots, V_{n-1})$, *where*

$$V_j = \sum_{i=0}^{n-1} v_i \alpha^{ij}, \text{ for } 0 \leq j \leq n-1.$$

When $q = 2^p$ for $p \in \mathbb{Z}^+$, the inverse GFFT of n-tuple $\mathbf{V} = (V_0, V_1, \ldots, V_{n-1}) \in \mathrm{GF}(q)^n$ is $\mathbf{v} = (v_0, v_1, \ldots, v_n)$, where

$$v_i = \sum_{j=0}^{n-1} V_j \alpha^{-ij}, \text{ for } 0 \leq i \leq n-1.$$

Associate the polynomial $v(x) = \sum_{i=0}^{n-1} v_i x^i$ to the vector $\mathbf{v}$. An important property of the GFFT is often used in this thesis is as follows.

Theorem 1.1 *[91, Theorem 8-13]*

1. α^j *is a zero of the polynomial* $v(x)$ *if and only if the* j*th component of* $\mathbf{V}$ *is zero.*

2. α^{-i} *is a zero of the polynomial* $V(x)$ *if and only if the* i*th component of* $\mathbf{v}$ *is zero.*

The evaluation map of a message vector $\mathbf{m} = (m_0, m_1, \ldots, m_{k-1})$ is actually the GFFT of the n-tuple $(m_0, m_1, \ldots, m_{k-1}, 0, \ldots, 0)$. By part 2 of Theorem 1.1, the resulting code polynomial has zeros $1, \alpha, \ldots, \alpha^{n-k-1}$.

Since a RS code is cyclic, it can be encoded via its generator polynomial $g(x)$. Let $\mathcal{C}$ be an (n,k) RS code over GF(q) with zeros $\alpha^b, \alpha^b+1, \ldots, \alpha^{n-k+b-1}$, where $\alpha \in \text{GF}(q)$, $\text{ord}(\alpha) = n$. Let $g(x) = \prod_{i=b}^{n-k+b-1}(x-\alpha^i) = x^{n-k} + \sum_{i=0}^{n-k-1} g_i x^i$. A codeword $\mathbf{c} = (c_0, c_1, \ldots, c_{n-1}) \in \mathcal{C}$ can be represented in the polynomial form as $c(x) = \sum_{i=0}^{n-1} c_i x^i$. The message vector $\mathbf{m} = (m_0, m_1, \ldots, m_{k-1})$ can be represented by the polynomial $m(x) = \sum_{i=0}^{k-1} m_i x^i$. Then $\mathbf{c}$ can be encoded as $c(x) = m(x)g(x)$. In addition, the RS codes are also linear block codes and can be encoded via a generator matrix $\mathbf{G}$ as showed in Section 1.1. For example, the generator polynomial encoding method is equivalent to encoding the same message vector $\mathbf{m}$ by the following $k \times n$ generator matrix.

$$\mathbf{G} = \begin{pmatrix} g_0 & g_1 & \cdots & g_{n-k} & 0 & \cdots & 0 \\ 0 & g_0 & \cdots & g_{n-k-1} & g_{n-k} & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & \cdots & \cdots & g_{n-k-1} \end{pmatrix}.$$

There are three important algebraic hard decision decoding algorithms for RS codes. They are the Perteson-Gorenstein-Zierler Algorithm (PGZA), BMA and EA. Given a received word, these algorithms either output a single codeword as the decoding result or output none (decoder failure). These decoding algorithms process the received vector $\mathbf{r}$ in three steps.

1. Compute the syndromes for $\mathbf{r}$.

2. Find the error locations.

3. Compute the error values and subtract the error values from $\mathbf{r}$.

All these algorithms first compute the syndromes of the received vector $\mathbf{r} = (r_0, r_1, \ldots, r_{n-1})$. The syndromes for $\mathbf{r}$ are computed as $S_j = \sum_{i=0}^{n-1} r_i \alpha^{ij}$ for $b \le j \le n-k+b-1$, given an (n,k) RS codes having zeros $\alpha^b, \alpha^{b+1}, \ldots, \alpha^{n-k+b-1}$.

Secondly, these algorithms try to find the error locations in the received word. Assume there are t errors in the received word. The coefficients σ_i of the error locator polynomial $\sigma(x) = \sum_{i=1}^{t} \sigma_i x^i + 1$ is solved for in this step. Then the error locations are identified as the exponents of the reciprocal of the zeros for the error locator polynomial. These algorithms use different methods in this step. The PGZA sets up a group of linear equations and solved these equations for the unknown coefficients of the error locator polynomial while the BMA uses shift register synthesis to solve for the error locator polynomial $\sigma(x)$. The EA uses a division algorithm to solve for the error locator polynomial $\sigma(x)$ from the key equation. For an (n, k) RS code with zeros $\alpha, \alpha^2, \ldots, \alpha^{n-k}$, the key equations is

$$\sigma(x)(1 + S(x)) = \Omega(x) \pmod{x^{n-k+1}},$$

where $S(x) = \sum_{i=1}^{n-k} S_i x^i$. After obtaining the error locations, the error values can be obtained by solving linear equations as in the PGZA or by Forney's procedure [26] as in the BMA and EA. Although all these decoding algorithms can correct up to $\lfloor \frac{n-k}{2} \rfloor$ errors with an (n, k) RS code, the BMA is the most successful one in terms of complexity. Later, all these algorithms are referred to as classical decoding algorithms for RS codes.

The algebraic list decoding algorithm of RS codes proposed by Sudan [83] and later improved by Guruswami and Sudan [33] was a significant step in decoding RS codes. It is referred to as the Guruswami-Sudan Algorithm (GSA) in this thesis. This algorithm assumes that the RS code was encoded by the evaluation map. The evaluation points are known to both encoder and decoder as $1, \alpha, \alpha^2, \ldots, \alpha^{n-1}$. The elements in the received vector $\mathbf{r}$ are paired with the evaluation points as $(1, r_0), (\alpha, r_1), (\alpha^2, r_2), \ldots, (\alpha^{n-1}, r_{n-1})$. A bivariate polynomial $Q(x, y)$ is obtained by interpolating each of these n points with certain multiplicity m. This $Q(x, y)$ is then factorized to find factors

of the form $y - f(x)$. All $f(x)$ such that $\deg(f(x)) < k$ are candidates of the transmitted message polynomial. The interpolation step was viewed as a constraint interpolation problem and a non-trivial interpolation algorithm based on Gröbner Bases (GB) was proposed in [61]. Later, based on the Fundamental Iterative Algorithm (FIA) in [23], Koetter presented a simpler interpolation algorithm with complexity $\mathcal{O}(n^2m^4)$ [55]. For the factorization step, Roth and Ruckenstein proposed an effective recursive algorithm in [74].

The GSA generated a list of most possible candidate messages instead of a unique candidate codeword as in classical decoding algorithms of RS codes. If the decoding is thought to be successful once the transmitted message is in the output list, the GSA can correct up to $\lfloor n - \sqrt{n(k-1)} \rfloor$ errors for an (n, k) RS code. Since $n - \sqrt{n(k-1)} > \frac{n-k}{2}$, the GSA can correct more errors than classical decoding algorithms for RS codes. Although the GSA is a list decoding algorithm [21], it was shown that its average list size was quite close to unity in [56]. This result is two-fold. First, it shows that the list size is unity with very high probability and the GSA is very close to a unique decoding algorithm for RS codes. Secondly, RS codes are highly non-perfect codes, a fact which may be further exploited to correct more errors.

Besides the above hard decision decoding algorithms, soft decision decoding algorithms are also studied. The generalized minimum distance algorithm in [28] and a variant, the Chase algorithm [10], make use of the soft channel output to assist the hard decision decoder. They can significantly improve the performance of RS codes although they are not Maximum Likelihood (ML) decoders. Also, a linear block code can be described by a trellis [92]. The Bahl-Cocke-Jelinik-Raviv algorithm [4] and the Viterbi algorithm can be applied to this trellis to perform ML decoding. The complexity of this algorithm depends on the number of nodes in the trellis. For an (n, k) RS code, the number of nodes in the trellis

is $\min(2^k, 2^{n-k})$. To reduce complexity, some algorithms were developed to find the minimal trellis of a code [59]. A suboptimal trellis-based decoding algorithm was also studied in [79]. Moreover, Vardy and Be'ery showed that RS codes could be represented as unions of cosets of binary BCH codes and developed a bit level soft decision decoding algorithm [89]. This algorithm is simpler than trellis-based soft decision decoding algorithms when the RS codes are low rate codes or long, high rate codes. Based on this code partitioning and the trellis of the coset, an ML and a suboptimal soft decision decoding algorithm were presented in [66], which significantly reduces decoding complexity. In addition, based on the GSA, Koetter and Vardy proposed a soft decision decoding algorithm (KVA) [43]. Unlike the GSA which assigns all interpolation points with the same multiplicity, the KVA assigns different multiplicities for the interpolation points according to the reliability of corresponding received symbols. The KVA converts this reliability information into multiplicities by an iterative algorithm. The application of this algorithm is discussed in [1, 31, 30, 94]. Different multiplicity assignment schemes are developed in [69, 18].

In [39] and [40], a soft decision decoding algorithm was proposed by considering the binary image representation of a RS code over GF(2^p). In this algorithm, the parity check matrix $\mathbf{H}$ of this RS code is first converted into its binary image $\mathbf{H}_b$ by representing each elements in $\mathbf{H}$ with its binary image. Then the algorithm iterates between two steps. In the first step, the reliability of each bit in the received vector is computed and sorted according to their amplitudes. The submatrix in $\mathbf{H}_b$ corresponding to the $p(n-k)$ most unreliable bits is then reduced to a sparse matrix by elementary matrix operations. In the second step, the bit reliability is updated by the sum-product algorithm. In this iterative algorithm, the reduction of the submatrix corresponding to the $p(n-k)$ most unreliable bits in $\mathbf{H}_b$ is quite important. With the reduced parity check

matrix, the effect of the least reliable bits is suppressed when the sum-product algorithm is successively applied because these bits only participate in a few parity check equations. Moreover, although short cycles may appear in the parity check matrix after matrix reduction, the performance improvement presented in [39, 40] was still impressive. From this result, we can see that room for improving the performance of RS codes may still exist. We may also infer that the sum-product algorithm can achieve good performance even with short cycles in the parity check matrix if the effect from the least reliable bits is suppressed. This iterative algorithm was later combined with the KVA in [19]. In this method, the bit reliability was refined after some iterations as in [39, 40]. Then the refined bit reliability was converted to symbol reliability and used to assign multiplicities for each interpolation point in the KVA.

Besides RS codes, some variants of RS codes are also interesting. Let $\mathcal{C}$ be an (n, k) RS code with zeros $1, \alpha, \alpha^2, \ldots, \alpha^{n-k-1}$ and $n = L \times N$ $(L, N \in \mathbb{Z}^+)$. According to [44], a received word of $\mathcal{C}$ could be folded as an $L \times N$ array. After the GFFT of the columns, the syndromes of the rows in the resulting array are some distribution of the syndromes of the original received word. If the resulting array is transmitted column by column in a burst error channel, the error locations found in a row can help the decoding of the successive rows since they share the same error pattern. Hence, FRS codes are effective in correcting burst errors.

RS codes are also used as constituent codes for some compound codes. In [75], RS codes were used to construct Nearly-MDS (NMDS) linear expander codes which were also linear time encodable and decodable. These codes are optimal in the sense that they asymptotically achieve the best tradeoff between code rate and minimum distance. Such an (n, k) code has relative minimum distance $\frac{n-k-\epsilon n}{n} = 1 - R - \epsilon$ and can correct up to a fraction of $\frac{1-R-\epsilon}{2}$ errors with a sufficiently small ϵ by the modified GMD algorithm proposed in [81]. The

construction of NMDS expander codes is based on the expander graphs [73]. The idea of graph codes was first proposed by Tanner in [84]. In [2], Alon *et al.* made use of the explicit construction of Ramanujan expander graph presented in [48, 54] to construct polynomial-time encodable asymptotically good codes for which both rate and minimum distance were bounded away from zero. In [80, 82], Sipser and Spielman constructed asymptotically good codes which could be both encoded and decoded in linear-time from expander graphs. The fraction of errors that could be corrected by expander codes was later improved in [95] and [81]. Surprisingly, it was shown in [5] that expander codes can achieve the capacity of the BSC under iterative decoding and decoding complexity grows linearly with code length. Moreover, NMDS and linear-time encodable/decodable linear expander codes over large alphabets were constructed in [32]. Using RS codes as the constituent codes, [75] and [3] studied codes with the same properties but over smaller alphabets. The codes from such constructions have long code length and good tradeoff between the code rate and the minimum distance.

1.3 Motivation, Objectives and Contributions

In [44], only (n, k) RS codes with zeros $1, \alpha, \alpha^2, \ldots, \alpha^{n-k-1}$ and n a composite number were used to construct FRS codes. In certain applications, RS codes may have other $n-k$ consecutive powers of α as zeros than $1, \alpha, \alpha^2, \ldots, \alpha^{n-k-1}$. It will be interesting to see if these RS codes can also be used to construct FRS codes and find the relation between the syndrome sequence of the original received vector and the syndrome sequence of the rows in the resulting array. It is also interesting to find out what are the row codes in the resulting array. In this thesis, we will show that all RS codes with codelength a composite number can be used to construct FRS codes. The T-transformation used in [44] is identified

as the GFFT of the columns in the folded array. The rows in the resulting array are GRS codes. A RS code with codelength a composite number is shortened and folded as an array. After the GFFT of the columns in the folded array, a Transformed Folded Shortened RS (TFSRS) code is obtained. Columns and rows of this code are GRS codes. When this array is transmitted column by column in a burst error channel, the rows may have the same error locations and most of the erroneous columns can be detected by the column codes. In addition, the error locations found by both column codes and row codes can help in the decoding of the successive rows. Such a cooperative decoding scheme is presented in this thesis with each row decoded by the GSA. The performance of this scheme is also analyzed.

If an RS code is folded and transformed at the transmitter side, the resulting array may be viewed as an Interleaved RS (IRS) codes [8]. Hence, each row in the code array can actually be encoded independently and each row in the received array can also be decoded independently by the algebraic list decoders in [33, 43]. However, the messages are assumed to be encoded by the evaluation map by such a decoder. If the row codes are not encoded by the evaluation map, the output list of this decoder is a coset of the candidate messages. Each of the messages in the output list of this decoder may be reencoded by the evaluation map. Then the candidate messages may be obtained from these codewords. This method is somewhat clumsy and requires much extra computation. In this thesis, we develop a algorithm to solve this problem. The generator matrix of an RS code is first extended and the extended generator matrix $[\bar{\mathbf{G}}]$ is decomposed as

$$[\bar{\mathbf{G}}] = \mathbf{F}^{-1} \times \mathbf{D} \times \mathbf{F},$$

where $\mathbf{F}$ is the GFFT matrix and $\mathbf{D}$ is a diagonal matrix as defined in Chapter 3. The codeword obtained by the evaluation map can also be expressed as the GFFT

of the extended message which will be defined in Chapter 3. Based on the above two results, an algorithm to retrieve the candidate messages from the output list of the GSA is derived. This algorithm is non-trivial and requires less computation than the previous one.

As in the previous generalization of the construction of FRS codes, some GRS codes can also be used to construct a folded code. After the GFFT is applied to the columns of the array, the rows of the resulting array are shown to be GRS codes. The zeros of these GRS codes are related to the zeros of the original GRS code. The syndrome sequences of these GRS codes are some distribution of the syndrome sequence of the original GRS code. These GRS codes may not have consecutive syndrome sequences. In [23, 22], the nonconsecutive syndrome sequences of general cyclic codes were exploited by the FIA. The nonconsecutive syndrome sequences are arranged as an array and the error locator polynomial characterizing the minimal initial set of linearly dependent columns is found by the FIA. The idea behind this method is to find those minimal initial set of linearly dependent columns before the unknown elements in the syndrome sequences are touched. Hence the process does not involve the unknown elements. All the unknown elements are at the tail end of the syndrome sequences in [23, 22] whereas the unknown elements in our case may be in the middle of the syndrome sequences. Careful study of the FIA shows that its solution is independent of the order of how the syndrome sequences are arranged in the array. Hence, the order of the syndrome sequences can be adjusted such that processing the unknown elements are deferred as much as possible. If the minimal initial set of linearly dependent columns can be found before the unknown elements are processed, the problem is solved. Otherwise, a set of nonlinear equations involving some unknown elements are derived and solved. In Chapter 4, this technique is presented and applied to decode some GRS codes as well as the folded codes

derived from GRS codes.

As the results of the GSA shows, RS codes are highly non-perfect codes. Given a Hamming sphere with radius greater than half of the minimum distance, the average number of valid codewords in this sphere is still small. It is interesting to increase the error-correction capability of RS codes further by making use of this property. Also, all the row codes in a FRS code array share the same error pattern when the code array is transmitted column by column in burst error channels. The length of these row codes is small compared with the length of the original RS codes. Based on these, a search-based list decoding algorithm for RS codes is presented in this thesis. The number of errors can be corrected in the list decoding sense is up to $n-k-1$ for an (n,k) RS codes. The syndrome sequence of a received vector are used to search along a tree structure for all the possible error locator polynomials. The tree structure is built up in advance. A path from the root node to a leaf node corresponds to an error locator polynomial. Some complexity reduction strategies are proposed to reduce the set of nodes to search. Decoding RS codes with shortening and puncturing are also studied. The performance, the average complexity and the average list size of this algorithm in an AWGN channel are analyzed. This algorithm can be applied to decode FRS codes transmitted column by column in a burst error channel.

In the classical decoding algorithms for RS codes, finding the error locator polynomial is the key step. Sticking to find a unique solution for the error locator polynomial $\sigma(x)$ leads to the classical bound on the number of errors that can be corrected. Motivated by the property that RS codes are highly non-perfect codes, we try to develop a list-type decoding algorithm which can correct more errors than the GSA while keeping the output list size small. Assume there are t errors in a received vector of an (n,k) RS code over GF(q). The coefficients of a valid error locator polynomial should satisfy the Generalized Newton's Identities (GNI) [22].

Given the syndrome sequence, $n-k-t$ linear equations can be obtained from the GNI. In addition, for each solution of $\sigma(x)$, there is a polynomial $h(x) = x^{n-t} + \sum_{j=1}^{n-t-1} h_{n-t-j}x^j \in \mathrm{GF}(q)[x]_{n-t}$ such that

$$x^{\deg(\sigma(x))}\sigma(x^{-1})h(x) = x^n - 1. \tag{1.1}$$

Here, $h(x)$ is the product of $n-t$ distinct linear factors and the set of zeros of $h(x)$ are disjoint with the set of the reciprocal of the zeros of $\sigma(x)$. From (1.1), n nonlinear equations with unknowns being the coefficients of $\sigma(x)$ and $h(x)$ can be obtained. By combining the above linear equations and nonlinear equations, $\sigma(x)$ of degree larger than $\lfloor \frac{n-k}{2} \rfloor$ may be found. The solution may not be unique and thus this algorithm is a list-type decoding algorithm. This algorithm can be applied to decode IRS codes and improve its decoding failure probability. This algorithm can also be applied to decode FRS codes transmitted column by column in a burst error channel.

The contributions of this thesis are as follows.

- Generalize the construction of the FRS codes and analyze the properties of the FRS codes.
- Construct the TFSRS codes and analyze their performance in burst error channels.
- Develop an algorithm to obtain the transmitted message vector from the output of the GSA when the RS code is encoded via the generator polynomial.
- Construct and decode the FGRS codes by a multisequences synthesis method.

- Develop a search-based list-type decoding algorithm for RS codes.
- Develop a decoding algorithm for RS codes based on Gröbner basis.

1.4 Organization of the Thesis

The remaining part of this thesis is organized as follows. In Chapter 2, we generalize the construction of FRS codes. The TFSRS code and a list decoding algorithm based on the GSA are also proposed in this chapter. The algorithm to retrieve the messages from the output of the algebraic list decoder is derived in Chapter 3 when the original RS code is not encoded by the evaluation map. Moreover, the algorithm for the multisequences synthesis is proposed in Chapter 4. The applications of this algorithm on decoding GRS codes and folded codes obtained from GRS codes are also studied in this chapter. In addition, the search-type list decoding algorithm for RS codes is presented and analyzed in Chapter 5. Further, another list-type decoding algorithm for RS codes based on GNI and the relation $x^{\deg(\sigma(x))}\sigma(x^{-1})h(x) = x^n - 1$ is proposed in Chapter 6. Application of this algorithm to IRS is studied. Decoding some special codes are also discussed in this chapter. Finally, this thesis is summarized and the conclusion is drawn in Chapter 7. The future research on this and related topics are also presented in this chapter.

Chapter 2

Generalization of FRS Codes and Decoding of TFSRS Codes

In this chapter, we show that FRS codes can be constructed from primitive RS codes with zeros $\{\alpha^b, \alpha^{b+1}, ..., \alpha^{d-2+b}\}$, where $0 \leq b \leq q-2$, and thus generalize the results of [44] where only the case $b = 0$ was considered. Key properties of FRS codes are also derived. We also introduce TFSRS codes and a list decoding algorithm based on the recursive use of the GSA. The estimations for the probability of successful decoding, decoder error and decoder failure are derived in this chapter.

2.1 Introduction

Interleavers are often used to protect data from burst errors by randomizing the burst errors over several codewords. Such a scheme, however, introduces delay because of buffering these codewords. On the other hand, given $n = L \times N$, $d = n-k$ and α being primitive in GF(q), an $L \times N$ FRS code can be constructed from an (n, k) RS code over GF(q) and with zeros $1, \alpha, \alpha^2, \ldots, \alpha^{d-2}$. The rows of the resulting FRS code are modified RS codes [44]. If an FRS code array is

transmitted column by column in burst error channels, data can be protected without interleavers and buffering delay. We characterize the burst error channel model used in this chapter by an $L \times N$ array $\mathbf{E}$ where a column may be a burst error with certain probability. In this model, a burst error can be any nonzero vector drawn from $\mathrm{GF}(q)^L \backslash \{\mathbf{0}\}$ according to a uniform distribution on it. We decode FRS codes by recursive application of the GSA instead of the classical decoding approach as in [44]. This is because the GSA can correct up to $\lceil N - s - \sqrt{(N-s)K} - 1 \rceil$ errors compared to $\lfloor \frac{N-K-s-1}{2} \rfloor$ errors under classical decoding if s erasure positions are known for a GRS code of length N and dimension $K+1$.

2.2 FRS Codes

Let $\mathcal{C}$ be a primitive $(n, k+1)$ RS code over $\mathrm{GF}(q)$ having zeros $\{\alpha^b, \alpha^{b+1}, \alpha^{b+2}, \ldots, \alpha^{(b+d-2)}\}$, where α is as previously defined, $d = n - k$, and $n = L \times N$ such that $N, L > 1$. Then a codeword $\mathbf{c} = (c_0, c_1, \cdots, c_{n-1})$ of $\mathcal{C}$ can be folded into the following $L \times N$ array.

$$\mathbf{C} = \begin{pmatrix} c_0 & c_1 & \cdots & c_{N-1} \\ c_N & c_{N+1} & \cdots & c_{2N-1} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n-2N+1} & c_{n-2N+2} & \cdots & c_{n-N-1} \\ c_{n-N} & c_{n-N+1} & \cdots & c_{n-1} \end{pmatrix}. \tag{2.1}$$

We recall that if the order of $\beta \in \mathrm{GF}(q)$ is $\mathrm{ord}(\beta) = L$, the GFFT of $(v_0, v_1, \ldots, v_{L-1}) \in \mathrm{GF}(q)^n$ is the vector $\mathbf{V} = (V_0, V_1, \ldots, V_{L-1})$ where

$$V_j = \sum\nolimits_{i=0}^{L-1} v_i \beta^{ij}, \tag{2.2}$$

for $0 \leq j \leq L-1$ [7]. Since $\mathrm{ord}(\alpha^N) = L$, we have $\beta = \alpha^N$. We then transform the columns of (2.1) by the GFFT and obtain the array $\mathbf{B}$ in (2.3).

$$\mathbf{B} = \begin{pmatrix} b_0 & b_1 & \cdots & b_{N-1} \\ b_N & b_{N+1} & \cdots & b_{2N-1} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n-2N+1} & b_{n-2N+2} & \cdots & b_{n-N-1} \\ b_{n-N} & b_{n-N+1} & \cdots & b_{n-1} \end{pmatrix}. \tag{2.3}$$

Because the GFFT is linear [7], the vector $(b_0, b_1, \ldots, b_{n-1})$ is also a codeword of a linear code of dimension $k+1$.

Lemma 2.1 *Let* $c(x) = \sum_{i=0}^{n-1} c_i x^i \in \mathrm{GF}(q)[x]$ *and* $\deg(c(x)) \leq n-1$. *For* $0 \leq b \leq q-2$ *and* $J \leq \frac{q-1}{n}$, *if* $c(x)$ *has distinct zeros* $\alpha^b, \alpha^{b+J}, \ldots, \alpha^{b+(d-2)J}$, *then* $\mathbf{c} = (c_0, c_1, \ldots, c_{n-1})$ *is a codeword of* $\mathrm{GRS}_{\mathbf{a},\mathbf{v}}(n, n-d+1)$, *where* $\mathbf{a} = (1, \alpha^J, \ldots, \alpha^{(n-1)J})$, $\mathbf{v} = (1, \alpha^b, \ldots, \alpha^{(n-1)b})$.

Proof: A parity check matrix for $\mathbf{c}$ can be

$$\begin{aligned} &\begin{pmatrix} 1 & \alpha^b & \cdots & \alpha^{(n-1)b} \\ 1 & \alpha^{b+J} & \cdots & \alpha^{(n-1)(b+J)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{b+(d-2)J} & \cdots & \alpha^{(n-1)(b+(d-2)J)} \end{pmatrix} \\ = &\begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & \alpha^J & \cdots & \alpha^{(n-1)J} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{(d-2)J} & \cdots & \alpha^{(n-1)(d-2)J} \end{pmatrix} \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & \alpha^b & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \alpha^{(n-1)b} \end{pmatrix}. \end{aligned} \tag{2.4}$$

Let $\mathbf{a} = \{1, \alpha^J, \ldots, \alpha^{(n-1)J}\}$ and $\mathbf{v} = \{1, \alpha^b, \cdots, \alpha^{(n-1)b}\}$. Since $nJ \leq q-1$, the components of $\mathbf{a}$ are all distinct elements in $\mathrm{GF}(q)$. It is obvious that elements in $\mathbf{v}$ are all nonzero. Thus, according to [53], $\mathbf{c}$ is a codeword of $\mathrm{GRS}_{\mathbf{a},\mathbf{v}}(n, n-d+1)$.

■

With Lemma 2.1, we obtain the following Theorem 2.2.

Theorem 2.2 *The rth row of the array in (2.3) is a codeword of* $\mathrm{GRS}_{\mathbf{a},\mathbf{v}}(N, K_r+1)$*, where* $\mathbf{a} = (1, \alpha^L, \ldots, \alpha^{(N-1)L})$*,* $\mathbf{v} = \{1, \alpha^{r+\lceil \frac{b-r}{L} \rceil L}, \ldots, \alpha^{(N-1)(r+\lceil \frac{b-r}{L} \rceil L)}\}$ *and*

$$K_r = N - (\lfloor \frac{d+b-2-r}{L} \rfloor - \lceil \frac{b-r}{L} \rceil + 1) - 1$$

for $0 \leq r \leq L-1$.

Proof: First, we proof the polynomial corresponding to the vector on the rth row of (2.3) has zeros α^{r+sL}, where

$$\lceil \frac{b-r}{L} \rceil \leq s \leq \lfloor \frac{b+d-2-r}{L} \rfloor.$$

From (2.1) and (2.2), the element at position (r, j) in (2.3) is

$$b_{rN+j} = \sum_{h=0}^{L-1} c_{hN+j} \beta^{hr}. \tag{2.5}$$

Then the polynomial $b^{(r)}(y)$ corresponding to the vector on the rth row of $\mathbf{B}$ in (2.3) is

$$b^{(r)}(y) = \sum_{j=0}^{N-1} b_{rN+j} y^j = \sum_{j=0}^{N-1} \sum_{h=0}^{L-1} c_{hN+j} \beta^{hr} y^j. \tag{2.6}$$

On the other hand, the corresponding polynomial for a a codeword $\mathbf{c} \in \mathcal{C}$ is

$$c(x) = \sum_{i=0}^{n-1} c_i x^i = \sum_{k=0}^{N-1} \sum_{h=0}^{L-1} c_{hN+k} x^{hN+k}. \tag{2.7}$$

The polynomial in (2.7) has zeros $\alpha^b, \alpha^{b+1}, \alpha^{b+2}, \ldots, \alpha^{(b+d-2)}$ as given at the beginning of this section. Supposing $s \in \mathbb{Z}^+$ and $x = \alpha^{r+sL}$ are zeros of the polynomial in (2.7), where

$$b \leq r + sL \leq b + d - 2. \tag{2.8}$$

Since $(\alpha^{r+sL})^{hN} = (\alpha^r)^{hN}(\alpha^{LN})^{hs} = (\alpha^r)^{hN}$,

$$\begin{aligned} c(\alpha^{r+sL}) &= \sum_{j=0}^{N-1}\sum_{h=0}^{L-1} c_{hN+j}(\alpha^{r+sL})^{hN+j} \\ &= \sum_{j=0}^{N-1}\sum_{h=0}^{L-1} c_{hN+j}(\alpha^{r+sL})^{hN}(\alpha^{r+sL})^{j} \\ &= \sum_{j=0}^{N-1}\sum_{h=0}^{L-1} c_{hN+j}(\alpha^{r})^{hN}(\alpha^{r+sL})^{j} \\ &= \sum_{j=0}^{N-1}\sum_{h=0}^{L-1} c_{hN+j}\beta^{hr}(\alpha^{r+sL})^{j} \\ &= b^{(r)}(\alpha^{r+sL}) \\ &= 0. \end{aligned} \tag{2.9}$$

So if $x = \alpha^{r+sL}$ is a zero of the polynomial in (2.7), $y = \alpha^{r+sL}$ is also zero of the polynomial in (2.6).

In addition, from (2.8), we have

$$\left\lceil \frac{b-r}{L} \right\rceil \le s \le \left\lfloor \frac{d+b-2-r}{L} \right\rfloor. \tag{2.10}$$

Since

$$\begin{aligned} & \left(r + \left\lfloor \frac{d+b-2-r}{L} \right\rfloor L\right) - \left(r + \left\lceil \frac{b-r}{L} \right\rceil L\right) \\ = & \left\lfloor \frac{d+b-2-r}{L} \right\rfloor L - \left\lceil \frac{b-r}{L} \right\rceil L \\ \le & \left(\frac{d+b-2-r}{L} - \frac{b-r}{L}\right) L \\ = & \; d-2 = d-1 = n-k-2 < n \end{aligned}$$

and $\mathrm{ord}(\alpha) = n$, $\alpha^{r+\lceil \frac{b-r}{L} \rceil L}, \alpha^{r+(\lceil \frac{b-r}{L} \rceil + 1)L}, \ldots, \alpha^{r+\lfloor \frac{d-b+2-r}{L} \rfloor L}$ are different elements in GF(q). Thus, the polynomial corresponding to the vector on the rth row of (2.3) has $\lfloor \frac{d+b-2-r}{L} \rfloor - \lceil \frac{b-r}{L} \rceil + 1$ distinct zeros.

Next, let $b = r + \lceil \frac{b-r}{L} \rceil L$ and $J = L$. From lemma 2.1, we have the theorem.

■

Note that $\lceil \frac{b-r}{L} \rceil \geq 0$ since $0 \leq r \leq L-1$ and $b \geq 0$. We also have the following corollaries to Theorem 2.2.

Corollary 2.3 *The rth row code of the array in (2.3) is MDS and has zeros, if any,* $\{\alpha^{r+\lceil \frac{b-r}{L} \rceil L}, \alpha^{r+(\lceil \frac{b-r}{L} \rceil+1)L}, \ldots, \alpha^{r+\lfloor \frac{d+b-2-r}{L} \rfloor L}\}$.

Proof: Since the rth row code of the array in (2.3) is a GRS code and GRS codes are MDS codes [53], the rth row code is MDS. The zeros is as shown in the proof of Theorem 2.2. ■

If we denote the minimum distance of the rth row code as d_r, $\bar{b} = b \pmod{L}$, $\bar{w} = d + b - 1 \pmod{L}$ and $z = \frac{d-1-(L-\bar{b})-\bar{w}}{L}$, we have the following corollary.

Corollary 2.4 *The minimum distance for the rth row code of the array in (2.3) is :*

$$d_r = \begin{cases} z+1, & \bar{b} \geq \bar{w}-1,\ \bar{w}-1 < r < \bar{b}; \\ z+2, & \bar{b} \geq \bar{w}-1,\ r \leq \bar{w}-1 \text{ or } r \geq \bar{b}; \\ z+3, & \bar{b} < \bar{w}-1,\ \bar{b} \leq r \leq \bar{w}-1; \\ z+2, & \bar{b} < \bar{w}-1,\ r < \bar{b} \text{ or } r > \bar{w}-1. \end{cases} \tag{2.11}$$

Proof: According to Corollary 1.3, the rth row code is MDS. Its minimum distance then is thus one more than the number of zeros of this row codes. Moreover, from the result of Theorem 2.2, the zeros of the original RS code $\mathcal{C}$ are distributed among the row codes of the array in (2.3) as in Fig. 2.1. Here, $\bar{b}$ is the index of the row having the zero α^b and $\bar{w}-1$ is the index of the row having the zero α^{d+b-2}. Two cases as Fig. 2.2 shown need be considered. First consider the case when $\bar{b} \geq \bar{w}-1$ as subfigure a. in Fig. 2.2. When $\bar{w}-1 < r < \bar{b}$, the number of the zeros of the rth row code is $\frac{d-2-(L-\bar{b})-\bar{w}}{L}$ which is z as defined.[1] The minimum distance of this row code is $z+1$. When $r \leq \bar{w}-1$ or $r \geq \bar{b}$, the number of the zeros of the rth row codes is $\frac{d-2-(L-\bar{b})-\bar{w}}{L} + 1 = z+1$ and its

[1]Note that the index of rows in (2.3) starts from 0.

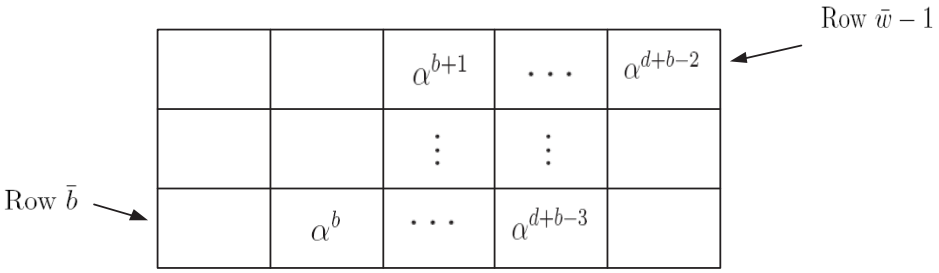

Figure 2.1: Zeros of the original RS code are distributed among row codes of FRS code array.

minimum distance is $z+2$. Next, consider the case when $\bar{b} < \bar{w}-1$ as subfigure b. in Fig. 2.2. When $\bar{b} \leq r \leq \bar{w}-1$, the number of the zeros of the rth row code is $\frac{d-2-(L-\bar{b})-\bar{w}}{L}+2 = z+2$ and its minimum distance is $z+3$. Moreover, if $r < \bar{b}$ or $r > \bar{w}-1$, the number of the zeros of this row code is $\frac{d-2-(L-\bar{b})-\bar{w}}{L}+1 = z+1$. It has minimum distance $z+2$. Thus, we have the corollary. ■

From this corollary, if we tune the parameters b and d properly, we can make any fraction of the L row codes with one more zero than the others.

Further, the rth row code is MDS. Its weight enumerator $A^{(r)}(u)$ can be found in [53]. The array in (2.3) can be viewed as a codeword of a linear block code with dimension $k+1$ and its weight enumerator $A(u)$ can be obtained as the corollary follows.

Corollary 2.5 *Let the rth row code in (2.3) have length N and dimension K_r+1 for $0 \leq r \leq L-1$. The array in (2.3) is a linear code with minimum distance $N - \max_r K_r$. The weight enumerator of this code is*

$$A(u) = \sum_{(w_0,w_1,\ldots,w_{L-1})} \left(\prod_{i=0}^{L-1} A^{(i)}(w_i)\right), \tag{2.12}$$

where $(w_0, w_1, \ldots, w_{L-1})$ is any n-tuple with nonnegative integer elements satisfying $\sum_{i=0}^{L-1} w_i = u$.

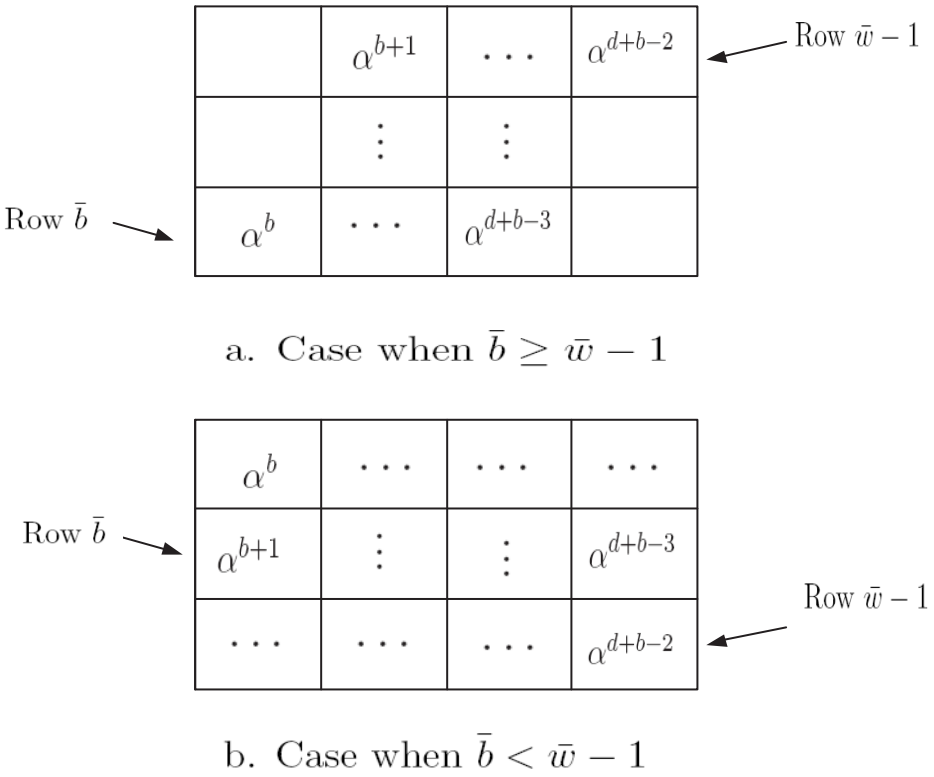

Figure 2.2: Two cases for zeros of FRS code array.

Proof: Denote the transformation which transform the array $\mathbf{C}$ in (2.1) to the array $\mathbf{B}$ in (2.3) as Θ. Since the GFFT is a linear transformation [7] and the transformation Θ consists of the GFFT of N columns, Θ is also linear transformation from $\mathrm{GF}(q)^{L\times N}$ to $\mathrm{GF}(q)^{L\times N}$, i.e. given $\mathbf{X}_1, \mathbf{X}_2 \in \mathrm{GF}(q)^{L\times N}$,

$$\gamma\Theta(\mathbf{X}_1) + \delta\Theta(\mathbf{X}_2) = \Theta(\gamma\mathbf{X}_1 + \delta\mathbf{X}_2). \tag{2.13}$$

In addition, since the GFFT is invertible, Θ is invertible and its inverse is denoted by Θ^{-1}. Denote the set of $\mathbf{C}$ as $\mathcal{C}_F$. Since $\mathcal{C}_F$ is obtained by folding codewords $\mathbf{c} \in \mathcal{C}$, $\mathcal{C}_F$ is a linear block code with dimension $k+1$. Also denote the set of $\mathbf{B}$ as $\mathcal{B}_F$. Then $\mathcal{B}_F = \{\Theta(\mathbf{C}) | \mathbf{C} \in \mathcal{C}_F\}$. It is obvious that $\mathbf{0} \in \mathcal{C}_F$ is transformed to $\mathbf{0} \in \mathcal{B}_F$ by Θ. Moreover, for any two $L \times N$ arrays $\mathbf{Y}_1, \mathbf{Y}_2 \in \mathcal{B}_F$ and any two

scalars $\gamma, \delta \in \mathrm{GF}(q)$, by (2.13),

$$\begin{aligned} \gamma\mathbf{Y}_1 + \delta\mathbf{Y}_2 &= \gamma\Theta(\Theta^{-1}(\mathbf{Y}_1)) + \delta\Theta(\Theta^{-1}(\mathbf{Y}_2)) \\ &= \Theta(\gamma\Theta^{-1}(\mathbf{X}_1) + \delta\Theta^{-1}(\mathbf{Y}_2)) \end{aligned} \tag{2.14}$$

Since $\Theta^{-1}(\mathbf{Y}_1), \Theta^{-1}(\mathbf{Y}_2) \in \mathcal{C}_F$ and $\mathcal{C}_F$ is linear, $\gamma\Theta^{-1}(\mathbf{X}_1) + \delta\Theta^{-1}(\mathbf{Y}_2) \in \mathcal{C}_F$. By definition of $\mathcal{B}_F$, $\Theta(\gamma\Theta^{-1}(\mathbf{X}_1) + \delta\Theta^{-1}(\mathbf{Y}_2)) \in \mathcal{B}_F$. Hence, $\mathcal{B}_F$ is also a linear subspace of $\mathrm{GF}(q)^{L\times N}$. Moreover, the GFFT is one-to-one mapping and so does Θ. Hence, by definition of $\mathcal{B}_F$, $|\mathcal{B}_F| = |\mathcal{C}_F| = q^{k+1}$. So $\mathcal{B}_F$ is a linear block code with dimension $k+1$.

The minimum distance of a linear code is the weight of the nonzero codeword with minimum weight. Since the rth row of $\mathbf{B}$ is a codeword of a GRS code with parameter $(N, K_r{+}1)$, its minimum distance is $N{-}K_r$. Let $\bar{j} = \arg_i \min(N - K_i)$. The codeword in $\mathcal{B}_F$ with minimum weight can be the codeword with all the rows being zero vector except the $\bar{j}$th row being the minimum weight codeword of $\bar{j}$th row code. This is shown as Table 2.1. The weight of this codeword is

Table 2.1: A codeword of $\mathcal{B}_F$ with minimum weight.

0	0	$\cdots$	0	0	0	0	0	0	0	0	0	0	0
$\vdots$													
A minimum weight codeword of the $\bar{j}$th row code													
$\vdots$													
0	0	$\cdots$	0	0	0	0	0	0	0	0	0	0	0

$N - \max_{0\le r\le L-1} K_r$.

For any $u \in \mathbb{Z}^+$,

$$A(u) = \sum_{w_0=0}^{u} \sum_{w_1=0}^{u-w_0} \cdots \sum_{w_{L-2}=0}^{u-\sum_{j=0}^{L-3} w_j} ((\prod_{i=0}^{L-2} A^{(i)}(w_i)) A^{(L-1)}(u - \sum_{j=0}^{L-2} w_j)), \tag{2.15}$$

which is (2.12). Here, $A(i)$ is the weight enumerator of an $(N, K_i + 1)$ MDS code as given in [53]. ■

Example 2.1 *FRS codes and their properties. Let $\mathcal{C}$ be a primitive $(15, 9)$ RS code over* GF(16) *with $b = 2$. Here, $L = 3$, $N = 5$ and $d = 7$. Let α be a primitive element in* GF(16)*, which is zero of $\alpha^4 + \alpha + 1$. Assume a message vector is $\mathbf{m} = (1, \alpha, \alpha^4, \alpha^2, \alpha^8, \alpha^5, \alpha^{10}, \alpha^3, 0)$. The generator polynomial is $g(x) = x^6 + \alpha^{11}x^5 + \alpha x^4 + \alpha^7 x^3 + \alpha^{10}x^2 + \alpha^{14}x + \alpha^{12}$ and the corresponding codeword $\mathbf{c} = (\alpha^{12}, \alpha^2, \alpha^2, \alpha^2, \alpha^9, \alpha^2, \alpha, 0, \alpha^{14}, \alpha, \alpha^{11}, \alpha^{14}, \alpha^{11}, \alpha^3, 0)$. After folding $\mathbf{c}$, we get the array*

$$\mathbf{C} = \begin{pmatrix} \alpha^{12} & \alpha^2 & \alpha^2 & \alpha^2 & \alpha^9 \\ \alpha^2 & \alpha & 0 & \alpha^{14} & \alpha \\ \alpha^{11} & \alpha^{14} & \alpha^{11} & \alpha^3 & 0 \end{pmatrix}. \tag{2.16}$$

Transforming array (2.16) column by column. we get the FRS codeword

$$\mathbf{B} = \begin{pmatrix} \alpha^8 & \alpha^{12} & \alpha^9 & \alpha^8 & \alpha^3 \\ \alpha^3 & \alpha & \alpha^3 & \alpha^9 & \alpha^5 \\ \alpha & \alpha^{14} & \alpha^5 & \alpha^7 & \alpha^2 \end{pmatrix}. \tag{2.17}$$

Here, $\lceil \frac{b-r}{L} \rceil = \lceil \frac{2-r}{3} \rceil$ and $\lfloor \frac{d-2+b-r}{L} \rfloor = \lfloor \frac{7-r}{3} \rfloor$. When $r = 0$, $r + \lceil \frac{2-r}{3} \rceil L = 3$ and $r + \lfloor \frac{7-r}{3} \rfloor L = 6$. Zeros for this row codes are $\{\alpha^3, \alpha^6\}$. When $r = 1$, $r + \lceil \frac{2-r}{3} \rceil L = 4$ and $r + \lfloor \frac{7-r}{3} \rfloor L = 7$. Zeros for this row codes are $\{\alpha^4, \alpha^7\}$. When $r = 2$, $r + \lceil \frac{2-r}{3} \rceil L = 2$ and $r + \lfloor \frac{7-r}{3} \rfloor L = 5$. Zeros for this row code are $\{\alpha^2, \alpha^5\}$.

In addition, $\bar{b} = b \pmod L = 2 \pmod 3 = 2$, $\bar{w} = d + b - 1 \pmod L = 7 + 2 - 1 \pmod 3 = 2$, $z = \frac{d-1-(L-\bar{b})-\bar{w}}{L} = 1$. $\bar{b} > \bar{w} - 1$. According to Corollary 2.4, for $r = 0, 1, 2$, $d_r = z + 2 = 3$.

Further, the folded array corresponding to the message vector $\mathbf{m}' = (\alpha^{12}, \alpha^{12}, \alpha^5, \alpha^{10}, \alpha^{11}, \alpha^9, 1, 0, 0)$ is

$$\mathbf{C}' = \begin{pmatrix} \alpha^9 & \alpha^2 & 1 & 0 & 0 \\ \alpha^9 & \alpha^2 & 1 & 0 & 0 \\ \alpha^9 & \alpha^2 & 1 & 0 & 0 \end{pmatrix}. \tag{2.18}$$

After transformation of columns of (2.18), we get

$$\mathbf{B}' = \begin{pmatrix} \alpha^9 & \alpha^2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \tag{2.19}$$

which is an FRS codeword with minimum weight.

2.3 TFSRS Codes

Let $(s_0, s_1, \ldots, s_{n-N-1})$ be a codeword of an $(n-N, k+1-N)$ shortened RS code over GF(q). We append N zero symbols to the end of this shortened RS code and fold it into the $L \times N$ array $\mathbf{S}$:

$$\mathbf{S} = \begin{pmatrix} s_0 & s_1 & \cdots & s_{N-1} \\ s_N & s_{N+1} & \cdots & s_{2N-1} \\ \vdots & \vdots & \ddots & \vdots \\ s_{n-2N+1} & s_{n-2N+2} & \cdots & s_{n-N-1} \\ 0 & 0 & \cdots & 0 \end{pmatrix}. \tag{2.20}$$

Applying the GFFT to each column of $\mathbf{S}$ yields the

$$\mathbf{U} = \begin{pmatrix} u_0 & u_1 & \cdots & u_{N-1} \\ u_N & u_{N+1} & \cdots & u_{2N-1} \\ \vdots & \vdots & \ddots & \vdots \\ u_{n-2N+1} & u_{n-2N+2} & \cdots & u_{n-N-1} \\ u_{n-N} & u_{n-N+1} & \cdots & u_{n-1} \end{pmatrix}, \tag{2.21}$$

where

$$u_{jN+l} = \sum_{i=0}^{L-1} s_{iN+l} \beta^{ij} \tag{2.22}$$

for $\beta \in$ GF(q), ord(β) $= L$. The array $\mathbf{U}$ is a codeword of a TFSRS code. An important property is given in the theorem follows.

Theorem 2.6 *The columns of* $\mathbf{U}$ *are codewords of a GRS code with zero* β*.*

Proof: Since each column of (2.21) is the GFFT of the corresponding column in (2.20) and the last element in each row of (2.20) is the zero element, according to property of the GFFT [91, Theorem 8-13, part 2], the polynomials corresponding to columns in (2.21) all have zero $\beta^{1-L} = \beta$. From Lemma 2.1, every column is a *GRS* code with one zero. ■

Thus, the burst errors can be detected by the column codes in a TFSRS code, if the array in (2.21) is transmitted column by column.

When a column of $\mathbf{U}$ is transmitted, one of $q^L - 1$ burst errors vector may occur. This error vector is drawn from a uniform distribution of $q^L - 1$ nonzero vectors of length L over GF(q) and $q^{L-1} - 1$ of them are valid column code codewords. So a burst error vector can be detected with probability

$$\frac{q^L - q^{L-1}}{q^L - 1} \approx 1 - \frac{1}{q}.$$

For large q, this detection probability is very close to 1 and does not depend on L.

Example 2.2 *Consider folding a shortened* $(15-5, 9-5)$ *RS code over* GF(16) *into a* 3×5 *array. The generator polynomial for the primitive* $(15, 9)$ *RS code with* $b = 2$ *over* GF(16) *is*

$$g(x) = x^6 + \alpha^{11}x^5 + \alpha x^4 + \alpha^7 x^3 + \alpha^{10}x^2 + \alpha^{14}x + \alpha^{12}.$$

In array form, the folded codeword corresponding to the message polynomial $x^3 + \alpha x^2 + \alpha^4 x + \alpha^2$ *is*

$$\mathbf{S} = \begin{pmatrix} \alpha^{14} & 0 & \alpha^9 & \alpha^{13} & 1 \\ \alpha^{14} & \alpha^9 & \alpha^{11} & \alpha^6 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \tag{2.23}$$

After the GFFT of columns, we get

$$\mathbf{U} = \begin{pmatrix} 0 & \alpha^9 & \alpha^2 & 1 & 0 \\ \alpha^9 & \alpha^{14} & \alpha^3 & \alpha^4 & \alpha^{10} \\ \alpha^4 & \alpha^4 & \alpha^5 & \alpha^{12} & \alpha^5 \end{pmatrix}. \tag{2.24}$$

It can be verified that the first, second and third row codes have zeros $\{\alpha^3, \alpha^6\}$, $\{\alpha^4, \alpha^7\}$ *and* $\{\alpha^2, \alpha^5\}$, *respectively, while the columns have a single zero at* $\beta = \alpha^5$.

2.4 List Decoding TFSRS Codes in a Burst Error Channel

Assume that a code array $\mathbf{U}$ of a TSFRS code is transmitted column by column in a burst error channel. The corresponding error array is

$$\mathbf{E} = \begin{pmatrix} e_0 & e_1 & \cdots & e_{N-1} \\ e_N & e_{N+1} & \cdots & e_{2N-1} \\ \vdots & \vdots & \ddots & \vdots \\ e_{n-N} & e_{n-N+1} & \cdots & e_{n-1} \end{pmatrix}. \tag{2.25}$$

The nonzero columns of $\mathbf{E}$ are the burst errors occurring during the transmission. They are uniformly distributed on $\mathrm{GF}(q)^L \backslash \{\mathbf{0}\}$. Therefore, a burst error may contain zeros in this burst error model. We denote the probability that a given column of $\mathbf{E}$ is nonzero as P_b.

From the channel model assumed in this chapter and the product-like code structure of the TFSRS codes, we propose a decoding algorithm for these codes. There are two main steps in this algorithm. In the first step, the column codes are used to detect possible burst error locations. In the second step, the rows in the received array are decoded by the GSA with erasure locations being the union of the error locations detected in the first step and the error locations found in the rows that have already been decoded.

Algorithm 2.1 –***List decoding of TFSRS codes.***

Input: received array given by $\mathbf{R} = \mathbf{U} + \mathbf{E}$.

Output: an estimate $\hat{\mathbf{S}}$ *of* $\mathbf{S}$.

Step 0: Initialize $\hat{\mathbf{E}}$*, the estimate of* $\mathbf{E}$*, as an* $L \times N$ *all-zero array.*

Step 1: Check all the columns of $\mathbf{R}$ *using the column codes and mark the positions of the burst errors. If more than* $\min_{0 \le i \le L-1}\{d_i\} - 1$ *(*d_i *is the minimum distance of the* i*th row code) burst errors are detected, declare decoder failure.*

Step 2:

- *Start with row code whose minimum distance is equal to* $\max_{0 \le i \le L-1}\{d_i\}$ *with erasure locations supplied by Step 1.*
- *Perform error and erasure decoding for the* i*th row code to generate a list of candidate codewords by the GSA. (Note: when there are* $d_i - 1$ *erasure positions in the* i*th row, only erasure decoding is performed for that row.)*
- *Choose the codeword nearest to the* i*th row of* $\mathbf{R}$*. Denote the corresponding error pattern by* $\hat{\mathbf{e}}_i$.
- *Update the estimation* $\hat{\mathbf{E}}$ *of the error array by replacing the* i*th row with* $\hat{\mathbf{e}}_i$*. Also, update the set of erasure locations for the next row.*

Step 3: After decoding all the rows, perform the inverse GFFT of columns in $\mathbf{R} - \hat{\mathbf{E}}$ *to obtain* $\hat{\mathbf{S}}$*. If it is a valid shortened RS codeword in array form. Output this array, otherwise, declare decoder failure.*

If n is large, we can shorten more than just one row to increase the minimum distance of the column codes. In this case, a decoding strategy will involve error detection and correction for both column and row codes.

2.5 Error-Correction Capability

The error-correction capability of our algorithm can be evaluated in terms of the probability of decoder failure P_f, the probability of decoder error P_e and the probability of successful decoding P_s. A decoder failure occurs, when the received array $\mathbf{R}$ cannot be decoded to any codeword. A decoder error occurs, when $\mathbf{R}$ is decoded to a codeword other than the one that was transmitted. Further, $\mathbf{R}$ is said to be decodable, if it can be decoded to a codeword. If the probability of receiving a decodable array is P_d, then

$$\begin{aligned} P_d + P_f &= 1, \\ P_s + P_e &= P_d. \end{aligned} \tag{2.26}$$

Assume that all possible codeword arrays of a TFSRS code are transmitted with equal probability. Since TFSRS codes are linear, we only consider the case that the all-zero codeword is transmitted. For simplicity, we assume $L|(d-1)$. Thus, by Corollary 2.3, all the row codes have the same number of zeros and dimension $K+1$. Then the rows in the received array are sequentially proceed from the first to the last in the second step of Algorithm 2.1. Hereafter, the rows of $\mathbf{U}$ are enumerated from 1 to L instead of 0 to $L-1$. In addition, if a burst error vector at a column of $\mathbf{E}$ happens to be a codeword of the column code, that burst error cannot be detected by the column code. Such a column of $\mathbf{E}$ is referred as an undetected burst position (UBP). From the channel model defined, an UBP occurs with probability

$$P_{ubp} = P_b \frac{q^{L-1}-1}{q^L-1},$$

while a detected burst occurs with probability

$$P_{dbp} = P_b(1 - \frac{q^{L-1}-1}{q^L-1}).$$

2.5.1 Probability P_s of Successful Decoding

The row codes in the TFSRS codes we consider are $(N, K+1)$ GRS codes. From [33, Theorem 16], each row code can correct up to e errors when s erasure positions are known, provided

$$e + s < N - \sqrt{(N-s)K}. \tag{2.27}$$

Lemma 2.7 *Let $T_c = e+s$. If e and s satisfy (2.27), then T_c is at its maximum, i.e. $N-K-1$, when $e=0$.*

Proof: Substituting $s = T_c - e$ into (2.27), we have

$$T_c < N - \sqrt{(N - T_c + e)K}.$$

Since $N \geq T_c$, solving above inequality, we have

$$N - T_c > \frac{K + \sqrt{K^2 + 4eK}}{2}$$

and

$$T_c < N - \frac{K + \sqrt{K^2 + 4eK}}{2}.$$

Hence $T_c = N - K - 1$, the maximum of T_c, when $e = 0$. ■

By Lemma 2.7, at most $N-K-1$ burst errors can be corrected by Algorithm 2.1, which occurs when all the $N-K-1$ burst error positions are detected in the first step. When the number of burst errors detected in the first step of Algorithm 2.1 is less than $N-K-1$, some UBP may be corrected in the second step of Algorithm 2.1. Let e_j be the number of errors corrected in jth row of the received array $\mathbf{R}$ for $j = 1, \ldots, L$. If there are s_{j-1} erasures for this row, then e_j more errors can be corrected for this row, where

$$e_j \leq \left\lceil N - s_{j-1} - \sqrt{(N - s_{j-1})K} - 1 \right\rceil. \tag{2.28}$$

Thus, the UBP's can be corrected provided the number of the UBP's does not exceed the RHS of (2.28) for $j = 1, \ldots, L$. Moreover, for the Lth row, we have the following lemma.

Lemma 2.8 *If a received array* $\mathbf{R}$ *can be decoded to a TFSRS codeword,* $e_L = 0$.

Proof: Supposing a received array $\mathbf{R}$ can be decoded to a TFSRS codeword $\bar{\mathbf{U}}$ with error array $\bar{\mathbf{E}}$, i.e. $\mathbf{R} = \bar{\mathbf{U}} + \bar{\mathbf{E}}$. If $e_L \neq 0$, any one of these e_L columns in $\mathbf{R}$ can be denoted as $\mathbf{r}^T = \bar{\mathbf{u}}^T + \bar{\mathbf{e}}^T$, where $\bar{\mathbf{u}}^T$ and $\bar{\mathbf{e}}^T$ are the corresponding columns in $\bar{\mathbf{U}}$ and $\bar{\mathbf{E}}$, respectively.

Since $\bar{\mathbf{u}}$ is a codeword of a GRS code with zero β by Theorem 2.6, it is linear and with minimum weight 2. Since those e_L are UBP's, $\mathbf{r}$ is also a valid codeword of the GRS code. Hence $\bar{\mathbf{e}}$ is a nonzero codeword of the GRS code with minimum weight 2.

But by definition, $\bar{\mathbf{e}}$ only has nonzero element at the Lth row and has weight 1. (Otherwise this erroneous column has been detected before decoding the Lth row and indicated as erasure during the decoding of the Lth row.) Hence $e_L = 0$. ■

By Lemma 2.8, only erasure decoding is needed for the Lth row of $\mathbf{R}$. We further denote the number of erasure positions detected in the first step of Algorithm 2.1 by s_0. We also associate with $\mathbf{E}$ the vector $\varepsilon = (s_0, e_1, \ldots, e_L)$, which we call the error pattern of $\mathbf{E}$. Clearly, ε corresponds to a class of correctable error arrays, if $s_0 \leq N - K - 1$, (2.28) holds for $j = 1, \ldots, L-1$, and $e_L = 0$.

Lemma 2.9 *The decoding output of the* L*th row under the GSA is a list with list size either 1 or 0.*

Proof: By Lemma 2.7, $s_0 \leq N - K - 1 < N - K$ if decoder failure is not declared. Supposing that there are $s_{j-1} < N - K$ erasure positions for the jth

row code for $1 \leq j \leq L$, by the GSA, we have

$$s_{j-1} + e_j < N - \sqrt{(N - s_{j-1})K} < N - \sqrt{(N - (N - K))K} = N - K.$$

Since $s_j = s_{j-1} + e_j < N - K$, with the same reasoning, we have

$$s_{j+1} = s_j + e_{j+1} < N - K. \tag{2.29}$$

Hence, we have

$$s_{L-1} = s_{L-2} + e_{L-1} < N - K$$

and

$$s_L = s_{L-1} + e_L < N - K.$$

By Lemma 2.8, we have $e_L = 0$. Thus, the number of erasures in the Lth row is $s_{L-2} < N - K$ and only erasure decoding is performed for the Lth row. The number of coordinates other than the erasures is at least $K + 1$ in this row. Recall that the Lth row of $\mathbf{U}$ is a codeword of a GRS code with dimension $K+1$ and a codeword of this code can be specified by any $K + 1$ coordinates. So the output list size is 1 when the $N - s_{L-1}$ coordinates other than the s_{L-1} erasures coincide with the corresponding coordinates of a valid codeword or 0 when they do not. ■

To be correctly decoded by the GSA, the erasures and errors in a row of $\mathbf{R}$ should satisfy (2.27). We consider the probability of an error pattern of $\mathbf{R}$ that can be corrected.

We first consider the probability of the burst errors found when decoding a row of $\mathbf{R}$.

Lemma 2.10 *Assume e_j errors are corrected when decoding jth row of $\mathbf{R}$ by Algorithm 2.1. Then the corresponding e_j burst errors occur with probability*

$$\binom{N - s_{j-1}}{e_j} P_{ubpc}(j),$$

where $P_{ubpc}(j) = (p_b \frac{(q-1)q^{L-j-1}}{q^L - 1})^{e_j}$.

Proof: Since there are $s_{j-1} = s_0 + \sum_{i=1}^{j-1} e_i$ erasures for the jth row of $\mathbf{R}$, the e_j errors in this row have $\binom{N-s_{j-1}}{e_j}$ choices.

Any of the e_j corresponding columns has $j-1$ zero symbols at the first $j-1$ entries in $\mathbf{E}$. The coordinate where this column intersects with the jth row can only be nonzero symbol in $\mathbf{E}$. The remaining $L-j$ entries in this column are recorded as erasures as in Algorithm 2.1. Since this column is a UBP, these $L-j$ entries can be any one of q^{L-j-1} vectors. Such a column occurs on condition that it is a UBP with probability

$$\frac{(q-1)q^{L-j-1}}{q^{L-1}-1}.$$

Hence, the probability of these e_j columns in $\mathbf{E}$ occurring is

$$\begin{aligned} & \binom{N-s_{j-1}}{e_j} \left(\frac{(q-1)q^{L-j-1}}{q^{L-1}-1}\right)^{e_j} (P_{ubp})^{e_j} \\ = & \binom{N-s_{j-1}}{e_j} \left(\frac{(q-1)q^{L-j-1}}{q^{L-1}-1}\right)^{e_j} \left(p_b \frac{q^{L-1}-1}{q^L-1}\right)^{e_j} \\ = & \binom{N-s_{j-1}}{e_j} \left(p_b \frac{(q-1)q^{L-j-1}}{q^L-1}\right)^{e_j} \end{aligned}$$

■

We next consider the probability of a correctable error pattern occurring.

Theorem 2.11 *A correctable error array with associated error pattern ε occurs with probability*

$$P_{ep}(\varepsilon) = \binom{N}{s_0} (P_{dbp})^{s_0} \prod_{j=1}^{L-1} \binom{N-s_{j-1}}{e_j} P_{ubpc}(j) (1-P_b)^{N-s_0-\sum_{j=1}^{L-1} e_j}. \qquad (2.30)$$

Proof: The s_0 erroneous columns are detected by the column code. There are $\binom{N}{s_0}$ choices for the positions of these detected burst errors and each of them occurs with probability P_{dbp}. Moreover, except the $s_0 + \sum_{j=1}^{L} e_j$ erroneous columns, the remaining columns are free of errors. The probability of this columns occurring is $(1-P_b)^{N-s_0-\sum_{j=1}^{L} e_j}$.

By Lemma 2.8 and Lemma 2.10, we have the probability of a correctable error pattern

$$P_{ep}(\varepsilon) = \binom{N}{s_0}(P_{dbp})^{s_0}\prod_{j=1}^{L-1}\binom{N-s_{j-1}}{e_j}P_{ubpc}(j)(1-P_b)^{N-s_0-\sum_{j=1}^{L-1}e_j}.$$

■

A received array can only be successfully decoded when its error pattern is within the error-correction capability and all the L correct row codewords are chosen from their output lists respectively. Before we derive the probability of successful decoding, we need to find the probability of choosing the correct codeword for each row given the error pattern is within the error-correction capability. In Algorithm 2.1, the codeword most close to the corresponding received row vector is chosen from the output list for this row. This strategy is better than randomly choosing a codeword from the list because the error vector for a row in the received array are more probable to have small Hamming weight than larger Hamming weight. We first define a notation for a sphere in the Hamming space.

Definition 2.1 *The sphere centered at a vector* $\mathbf{r}$ *and with radius* w *in an* N*-dimensional* q*-ary space is denoted as* $S_{q^N}(\mathbf{r}, w)$.

Theorem 2.12 *Given a correctable error array with associated error pattern* ε*, the probability of successful decoding can be tightly estimated by*

$$P_{cc}(\varepsilon) = \prod_{j=1}^{L-1}(1 - L(N - s_{j-1}, e_j)), \tag{2.31}$$

where, according to [13, 56],

$$L(N, w) = \overline{L}(w, w-1) = \frac{D(w, w-1)}{\binom{N}{w}(q-1)^w}, \tag{2.32}$$

$$D(w, w-1) = \binom{n}{w}\sum_{j=0}^{2w-d-1}(-1)^j N_j,$$

$$V_N(w-1) = \sum_{i=0}^{w-1} \binom{N}{i} (q-1)^i,$$

when $0 \leq j < w-d$,

$$N_j = \binom{w}{j} \left[q^{w-d+1-j} V_n(w-1) - \sum_{i=0}^{w-1} \binom{w-j}{i} (q-1)^i \right],$$

when $w-d+1 \leq j \leq 2w-d-1$,

$$\begin{aligned} N_j &= \binom{w}{j} \Bigg[\sum_{u=d-w+j}^{w-1} \binom{N-w+j}{u} \sum_{i=0}^{u-d+w-j} (-1)^i \binom{u}{i} (q^{u-d+w-j-i+1} - 1) \\ &\quad \times \sum_{s=u}^{w-1} \binom{w-j}{s-u} (q-1)^{s-u} \Bigg]. \end{aligned}$$

Proof: Since each row code is linear, it is sufficient to consider the all-zero codeword is transmitted. In Fig. 2.3, a received vector $\mathbf{r}$ with weight w will be successfully decoded if there are no false codewords in $S_{q^N}(\mathbf{r}, w-1)$. Since $\mathbf{r}$ has Hamming weight w, the true codeword, all-zero codeword, is the one most close to $\mathbf{r}$ in $S_{q^N}(\mathbf{r}, w)$ and chosen as the decoding output for this row code according to Algorithm 2.1.

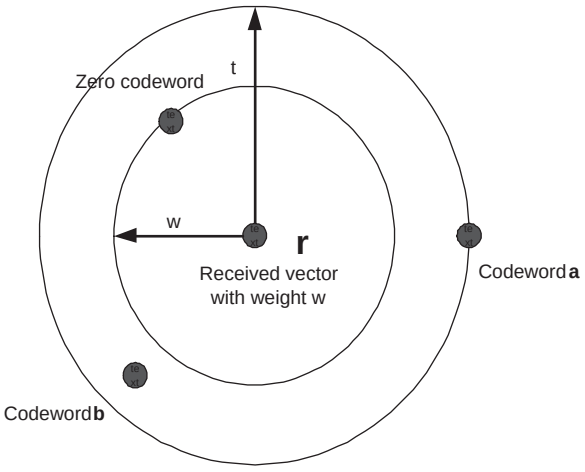

Figure 2.3: Error pattern with Hamming weight w decoded to all-zero codeword (t is the error-correction capability).

Firstly, we consider the probability of some false codewords being in $S_{q^N}(\mathbf{r}, w-1)$. According to [56], for an MDS code with length N and dimension

$K+1$, the average number of false codewords in $S_{q^N}(\mathbf{r}, t)$, where $\mathbf{r}$ has Hamming weight u, is

$$\overline{L}(u,t) = \frac{D(u,t)}{\binom{N}{u}(q-1)^u}, \tag{2.33}$$

where $D(u,t)$ is the total number of possible received vectors with Hamming weight u that can be decoded to the all-zero codeword, $\binom{N}{u}(q-1)^u$ is the number of words with Hamming weight u.[2] Thus if all the $\mathbf{r}$ with Hamming weight w are received with the same probability[3], the average number of false codewords in $S_{q^N}(\mathbf{r}, w-1)$ is

$$L(N,w) = \overline{L}(w, w-1) = \frac{D(w, w-1)}{\binom{N}{w}(q-1)^w}. \tag{2.34}$$

On the other hand, supposing the number of false codewords in $S_{q^N}(\mathbf{r}, w-1)$ is l_f, the probability of this event is P_{l_f} and the probability of some false codewords in $S_{q^N}(\mathbf{r}, w-1)$ is $P(l_f > 0)$, the average number of false codewords in $S_{q^N}(\mathbf{r}, w-1)$ can be expressed as

$$\begin{aligned} L(N,w) = \overline{L}(u,t) &= \sum_{l_f \in \mathbb{Z}^+} l_f P_{l_f} = \sum_{l_f \in \mathbb{Z}^+} P_{l_f} + \sum_{l_f \geq 2, l_f \in \mathbb{Z}^+} (l_f - 1) P_{l_f} \\ &= P(l_f > 0) + \sum_{l_f \geq 2, l_f \in \mathbb{Z}^+} (l_f - 1) P_{l_f}. \end{aligned} \tag{2.35}$$

It is known that the GSA almost always produce either a single candidate or an empty list [14]. From [60], the probability of a list with multiple candidates becomes smaller when code length N and alphabet size q increase. Moreover, the list size of the GSA is upper bounded [33]. This is also the case for codewords in $S_{q^N}(\mathbf{r}, w-1)$ when $0 \leq w \leq \lceil N - \sqrt{NK} - 1 \rceil$. Since all the codewords in $S_{q^N}(\mathbf{r}, w-1)$, if any, are false codewords by definition, l_f is also upper bounded and the probability of $l_f > 1$ is very small. So $\sum_{l_f \geq 2, l_f \in \mathbb{Z}^+} (l_f - 1) P_{l_f}$ contributes

[2] It is easy to see that $\overline{L}(u,t) \leq 1$.

[3] This is satisfied in our burst error channel model.

little to $L(N, w)$ compared with $P(l_f > 0)$, especially when N and q are large. Hence,

$$P(l_f > 0) \approx L(N, w) \leq 1. \tag{2.36}$$

From [13, 56], when $2w-1 < N-K = d$, there are no codewords in $S_{q^N}(\mathbf{r}, w-1)$. When $2w-1 \geq N-K = d$, the expression of $D(w, w-1)$ can be obtained from [13] with slight change in notation.

According to Algorithm 2.1, there are s_{j-1} erasure positions for jth row code. Since this row code is MDS, the remaining $N - s_{j-1}$ coordinates is also a codeword of an MDS code and the above analyse is applicable with the code length becomes $N - s_{j-1}$. Assume the correctable error pattern ε occurs and the first $j-1$ rows are correctly decoded. The probability of choosing the correct codeword for jth row code is $1 - L(N - s_{j-1}, e_j)$.

Further, by Lemma 2.8, when the first $L-1$ rows of the received array are successfully decoded, the Lth row decoding will output a list of size 1. ■

Thus, we sum up on all correctable error patterns and have the probability of successful decoding.

Theorem 2.13 *Let* $M_0 = N-K-1$ *and* $M_j = \lceil N - s_{j-1} - \sqrt{(N-s_{j-1})K} - 1 \rceil$ *for* $1 \leq j \leq L-1$. *The successful decoding probability* P_s *is*

$$P_s = \sum_{s_0=0}^{M_0} \sum_{e_1=0}^{M_1} \cdots \sum_{e_{L-1}=0}^{M_{L-1}} P_{ep}(\varepsilon) P_{cc}(\varepsilon). \tag{2.37}$$

2.5.2 Probability of Decodable Words P_d

Before deriving P_d, we need the following lemma about shortening GRS codes.

Lemma 2.14 *Let* C *be a GRS code over* $\mathrm{GF}(q)$ *of length* N *and dimension* $K+1$. *If* $\psi = \{0, 1, 2, \ldots, N-1\}$, $\iota \subset \psi$ *and* $|\iota| \leq K+1$, *the number of codewords in* C

with $c_i = 0$*, where* $i \in \eta \subseteq \psi \backslash \iota$*, is*

$$N_C = \begin{cases} q^{K+1-|\eta|}, & K+1 > |\eta|; \\ 1, & K+1 \leq |\eta|. \end{cases} \tag{2.38}$$

Proof: If we puncture an MDS code C by coordinate set ι, the resulting code C_1 is an MDS code with length $N_1 = N - |\iota|$ and dimension $K+1$ [91]. There are q^{K+1} codewords for code C_1.

If $K+1 \leq |\eta|$, since any $K+1$ coordinates can be information coordinates for an MDS code, only all-zero codeword satisfies $c_i = 0$, where $i \in \eta \subseteq \psi \backslash \{\iota\}$. If $K+1 > |\eta|$, We can shorten C_1 by the coordinate set η, the resulting code is an MDS code with dimension $K+1-|\eta|$ [91]. There are $q^{K+1-|\eta|}$ codewords in this code and each of them corresponds to a codeword in C satisfying the requirement.

■

Let h denote a coordinate at the jth row of an $L \times N$ array, $\mathcal{A}$. Then the column specified by h in $\mathcal{A}$ is denoted by $\langle h \rangle_{\mathcal{A}}$, the vector obtained by deleting the jth entry to the Lth entry of $\langle h \rangle_{\mathcal{A}}$ is denote by $\langle \overline{h} \rangle_{\mathcal{A}}$ and the vector obtained by deleting the 1st entry to the jth entry of $\langle h \rangle_{\mathcal{A}}$ is denoted by $\langle \underline{h} \rangle_{\mathcal{A}}$. We also denote the element at (j, h) of $\mathcal{A}$ as $\mathcal{A}_{jh}$. Let $a_{j1}, a_{j2}, \ldots, a_{j5}$ be disjoint sets of coordinates of the jth row of $\mathcal{A}$. Correction of $\mathbf{E}_{jh}$, occurs in one of the following cases.

- For $h \in a_{j1}$, $\mathbf{U}_{jh} = 0, \mathbf{R}_{jh} \in \mathrm{GF}(q) \backslash \{0\}$.
- For $h \in a_{j2}$, $\mathbf{U}_{jh} \in \mathrm{GF}(q) \backslash \{0\}, \mathbf{R}_{jh} \in \mathrm{GF}(q) \backslash \{0, \mathbf{U}_{jh}\}$.
- For $h \in a_{j3}$, $\mathbf{U}_{jh} \in \mathrm{GF}(q) \backslash \{0\}, \mathbf{R}_{jh} = 0, \langle \overline{h} \rangle_{\mathbf{R}} = \langle \overline{h} \rangle_{\mathbf{U}} = \mathbf{0}, \langle \underline{h} \rangle_{\mathbf{R}} \neq \mathbf{0}$.
- For $h \in a_{j4}$, $\mathbf{U}_{jh} \in \mathrm{GF}(q) \backslash \{0\}, \mathbf{R}_{jh} = 0, \langle \overline{h} \rangle_{\mathbf{R}} = \langle \overline{h} \rangle_{\mathbf{U}} \neq \mathbf{0}$.
- For $h \in a_{j5}$, $\mathbf{U}_{jh} \in \mathrm{GF}(q) \backslash \{0\}, \langle h \rangle_{\mathbf{R}} = \mathbf{0}$.

A decodable received array $\mathbf{R}$ is associated with the correction pattern $A = A(\mathbf{R}) = \{f_0, A_1, A_2, A_3, \ldots, A_{L-1}, \theta, \chi\}$. Here, f_0 is the set of detected burst errors obtained at the end of the first step of Algorithm 2.1 and $A_j = \{a_{j1}, a_{j2}, a_{j3}, a_{j4}, a_{j5}\}$ is the set of positions of the errors corrected in the jth row. Thus, the set of erasure locations for the jth row is $f_j = f_0 \cup A_1 \cup \ldots \cup A_{j-1}$. Further, θ is the set of positions where the columns in $\mathbf{R} - \hat{\mathbf{E}}$ contain uncorrected burst errors in the end of decoding. The set χ denotes the positions where columns in $\mathbf{R} - \hat{\mathbf{E}}$ are without errors. The cardinality of the sets f_0, A_j , a_{ji}, θ and χ are respectively denoted by t_0, t_j, t_{ji}, t_θ and t_χ. By Lemma 2.8, we have $a_{L1}, a_{L2}, \ldots, a_{L5}$ are all empty. After decoding the jth row of $\mathbf{R}$, the coordinates specified by

$$z_j = \bigcup_{i=1}^{j-1} (a_{i3} \cup a_{i5}) \cup \chi_j \cup a_{j1}$$

in the jth row of the array $\mathbf{R} - \hat{\mathbf{E}}$ should be zeros.

Given a correction pattern A, we want to find the number of the TFSRS codewords $N_{cw}(A)$ that $\mathbf{R}$ may be decoded to via A. These codewords should satisfy

$$(\mathbf{R} - \hat{\mathbf{E}})_{ji} = 0, \forall i \in z_j, 1 \leq j \leq L. \tag{2.39}$$

And from the cases of correction as previously analyzed, none of them should satisfy any of the conditions in $Y = \{cond(1), cond(2), cond(3)\}$, where

- $cond(1)$: $\mathbf{U}_{jh} = 0$ when $h \in \cup_{i=2}^{5} a_{ji}, 1 \leq j \leq L$;
- $cond(2)$: $\langle \overline{h} \rangle_{\mathbf{U}} = \mathbf{0}$ when $h \in \cup_{j=1}^{L} a_{j4}$;
- $cond(3)$: $\langle h \rangle_{\mathbf{U}} = \mathbf{0}$ when $h \in \theta$.

We first find the number of the TFSRS codewords satisfying (2.39). To this end, the jth row code is punctured by the coordinates specified by f_j and then

shortened by the coordinates specified by z_j for $1 \leq j \leq L$. By Lemma 2.14, the number of codewords of the jth row code after above processing is

$$N(j, A) = \begin{cases} q^{K+1-|z_j|}, & K+1 \geq |z_j|; \\ 1, & K+1 < |z_j|. \end{cases}$$

Thus, the number of the TFSRS codewords satisfying the constraints in (2.39) is

$$N_{\Omega}(A) = \prod_{j=1}^{L} N(j, A). \tag{2.40}$$

Let Ω be a set of codewords of the TFSRS code. Some codewords in Ω may satisfy one or more conditions in Y. Denote the number of codewords in Ω satisfying a subset $\Pi \subseteq Y$ by $N(\Pi)$. By the inclusion and exclusion principle [36], the number of codewords in Ω that satisfy none of the conditions in Y is

$$N(0) = |\Omega| + \sum_{i=1}^{|Y|} (-1)^i \sum_{|\Pi|=i} N(\Pi). \tag{2.41}$$

A codeword satisfying (2.39) may also satisfy one or more conditions in Y. So we need to compute $N_{cw}(A)$ by (2.41) as follows. If the jth row of $\mathbf{R} - \hat{\mathbf{E}}$ satisfies $\Pi \subseteq Y$, some coordinates of this row besides those specified by z_j are also zeros. Let $z_{\Pi}(j)$ denote the set of these additional zeros coordinates. We also denote the number of codewords for the jth row code satisfying both (2.39) and Π by $\delta(\Pi, j, A)$. It can be computed from Lemma 2.14 by viewing the coordinates specified by z_j and $z_{\Pi}(j)$ as shortened ones. Thus the number of the TFSRS codewords satisfying both (2.39) and Π is $\zeta(\Pi, A) = \prod_{j=1}^{L} \delta(\Pi, j, A)$. Further, the number of codewords of the TFSRS code satisfying (2.39) and y ($1 \leq y \leq |Y|$) conditions of Y can be computed as $N_y(A) = \sum_{|\Pi|=y} \zeta(\Pi, A)$. By (2.41), we have

$$N_{cw}(A) = N_{\Omega}(A) + \sum_{y=1}^{|Y|} (-1)^y N_y(A). \tag{2.42}$$

Lemma 2.15 *Given the correction pattern A and one of the $N_{cw}(A)$ codewords $\mathbf{U}$ of the TFSRS code, a received array $\mathbf{R}$ which can be decoded to $\mathbf{U}$ via A occurs with probability*

$$P_{\mathbf{U}}(A) = (1-P_b)^{\sum_{j=1}^{L-1} t_{j5}+t_\chi}(P_{dbp})^{t_0}(\frac{P_{ubp}}{q^{L-1}-1})^{t_\theta}\prod_{j=1}^{L-1}\prod_{i=1}^{4}\lambda_{ji}(j,t_{ji}), \tag{2.43}$$

where

$$\lambda_{j1}(j,t_{j1}) = (\frac{(q-1)q^{L-j-1}}{q^{L-1}-1}P_b)^{t_{j1}},$$

$$\lambda_{j2}(j,t_{j2}) = (\frac{(q-2)q^{L-j-1}}{q^{L-1}-1}P_b)^{t_{j2}},$$

$$\lambda_{j3}(j,t_{j3}) = (\frac{q^{L-j-1}-1}{q^{L-1}-1}P_b)^{t_{j3}},$$

$$\lambda_{j4}(j,t_{j4}) = (\frac{q^{L-j-1}}{q^{L-1}-1}P_b)^{t_{j4}}.$$

Proof: Some received array $\mathbf{R}$ will be decoded to $\mathbf{U}$, given A and $\mathbf{U}$. If any of these words received, columns in $\mathbf{E}$ corresponding to $\sum_{j=1}^{L-1} a_{j5}+\chi$ are all zero columns, columns in $\mathbf{E}$ corresponding to $\sum_{j=1}^{L-1}\sum_{i=1}^{4} a_{ji}+\theta$ are undetected bursts and columns in $\mathbf{E}$ corresponding to f_0 are detected bursts.

Among those undetected bursts, symbols in columns θ are determined by the codeword $\mathbf{U}$ and each of such nonzero columns occurs with probability $\frac{1}{q^{L-1}-1}$ on condition that it is a undetected burst. Hence these undetected bursts occur with probability $(\frac{p_{ubp}}{q^{L-1}-1})^{t_\theta}$.

Since entries in $\langle \underline{h} \rangle_{\mathbf{E}}$ for each $h \in a_{j1}$ are nonzero symbols in GF(q), the corresponding t_{j1} undetected bursts happen with probability

$$\lambda_{j1}(j,t_{j1}) = (\frac{(q-1)q^{L-j-1}}{q^{L-1}-1}P_b)^{t_{j1}}.$$

By similar reasoning, t_{j2} undetected bursts corresponding to a_{j2} occur with probability

$$\lambda_{j2}(j,t_{j2}) = (\frac{(q-2)q^{L-j-1}}{q^{L-1}-1}P_b)^{t_{j2}},$$

t_{j3} undetected bursts corresponding to a_{j3} occur with probability

$$\lambda_{j3}(j, t_{j3}) = (\frac{q^{L-j-1}-1}{q^{L-1}-1}P_b)^{t_{j3}},$$

and t_{j4} undetected bursts occur with probability

$$\lambda_{j4}(j, t_{j4}) = (\frac{q^{L-j-1}}{q^{L-1}-1}P_b)^{t_{j4}}.$$

There are also $\sum_{j=1}^{L-1} t_{j5} + t_\chi$ zero columns in E, which occur with probability

$$(1-P_b)^{\sum_{j=1}^{L-1} t_{j5}+t_\chi}.$$

In addition, the t_0 detected bursts occur with probability $(P_{dbp})^{t_0}$.

Hence, the probability of these words received with probability

$$P_{\mathbf{U}}(A) = (1-P_b)^{\sum_{j=1}^{L-1} t_{j5}+t_\chi}(P_{dbp})^{t_0}(\frac{P_{ubp}}{q^{L-1}-1})^{t_\theta}\prod_{j=1}^{L-1}\prod_{i=1}^{4}\lambda_{ji}(j, t_{ji}).$$

■

Hence, the probability of $\mathbf{R}$ being decoded to one of the $N_{cw}(A)$ codewords is

$$P(A) = N_{cw}(A)P_{\mathbf{U}}(A). \tag{2.44}$$

From (2.40), (2.42), (2.43) and (2.44), we can see that for two distinct error correction patterns A and A', if $T = T'$, then $P(A) = P(A')$. All the possible A's can be partitioned such that elements in each class have the same T and $P(A)$ which is denoted as $P(T)$.

Example 2.3 *We use a simple example to illustrate the computation of* $P(A)$. *Supposing* $q = 16, n = 15, N = 5, L = 3, K+1 = 3$, *a certain* A *with* $(t_0, t_1, t_2, t_3, t_\theta, t_\chi) = (1, 0, 1, 0, 1, 2)$ *and* $(t_{21}, t_{22}, t_{23}, t_{24}, t_{25}) = (0, 0, 0, 1, 0)$, $|z_1| = |z_2| = |z_3| = 2$, $N(1, A) = N(2, A) = N(3, A) = 16$ *and* $N_0(A) = 16^3$. *The conditions are* $\pi_1 : \mathbf{U}_{2h}$ *for* $h \in a_{24}$, $\pi_2 : \langle\overline{h}\rangle_{\mathbf{U}} = \mathbf{0}$ *for* $h \in a_{24}$ *and* $\pi_3 : \langle h\rangle_{\mathbf{U}} = \mathbf{0}$ *for* $h \in \theta$.

With condition π_1 *, we have* $\delta(\pi_1, 1, A) = \delta(\pi_1, 2, A) = \delta(\pi_1, 3, A) = 1$*, so* $\zeta(\pi_1, A) = 1$*. We also can get* $\zeta(\pi_2, A) = 16^2$*,* $\zeta(\pi_3, A) = 16^2$ *with* π_2 *and* π_3*, respectively.*

With condition sets $\pi_1 + \pi_2$*, we have* $\zeta(\pi_1 + \pi_2, A) = 16$*. We also have* $\zeta(\pi_2 + \pi_3, A) = 1$ *with with condition sets* $\pi_2 + \pi_3$ *and* $\zeta(\pi_1 + \pi_3, A) = 1$ *with condition sets* $\pi_1 + \pi_3$*.*

With condition set $\pi_1 + \pi_2 + \pi_3$*, we have* $\zeta(\pi_1 + \pi_2 + \pi_3, A) = 1$*.*

So by (2.42), $N_{cw} = 16^3 - (16^2 + 16^2 + 1) + (16 + 2) - 1 = 3600$ *and by (2.43),* $P_{\mathbf{u}}(A) = (1 - P_b)^2 P_{dbp}(\frac{P_{ubp}}{q^{L-1}-1})(\frac{q^{L-3}}{q^{L-1}-1} P_b) = 8.8014 \times 10^{-13}$*. Consequently,* $P(A) = N_{cw}(A) P_{\mathbf{u}}(A) = 3.1685 \times 10^{-9}$*.*

Lemma 2.16 *Given* $T^* = \{t_1, t_2, \ldots, t_{L-1}, t_\chi, t_0\}$*, denote the summation on all the* t_{ji} *satisfied* $\sum_{i=0}^{5} t_{ji} = t_j$ *for* $1 \le j \le L-1$ *as* $\sum_{t_{ji}|T^*}$*. The probability of observing a received array* $\mathbf{R}$ *which may be decoded by correction patterns satisfying the constraints* $\sum_{i=1}^{5} t_{ji} = t_j$ *imposed by* T^**, is*

$$\begin{aligned} P'(T) &= \binom{N}{t_0} \prod_{i=1}^{L-1} \binom{N - \sum_{j=0}^{i-1} t_j}{t_i} \binom{N - \sum_{j=0}^{L-1} t_j}{t_\chi} \\ &\quad \sum_{t_{ji}|T'} \prod_{j=1}^{L-1} \prod_{v=1}^{4} \binom{t_j - \sum_{i=1}^{v-1} t_{ji}}{t_{jv}} P(T). \end{aligned} \tag{2.45}$$

Proof: For t_0 columns with detected burst errors by column code, there are $\binom{N}{t_0}$ choices for their positions. Also, there are $\prod_{i=1}^{L-1} \binom{N - \sum_{j=0}^{i-1} t_j}{t_i}$ choices for $(t_1, \ldots, t_{L-1})$ and $\binom{N - \sum_{j=0}^{L-1} t_j}{t_\chi}$ choices for t_χ columns without errors. Further, given t_j, there are $\prod_{v=1}^{4} \binom{t_j - \sum_{i=1}^{v-1} t_{ji}}{t_{jv}}$ choices for $t_{jv} (1 \le v \le 5)$. These choices are independent of each other. We then have the lemma. ■

According to Algorithm 2.1, at most $N-K-1$ bursts errors can be corrected. In addition, if $M_i = \lceil N - \sum_{j=0}^{i-1} t_j - \sqrt{(N - \sum_{j=0}^{i-1} t_j)K} - 1 \rceil$, at most M_i errors can be corrected in the ith row using Algorithm 2.1. Summing up on all possible T, we have the following theorem for P_d.

Theorem 2.17 *The probability of receiving a decodable array* $\mathbf{R}$ *is*

$$P_d = \sum_{t_0=0}^{N-K-1} \sum_{t_1=0}^{M_1} \sum_{t_2=0}^{M_2} \cdots \sum_{t_{L-1}=0}^{M_{L-1}} \sum_{t_\chi=0}^{N-\sum_{j=0}^{L-1} t_j} P(T^*). \tag{2.46}$$

According to (2.26), we have $P_e = P_d - P_s$ and $P_f = 1 - P_d$.

2.6 Summary

We have extended the results in [44] about FRS codes. We generalize the construction of FRS code. We also derive some important properties of FRS codes. Based on the derivation, we can see that FRS codes can also be constructed from any non-primitive RS code with codelength a composite number.

In addition, to detect phased bursts effectively, we propose TFSRS codes and a decoding algorithm based on the GSA. We also derive estimates of the probability of successful decoding, decoder error and decoder failure of our scheme.

Chapter 3

Retrieving Messages from Output List of the GSA

In this chapter, we present a transform that enables the generator-matrix-based RS coded data to be recovered under the interpolation-based list decoder of [33, 43]. The transform matrix needs to be computed only once and the method introduces an average computational overhead of k^2 field multiplications to the decoding process, given a code of dimension k.

3.1 Introduction

RS codes are an important code family and have been adopted in a wide range of applications such as compact discs (CDs), digital video broadcasting and high definition TV. Denote a cyclic subgroup of $\mathrm{GF}(q)\backslash\{0\}$ of order n and its generator as Φ and α respectively. Then an (n, k) RS code over $\mathrm{GF}(q)$ with zeros $\alpha, \alpha^2, \ldots, \alpha^{n-k}$ is defined as

$$\{(c_0, c_1, \ldots, c_{n-1}) | c_i = m(\alpha^i),\ 0 \leq i \leq n-1,\ m(x) = \sum_{j=0}^{k-1} m_j x^j \in \mathrm{GF}(q)[x]\}.$$

This definition implies the evaluation map encoding method as in Section 1.2. Assuming the messages are encoded via the evaluation map, the interpolation-based list decoding algorithms in [33, 43] can correct far more errors than the classical decoding algorithms. It is advantageous to incorporate such a decoder in a system employing RS codes. For example, the error performance of the system can be maintained even at much lower SNR by incorporating such a decoder. However, RS codes are often encoded via their generator polynomial in existing applications. When such RS codes are decoded by an interpolation-based list decoding algorithm, its output list may not include the original message. Thus, a method for retrieving the original message from the output list of the interpolation-based list decoder is needed.

In this chapter, a more general solution to the above problem is presented. Let $\mathbf{G}_a$ be an generator matrix of a RS code. We consider retrieving the messages coded via $\mathbf{G}_a$ from the output list of an interpolation-based list decoding algorithm. Since $\mathbf{G}_a$ is arbitrary, the basis transformation used in this approach is different from the one used in [34]. Especially, if messages are originally encoded as codewords of a narrow-sense RS code, no basis transformation is required.

In the remaining part of this chapter, we first develop three lemmas. Based on these lemmas, we further derive the main result - Theorem 3.4 and therefore propose an algorithm to solve the aforementioned problem in Section 3.3.

3.2 Lemmas Leading to the Main Result

Without loss of generality, let q be a fixed power of 2. Let $g(x) = \prod_{i=b}^{n-k-1+b}(x - \alpha^i) = \sum_{i=0}^{n-k} g_i x^i$ be the generator polynomial of an (n, k) RS code $\mathcal{C}$ over GF(q). Here b is not assume to be 1 and therefore $\mathcal{C}$ may not be a narrow-sense RS code.

Also, we know that the matrix

$$\mathbf{G} = \begin{pmatrix} g_0 & g_1 & \cdots & g_{n-k} & 0 & \cdots & 0 \\ 0 & g_0 & \cdots & g_{n-k-1} & g_{n-k} & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & \cdots & \cdots & \cdots & g_{n-k} \end{pmatrix} \tag{3.1}$$

is a generator matrix of $\mathcal{C}$. We next derive a transformation to convert $\mathcal{C}$ to a narrow-sense RS code as in Lemma 3.1.

Lemma 3.1 *If* $\mathbf{c} = (c_0, c_1, \ldots, c_{n-1}) \in \mathcal{C}$, *then* $\bar{\mathbf{c}} = \mathbf{c} \times \mathbf{W}$ *is a codeword of a narrow-sense* (n, k) *RS code* $\bar{\mathcal{C}}$ *over* GF(q), *where*

$$\mathbf{W} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & \alpha^{(b-1)} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \alpha^{(b-1)(n-1)} \end{pmatrix}.$$

Proof: Let $c(x)$ and $\bar{c}(y)$ be the code polynomials for $\mathbf{c}$ and $\bar{\mathbf{c}}$, respectively. Then we have

$$c(x) = \sum_{i=0}^{n-1} c_i \alpha^{(b-1)i} \left(\frac{x}{\alpha^{b-1}}\right)^i = \sum_{i=0}^{n-1} \bar{c}_i y^i = \bar{c}(y), \tag{3.2}$$

where $\bar{c}_i = c_i \alpha^{(b-1)i}$, $y = \frac{x}{\alpha^{b-1}}$. Since $c(x)$ has zeros $\alpha^b, \alpha^{b+1}, \ldots, \alpha^{n-k-1+b}$, we have that $\bar{c}(y)$ has zeros $\alpha, \alpha^2, \ldots, \alpha^{n-k}$ and therefore $\bar{\mathbf{c}}$ is a narrow-sense RS code. Given $\mathbf{W}$ as in the lemma, the transformation from $\mathbf{c}$ to $\bar{\mathbf{c}}$ in (3.2) can be expressed as $\bar{\mathbf{c}} = \mathbf{c} \times \mathbf{W}$. ■

It is easy to see that $\mathbf{W}$ will be an identity matrix and above transformation is not needed if $\mathcal{C}$ is a narrow-sense RS code. Let $\bar{g}(y) = \sum_{i=0}^{n-k} \bar{g}_i y^i$ where $\bar{g}_i = g_i \alpha^{(b-1)i}$. From the proof of Lemma 3.1, we know that $\bar{g}(y)$ has zeros

$\alpha, \alpha^2, \ldots, \alpha^{n-k}$ and is a code polynomial of $\bar{\mathcal{C}}$. Then the following matrix

$$\bar{\mathbf{G}} = \begin{pmatrix} \bar{g}_0 & \bar{g}_1 & \cdots & \bar{g}_{n-k} & 0 & \cdots & 0 \\ 0 & \bar{g}_0 & \cdots & \bar{g}_{n-k-1} & \bar{g}_{n-k} & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & \cdots & \cdots & \cdots & \bar{g}_{n-k} \end{pmatrix}, \tag{3.3}$$

is a generator matrix for $\bar{\mathcal{C}}$, since the rows of $\bar{\mathbf{G}}$ span a vector space over GF(q) of dimension k. Let $[\mathbf{U}]$ denote the $n \times n$ matrix obtained by appending $n-k$ rows to a $k \times n$ matrix ($n > k$) $\mathbf{U}$ such that each additional row is a right cyclic shift of the previous row by one position. Lemma 3.2 shows the relation between $[\bar{\mathbf{G}}]$ and $[\mathbf{G}]$.

Lemma 3.2 $[\mathbf{G}] \times \mathbf{W} = \mathbf{W} \times [\bar{\mathbf{G}}]$.

Proof: Denote the first row of $[\mathbf{G}]$ and $[\bar{\mathbf{G}}]$ by $(g_0, g_1, \ldots, g_{n-1})$ and $(\bar{g}_0, \bar{g}_1, \ldots, \bar{g}_{n-1})$, respectively, where $g_j = \bar{g}_j = 0$ for $n-k+1 \leq j \leq n-1$. From the definitions of $[\mathbf{G}]$ and $[\bar{\mathbf{G}}]$, we have that the respective elements of $[\mathbf{G}]$ and $[\bar{\mathbf{G}}]$ at the $((s+1),(t+1))$ are $g_{t-s \bmod n}$ and $\bar{g}_{t-s \bmod n} = g_{t-s \bmod n}\alpha^{(b-1)t}/\alpha^{(b-1)s}$ for $0 \leq s, t \leq n-1$.

Thus, $[\bar{\mathbf{G}}]$ can be obtained by multiplying the $(t+1)$th column of $[\mathbf{G}]$ by $\alpha^{(b-1)t}$ and dividing the $(s+1)$th row of the resulting matrix by $\alpha^{(b-1)s}$ for $0 \leq s, t \leq n-1$. In matrix form, these operations can be expressed as $[\bar{\mathbf{G}}] = \mathbf{W}^{-1} \times [\mathbf{G}] \times \mathbf{W}$. ■

Let $\mathbf{F}$ and $\mathbf{F}^{-1}$ denote the n-point GFFT and inverse GFFT matrices over GF(q), i.e.

$$\mathbf{F} = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & \alpha & \cdots & \alpha^{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{n-1} & \cdots & \alpha^{(n-1)(n-1)} \end{pmatrix}, \tag{3.4}$$

$$\mathbf{F}^{-1} = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & \alpha^{-1} & \cdots & \alpha^{-(n-1)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{-(n-1)} & \cdots & \alpha^{-(n-1)(n-1)} \end{pmatrix}. \tag{3.5}$$

The matrix $[\bar{\mathbf{G}}]$ can be decomposed in terms of $\mathbf{F}$ and $\mathbf{F}^{-1}$ as Lemma 3.3 follows.

Lemma 3.3 $[\bar{\mathbf{G}}] = \mathbf{F}^{-1} \times \mathbf{D} \times \mathbf{F}$ *where* $\mathbf{D}$ *is an* $n \times n$ *diagonal matrix such that its main diagonal is the inverse GFFT of the first row of* $[\bar{\mathbf{G}}]$.

Proof: Denote the inverse GFFT of the first row of $[\bar{\mathbf{G}}]$ by $\mathbf{g}(1) = (G_0, G_1, \ldots, G_{n-1})$. Since the $(i+1)$th row of $[\bar{\mathbf{G}}]$ is the right cyclic shift of the first row of $[\bar{\mathbf{G}}]$ by i positions, the inverse GFFT of the $(i+1)$th row of $[\bar{\mathbf{G}}]$ is

$$\mathbf{g}(i+1) = (G_0, G_1/\alpha^i, G_2/\alpha^{2i}, \ldots, G_{n-1}/\alpha^{(n-1)i})$$

by the modulation property of GFFT [7, Figure 6.1]. Thus, the inverse GFFT of the rows of $[\bar{\mathbf{G}}]$ in matrix form is

$$\begin{aligned} [\bar{\mathbf{G}}] \times \mathbf{F}^{-1} &= \begin{pmatrix} G_0 & G_1 & G_2 & \cdots & G_{n-1} \\ G_0 & G_1/\alpha & G_2/\alpha^2 & \cdots & G_{n-1}/\alpha^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ G_0 & G_1/\alpha^{n-1} & G_2/\alpha^{2(n-1)} & \cdots & G_{n-1}/\alpha^{(n-1)(n-1)} \end{pmatrix} \\ &= \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \alpha^{-1} & \alpha^{-2} & \cdots & \alpha^{-(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{-(n-1)} & \alpha^{-2(n-1)} & \cdots & \alpha^{-(n-1)(n-1)} \end{pmatrix} \begin{pmatrix} G_0 & 0 & \cdots & 0 \\ 0 & G_1 & \cdots & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & \cdots & G_{n-1} \end{pmatrix} \\ &= \mathbf{F}^{-1} \times \begin{pmatrix} G_0 & 0 & \cdots & 0 \\ 0 & G_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & G_{n-1} \end{pmatrix} = \mathbf{F}^{-1} \times \mathbf{D}. \end{aligned} \tag{3.6}$$

Then, the lemma is obtained by left multiplying both sides of (3.6) by $\mathbf{F}$. ■

Since $\alpha, \alpha^2, \ldots, \alpha^{n-k}$ are zeros of $\bar{g}(x)$, the last $n-k$ elements of $\mathbf{g}(1)$ and the last $n-k$ elements in the main diagonal of $\mathbf{D}$ are all zeroes by the property 2 of [91, Theorem 8-13]. Moreover, by the translation property of the GFFT [7, Figure 6.1], the inverse GFFT $(G_0, G_1, \ldots, G_{k-1}, 0, \ldots, 0)$ of the n-tuple $(\bar{g}_0, \bar{g}_1, \ldots, \bar{g}_{n-k}, 0, \ldots, 0)$ is the right cyclic shift of the inverse GFFT of the n-tuple $(g_0, g_1, \ldots, g_{n-k}, 0, \ldots, 0)$ by $b-1$ positions since $\bar{g}_i = g_i \alpha^{(b-1)i}$.

3.3 The Main Result

Let $\mathbf{A}$ be some $k \times k$ basis transformation matrix. Then $\mathbf{G}_a = \mathbf{A} \times \mathbf{G}$ is a generator matrix of RS code $\mathcal{C}$. In this section, we propose a method to retrieve the messages encoded via $\mathbf{G}_a$ from the output list of an interpolation-based list decoding algorithm. We first define some notations for the derivation follows. Let $\tilde{\mathbf{A}} = (\ \mathbf{A}\ \ \mathbf{0}\ )$ be a $k \times n$ matrix where $\mathbf{0}$ is a $k \times (n-k)$ all-zero matrix. Let $\mathbf{U}_{i \times j}$ denote the $i \times j$ upper-left submatrix of $\mathbf{U}$. In addition, since the evaluation map may be viewed as the n-point GFFT over GF(q), the relation between a codeword $\bar{\mathbf{c}}$ of $\bar{\mathcal{C}}$ and its corresponding message vector $(f_0, f_1, \ldots, f_{k-1}) \in GF(q)^k$ can be expressed as $\bar{\mathbf{c}} = \mathbf{f} \times \mathbf{F}$ where $\mathbf{f} = (f_0, f_1, \ldots, f_{k-1}, 0, \ldots, 0)$. We then have the following theorem.

Theorem 3.4 *Let* $\mathbf{m} \in \mathrm{GF}(q)^k$ *be encoded as* $\mathbf{c} \in \mathcal{C}$ *via the generator matrix* $\mathbf{G}_a$. *Then* $\mathbf{m}^T = (\mathbf{A}^T)^{-1} \times (\mathbf{W}_{k \times k})^{-1} \times (\mathbf{F}^{-1}_{k \times k})^{-1} \times (\mathbf{D}_{k \times k})^{-1} \times (\mathbf{f}_{1 \times k})^T$.

Proof: From Lemmas 3.1 to 3.3,

$$\begin{aligned} \bar{\mathbf{c}} &= \mathbf{c} \times \mathbf{W} = \mathbf{m} \times \mathbf{G}_a \times \mathbf{W} = \mathbf{m} \times \mathbf{A} \times \mathbf{G} \times \mathbf{W} \\ &= \mathbf{m} \times \tilde{\mathbf{A}} \times [\mathbf{G}] \times \mathbf{W} = \mathbf{m} \times \tilde{\mathbf{A}} \times \mathbf{W} \times [\bar{\mathbf{G}}] \\ &= \mathbf{m} \times \tilde{\mathbf{A}} \times \mathbf{W} \times \mathbf{F}^{-1} \times \mathbf{D} \times \mathbf{F}. \end{aligned} \tag{3.7}$$

Since $\bar{\mathbf{c}} = \mathbf{f} \times \mathbf{F}$, we have $\mathbf{f} = \mathbf{m} \times \tilde{\mathbf{A}} \times \mathbf{W} \times \mathbf{F}^{-1} \times \mathbf{D}$. Moreover, since $\mathbf{F}$, $\mathbf{F}^{-1}$, $\mathbf{D}$ and $\mathbf{W}$ are symmetric, we have

$$\mathbf{f}^T = \mathbf{D} \times \mathbf{F}^{-1} \times \mathbf{W} \times \tilde{\mathbf{A}}^T \times \mathbf{m}^T. \tag{3.8}$$

The last $n-k$ elements in the main diagonal line of $\mathbf{D}$ and in the column vector $\tilde{\mathbf{A}}^T \times \mathbf{m}^T$ are all zeros. Thus, the last $n-k$ equations in (3.8) are all trivial equations $0 = 0$ and we can reduce (3.8) to

$$(\mathbf{f}^T)_{k\times 1} = \mathbf{D}_{k\times k} \times (\mathbf{F}^{-1} \times \mathbf{W})_{k\times k} \times (\tilde{\mathbf{A}}^T \times \mathbf{m}^T)_{k\times 1}.$$

Since $\mathbf{W}$ is diagonal, $(\mathbf{F}^{-1} \times \mathbf{W})_{k\times k} = \mathbf{F}^{-1}_{k\times k} \times \mathbf{W}_{k\times k}$. Moreover, $(\tilde{\mathbf{A}}^T \times \mathbf{m}^T)_{k\times 1} = \mathbf{A}^T \times \mathbf{m}^T$ and $\mathbf{F}^{-1}_{k\times k}$, $\mathbf{D}_{k\times k}$, $\mathbf{W}_{k\times k}$ and $\mathbf{A}$ are all invertible. Thus, we have

$$\mathbf{m}^T = (\mathbf{A}^T)^{-1} \times (\mathbf{W}_{k\times k})^{-1} \times (\mathbf{F}^{-1}_{k\times k})^{-1} \times (\mathbf{D}_{k\times k})^{-1} \times (\mathbf{f}_{1\times k})^T.$$

■

The vectors $\mathbf{f}_{1\times k}$ and $\mathbf{m}$ can be viewed as an element in the output list and the corresponding candidate for original message, respectively. We then present an algorithm to retrieve the messages encoded via any generator-matrix of a RS code from the output list of an interpolation-based list decoding algorithm.

Algorithm 3.1

Input: The zeros $(\alpha^b, \alpha^{b+1}, \ldots, \alpha^{n-k-1+b})$ *of* $\mathcal{C}$ *and its generator matrix* $\mathbf{G}_a$.
Output: The desired messages corresponding to the elements of the output list.
Precomputation (to be performed only once):

i. Compute $g(x) = \prod_{i=0}^{n-k-1}(x - \alpha^{b+i}) = \sum_{i=0}^{n-k} g_i x^i$ *and construct the matrix* $\mathbf{G}$ *for which the* $(i+1)^{th}$ *row,* $0 \le i \le n-1$, *is the right cyclic shift of the* n*-tuple* $(g_0, g_1, \ldots, g_{n-k}, 0, \ldots, 0)$ *by* i *positions.*

ii. Find $\mathbf{A}$ *such that* $\mathbf{G}_a = \mathbf{A} \times \mathbf{G}$ *and* $(\mathbf{A}^T)^{-1}$. *(Note: The matrix* $\mathbf{A}$ *can be easily found using standard techniques in linear algebra since* $\mathbf{G}$ *is in row echelon form.)*

iii. Compute the inverse GFFT of the n-tuple $(g_0, g_1, \ldots, g_{n-k}, 0, \ldots, 0)$. *Then right cyclic shift the resultant vector by* $b-1$ *positions to obtain* $(G_0, G_1, \ldots, G_{k-1}, 0, \ldots, 0)$.

iv. Set $(\mathbf{D}_{k\times k})^{-1} = \mathrm{Diag}(G_0^{-1}, G_1^{-1}, \ldots, G_{k-1}^{-1})$ *and* $(\mathbf{W}_{k\times k})^{-1} = \mathrm{Diag}(1, \alpha^{-(b-1)}, \alpha^{-2(b-1)}, \ldots, \alpha^{-(k-1)(b-1)})$.

v. Compute $(\mathbf{F}_{k\times k}^{-1})^{-1}$ *and* $\mathbf{B} = (\mathbf{A}^T)^{-1} \times (\mathbf{W}_{k\times k})^{-1} \times (\mathbf{F}_{k\times k}^{-1})^{-1} \times (\mathbf{D}_{k\times k})^{-1}$.[1]

List decoding & message recovery:

1. *Compute* $\bar{\mathbf{r}} = \mathbf{r} \times \mathbf{W} = (r_0, r_1\alpha^{(b-1)}, \ldots, r_{n-1}\alpha^{(n-1)(b-1)})$ *where* $\mathbf{r}$ *is the hard-decision received vector.*

2. *List decode* $\bar{\mathbf{r}}$.

3. *If the output list is not empty, then for each element* $\mathbf{f}_{1\times k}$ *in this list, return* $\mathbf{B} \times (\mathbf{f}^T)_{k\times 1}$.

We have a few remarks for the above results. First, since the average list size of the interpolation-based algorithms have been shown very close to unity [56, 43], about $k^2 + n - 1$ multiplications is introduced by Step 1) and 3) on average. For code rate of practical interest, an average overhead of $O(k^2)$ multiplications is incurred besides the computations incurred by Step 2). Second, if $b = 1$, since $\mathbf{W}$ is reduced to an identity matrix, $\mathbf{W}_{k\times k}$ can be omitted in the computation of $\mathbf{B}$. Finally, if the messages are encoded via the generator polynomial $g(x)$, $\mathbf{A}$ is reduced to an identity matrix.

Example 3.1 *Let* $\mathcal{C}$ *be a* $(7, 4)$ *RS code over* GF(8) *with zeros* $\alpha^2, \alpha^3, \alpha^4$. *Its*

[1] Since $\mathbf{F}_{k\times k}^{-1}$ is symmetric, its inverse can be computed by eigenvalue decomposition.

generator matrix is

$$\mathbf{G}_a = \begin{pmatrix} \alpha^5 & \alpha & \alpha^3 & \alpha & \alpha^3 & \alpha^2 & \alpha \\ \alpha^6 & 0 & \alpha^4 & \alpha^3 & \alpha^6 & 1 & \alpha^2 \\ \alpha^6 & \alpha^2 & \alpha^2 & \alpha^2 & 0 & \alpha^5 & \alpha^6 \\ \alpha^4 & \alpha^6 & \alpha^3 & \alpha^2 & 1 & 0 & \alpha \end{pmatrix}.$$

Following Algorithm 3.1, we obtain

$$(\mathbf{A}^T)^{-1} = \begin{pmatrix} \alpha^2 & 1 & \alpha^2 & 0 \\ \alpha^2 & \alpha & \alpha^2 & \alpha \\ \alpha^3 & \alpha^6 & \alpha^5 & \alpha^5 \\ \alpha^6 & \alpha^3 & \alpha^2 & \alpha \end{pmatrix}.$$

Applying the inverse GFFT to $(g_0, g_1, g_2, g_3, 0, 0, 0) = (\alpha^2, \alpha^3, 1, 1, 0, 0, 0)$ *and right cyclic shifting the resulting vector by* 1 *position yields* $(G_0, G_1, G_2, G_3, 0, 0, 0) = (\alpha^6, \alpha^5, 1, \alpha^5, 0, 0, 0)$ *and so* $(\mathbf{D}_{4\times4})^{-1} = \mathrm{Diag}(\alpha, \alpha^2, 1, \alpha^2)$. *Now,* $(\mathbf{W}_{4\times4})^{-1} = \mathrm{Diag}(1, \alpha^6, \alpha^5, \alpha^4)$ *and*

$$(\mathbf{F}_{4\times4}^{-1})^{-1} = \begin{pmatrix} \alpha^4 & \alpha^3 & \alpha^5 & \alpha^3 \\ \alpha^3 & 1 & 0 & \alpha \\ \alpha^5 & 0 & \alpha^3 & \alpha^2 \\ \alpha^3 & \alpha & \alpha^2 & \alpha^6 \end{pmatrix}.$$

Hence,

$$\mathbf{B} = (\mathbf{A}^T)^{-1} \times (\mathbf{W}_{k\times k})^{-1} \times (\mathbf{F}_{k\times k}^{-1})^{-1} \times (\mathbf{D}_{k\times k})^{-1} = \begin{pmatrix} \alpha^5 & \alpha^3 & \alpha & \alpha \\ \alpha^4 & \alpha^5 & \alpha^3 & 1 \\ \alpha^5 & \alpha^2 & 1 & \alpha \\ \alpha & \alpha & \alpha^2 & \alpha \end{pmatrix}. \quad (3.9)$$

Suppose the codeword $\mathbf{c} = (\alpha^2, 0, \alpha, 0, 0, \alpha^3, \alpha^6)$ *is transmitted and received as* $\mathbf{r}$. *If list decoding the vector* $\bar{\mathbf{r}} = \mathbf{r} \times \mathbf{W}$ *is successful,* $\mathbf{f} = (\alpha, 0, \alpha^5, 1, 0, 0, 0)$ *will be in the output list. We can recover the original message* $\mathbf{m}^T = \mathbf{B} \times \mathbf{f}_{4\times1}^T = (\alpha^3, \alpha^2, 0, \alpha^5)^T$. *It can be verified that* $\mathbf{c} = \mathbf{m} \times \mathbf{G}_a$.

3.4 Summary

We have established a relationship between codewords resulting from generator-matrix-based encoding, and codewords obtained via the evaluation map. We have further derived from this relationship, an algorithm for recovering generator-matrix-based coded data under interpolation-based list decoding.

Chapter 4

Synthesis of Multisequences Having Unknown Elements in the Middle and Decoding Applications

In this chapter, we first propose an algorithm incorporating the FIA to solve the multisequences synthesis problem when the sequences are nonconsecutive. We then show that GRS codes with nonconsecutive syndromes can be decoded by this algorithm. We also show that Folded GRS (FGRS) codes can be constructed from GRS codes but their row codes may have nonconsecutive syndrome sequences. The proposed algorithm may be applied to decode such FGRS codes with row codes having nonconsecutive syndromes associated. FGRS codes have the potential to be deployed in data storage applications and wireless communication systems where burst errors tend to occur owing to uneven media surface and deep fading caused by multipath transmission in respective systems.

4.1 Introduction

The FIA [23] can be applied to find the minimal initial set of linearly dependent columns in an array with known elements. It can also solve the multisequences synthesis problem when the sequences contain unknown elements in the tail end.

Example 4.1 *Each rows in the array in (4.1) specifies a sequences over* GF(2). *The FIA can find the minimal initial set of linearly dependent columns in (4.1).*

$$\begin{pmatrix} 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 \end{pmatrix} \tag{4.1}$$

It generates the polynomial $\sigma(x) = \sigma_2 x^2 + \sigma_1 x + 1 = x^2 + x + 1$ *which annihilates the first three columns of (4.1). i.e.,*

$$1 \times \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \end{pmatrix} + \sigma_1 \times \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \end{pmatrix} + \sigma_2 \times \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} = \mathbf{0}. \tag{4.2}$$

Applying the FIA, we can find the error locator polynomials from the syndrome sequences of BCH codes, RS codes [23] and FRS codes [44].

However, the FIA may fail when nonconsecutive sequences are involved. By nonconsecutive sequence, we mean a sequence with unknown elements in the middle of it. The FIA fails in this case because it may need the values for the unknown elements to compute discrepancies and update $\sigma(x)$.

Example 4.2 *Consider finding the minimal initial set of linearly dependent columns in (4.3) which is over* GF(2). *Each of the two sequences* $(1, 1, w, 1, 0)$

and $(1, w, 1, 1, 0)$ *has an element with unknown value* w*.*

$$\begin{pmatrix} 1 & 1 & w & 1 & 0 \\ 1 & w & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 \end{pmatrix} \tag{4.3}$$

The FIA can only process the elements at $(0, 0)$ *and* $(1, 0)$ *and fails to find the minimal initial set of linearly dependent columns in (4.3). But we can see that the solution for Example 4.1 is a solution for this example if* $w = 0$*.*

4.2 Synthesizing Multisequences with Unknown Elements in the Middle

To solve the problem in Example 4.2, since $\deg(\sigma(x)) \geq 1$, we begin with the assumption $L := \deg(\sigma(x)) = 1$ and the sub-array of (4.3).

$$\begin{pmatrix} 1 & 1 \\ 1 & w \\ 0 & 1 \\ 1 & 1 \end{pmatrix} \tag{4.4}$$

Since the 2×2 sub-array in (4.4),

$$\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}, \tag{4.5}$$

has full row and column rank, we increase L by 1 and set $\sigma(x) := \sigma_2 x^2 + \sigma_1 x + 1$. For the columns of the following sub-array,

$$\begin{pmatrix} 1 & 1 & w \\ 1 & w & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{pmatrix}, \tag{4.6}$$

to be linearly dependent, it is required that

$$\begin{aligned} w + \sigma_1 w + \sigma_2 &= 0 \\ 1 + \sigma_1 w + \sigma_2 &= 0 \\ 1 + \sigma_1 &= 0 \\ \sigma_1 + \sigma_2 &= 0. \end{aligned} \tag{4.7}$$

An obvious solution to (4.7) involves setting $\sigma_1 = 1, \sigma_2 = 1$ and $w = 0$.

A careful study on the FIA shows that permutating the rows of an array will not change the final solution for the minimal initial set of linearly dependent columns of the array. If we order the rows of the array in (4.3) as

$$\begin{pmatrix} 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & w & 1 & 0 \\ 1 & w & 1 & 1 & 0 \end{pmatrix}, \tag{4.8}$$

we have $\sigma(x) = x^2+x+1$ by applying the FIA to the array in (4.8). The unknown elements are not involved in the computation of $\sigma(x)$. Also, from the last two rows of the following sub-array of the array in (4.8),

$$\begin{pmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & w \\ 1 & w & 1 \end{pmatrix}, \tag{4.9}$$

we have the equations

$$\begin{aligned} w + 1 + 1 &= 0 \\ 1 + w + 1 &= 0, \end{aligned} \tag{4.10}$$

which imply $w = 0$. Thus, the minimal initial set of linearly dependent columns in (4.8) consists of the first 3 columns with $w = 0$.

From these examples, we can derive a general procedure for the synthesis of the multisequences with unknown elements in the middle. Denote by $\mathbf{A}$, the array whose rows are given sequences including unknown elements in the middle. Denote the sub-array consists of the first $L+1$ columns of $\mathbf{A}$ by $\mathbf{A}_{L+1}$ with given L. Also, denote by Order($\mathbf{A}_{L+1}$) a procedure which permutes the rows in $\mathbf{A}_{L+1}$ such that all the rows containing unknown elements are bottom-most rows in the resulting array, denoted by $\mathbf{B}_{L+1}$. We further denote the largest sub-array of $\mathbf{B}_{L+1}$ by $\mathbf{U}_{L+1}$ which has $L+1$ columns and no unknown elements. It is possible that $\mathbf{U}_{L+1}$ does not exist.

Assume that $\mathbf{U}_{L+1}$ exists for certain L. To make use of the rows in $\mathbf{A}_{L+1}$ which do not contain unknown elements, we process the rows of $\mathbf{A}_{L+1}$ by Order($\mathbf{A}_{L+1}$) and obtain $\mathbf{B}_{L+1}$. From $\mathbf{B}_{L+1}$, we have $\mathbf{U}_{L+1}$. The FIA can be applied to $\mathbf{U}_{L+1}$ and we denote such a operation by FIA($\mathbf{U}_{L+1}$). If the FIA fails to generate the $\sigma(x)$ from $\mathbf{U}_{L+1}$, then $\mathbf{U}_{L+1}$ must have full column rank. We therefore increase L by one and the above steps are repeated. Otherwise, if a particular $\sigma(x)$ with degree L is generated, by the FIA, a group of equations Ψ similar to (4.10), involving the values of the unknown elements in $\mathbf{B}_{L+1}$, are derived. If $\deg(\sigma(x)) < L$, we ignore the $\sigma(x)$ generated and set $\sigma(x) := \sum_{i=1}^{L} \sigma_i x^i + 1$ with $\sigma_L \neq 0$. We then derive a group of equations Ψ involving the unknown elements in $\mathbf{B}_{L+1}$ and the σ_i's where $1 \leq i \leq L$.

If $\mathbf{U}_{L+1}$ does not exist, we also set $\sigma(x) := \sum_{i=1}^{L} \sigma_i x^i + 1$ with $\sigma_L \neq 0$ and derive Ψ involving the unknown elements in $\mathbf{B}_{L+1}$ and the σ_i's.[1]

If a solution to Ψ exists, then $\sigma(x)$ specifies the minimal initial set of linearly dependent columns of $\mathbf{A}$. If no solution exists, we increase L by one and repeat the above steps.

The procedure fails when all the columns of $\mathbf{A}$ are linearly independent. The

[1]The set of equations Ψ can be solved using standard techniques.

following algorithm explicitly summarizes the above procedure.

Algorithm 4.1

Input: An array $\mathbf{A} \in \mathrm{GF}(q)^{m \times n}$ *whose rows are the prescribed sequences.*

Output: $\sigma(x) \in \mathrm{GF}(q)[x]$.

Initialize $L := 1$.

1. *If* $L > n-1$, *declare failure and exit; otherwise, apply* $\mathrm{Order}(\mathbf{A}_{L+1})$.
2. *If* $\mathbf{U}_{L+1}$ *exists, invoke* $\mathrm{FIA}(\mathbf{U}_{L+1})$ *to generate* $\sigma(x)$. *Otherwise, go to 4).*
3. *If* $FIA(\mathbf{U}_{L+1})$ *fails, go to step 7). Otherwise, if* $\deg(\sigma(x)) = L$, *construct* Ψ *with unknowns being the unknown elements in* $\mathbf{B}_{L+1}$ *and go to step 5).*
4. *Set* $\sigma(x) := \sum_{i=1}^{L} \sigma_i x^i + 1$ *and construct* Ψ *with unknowns being the unknown elements in* $\mathbf{B}_{L+1}$ *and* σ_i*'s* $(1 \le i \le L)$.
5. *If no solution to* Ψ *exists, go to step 7).*
6. *Output* $\sigma(x)$ *and exit.*
7. $L := L + 1$, *go to step 1).*

4.3 Decoding GRS Codes

Let α be primitive in $\mathrm{GF}(q = p^m)$, $\mathbf{v} = (v_0, v_1, \ldots, v_{n-1})$ where $v_0, v_1, \ldots, v_{n-1} \in \mathrm{GF}(q) \backslash \{0\}$ and $\mathbf{a} = (\alpha^{i_0}, \alpha^{i_1}, \ldots, \alpha^{i_{n-1}})$ such that $\alpha^{i_0}, \alpha^{i_1}, \ldots, \alpha^{i_{n-1}}$ are distinct elements in $\mathrm{GF}(q)$. The $\mathrm{GRS}_{\mathbf{a},\mathbf{v}}(n, k+1)$ code over $\mathrm{GF}(q)$ encodes an information vector $(f_0, f_1, \ldots, f_k) \in \mathrm{GF}(q)^{k+1}$ to a codeword $\mathbf{c} = (c_0, c_1, \ldots, c_{n-1})$ of this GRS code, where $c_j = v_j f(\alpha^{i_j}) \in \mathrm{GF}(q)$ for $0 \le j \le n-1$ and $f(x) = \sum_{l=0}^{k} f_l x^l \in \mathrm{GF}(q)[x]_{k+1}$. Without loss of generality, let $0 \le i_j \le q-2$. We call the set $\{\alpha^{i_0}, \alpha^{i_1}, \ldots, \alpha^{i_{n-1}}\}$ the support set of $\mathrm{GRS}_{\mathbf{a},\mathbf{v}}(n, k+1)$.

Let $\mathbf{r} = (r_0, r_1, \ldots, r_{n-1})$. The transmitted codeword can be recovered as $\mathbf{c} = \mathbf{r} - \mathbf{e}$ if the error vector $\mathbf{e} = (e_0, e_1, \ldots, e_{n-1})$ can be recovered from the syndrome sequence $\mathbf{s}$ in the decoding.

Let $\text{ord}(\beta) = n$. The GFFT of the error vector $\mathbf{e}$ is $E = (E_0, E_1, \ldots, E_{n-1})$ where $E_j = \sum_{i=0}^{n-1} e_i \beta^{ij}$ as defined in [7]. Let Φ be a multiplicative cyclic subgroup of $\text{GF}(q)\backslash\{0\}$ and β its generator. If Φ is the support set of $\text{GRS}_{\mathbf{a},\mathbf{v}}(n, k+1)$ and β^j is a zero of the code, the jth syndrome is computed as

$$S_j = \sum_{i=0}^{n-1} r_i \beta^{ij} = \sum_{i=0}^{n-1} (c_i + e_i)\beta^{ij} = \sum_{i=0}^{n-1} e_i \beta^{ij} = E_j. \tag{4.11}$$

Thus, if we can recover the vector E from the syndrome sequence and then obtain $\mathbf{e}$ by the inverse GFFT of E, we can also recover the transmitted codeword.

If the zeros of $\text{GRS}_{\mathbf{a},\mathbf{v}}(n, k+1)$ are consecutive powers of β, its syndrome sequence is consecutive and E can be recovered by the Generalized Iterative Algorithm for Multiple Sequences (GIAMS) as in [23]. When the number of errors v in the received vector does not exceed the corresponding Hartmann-Tzeng bound [38], the GIAMS may be used to find the error locator polynomial $\sigma(x) = \sum_{i=1}^{v} \sigma_i x^i + 1$ by viewing E as a linear recurring sequence.

However, if the zeros of $\text{GRS}_{\mathbf{a},\mathbf{v}}(n, k+1)$ code are not consecutive powers of β, the syndrome sequence will have unknown elements. To find $\sigma(x)$, Algorithm 4.1 can be applied. In this application, the rows of $\mathbf{A}$ are the left-shifts of the syndrome sequence. We give an example of this application as follows.

Example 4.3 *Let α be primitive in* $\text{GF}(8)$. *Let* $\mathbf{a} = (1, \alpha, \ldots, \alpha^6)$ *and* $\mathbf{v} = (1, 1, 1, 1, 1, 1, 1)$. *The* $\text{GRS}_{\mathbf{a},\mathbf{v}}(7, 3)$ *over* $\text{GF}(8)$ *has zeros* $\alpha, \alpha^3, \alpha^4, \alpha^6$. *Suppose a codeword is transmitted and received as* $\mathbf{r} = (0, 0, \alpha, 0, 1, 0, 0)$. *Using (4.11), the known elements of E are* $E_1 = \alpha^6, E_3 = \alpha^4, E_4 = 0, E_6 = \alpha^4$. *Due to the unknown elements E_2 and E_5, as in [23], we arrange the syndrome sequence and its left-shifts in the following array, where X can be viewed as a wildcard and E_2*

and E_5 represent the unknown elements in the syndrome sequence.

$$\mathbf{A} = \begin{pmatrix} \alpha^6 & E_2 & \alpha^4 & 0 & E_5 & \alpha^4 \\ E_2 & \alpha^4 & 0 & E_5 & \alpha^4 & X \\ \alpha^4 & 0 & E_5 & \alpha^4 & X & X \\ 0 & E_5 & \alpha^4 & X & X & X \\ E_5 & \alpha^4 & X & X & X & X \\ \alpha^4 & X & X & X & X & X \end{pmatrix}. \tag{4.12}$$

We apply Algorithm 4.1 to find the error locator polynomial.

Initially, $L = 1$,

$$\mathbf{B}_{L+1} = \begin{pmatrix} \alpha^4 & \alpha^6 & E_2 & 0 & E_5 & \alpha^4 \\ 0 & E_2 & \alpha^4 & E_5 & \alpha^4 & X \end{pmatrix}^T$$

and $\mathbf{U}_{L+1} = \begin{pmatrix} \alpha^4 & 0 \end{pmatrix}$.

Applying the FIA to $\mathbf{U}_{L+1}$ to generate $\sigma(x) = \sigma_1 x + 1$ results in failure, since $0 \times 1 + \alpha^4 \times \sigma_1 = 0$ implies $\sigma_1 = 0$. So we increase L to 2.

Now,

$$\mathbf{A}_{L+1} = \mathbf{B}_{L+1} = \begin{pmatrix} \alpha^6 & E_2 & \alpha^4 & 0 & E_5 & \alpha^4 \\ E_2 & \alpha^4 & 0 & E_5 & \alpha^4 & X \\ \alpha^4 & 0 & E_5 & \alpha^4 & X & X \end{pmatrix}^T.$$

$\mathbf{U}_{L+1}$ does not exist in this case, since all the rows in $\mathbf{A}_3$ contain unknown elements. We therefore set $\sigma(x) := \sigma_2 x^2 + \sigma_1 x + 1$, derive the set Ψ of equations in (4.13). We then solve for the solution to the σ_i's and the unknown elements in $\mathbf{B}_{L+1}$. From the first four rows of $\mathbf{B}_{L+1}$, we have

$$\begin{aligned} \alpha^4 + \sigma_1 E_2 + \sigma_2 \alpha^6 &= 0 \\ \sigma_1 \alpha^4 + \sigma_2 E_2 &= 0 \\ E_5 + \sigma_2 \alpha^4 &= 0 \\ \alpha^4 + \sigma_1 E_5 &= 0. \end{aligned} \tag{4.13}$$

From (4.13), we have

$$\alpha^2 \sigma_2^4 + \sigma_2^3 + 1 = 0.$$

The LHS of the last equality is a polynomial in $\mathrm{GF}(8)[\sigma_2]$. *Its coefficients in vector form is* $(1,0,0,1,\alpha^2,0,0) \in \mathrm{GF}(8)^7$. *The GFFT of this vector is* $(\alpha^2,\alpha^5,\alpha^5,\alpha^2,0,1,0)$. *By property of GFFT [7, GFFT Property 6],* σ_2 *has two solutions,* α^4 *and* α^6. *The corresponding solutions for* σ_1 *are* α^3 *and* α, *respectively.*

The polynomial $\alpha^4x^2+\alpha^3x+1$ *is not a valid error locator polynomial because it cannot be factorized into distinct linear factors over the support set* $\mathrm{GF}(8)\backslash\{0\}$ *of this code. The other solution* $\sigma(x) = \alpha^6x^2 + \alpha x + 1$ *can be factorized as* $\sigma(x) = (\alpha^2x+1)(\alpha^4x+1)$ *which indicates the error locations* α^2 *and* α^4.

Substituting $\sigma_1 = \alpha$ *and* $\sigma_2 = \alpha^6$ *into (4.13) yields* $E_2 = \alpha^6$ *and* $E_5 = \alpha^3$. *We also have* $E_0 = \alpha^3$. *Then* E *turns out to be* $(\alpha^3,\alpha^6,\alpha^6,\alpha^4,0,\alpha^3,\alpha^4)$. *Its inverse GFFT is* $\mathbf{e} = (0,0,\alpha,0,1,0,0)$. *Hence, the transmitted codeword is* $\mathbf{r}-\mathbf{e} = (0,0,0,0,0,0,0)$.

When the support set $\hat{\Phi}$ of a GRS code over $\mathrm{GF}(q)$ is not a cyclic subgroup of $\mathrm{GF}(q)\backslash\{0\}$, we can always find a minimal cyclic subgroup Φ in $\mathrm{GF}(q)\backslash\{0\}$ with generator β such that $\hat{\Phi} \subset \Phi$. If the zeros of this code are not consecutive powers of β, we can view this code as a shortened code of a GRS code which has support set Φ. This reduces the problem to the case studied above.

Moreover, according to the results in [23], the solution to Ψ corresponding to a valid error locator polynomial is unique when we decode up to the Hartmann-Tzeng bound.

4.4 Folded GRS Codes From GRS Codes

Let $\mathcal{C}$ be a primitive $(n, k+1)$ RS code over $\mathrm{GF}(q)$ where $n = L \times N$ such that $L, N > 1$. We can fold a codeword $(c_0, c_1, \ldots, c_{n-1})$ of $\mathcal{C}$ into the following

$L \times N$ array.

$$\begin{pmatrix} c_0 & c_1 & \cdots & c_{N-1} \\ c_N & c_{N+1} & \cdots & c_{2N-1} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n-2N+1} & c_{n-2N+2} & \cdots & c_{n-N-1} \\ c_{n-N} & c_{n-N-1} & \cdots & c_{n-1} \end{pmatrix}. \tag{4.14}$$

After applying the GFFT to each column of the array in (4.14), we have the FRS code given by

$$\begin{pmatrix} b_0 & b_1 & \cdots & b_{N-1} \\ b_N & b_{N+1} & \cdots & b_{2N-1} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n-2N+1} & b_{n-2N+2} & \cdots & b_{n-N-1} \\ b_{n-N} & b_{n-N-1} & \cdots & b_{n-1} \end{pmatrix}. \tag{4.15}$$

Let the order of α be $\mathrm{ord}(\alpha) = n$. Generally, if a q-ary (n, k) RS code has zeros $\alpha^b, \alpha^{b+1}, \ldots, \alpha^{b+d-2}$, each row of the array in (4.15) is a GRS code [96, Theorem 2] and the zeros of the rth $(0 \leq r \leq L-1)$ row code are $\alpha^{r+\lceil \frac{b-r}{L} \rceil L}, \alpha^{r+(\lceil \frac{b-r}{L} \rceil+1)L}, \ldots, \alpha^{r+\lfloor \frac{d+b-2-r}{L} \rfloor L}$ [96, Corollary 4].

In the following, we will show that any GRS code over GF(q) can be transformed to an $L \times N$ FGRS code if its support set is a subset of a cyclic subgroup Φ of GF(q)$\backslash\{0\}$ and $|\Phi| = n = L \times N$.

Let $\hat{\mathcal{C}}$ be a GRS code over GF(q) with length $n'(\leq n)$, dimension $k+1$ and support set $\hat{\Phi}$. Assume $\beta^l \in GF(q)\backslash\{0\}$ be a zero of $\hat{\mathcal{C}}$. There is a minimal cyclic subgroup $\Phi \subseteq \mathrm{GF}(q)\backslash\{0\}$ of order n with generator β such that $\hat{\Phi} \subseteq \Phi$ and $\beta^l \in \Phi$. Then code $\hat{\mathcal{C}}$ can be extended to a new GRS code, denoted by $\mathcal{C}$, by inserting zero symbols at coordinates specified by $\Phi - \hat{\Phi}$. This new GRS code has codelength n and the same dimension as $\hat{\mathcal{C}}$. Assume that $\hat{\mathbf{c}} \in \hat{\mathcal{C}}$ is extended to $\mathbf{c} \in \mathcal{C}$ as above. Denote their code polynomial as $\hat{c}(x) = \sum_{i=0}^{n'-1} \hat{c}_i x^i$

and $c(x) = \sum_{i=0}^{n-1} c_i x^i$, respectively, where $\hat{c}_i$ and c_i are the respective ith code symbols of $\hat{\mathbf{c}}$ and $\mathbf{c}$. By the above extension, we have

$$\begin{cases} c_i = \hat{c}_i, & \text{if } \beta^i \in \hat{\Phi}; \\ c_i = 0, & \text{if } \beta^i \in (\Phi - \hat{\Phi}). \end{cases} \tag{4.16}$$

Thus, we have $c(\beta^l) = \sum_{i=0}^{n-1} c_i \beta^{li} \sum_{i=0}^{n'-1} \hat{c}_i \beta^{li} = 0$ and $\mathcal{C}$ has the same zeros as $\hat{\mathcal{C}}$. If $n = L \times N$, $\mathcal{C}$ can be folded into an array as in (4.14) and there is an element $\theta \in \Phi$ of order L in GF(q). The columns of (4.14) can be transformed by the L-point GFFT and an FGRS code corresponding to the original code $\hat{\mathcal{C}}$ can be obtained. From above, we have Theorem 4.1.

Theorem 4.1 *If the extended support set Φ of a GRS code satisfies $|\Phi| = L \times N$, where $L, N \in \mathbb{Z}^+$, an $L \times N$ FGRS code can be constructed from this GRS code.*

A property of the resulting FGRS code is given by Theorem 4.2.

Theorem 4.2 *Let a GRS code $\hat{\mathcal{C}}$ over* GF(q) *of length n' have $\beta^l \in$ GF(q)$\backslash\{0\}$ as a zero where* ord$(\beta) = n = L \times N \geq n'$. *Also, let $\mathbf{B}$ be the $L \times N$ FGRS code constructed from $\hat{\mathcal{C}}$. The $\bar{l}$th row code of $\mathbf{B}$ has zero β^l where $\bar{l} = l \pmod{L}$.*

Proof: Assume the FGRS code array $\mathbf{B}$ as in (4.15) is obtained from the GRS code $\hat{\mathcal{C}}$ and $\theta = \beta^N$. Thus, we have ord$(\theta) = L$. By the definition of the GFFT, we have $b_{sN+j} = \sum_{i=0}^{L-1} c_{iN+j} \theta^{si}$. The sth row code of $\mathbf{B}$ in polynomial form is

$$\begin{aligned} b_s(y) &= \sum_{j=0}^{N-1} b_{sN+j} y^j = \sum_{j=0}^{N-1} \sum_{i=0}^{L-1} c_{iN+j} \theta^{si} y^j \\ &= \sum_{i=0}^{L-1} \sum_{j=0}^{N-1} c_{iN+j} \beta^{sNi} y^j. \end{aligned} \tag{4.17}$$

Assume $l = \bar{l} + hL$. Since β^l is a zero of $\hat{\mathcal{C}}$ and also a zero of $\mathcal{C}$, we have

$$\begin{aligned} 0 &= c(x)|_{x=\beta^l} = \sum_{i=0}^{L-1}\sum_{j=0}^{N-1} c_{iN+j}x^{iN+j}|_{x=\beta^l} \\ &= \sum_{i=0}^{L-1}\sum_{j=0}^{N-1} x_{iN+j}(\beta^{\bar{l}+hL})^{iN+j} \\ &= \sum_{i=0}^{L-1}\sum_{j=0}^{N-1} c_{iN+j}\beta^{\bar{l}Ni}\beta^{hiLN}(\beta^l)^j \\ &= \sum_{i=0}^{L-1}\sum_{j=0}^{N-1} c_{iN+j}\beta^{\bar{l}Ni}(\beta^l)^j. \end{aligned} \tag{4.18}$$

The last equality is because $\mathrm{ord}(\beta) = L \times N$. Comparing (4.17) and (4.18), we have $b_s(\beta^l) = 0$, when $s = \bar{l}$. ■

According to Theorem 4.2, the zeros of the original code $\hat{\mathcal{C}}$ are distributed among the row codes of the corresponding FGRS code $\mathbf{B}$. As in [96, Lemma 1], all the row codes have the same support set $\{1, \gamma^1, \ldots, \gamma^{N-1}\}$ where $\mathrm{ord}(\gamma) = N$. However, the zeros of the row codes except the 0th row code may be not from this support set.

Example 4.4 *A* $(15, 6)$ *RS code over* GF(16) *has zeros* $1, \beta^1, \ldots, \beta^8$ *where* β *is primitive in* GF(16)*. This code can be folded into a* 3×5 *FRS code. The* 0*th row has zeros* $\{1, \beta^3, \beta^6\} = \{1, \gamma, \gamma^2\}$*, the* 1*st row has zeros* $\{\beta, \beta\gamma, \beta\gamma^2\}$ *and the* 2*nd row has zeros* $\{\beta^2, \beta^2\gamma, \beta^2\gamma^2\}$ *where* $\gamma = \beta^3$*. But the support set of the row codes of this FRS code is a cyclic subgroup* $\Phi = \{1, \gamma, \gamma^2, \gamma^3, \gamma^4\}$*.*

Let the code polynomial of the codeword $\check{\mathbf{c}} = (\check{c}_0, \check{c}_1, \ldots, \check{c}_{N-1})$ of the vth row code $\check{\mathcal{C}}$ be $\check{c}(x) = \sum_{j=0}^{N-1} \check{c}_j x^j$. If $\beta^v\gamma^i$ is a zero of this row code, we have $\check{c}(x) = \sum_{j=0}^{N-1} \check{c}_j(\beta^v\gamma^i)^j = \sum_{j=0}^{N-1}(\check{c}_j\beta^{vj})\gamma^{ij}$. Such a row code can be mapped to another code $\mathcal{C}'$ where $\mathbf{c}' = (\check{c}_0, \beta^v\check{c}_1, \ldots, \beta^{v(N-1)}\check{c}_{N-1}) \in \mathcal{C}'$. The code $\mathcal{C}'$ has the same support set Φ and its zeros are from Φ as well. The mapping of $\check{\mathcal{C}}$ to $\mathcal{C}'$ and its inverse is denoted as $\mathcal{T}$ and $\mathcal{T}^{-1}$, respectively. Thus, all the L $\mathcal{T}$-mapped row

codes in the FGRS code array can be decoded as codes with zeros in the same support set.

Suppose that the array in (4.14) is transmitted column by column in a burst error channel. We can obtained the syndromes for the $\mathcal{T}$-mapped row codes of the FGRS code from the syndromes for the original code $\hat{\mathcal{C}}$. This is shown as follows. If β^l $(l = \bar{l} + Lh)$ is a zero of the original GRS code $\hat{\mathcal{C}}$ and the corresponding syndrome is S_l, by Theorem 4.2, β^l is a zero of the $\bar{l}$th row code in the corresponding FGRS code array. Let the corresponding syndrome for the $\mathcal{T}$-mapped row code be denoted by $S_l^{(\bar{l})}$ and let $r_{lN+j}^{(b)}$ be the entry at (l, j) in the array obtained by the GFFT of each column in the received array and $\gamma = \beta^L$.

$$\begin{aligned}
E_l &= S_l = \sum_{i=0}^{n'-1} \hat{r}_i(\beta^l)^i = \sum_{i=0}^{n-1} r_i(\beta^l)^i \\
&= \sum_{i=0}^{L-1}\sum_{j=0}^{N-1} r_{iN+j}(\beta^l)^{iN+j} = \sum_{j=0}^{N-1}\sum_{i=0}^{L-1} r_{iN+j}(\beta^N)^{li}\beta^{lj} \\
&= \sum_{j=0}^{N-1} r_{lN+j}^{(b)}(\beta^l)^j = \sum_{j=0}^{N-1} r_{lN+j}^{(b)}(\beta^{\bar{l}+hL})^j \\
&= \sum_{j=0}^{N-1}(r_{lN+j}^{(b)}\beta^{\bar{l}j})(\beta^L)^{hj} = \sum_{j=0}^{N-1}(r_{lN+j}^{(b)}\beta^{\bar{l}j})\gamma^{hj} \\
&= S_l^{(\bar{l})} = E_l^{(\bar{l})}. \qquad (4.19)
\end{aligned}$$

If the support set of the original GRS code is a cyclic subgroup with generator β and its zeros are consecutive powers of β, the syndrome sequences for the $\mathcal{T}$-mapped row codes are also consecutive. Thus, E can be recovered by the GIAMS. However, if its zeros are not consecutive powers of β, the syndromes for the $\mathcal{T}$-mapped row codes may not be consecutive.

Example 4.5 *Let* $\mathbf{a} = (1, \alpha, \alpha^2, \ldots, \alpha^{14})$ *and* $\mathbf{v} = (1, 1, 1, \ldots, 1)$ *where* α *is primitive in* GF(16). *A* $\mathrm{GRS}_{\mathbf{a},\mathbf{v}}(15, 3)$ *code over* GF(16) *has zeros* $1, \alpha, \alpha^2, \alpha^3, \alpha^4, \alpha^5, \alpha^7, \alpha^8, \alpha^9, \alpha^{11}, \alpha^{12}, \alpha^{13}$. *A* 3×5 *FGRS code can be constructed*

Row 0	S_0	S_3	E_6	S_9	S_{12}
Row 1	S_1	S_4	S_7	E_{10}	S_{13}
Row 2	S_2	S_5	S_8	S_{11}	X

Figure 4.1: Nonconsecutive syndrome sequences of row codes.

from this code. The syndromes of the $\mathcal{T}$-mapped row codes are shown in Fig. 4.1 where E_6 and E_{10} are unknown elements in the middle of the sequences.

The algorithm in [23] cannot recover E. However, it can be recovered by Algorithm 4.1. We show this by Example 4.6.

Example 4.6 *Continuing with Example 4.5, supposing the FGRS code is transmitted column by column and the received array is*

$$\begin{pmatrix} \alpha^3 & 0 & \alpha^{10} & \alpha^4 & 0 \\ 1 & 0 & 0 & \alpha^7 & 0 \\ \alpha & 0 & \alpha^{12} & \alpha^8 & 0 \end{pmatrix} \tag{4.20}$$

After GFFT of columns and mapping rows by $\mathcal{T}$, it becomes to

$$\begin{pmatrix} \alpha^7 & 0 & \alpha^3 & \alpha^{13} & 0 \\ 0 & 0 & \alpha^8 & \alpha^5 & 0 \\ \alpha^4 & 0 & \alpha^8 & 1 & 0 \end{pmatrix} \tag{4.21}$$

The syndromes for the row codes in (4.21) are

$$\begin{pmatrix} \alpha^{11} & \alpha^9 & E_6 & 0 & \alpha^{10} \\ \alpha^4 & 0 & \alpha^4 & E_{10} & \alpha^9 \\ \alpha^{10} & 0 & \alpha^{13} & \alpha & X \end{pmatrix} \tag{4.22}$$

where E_6 and E_{10} are unknown elements. We arrange these sequences and their left-shifts to form the array in (4.23) where X is a wildcard.

Start with $L = 1$, we have

$$\mathbf{U}_{L+1} = \begin{pmatrix} \alpha^{11} & \alpha^4 & \alpha^{10} & 0 & 0 & \alpha^{13} & 0 \\ \alpha^9 & 0 & 0 & \alpha^4 & \alpha^{13} & \alpha^4 & \alpha^{10} \end{pmatrix}^T.$$

By the FIA, $\mathbf{U}_{L+1}$ has full column rank, so increase L to 2. Now,

$$\mathbf{U}_{L+1} = \begin{pmatrix} \alpha^4 & 0 & \alpha^4 \\ \alpha^{10} & 0 & \alpha^{13} \\ 0 & \alpha^{13} & \alpha \end{pmatrix}.$$

$$\mathbf{A} = \begin{pmatrix} \alpha^{11} & \alpha^9 & E_6 & 0 & \alpha^{10} \\ \alpha^4 & 0 & \alpha^4 & E_{10} & \alpha^9 \\ \alpha^{10} & 0 & \alpha^{13} & \alpha & X \\ \alpha^9 & E_6 & 0 & \alpha^{10} & X \\ 0 & \alpha^4 & E_{10} & \alpha^9 & X \\ 0 & \alpha^{13} & \alpha & X & X \\ E_6 & 0 & \alpha^{10} & X & X \\ \alpha^4 & E_{10} & \alpha^9 & X & X \\ \alpha^{13} & \alpha & X & X & X \\ 0 & \alpha^{10} & X & X & X \\ E_{10} & \alpha^9 & X & X & X \\ \alpha & X & X & X & X \\ \alpha^{10} & X & X & X & X \\ \alpha^9 & X & X & X & X \end{pmatrix}, \tag{4.23}$$

It is also full column rank and we increase L to 3.

At this step,

$$\mathbf{U}_{L+1} = \begin{pmatrix} \alpha^{10} & 0 & \alpha^{13} & \alpha \end{pmatrix}.$$

The $\sigma(x)$ obtained from $\mathbf{U}_{L+1}$ by the FIA has degree 1, which is less than $L = 3$.

Therefore, we set $\sigma(x) = \sigma_3 x^3 + \sigma_2 x^2 + \sigma_1 x + 1$ *and derive* Ψ *as*

$$\begin{aligned} E_6\sigma_1 + \alpha^9\sigma_2 + \alpha^{11}\sigma_3 &= 0 \\ E_{10} + \alpha^4\sigma_1 + \alpha^4\sigma_3 &= 0 \\ \alpha + \alpha^{13}\sigma_1 + \alpha^{10}\sigma_3 &= 0 \\ \alpha^{10} + E_6\sigma_2 + \alpha^9\sigma_3 &= 0 \\ \alpha^9 + E_{10}\sigma_1 + \alpha^4\sigma_2 &= 0 \end{aligned} \tag{4.24}$$

Solving Ψ*, we have* $\sigma_3 = 1$*,* $\sigma_2 = \alpha^{10}$ *and* $\sigma_1 = \alpha^{10}$*. So* $\sigma(x) = x^3 + \alpha^{10}x^2 + \alpha^{10}x + 1$ *and the error array can be recovered as*

$$\begin{pmatrix} \alpha^7 & 0 & \alpha^3 & \alpha^{13} & 0 \\ 0 & 0 & \alpha^8 & \alpha^5 & 0 \\ \alpha^4 & 0 & \alpha^8 & 1 & 0 \end{pmatrix} \tag{4.25}$$

After applying $\mathcal{T}^{-1}$ *to the rows and inverse GFFT to the columns of array in (4.25), we have*

$$\begin{pmatrix} \alpha^3 & 0 & \alpha^{10} & \alpha^4 & 0 \\ 1 & 0 & 0 & \alpha^7 & 0 \\ \alpha & 0 & \alpha^{12} & \alpha^8 & 0 \end{pmatrix}. \tag{4.26}$$

Hence, the all-zero codeword is the transmitted FGRS codeword and 3 *burst errors are corrected.*

4.5 Conclusion

In this chapter, an algorithm for the synthesis of multisequences with unknown elements in the middle is proposed. This algorithm is applied to decode GRS codes with nonconsecutive syndromes, which could not be solved by sequence synthesis method before. We also show that folded codes can be constructed from GRS codes besides RS codes as in [96] and all the row codes in the resulting FGRS codes can be viewed as equivalent GRS codes with zeros from the same

support set. Therefore, the proposed algorithm can also be used to decode FGRS codes.

We note that the proposed algorithm may involve nonlinear equations. The proposed algorithm minimizes the degree of these nonlinear equations by putting the process of these unknown elements off. However, bounding the complexity of the proposed algorithm is difficult due to the random occurrence of the errors. Nevertheless, when the number of the unknown elements involved in the nonlinear equations is small, the complexity is reasonable.

Chapter 5

A Search-Based List Decoding Algorithm for RS codes

In this chapter, we propose a search-based list decoding algorithm which can correct up to $n-k-1$ errors with an (n, k) RS code. The performance, complexity and average list size of this search procedure are analyzed when the RS code is transmitted in an AWGN channel with BPSK signaling.

5.1 Introduction

Given an (n, k) RS code, the GSA can correct up to $\lceil n - \sqrt{n(k-1)} - 1 \rceil$ errors in polynomial time. This error-correction capability is far more than that of the classical decoding algorithms. This algorithm outputs a list of the most possible candidate messages. The average list size was shown very close to unity [56]. Motivated by these results of the GSA, we propose a search-based list decoding algorithm for RS codes. This algorithm can correct up to $n-k-1$ errors (in the list decoding sense). The idea of this decoding technique is from the fact that any solution[1] to the classical key equation leads to a possible solution for decoding

[1] An error locator polynomial and the corresponding error evaluator polynomial.

a received vector of a RS code. If and only at most $\lfloor \frac{n-k}{2} \rfloor$ errors occur, the minimal solution to the key equation is unique and coincides with the error locator polynomial. However, the proposed algorithm in this chapter does not stick to find the minimal solution and can correct beyond the classical error-correction bound.

Let $\mathbf{s} = (S_1, S_2, \ldots, S_{n-k}) \in \mathrm{GF}(q)^{n-k}$ be the syndrome sequence for a received vector of a (n, k) RS code. We denote by $\mathrm{Anni}(\mathbf{s})$, the set of polynomials $f(x) = \sum_{l=1}^{\deg(f(x))} f_l x^l + 1 \in \mathrm{GF}(q)[x]$, which annihilate $\mathbf{s}$ as

$$\sum_{j=0}^{\deg(f(x))-1} S_{i+j} f_{\deg(f(x))-j} + S_{i+\deg(f(x))} = 0, \forall 1 \geq i \geq n-k-\deg(f(x)). \quad (5.1)$$

A list of candidate error locator polynomials in $\mathrm{Anni}(\mathbf{s})$ are constructed by a search routine in the proposed algorithm. For each element in this list, the corresponding error values are then computed by Forney's method. The candidate error locator polynomials with degree up to $n-k-1$ are constructed in the proposed algorithm, the error-correction capability of this algorithm is therefore $n-k-1$ and larger than that of the GSA. We proceed to present this decoding approach in detail.

5.2 Search-Based List Decoding

5.2.1 The Search Tree

Let $\alpha \in \mathrm{GF}(q)$, $\mathrm{ord}(\alpha) = n$ and $\mathcal{C}$ an (n, k) RS code over $\mathrm{GF}(q)$ with zeros $\alpha, \alpha^2, \ldots, \alpha^{d-1}$, where $d = n-k+1$. Further, Let $\mathcal{Z} = \{1, \alpha, \ldots, \alpha^{n-1}\}$ which are all the zeros of $X^n - 1$ in $\mathrm{GF}(q)$. Denote the following set of polynomials by Ξ.

$$\Xi = \{f(x) | f(x) \in GF(q)[x]_{d-1}, f(x) | (x^n - 1)\}.$$

Thus, each polynomial from Ξ can be factorized as product of distinct linear factors over $\mathrm{GF}(q)$ and all the zeros of this polynomial are from $\mathcal{Z}$. The algorithm

we will present in this chapter is to find the polynomials in $\Xi \cap \mathrm{Anni}(\mathbf{s})$ for a given syndrome sequence. Such polynomials are called *candidate error locator polynomials*. Each of them is associated with an estimate $\hat{\mathbf{e}}$ of the error pattern. Let β primitive in $\mathrm{GF}(q)$ and denote the support of $\hat{\mathbf{e}}$ by $\mathrm{Supp}(\hat{\mathbf{e}})$. Then a candidate error locator polynomial can be written as

$$\sigma(x) = \prod_{\alpha_j \in \mathrm{Supp}(\hat{\mathbf{e}})} (1 - \alpha_j x) = \sum_{i=0}^{|\mathrm{Supp}(\hat{\mathbf{e}})|} \beta^{j_i} x^i,$$

where $\beta^{j_0} = 1$ by definition. As special cases, when the number of errors is not greater than $\lfloor (d-1)/2 \rfloor$, there is only one element in $\Xi \cap \mathrm{Anni}(\mathbf{s})$.

The elements of $\Xi \cap \mathrm{Anni}(\mathbf{s})$ can be found by searching a tree structure constructed in advance. This tree has $d-1$ levels which are labeled from 0 to $d-2$. Each node at the level v of the tree represents a particular polynomial in $\mathrm{GF}(q)[x]$ of degree v. Those nodes representing elements of Ξ are called *check nodes*[2]. Each node on the tree is labeled by a integer outside parenthesis. Such labels of the nodes along the path linking the root node (at 0th level) to a check node specify the coefficients of the element of Ξ represented by this check node. Specifically, the label of the node at the level i on this path is the exponent j_i of the coefficient β^{j_i} for the monomial x^i of this element. The path as mentioned above is later referred to by the concatenation of those integer labels. Each check node has another label in parenthesis. It is the exponents of the reciprocal of the zeros of the element of Ξ this check node represents.

Example 5.1 *Fig. 5.1 shows the structure of the tree just described for* $\mathcal{C}$*, a* $(7, 4)$ *RS code over* $\mathrm{GF}(8)$*. The path* 0-4-3 *corresponds to the element* $\sigma(x) = 1 + \alpha^4 x + \alpha^3 x^2 \in \Xi$*. Assume the all-zero codeword is transmitted and* $(0, \alpha^2, \alpha^4, 0, 0, 0, 0)$ *is received. The corresponding syndrome sequence is* $\mathbf{s} = (\alpha^4, \alpha^2, \alpha^2)$*. One checks that* $\sigma(x)$ *satisfies (5.1) so that* $\sigma(x) \in \Xi \cap \mathrm{Anni}(\mathbf{s})$ *and is a candidate error*

[2]Since $\Xi \subset \mathrm{GF}(q)[x]$, not all the nodes are elements of Ξ.

locator polynomial. From Fig. 5.1, the reciprocals of its zeros are α *and* α^2 *which indicates the locations of the errors introduced by the channel, as desired. One further checks that* $1 + \alpha^6 x + x^2$ *is also a candidate error locator polynomial corresponding to the path* 0-6-0*.*

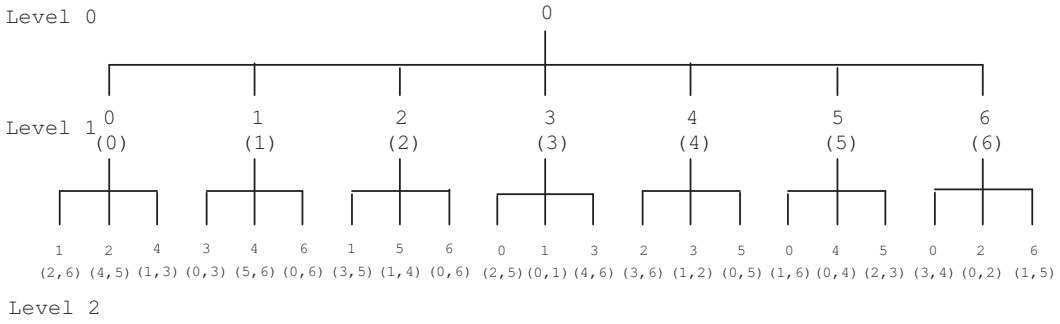

Figure 5.1: Tree structure for a $(7, 4)$ **RS code.**

To find all the candidate error locator polynomials of degree up to $d-2$, we need to check the linear constraints in (5.1) for each check node on the tree. There are $\binom{n}{i}$ check nodes at the level i of the tree. Thus, $\sum_{i=1}^{d-2} \binom{n}{i}$ check nodes need to be processed. Such a procedure will have exorbitantly large computational complexity. We thus propose several ways to reduce the complexity in next subsection.

5.2.2 Complexity Reduction Strategies

In this subsection, two ways are presented to reduce the complexity of the search procedure. The first one is due to Lemma 5.1 below.

Lemma 5.1 *Consider the code* $\mathcal{C}$ *and associated search tree as described in Section 5.2.1. If a check node at level* j $(0 < j < d-1)$ *corresponds to a candidate error locator polynomial, then there are no check nodes at level* i $(0 < i < d - j)$ *corresponding to other candidate error locator polynomials.*

Proof: Denote the candidate error locator polynomial represented by a check node at level j as $\sigma^{(1)}(x)$. Let $\mathbf{r}$ be decoded to a codeword $\mathbf{c}^{(1)}$ with $\sigma^{(1)}(x)$. Assume another candidate error locator polynomial represented by another check node at level i is found, where $0 < i < d - j$. Denote it as $\sigma^{(2)}(x)$ which decodes $\mathbf{r}$ to another codeword $\mathbf{c}^{(2)}$. Then

$$d(\mathbf{r}, \mathbf{c}^{(1)}) + d(\mathbf{r}, \mathbf{c}^{(2)}) \geq d(\mathbf{c}^{(1)}, \mathbf{c}^{(2)}) \geq d. \tag{5.2}$$

Since $\sigma^{(1)}(x)$ is a candidate error locator polynomial with degree j, $\mathbf{r}$ is decoded to $\mathbf{c}^{(1)}$ by correcting j symbols in $\mathbf{r}$ and $d(\mathbf{r}, \mathbf{c}^{(1)}) = j$. Similarly, we have $d(\mathbf{r}, \mathbf{c}^{(2)}) = i$. According to (5.2), we have $j + i \geq d(\mathbf{c}^{(1)}, \mathbf{c}^{(2)}) \geq d$. But from $0 < i < d - j$, we have $0 < i + j < d$. Thus, $\sigma^{(2)}(x)$ does not exist. There are no check nodes representing candidate error locator polynomials on level i for $0 < i < d - j$ in this case. ■

By Lemma 5.1, we can skip level i for $0 < i < d - j$, if we have found a candidate error locator polynomial at level j.

Example 5.2 *If $\mathcal{C}$ is a* $(63, 57)$ *RS code over* GF(64)*, the corresponding search tree has* 5 *levels. By Lemma* 5.1*, if a candidate error locator polynomial at level* 1 *is found, we can skip all other check nodes at level i for $0 < i < 6$, i.e., we can terminate the search immediately. In such a situation, we need search at most $\sum_{j=1}^{1} \binom{63}{j} = 63$, instead of $\sum_{j=1}^{5} \binom{63}{j} = 7666239$.*

Lemma 5.2 below shows another way to reduce the search complexity.

Lemma 5.2 *For the code $\mathcal{C}$ and associated search tree as described in Section 5.2.1, suppose we have two candidate error locator polynomials, one at level i, the other at level j such that $i + j \geq d$. Then, the cardinality $|z^{(i)} \cup z^{(j)}|$ of $z^{(i)} \cup z^{(j)}$ is at least d where $z^{(i)}$ and $z^{(j)}$ denote the supports of the respective error pattern.*

Proof: Assume two candidate error locator polynomials correct a received vector to two distinct codewords $\mathbf{c}^{(1)}$ and $\mathbf{c}^{(2)}$ respectively. Let $z^{(i)}$ and $z^{(j)}$ be the supports of the error patterns associated with these two error locator polynomials, respectively. Then $d(\mathbf{c}^{(1)}, \mathbf{c}^{(2)})$ is at most $|z^{(i)} \cup z^{(j)}|$. We have $|z^{(i)} \cup z^{(j)}| \geq d(\mathbf{c}^{(1)}, \mathbf{c}^{(2)}) \geq d$. ■

Assume a candidate error locator polynomial whose associated error pattern has support set $z^{(i)}$ is found at level i. Any check node representing an element of Ξ with reciprocals of zeros $z^{(j)}$ at level j can be skipped if $i + j \geq d$ and $|z^{(i)} \cup z^{(j)}| < d$. Let $\delta = i + j - d$ and suppose $i \leq j$. If $l = |z^{(i)} \cap z^{(j)}| > \delta$, then

$$|z^{(i)} \cup z^{(j)}| = i + j - l < i + j - \delta = d. \tag{5.3}$$

Thus,

$$\sum_{l=\delta+1}^{i} \binom{i}{l} \binom{n-i}{j-l} \tag{5.4}$$

check nodes can be skipped at level j, given a candidate error locator polynomial is found at level i.

Example 5.3 *Let $\mathcal{C}$ be a $(15, 10)$ RS code for which the tree structure has 5 levels. Suppose we have found a candidate error locator polynomial at level 2. Since the criterion to skip nodes implied by Lemma 5.1 is not satisfied, we still need to search through level 4 where there are $\binom{15}{4} = 1365$ check nodes. By (5.4), we can skip $\sum_{l=1}^{2} \binom{2}{l}\binom{15-2}{4-l} = 650$ of them. The number of check nodes to process is reduced by almost half at level 4.*

5.2.3 The Decoding Algorithm

Assume the search tree for the code $\mathcal{C}$ is available. We give an explicitly exposition of the proposed decoding method for $\mathcal{C}$ in Algorithm 5.1 below. The complexity reduction strategies proposed in previous section are incorporated in this algorithm.

Algorithm 5.1 ***Search-Based List Decoding***

- *Step 1: Compute the syndrome sequence* $\mathbf{s}$*, given the received vector* $\mathbf{r}$*.*
- *Step 2: Initialize* $l := 1$*,* $l_{min} := d-2$ *and* $\mathcal{L} := \{\}$*. (Note:* l_{min} *denotes the level where the first candidate error locator polynomial is found.)*
  - *(a) For every check node at level* l*, check the set of reciprocals of the zeros of the corresponding element of* Ξ *and skip that node according to Lemma* 5.2*, if possible.*
  - *(b) For each check node at the* l*th level which could not be skipped in Step 2(a), determine if the corresponding constraints, as specified by (5.1), are satisfied. For each check nodes where the corresponding constraints are satisfied, place the associated set of reciprocals of the zeros of the corresponding element of* Ξ*, in* $\mathcal{L}$*. If the first candidate error locator polynomial is found in the current level, then set* $l_{min} := l$*.*
  - *(c) Set* $l := \max(l+1, d-l_{min})$*.(Note: this assignment is due to Lemma* 5.1*.) If* $l < d-2$*, return to Step 2(a), else exit Step 2.*
- *Step 3: For each element of* $\mathcal{L}$*, recover the corresponding estimate* $\hat{\mathbf{e}}$ *of the error pattern induced by the channel and output* $\mathbf{r} - \hat{\mathbf{e}}$*.*

5.3 Decoding Shortened and Punctured RS Codes

In this section, we analyze decoding shortened and punctured RS codes by the proposed search-based list decoding algorithm.

- **Shortening**

  Assume ϕ is a subset of the support set of an (n, k) RS code and $|\phi| < k$. By shortening by ϕ, we mean the resulting code is a subset of the original

RS code such that the codewords of the resulting code have zero symbols at coordinate subset ϕ. Since RS codes are MDS codes, an (n, k) RS code is shortened to an $(n-t, k-t)$ MDS code, when $|\phi| = t < k$. The minimum distance of the $(n-t, k-t)$ code is $n-t-(k-t)+1 = n-k+1 = d$. The tree structure for the resulting code has the same number of levels as that for the original code. However, the number of elements in the support set of the $(n-t, k-t)$ code is reduced to $n-t$. The number of check nodes in the tree is also reduced to $\sum_{i=1}^{d-2} \binom{n-t}{i}$ from $\sum_{i=1}^{d-2} \binom{n}{i}$.

- **Puncturing**

  Assume ψ is a subset of the support set of an (n, k) RS code and $|\psi| = s < n-k$. By puncturing, we mean the resulting code has the same number of codewords as the original RS code and they are obtained by deleting the symbols at coordinate set ψ of the codewords in the original RS code. Hence, the resulting code is an $(n-s, k)$ MDS code. The minimum distance of the resulting $(n-s, k)$ code is $d = n-s-k+1$. The search tree for this code has $n-s-k-1$ levels. Since the length of the resulting code is reduced to $n-s$, the number of check nodes in the tree is also reduced to $\sum_{i=1}^{n-s-k-1} \binom{n-s}{i}$ from $\sum_{i=1}^{n-k-1} \binom{n}{i}$.

If s coordinates in a received vector are detected as erasures, the received vector can be decoded as if these s coordinates are punctured. The complexity will decrease since the number of check nodes to process reduces. This is especially important for the decoding of FRS or TFSRS codes transmitted in burst error channels as in Chapter 2.

Example 5.4 *In compact disk system,* $(32, 28)$ *RS code is applied [91]. It is obtained from shortening* $(255, 251)$ *RS code over* GF(256)*. Our algorithm can correct up to* 3 *symbol errors with this code. The search tree for the* $(32, 28)$

RS codes has 4 levels. To decode a received vector of this code, for the worst case, we need to search all the check nodes in the tree. The number of the check nodes is $\sum_{i=1}^{3}\binom{32}{i} = 5488$. *If an erasure position can be identified, in the worst case, we need to search the tree to level 2 and the number of the check nodes is* $\sum_{i=1}^{2}\binom{31}{i} = 496$, *which is less than* $\frac{1}{10}$ *of the previous one.*

5.4 Performance-Complexity-List-Size Analysis

5.4.1 Word-Error-Rate Performance

The proposed algorithm can correct more errors than the BMA and the GSA. To illustrate the advantageous error correcting capability of the proposed algorithm, it is enough to compare the performance of these algorithms in AWGN channels with simple BPSK signaling.

We compare the Word-Error-Rate (WER) of the proposed list decoding algorithm against that of the BMA as well as the GSA. We consider transmitting the codewords of a RS code in an AWGN channel by BPSK signaling. The equivalent BSC has crossover probability

$$p = Q(\sqrt{\frac{2RE_b}{N_0}}) \tag{5.5}$$

where E_b is the received bit energy of an information bit, R is the code rate, N_0 is the single-sided noise spectral density and $Q(\cdot)$ is the Q-function. For a 2^m-ary code $\mathcal{C}$, the code symbol error probability is computed as

$$p_s = 1 - (1-p)^m \tag{5.6}$$

and the probability of a received vector containing v errors is computed as

$$P_v = \binom{n}{v}(p_s)^v(1-p_s)^{n-v}. \tag{5.7}$$

The WERs of the proposed list decoding algorithm, BMA and GSA are then computed as $\sum_{v=t+1}^{n} P_v$ where t is equal to $n-k-1$, $\lfloor (n-k)/2 \rfloor$ and $\lceil n - \sqrt{n(k-1)} - 1 \rceil$, respectively. The WERs of these three decoders for two RS codes is shown in Fig. 5.2. We can see that at a WER of 10^{-6}, the proposed decoding algorithm offers an additional coding gain of about 1 dB over the GSA for both codes.

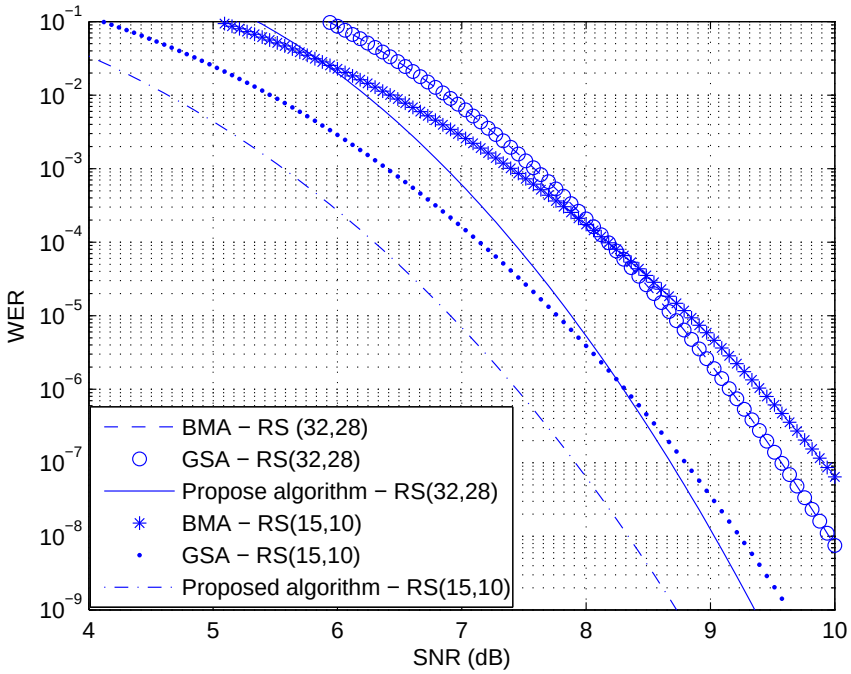

Figure 5.2: WERs of the BMA, the GSA and the proposed list decoding algorithm when applied to a $(32, 28)$ **RS code over** GF(256) **and a** $(15, 10)$ **RS code over** GF(16)**.**

5.4.2 Bounding The Average Complexity

The Step 2 in Algorithm 5.1 is the most computationally intensive step of this algorithm. In this subsection, we derive an upper bound for the average complexity of this step in terms of the field multiplication.

From the previous description, evaluating $n-k-v$ constraints are required to process a check node at level v on the tree. Since this check node represents a polynomial of degree v and with constant term 1, evaluating each constraint requires v multiplications. Thus, to processing a check node at level v, at most $M_v = (n-k-v)v$ multiplications are required.

There are $\binom{n}{v}$ check nodes in the vth level. According to the channel model considered in Section 5.4.1, error pattern with the same weight occur with the same probability. Thus, if a received vector has v errors, on average, half of the check nodes at level v need to be processed to find the desired error locator polynomial. If $v = 1$, searching half of the check nodes at level 1 will yield the desired error locator polynomial on average. Also, Algorithm 5.1 will exit Step 2 after that according to Lemma 5.1. Thus, the average complexity of Step 2 in this case can be upperbounded by

$$\kappa_1 = \frac{1}{2} P_1 \binom{n}{1} M_1 = \frac{n^2}{2} p_s (1-p_s)^{n-1} M_1.$$

If $2 \leq v \leq \lfloor \frac{n-k}{2} \rfloor$, we need to search through level 1 to level $v-1$ in the worst case. This requires an average number of multiplications at most $P_v \sum_{i=1}^{v-1} \binom{n}{i} M_i$. The desired error locator polynomial can be found at level v in this case. Since error patterns of the same weight occur with equal probability, the average number of multiplication required for finding the desired error locator polynomial at level v is at most $\frac{1}{2} P_v \binom{n}{v} M_v$. Since $v + v < d$, the remaining check nodes at level v can be skipped by Lemma 5.2. Further, due to Lemma 5.1, levels $\max(v+1, d-v) = d-v$ to $d-2$ need to be searched after searching level v. By (5.4), from level $d-v$ to $d-2$, only

$$\sum_{j=d-v}^{d-2} \left[\binom{n}{j} - \sum_{l=v+j-d+1}^{v} \binom{v}{l} \binom{n-v}{j-l} \right]$$

check nodes need to be processed. The average number of multiplications required

for searching these levels is upperbounded by

$$P_v \sum_{j=d-v}^{d-2} \left[\binom{n}{j} - \sum_{l=v+j-d+1}^{v} \binom{v}{l} \binom{n-v}{j-l} \right] M_j.$$

Thus, when $2 \leq v \leq \lfloor \frac{n-k}{2} \rfloor$, the average complexity of Step 2 can be upperbounded by

$$\kappa_2 = \sum_{v=2}^{\lfloor \frac{n-k}{2} \rfloor} P_v \left\{ \sum_{i=1}^{v-1} \binom{n}{i} M_i + \frac{1}{2} \binom{n}{v} M_v + \sum_{j=d-v}^{d-2} \left[\binom{n}{j} - \sum_{l=v+j-d+1}^{v} \binom{v}{l} \binom{n-v}{j-l} \right] M_j \right\}.$$

If $\lfloor \frac{n-k}{2} \rfloor + 1 \leq v \leq n-k-1$, the analysis of complexity is similar as the previous case except two differences. The first one is that since $v + v \geq d$ in this case, the remaining check nodes at level v need to be processed after we have found the desired error locator polynomial. By (5.4), among these remaining check nodes, only

$$\frac{1}{2} \left(\binom{n}{v} - \sum_{l=2v-d+1}^{v-1} \binom{v}{l} \binom{n-v}{v-l} \right)$$

need to be searched on average. The second is that after searching level v, levels $\max(v+1, d-v) = v+1$ to $d-2$ need to be searched. Thus, in this case, the average complexity of Step 2 can be upperbounded by

$$\begin{aligned}
\kappa_3 = & \sum_{v=\lfloor \frac{n-k}{2} \rfloor + 1}^{d-2} P_v \left\{ \sum_{i=1}^{v-1} \binom{n}{i} M_i + \frac{1}{2} \binom{n}{v} M_v + \frac{1}{2} \left[\binom{n}{v} - \sum_{l=2v-d+1}^{v-1} \binom{v}{l} \binom{n-v}{v-l} \right] M_v \right. \\
& \left. + \sum_{j=v+1}^{d-2} \left[\binom{n}{j} - \sum_{l=v+j-d+1}^{v} \binom{v}{l} \binom{n-v}{j-l} \right] M_j \right\} \\
= & \sum_{v=\lfloor \frac{n-k}{2} \rfloor + 1}^{d-2} P_v \left\{ \sum_{i=1}^{v} \binom{n}{i} M_i + \sum_{j=v+1}^{d-2} \left[\binom{n}{j} - \sum_{l=v+j-d+1}^{v} \binom{v}{l} \binom{n-v}{j-l} \right] M_j \right. \\
& \left. - \frac{1}{2} M_v \sum_{l=2v-d+1}^{v-1} \binom{v}{l} \binom{n-v}{v-l} \right\}. \qquad (5.8)
\end{aligned}$$

Finally, if $n-k \leq v \leq n$, all the $d-2$ levels need to be searched without skipping any nodes in the worst case. The average complexity of Step 2 in this

case can be upperbounded by

$$\kappa_4 = \sum_{v=d-1}^{n} P_v \sum_{j=1}^{d-2} \binom{n}{j} M_j.$$

The average complexity of Step 2 can therfore be upperbounded by

$$\kappa = \kappa_1 + \kappa_2 + \kappa_3 + \kappa_4.$$

The κ vs. SNR curves for the two RS codes considered in Section 5.4.1 are shown in Fig. 5.3. Combining Fig. 5.2 and Fig. 5.3, we see that at a WER of 10^{-6},

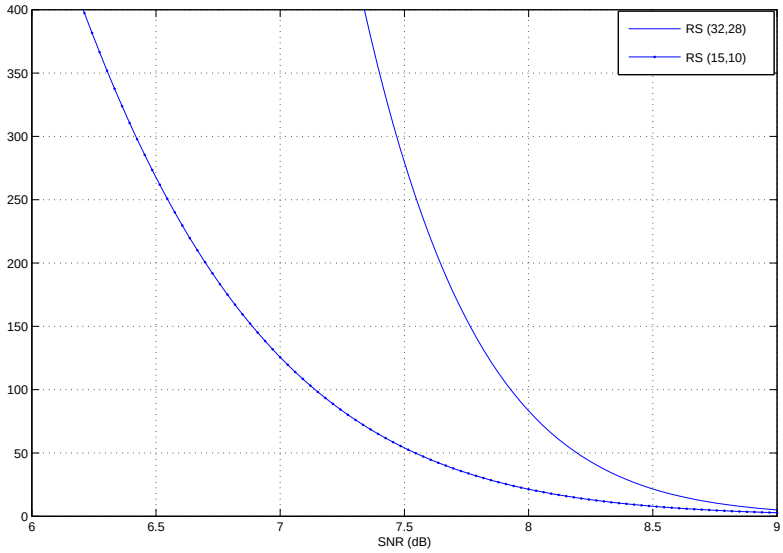

Figure 5.3: Complexity of Step 2 for decoding a $(32, 28)$ RS code (shortened from a $(255, 251)$ RS code) over $\mathrm{GF}(256)$ and a $(15, 10)$ RS code over $\mathrm{GF}(16)$.

the average number of multiplications incurred by Step 2 is less than n^2.

5.4.3 The Average List Size

Denote the Hamming sphere of radius $d-2$ centers at an vector $\mathbf{v} \in \mathrm{GF}(q)^n$ as $\mathcal{S}_{\mathbf{v}}(d-2)$. We estimate the average number of codewords contain in $\mathcal{S}_{\mathbf{v}}(d-2)$ in

this subsection. This quantity coincides with the average list size of the proposed list decoder. Since $\mathcal{C}$ is linear, we may assume that the *causal* codeword (i.e. the transmitted codeword) is the all-zero codeword. The corresponding received vector is denoted by $\mathbf{r}$.

When v, the weight of $\mathbf{r}$, is equal to unity, only one codeword is in $\mathcal{S}_{\mathbf{v}}(d-2)$ by Lemma 5.1. Clearly, this codeword is the causal one.

When $v \geq 2$, it has been shown in [55] that the average number of *noncausal* codewords in $\mathcal{S}_{\mathbf{r}}(d-2)$ can be closely estimated by

$$\overline{L}(d-2) = \frac{1}{q^{n-k}} \sum_{i=0}^{d-2} \binom{n}{i} (q-1)^i. \tag{5.9}$$

An estimate of the average number of codewords (causal and noncausal) in $\mathcal{S}_{\mathbf{r}}(d-2)$ is thus $\overline{L}(d-2)+1$ and $\overline{L}(d-2)$ for $d-2 \geq v \geq 2$ and $n \geq v \geq d-1$, respectively.

Thus, the average list size $A(n,k,q)$ of the proposed decoding algorithm when applied to decode $\mathcal{C}$ can be closely estimated by

$$\begin{aligned} A(n,k,q) &= (P(0)+P(1)) \times 1 + \sum_{v=2}^{d-2} P(v)(\overline{L}(d-2)+1) + \sum_{v=d-1}^{n} P(v)\overline{L}(d-2) \\ &= \sum_{v=0}^{d-2} P(v) + \sum_{v=2}^{n} P(v)\overline{L}(d-2) \end{aligned} \tag{5.10}$$

Fig. 5.4 shows the curves of $A(n,k,q)$ vs. SNR for the proposed decoding algorithm applied to decode a $(32,28)$ RS code over GF(256) and a $(15,10)$ RS code over GF(16) . From Fig. 5.2 and Fig. 5.4, we can see that at a WER of 10^{-6}, the estimated average list size is less than 3 for the latter code and less than 2 for the former code. For the purpose of comparison, Fig. 5.4 also shows the curves of the average list size of the GSA for these codes[3]. We can see that the average list size for both decoding algorithms are comparable at WERs of practical interest, for these two codes.

[3]Using additional results from [55]

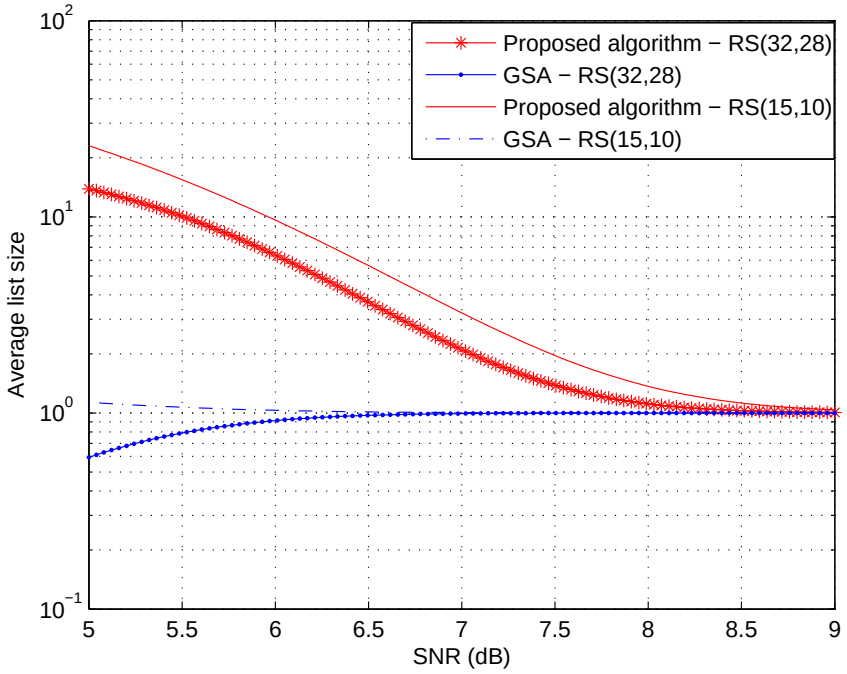

Figure 5.4: Average list size for a $(32, 28)$ RS code over $\mathrm{GF}(256)$ and a $(15, 10)$ RS code over $\mathrm{GF}(16)$ under the proposed decoding algorithm and the GSA. (Note that the estimated average list size of the GSA for the former code is less than 1 when SNR is less than 7 dB, due to the highly non-perfect nature of the code.)

5.5 Conclusion

Given an (n, k) RS code, we present a search-based list decoding algorithm capable of correcting up to $n - k - 1$ errors in this chapter. Its error-correction capability exceeds that of the GSA for a wide range of code parameters, although with larger complexity. we show that for short, high rate RS codes, it is possible that the average complexity of the proposed search procedure is less than n^2 at WERs of practical interests. This decoding algorithm can be applied to FRS and TSFRS codes presented in [96]. Choosing the dimension of the code array properly, we thus can apply the proposed algorithm with reasonable complexity

at practical WERs. In addition, the proposed algorithm is also applicable to some GRS codes and its subfield subcodes with consecutive syndrome sequence from the derivation.

The complexity of this algorithm is related with the number of nodes to be processed on the tree. For a low rate code, there are more levels in the corresponding tree structure compared with that of a high rate code with the same length. Hence, the number of nodes to be processed may increase and the complexity may become prohibitive. For a code with long length, there may be more nodes on each level to be processed in the corresponding tree structure compared with that of a code with short length and the same minimum distance. The overall number of nodes to be processed may increase and the complexity may also becomes undesirable. Nevertheless, the row codes in a FRS code are often short codes with high rate for practical interest. They can be decoded by the proposed algorithm with reasonable complexity.

Chapter 6

Decoding RS Codes with Gröbner Bases Method and Its Applications

In this chapter, we present a list-type decoding algorithm for high rate RS codes and its applications. This algorithm is based on Generalized Newton's Identities (GNI) and Gröbner Basis (GB) induced by the orthogonal relation defined in Section 6.2. Decoding IRS codes and some cyclic codes over GF(8) are also studied.

6.1 Introduction

RS codes are important linear block codes with abundant algebraic structures. The classical decoding algorithms of RS codes can decode an (n, k) RS code with up to $\lfloor \frac{n-k}{2} \rfloor$ errors and output a single codeword if the decoding is successful. These classical decoding algorithms are related with the GNI and the number of independent linear equations derived from the GNI limits their error-correction capability. On the other hand, the GSA can decode an (n, k) RS code with up

to $\lfloor n - \sqrt{n(k-1)} \rfloor$ errors and produce a short list of most possible messages. It is also shown that the average list size is very close to unity [56], which means the output list is a singleton in almost all the time. When a code array of an IRS or FRS code is transmitted column by column over a burst error channel, all the row codes share the same error pattern. Then, far more errors can probabilistically be corrected than the classical bound via the collaborations of the row codes as in [8, 44, 9, 76]. The decoding algorithms in [44, 76] are based on the equations from the GNI. The decoding algorithm in [8, 9] is based on the simultaneous polynomial reconstruction.

The GB method is applied to solve the key equation of cyclic codes in [25]. It is also applied to decode BCH codes with up to $\lfloor \frac{d_t-1}{2} \rfloor$, where d_t is the true minimum distance of the BCH codes, in [12] and decode cyclic codes in [11]. This method is further studied in [47, 24, 62]. In these papers, polynomial ideals are constructed according to the GNI and the equation $x^{q-1} + 1 = 0$, where x is a nonzero element in GF(q). The variables in these ideals are the error locations and the corresponding error values. To decode a received vector, we need find the GB of these ideals. However, since finding the GB of an ideal is hard, the complexity is prohibiting.

In this chapter, we propose a decoding algorithm based on the equations from the GB and the GNI. Instead of solving for GB for each received vector, we compute the GB in advance. The variables in this ideal are the elementary symmetric functions of the error locations.

In the following part, we review the GNI and the relation $x^{\deg(\sigma(x))}\sigma(x^{-1})h(x) = x^n - 1$ (This relation will be explained later.). A polynomial ideal is constructed. Then a decoding strategy for correcting more errors than the classical decoding bound is proposed. We show that an $(n, n-3)$ RS code can correct up to 2 errors by this strategy with

complexity $\mathcal{O}(n^2)$ and an $(n, n-4)$ RS code can correct up to 3 errors by this strategy with complexity $\mathcal{O}(n^3)$. In addition, a decoding algorithm to correct up to decoding $\lfloor \frac{(d_{min}-1)}{2} \rfloor + 1$ errors with an (n, k) RS codes is proposed. The result can be applied for decoding IRS codes to increase the probability of successful decoding.

Combining the above method with the decomposition of the syndromes S_1, S_2, S_4, new decoding algorithms for $(7, 3)$ RS code and $(7, 4)$ cyclic codes over GF(8) correcting up to 3 errors are presented. If the error values are restricted, decoding some cyclic codes over GF(8) is studied.

6.2 The GNI and the Relation $x^{\deg(\sigma(x))}\sigma(x^{-1})h(x) = x^n - 1$

Without loss of generality, assume that $\mathcal{C}$ is an (n, k) RS code over GF(q) with consecutive zeros $\alpha, \alpha^2, \ldots, \alpha^{(n-k)}$, where α is the generator of a cyclic subgroup $\mathcal{G}$ in GF(q) and ord(α) $= n$. Let $[n] = \{0, 1, \ldots, n-1\}$. Assume a codeword $\mathbf{c} \in \mathcal{C}$ is transmitted over a noisy channel and the received vector is $\mathbf{r} = \mathbf{c} + \mathbf{e}$, where $\mathbf{e} = (e_0, e_1, \ldots, e_{n-1})$ is the error vector. Let $\bar{\mathbf{I}} = \{i | i \in [n], e_i \neq 0\}$, $|\bar{\mathbf{I}}| = t$ and $\mathbf{I} = [n] \backslash \bar{\mathbf{I}}$. The error locator polynomial for $\mathbf{r}$ can be defined as $\sigma(x) = x^{-t} \prod_{i \in \bar{\mathbf{I}}} (x - \alpha^i) = \sum_{i=0}^{t} \sigma_i x^i$ where $\sigma_0 = 1$. The zeros of the monic polynomial $\phi(x) = x^t \sigma(x^{-1}) = \sum_{i=0}^{t} \sigma_{t-i} x^i$ are α^i where $i \in \bar{\mathbf{I}}$. Then σ_j, the $(t-j)$th coefficient of $\phi(x)$, is the jth elementary symmetric function [86] of α^i for $1 \leq j \leq t$ and $i \in \bar{\mathbf{I}}$. The coefficients of $\phi(x)$ and $\sigma(x)$ are in reverse order.

In the classical decoding algorithms and the algorithm in [25], the error locator polynomial need be found first. When the number of errors $t \leq \lfloor \frac{d_{min}-1}{2} \rfloor$, a unique solution for error locator polynomial can be determined from the syndrome

sequence $S_1, S_2, \ldots, S_{n-k}$ via the GNI

$$S_j = \sum_{i=1}^{t} \sigma_i S_{j-i}, \tag{6.1}$$

where $n - k \geq j \geq t$. Since $\sigma_0 = 1$, we have

$$\sum_{i=0}^{t} \sigma_i S_{j-i} = 0, \ n - k \geq j \geq t. \tag{6.2}$$

When $t \leq \lfloor \frac{n-k}{2} \rfloor$, a unique solution for $\sigma(x)$ such that $\deg(\sigma(x)) = t$ can be solved from (6.2). When $n - k - 1 \geq t > \lfloor \frac{n-k}{2} \rfloor$, there are t unknowns and $n - k - t$ equations in (6.2). Since $n - k - t < t$, there are q^{2t+k-n} possible solutions for $\sigma(x)$ from (6.2). However, not all of them are valid solutions of $\sigma(x)$ and checking them one by one is not effective. By valid solution of $\sigma(x)$, we mean the solutions for $\sigma(x)$ that can be factorized into distinct linear factors over $\mathcal{G}$ and satisfies (6.2). Moreover, since we do not know the number of errors t in advance, any valid solutions for $\sigma(x)$ from (6.2) can be the true one if there are more than one.

To find the valid solutions for $\sigma(x)$ from the syndromes for $\mathbf{r}$ when

$$n - k - 1 \geq t > \lfloor \frac{n-k}{2} \rfloor,$$

the relation follows may be useful. Since $\mathrm{ord}(\alpha) = n$, there is a polynomial $h(x) = \prod_{i \in \mathbf{I}}(x - \alpha^i) = \sum_{i=0}^{n-t} h_{n-t-i} x^i$, where $h_0 = 1$ and $|\mathbf{I}| = n - t$, such that $h(x)\phi(x) = 0 \pmod{x^n - 1}$. We refer this relation to orthogonal relation later. Denote the coefficient of x^i in $x^n - 1$ as $(x^n - 1)_i$. Let

$$\lambda(s) = \{i | \max(0, s - t) \leq i \leq \min(s, n - t)\},$$

$$\theta(s) = \{j | \max(0, s - (n - t)) \leq j \leq \min(s, t)\}.$$

From the orthogonal relation, we have the following n equations.

$$\begin{aligned}
(x^n-1)_{n-1} = \sum_{\substack{i+j=n-1,\\ i\in\lambda(n-1),\\ j\in\theta(n-1)}} h_{n-t-i}\sigma_{t-j} &= 0\\
(x^n-1)_{n-2} = \sum_{\substack{i+j=n-2,\\ i\in\lambda(n-2),\\ j\in\theta(n-2)}} h_{n-t-i}\sigma_{t-j} &= 0\\
&\vdots\\
(x^n-1)_1 = \sum_{\substack{i+j=1,\\ i\in\lambda(1),\\ j\in\theta(1)}} h_{n-t-i}\sigma_{t-j} &= 0\\
(x^n-1)_0 - 1 = h_{n-t}\sigma_t - 1 &= 0. \qquad (6.3)
\end{aligned}$$

If $\boldsymbol{\sigma} = (\sigma_t, \sigma_{t-1}, \ldots, \sigma_1, 1)$ satisfies the equations in (6.3), the polynomial $\sigma(x)$ can be factorized into distinct linear factors over $\mathcal{G}$. If a solution for $\sigma(x)$ also satisfies the equations in (6.2), it will be a valid solution for the error locator polynomial.

Since

$$(x^n-1)_s = \sum_{\substack{i+j=s,\\ s-t\,\le\, i\,\le\, \min(s,n-t)\\ \max(0,s-(n-t))\,\le\, j\,\le\, t,}} h_{n-t-i}\sigma_{t-j} = 0$$

for $t \le s \le n-1$ and $\sigma_0 = 1$,

$$h_{n-t-(s-t)} = h_{n-s} = \sum_{\substack{i+j=s,\\ s-t\,<\, i\,\le\, \min(s,n-t)\\ \max(0,s-(n-t))\,\le\, j\,<\, t,}} -h_{n-t-i}\sigma_{t-j}. \qquad (6.4)$$

Let $i' = s - t + i$. Then $n - t - i = n - s - i'$, $j = s - i = t - i'$ and $i' \in \boldsymbol{\eta}$, where $\boldsymbol{\eta} = \{i' | 0 < i' \le \min(t - \max(0, s-(n-t)), \min(s, n-t) - (s-t)) = \min(t, n-s)\}$. When $s = n-1$, (6.4) simplifies to

$$h_1 = \sigma_0 h_1 = -h_0\sigma_1 = -\sigma_1, \qquad (6.5a)$$

Similarly, for $t \leq s < n-1$,

$$h_{n-s} = \sum_{\substack{i+j=s, \\ s-t \,<\, i \,\leq\, \min(s,n-t) \\ \max(0,s-(n-t)) \,\leq\, j \,<\, t,}} -h_{n-t-i}\sigma_{t-j} = \sum_{i' \in \boldsymbol{\eta}} -h_{n-s-i'}\sigma_{i'}. \tag{6.5b}$$

So each h_{n-s} is a linear function of $h_{n-t-i'}$ for $i' \in \boldsymbol{\eta}$. Since $h_0 = 1$ and $h_1 = -\sigma_1$, each of the h_i's, $1 \leq i \leq n-t$, can be expressed as a multivariate polynomial on σ_j for $1 \leq j \leq t$. Substituting (6.5) into the left hand side of the last t equations in (6.3) yields t non-linear constraints on $\sigma_t, \sigma_{t-1}, \ldots, \sigma_1$, denoted as $\mathcal{F}$. Each monomial in these non-linear constraints has binary coefficient.

The equations in $\mathcal{F}$ are a group of multivariate polynomial equations. There are t variables (i.e. unknowns) $\sigma_t, \sigma_{t-1}, \ldots, \sigma_1$ and t equations in $\mathcal{F}$. Since the t zeros of any valid $\sigma(x)$ are distinct elements in $\mathcal{G}$, there are $\binom{n}{t}$ solutions for the unknowns. The left hand side of the equations in $\mathcal{F}$ form a zero dimensional multivariate polynomial system. This system $\mathcal{F}$ can be solved by reducing it into a triangular form by a GB approach, where t reduced basis elements can be obtained assuming the ordering $\sigma_1 > \sigma_2 > \ldots > \sigma_t$. We express $\mathcal{F}$ in the triangular form with t reduced basis elements as the LHS of the following equations.

$$\begin{aligned} f_{t-1}(\sigma_t, \sigma_{t-1}, \ldots, \sigma_1) &= 0 \\ f_{t-2}(\sigma_t, \sigma_{t-1}, \ldots, \sigma_2) &= 0 \\ &\vdots \\ f_1(\sigma_t, \sigma_{t-1}) &= 0 \\ f_0(\sigma_t) &= 0. \end{aligned} \tag{6.6}$$

The computation complexity of the base elements is exponential. But they only need to be computed once in advance. The monomials in the elements also have binary coefficients.

Example 6.1 *Let* $q = 8$, $n = 7$, $\deg(\sigma(x)) = 3$ *and* $\deg(h(x)) = 4$. *Assume* α *is the primitive element in* GF(8). *Here* $\mathcal{G} = \{1, \alpha, \alpha^2, \ldots, \alpha^6\}$ *and* $x^3\sigma(x^{-1})h(x) = x^7 - 1$. *From (6.3)*

$$\begin{aligned}
\sigma_1 + h_1 &= 0 \\
\sigma_2 + \sigma_1 h_1 + h_2 &= 0 \\
\sigma_3 + \sigma_2 h_1 + \sigma_1 h_2 + h_3 &= 0 \\
\sigma_3 h_1 + \sigma_2 h_2 + \sigma_1 h_3 + h_4 &= 0 \\
\sigma_3 h_2 + \sigma_2 h_3 + \sigma_1 h_4 &= 0 \\
\sigma_3 h_3 + \sigma_2 h_4 &= 0 \\
\sigma_3 h_4 + 1 &= 0. \qquad (6.7)
\end{aligned}$$

We only want to solve $\sigma(x)$. *So we eliminate the coefficients of* $h(x)$ *from (6.7) using (6.5) and obtained the following equations.*

$$\begin{aligned}
\sigma_1^5 + \sigma_1\sigma_2^2 + \sigma_1^2\sigma_3 &= 0 \\
\sigma_1^4\sigma_2 + \sigma_1^3\sigma_3 + \sigma_1^2\sigma_2^2 + \sigma_2^3 + \sigma_3^2 &= 0 \\
\sigma_1^4\sigma_3 + \sigma_1^2\sigma_2\sigma_3 + \sigma_2^2\sigma_3 + 1 &= 0. \qquad (6.8)
\end{aligned}$$

The following equations are from the reduced GB for the LHS of (6.8)[1].

$$\begin{aligned}
f_2(\sigma_3, \sigma_2, \sigma_1) &= \sigma_1 + \sigma_2^3\sigma_3^3 + \sigma_3^5 = 0 \\
f_1(\sigma_3, \sigma_2) &= \sigma_2^5 + \sigma_2^4\sigma_3^3 + \sigma_2^3\sigma_3^6 + \sigma_2\sigma_3^5 = 0 \\
f_0(\sigma_3) &= \sigma_3^7 + 1 = 0 \qquad (6.9)
\end{aligned}$$

6.3 Decoding $(n, n-3)$ and $(n, n-4)$ RS Codes

Although it is shown in [34] that decoding a restricted class of RS codes up to $n - k - 1$ is NP-hard, correcting high rate RS codes up to $n - k - 1$ errors is still feasible. High rate codes are more interested than low rate RS codes in data

[1] computed by MAPLE.

storage and wireless communication systems. The equations in the triangular form of $\mathcal{F}$ can be used to decode an (n, k) RS codes with up to $n-k-1$ errors.

In this section, we develop hard decision decoding algorithms for $(n, n-3)$ and $(n, n-4)$ RS codes by using the equations in the triangular form of $\mathcal{F}$ and equations in the GNI. We show these algorithms are advantageous in decoding complexity or error-correction capability when compared with the classical decoding algorithms and the GSA. These decoding algorithms can be generalized to decode an (n, k) RS code with up to $\lfloor \frac{n-k}{2} \rfloor + 1$ errors.

6.3.1 Outline of the Decoding Algorithm and List Size

In the description of the outline of the decoding algorithms for $(n, n-3)$ and $(n, n-4)$ RS codes, Lemma 5.2 can serve as a criteria to terminate the decoding process after Step 1 as well as to rule out some plausible error locator polynomials obtained in Step 2 for $(n, n-4)$ RS codes. This is because Lemma 5.2 implies two things. First, there is no codeword $\mathbf{c}$ such that $d(\mathbf{c}, \mathbf{r}) < n-k+1-l'$, if a valid error locator polynomial of degree l' is identified. Second, if an error locator polynomial $\sigma(x)$ with associated error location set W_1 is valid, another error locator polynomial $\sigma'(x)$ with associated error location set W_2 and $|W_1| < |W_2|$ should satisfy $|W_1 \cup W_2| \geq n-k+1$. Otherwise, $\sigma'(x)$ is a plausible error locator polynomial.

Both decoding algorithms consist of three steps. The classical decoding algorithm is employed to find the possible error locator polynomial with $t \leq \lfloor \frac{n-k}{2} \rfloor$ errors in Step 1, if there is. In Step 2, since $n-k-1 = \lfloor \frac{n-k}{2} \rfloor + 1$ for $(n, n-3)$ and $(n, n-4)$ RS codes, we look for error locator polynomials for $\lfloor \frac{n-k}{2} \rfloor + 1$ errors. The algorithm terminates if an error locator polynomial with degree 1 is found in Step 1 according to Lemma 5.2 because $n-k-1 < (n-k+1)-1$.

Otherwise we make use of the equations in the triangular form of $\mathcal{F}$ to solve for the possible error locator polynomials for $n-k-1$ errors. For $(n, n-4)$ RS codes, if an error locator polynomial $\sigma(x)$ with degree 2 and error location set W_1 is found in Step 1, error locator polynomials with error location set W_2 such that $|W_1 + W_2| < n-k+1$ may be found in Step 2. These error locator polynomials are plausible according to Lemma 5.2 and thus should be ruled out. In Step 3, the error values are computed by Forney's algorithm for each valid error locator polynomial obtained in the previous steps and subtracted from $\mathbf{r}$.

Since the decoding radius is larger than $\lfloor \frac{n-k}{2} \rfloor$, there may be more than one valid error locator polynomials. Hence, the decoding algorithms are list-type. Since the list size varies with different syndrome sequences, it is difficult to calculate the list size for all cases. However, a method to estimate the average list size is proposed in [56] for a bounded distance decoder. For an (n, k) RS code over GF(q), the average number of noncausal codewords in a Hamming sphere with radius $t \leq n-k-1$ can be estimated as

$$\bar{L}(t) = \frac{1}{q^{n-k}} \sum_{i=0}^{t} \binom{n}{i} (q-1)^i.$$

It is shown that this estimation is quite accurate in [56]. So, assume $\bar{t}$ is the number of errors occur in $\mathbf{r}$, the average list size in our case can be estimated as $\bar{L}(t) + p(\bar{t} \leq t)$.

6.3.2 Decoding $(n, n-3)$ RS Codes with up to 2 Errors

For an $(n, n-3)$ RS code, the classical decoding algorithm can correct one error only. The GSA can decoding up to 2 errors. The complexity of the GSA with Koetter's interpolation algorithm is $\mathcal{O}(r^4 n^2)$, where r is the multiplicity for each interpolation point in the GSA [55].

For an $(n, n-3)$ RS codes, up to $n-k-1 = 2$ errors can be corrected by the

algorithm outlined in subsection 6.3.1. We give the details for finding the valid error locator polynomials with degree 2 as follows. Let $\sigma(x) = \sigma_2 x^2 + \sigma_1 x + 1$, the underlying field have cardinality q and $L = \frac{q-1}{|\mathcal{G}|} = \frac{q-1}{n}$. If there is a solution for $\sigma(x)$ with error location set $W_1 = \{w_1, w_2\}$, $\sigma_2 = \alpha^{w_1} \times \alpha^{w_2} \in \mathcal{G}$. Let the syndromes be S_1, S_2, S_3. From (6.2), $S_1\sigma_2 + S_2\sigma_1 + S_3 = 0$. We consider the following four cases.

If $S_2 = 0$ and $S_1 = 0$, it cannot has two errors and decoding fails. Since $e(x)|_{x=\alpha} = e(\alpha) = S_1 = 0$ and $e(x)|_{x=\alpha^2} = e(\alpha^2) = S_2 = 0$, $\mathbf{e}$ is a codeword of a RS codes with zeros α and α^2 in this case. Assume i errors occur in $\mathbf{r}$ with probability p_i. The probability of decoding failure in this case is $\sum_{i=3}^{n} A_i p_i$ where A_i are the number of i-weight codewords in an $(n, n-2)$ RS codes, which is available from the weight distribution of RS codes [91].

If $S_2 = 0$ and $S_1 \neq 0$, $\sigma_2 = -\frac{S_3}{S_1}$. Since monomials in $f_1(\sigma_2, \sigma_1)$ have unit coefficients and $\sigma_2^n = 1$, $f_1(\sigma_2, \sigma_1) = \sum_{i=0}^{q} \sum_{j=0}^{n} a_{i,j} \sigma_2^j \sigma_1^i = 0$ where $a_{i,j} \in \mathrm{GF}(2)$. Substituting $\sigma_2 = -\frac{S_3}{S_1}$ into $\sum_{j=0}^{n} a_{i,j} \sigma_2^j$ requires at most n multiplications for each i. Solving $f_1(\sigma_2, \sigma_1) = 0$ requires at most q^2 multiplications since $\sigma_1 \in \mathrm{GF}(q)$. The over all complexity in number of multiplications is at most $\mathcal{O}(qn+q^2)$ which is $\mathcal{O}((L^2 + L)n^2)$.

If $S_2 \neq 0$ and $S_1 \neq 0$, $\sigma_1 = -\frac{S_1}{S_2}(\sigma_2 + \frac{S_3}{S_1})$. Computing all the σ_1^j for $1 \leq j \leq q$ at most requires $q(q-1)/2 + q - 1$ multiplications. Substitute σ_1^j into $f_1(\sigma_2, \sigma_1) = \sum_{j=0}^{n} \sum_{i=0}^{q} a_{i,j} \sigma_1^i \sigma_2^j = 0$ and solve for σ_2 from the resultant equation. This requires at most n^2 multiplications. Back substituting solutions for σ_2 to $\sigma_1 = -\frac{S_1}{S_2}(\sigma_2 + \frac{S_3}{S_1})$, we have the corresponding σ_1. The over all complexity is at most $\mathcal{O}((\frac{L^2}{2} + 1)n^2)$.

If $S_2 \neq 0$ and $S_1 = 0$, $\sigma_1 = -\frac{S_3}{S_2}$. Computing $\sum_{i=0}^{q} a_{i,j} \sigma_1^i$ requires at most q multiplications for each $0 \leq j \leq n$. Solving for σ_2 from the resultant $f_1(\sigma_2, \sigma_1) = 0$ needs at most n^2 multiplications. The overall complexity is at most $\mathcal{O}((L+1)n^2)$

in this case.

The decoding complexity is upper bounded by $\mathcal{O}((L+L^2)n^2)$. Here, we assume $f_1(\sigma_2, \sigma_1) = 0$ dense which means all the monomials $\sigma_1^i \sigma_2^j$ for $0 \leq i \leq q-2$ and $0 \leq j \leq n-1$ have unit coefficients. The real computation complexity may be far less than this bound, because $f_1(\sigma_2, \sigma_1)$ may not be dense as shown in Example 6.2. At least, this algorithm achieves the same error-correction capability and less complexity for $(n, n-3)$ RS codes compared with the GSA when $L + L^2 < r^4$.

Example 6.2 *Let a (7,4) RS codes over* GF(8) *have zeros* $\alpha, \alpha^2, \alpha^3$*, where* α *is the primitive element in* GF(8)*. Assume a received vector* $\mathbf{r} = (0, 1, 0, 0, 0, \alpha^4, 0)$*. The syndrome sequence of this received vector is* $(\alpha^5, \alpha^4, \alpha^6)$*. First let* $t \leq 1$*. The BMA cannot find any solution for error locator polynomial. Then let* $t = 2$ *and* $\sigma(x) = \sigma_2 x^2 + \sigma_1 x + 1$*. By (6.2),* $\alpha^6 + \alpha^4 \sigma_1 + \alpha^5 \sigma_2 = 0$ *and rearranging, we have*

$$\sigma_1 = \alpha \sigma_2 + \alpha^2. \tag{6.10}$$

The triangular form basis for the orthogonal relation are two polynomial equations.

$$f_1(\sigma_2, \sigma_1) = \sigma_1^3 + \sigma_1^2 \sigma_2^4 + \sigma_2^5 = 0 \tag{6.11}$$

$$f_0(\sigma_2) = \sigma_2^7 + 1 = 0. \tag{6.12}$$

Since the solutions for (6.12) are all the nonzero elements in GF(8)*, we only need to solve (6.11). Substitute (6.10) into (6.11),*

$$\alpha^2 \sigma_2^6 + \sigma_2^5 + \alpha^4 \sigma_2^4 + \alpha^3 \sigma_2^3 + \alpha^4 \sigma_2^2 + \alpha^5 \sigma_2 + \alpha^6 = 0.$$

The roots of this univariate equation in the indeterminate σ_2 *are* α^3 *and* α^6*. The corresponding solution* σ_1 *are* α *and* α^6 *respectively. So there are two solutions for the error locator polynomial,*

$$\sigma(x) = \alpha^3 x^2 + \alpha x + 1$$

and

$$\sigma(x) = \alpha^6 x^2 + \alpha x + 1.$$

The error locations are $(0, 3)$ *and* $(1, 5)$*, respectively. The corresponding error values are* (α, α^6) *and* $(1, \alpha^4)$*. Hence this received vector is decoded as two candidate codewords* $\mathbf{c}_1 = (\alpha, 1, 0, \alpha^6, 0, \alpha^4, 0)$ *and* $\mathbf{c}_2 = (0, 0, 0, 0, 0, 0, 0)$.

The method can be applied to decode (n, k) RS codes with up to $t = \lfloor \frac{n-k}{2} \rfloor + 1 = \frac{n-k+1}{2}$ errors if $n-k$ is odd. From (6.2), there are $n-k-\frac{n-k+1}{2} = \frac{n-k-1}{2}$ linear equations for σ_i, $1 \leq i \leq t$. All the σ_i can be expressed as linear functions of σ_t. One of these equations only involves σ_t, σ_{t-1}. Combining this linear equation with $f_1(\sigma_t, \sigma_{t-1}) = 0$, we can find the solutions for (σ_t, σ_{t-1}). Back substituting each solution for (σ_t, σ_{t-1}) into the linear equations gives the corresponding candidate $\sigma(x)$. The validity of these $\sigma(x)$ can be checked by factorizing each resulting $\sigma(x)$, which requires at most $qtn = Ltn^2$ multiplications. The complexity is thus $\mathcal{O}((L^2 + L + Lt)n^2)$.

6.3.3 Decoding $(n, n-4)$ RS Codes with up to 3 Errors

When $n \geq 9$, both the BMA and the GSA can decoding $(n, n-4)$ codes with at most 2 errors with complexity $\mathcal{O}(n^2)$ and $\mathcal{O}(r^4 n^2)$ respectively. The algorithm outlined in subsection 6.3.1 can correct an $(n, n-4)$ RS codes with up to 3 errors with complexity $\mathcal{O}((L^2+L)n^3)$. When $t \leq 2$, the possible error locator polynomial can be found by classical decoding algorithm. When $t = 3$, the detailed decoding algorithm is as follows.

Let the syndromes for $\mathbf{r}$ be S_1,S_2, S_3, S_4 and the error locator polynomial be $\sigma(x) = \sigma_3 x^3 + \sigma_2 x^2 + \sigma_1 x + 1$. In this case, $\sigma_3 \in \mathcal{G}$ and $\sigma_2, \sigma_1 \in \mathrm{GF}(q)$. From (6.2), $S_4 + S_3\sigma_1 + S_2\sigma_2 + S_1\sigma_3 = 0$.

If $S_3 \neq 0$,

$$\sigma_1 = -\frac{S_1\sigma_3 + S_2\sigma_2 + S_4}{S_3}. \tag{6.13}$$

The equation $f_1(\sigma_3, \sigma_2) = 0$ can be written as

$$\sum_{i=0}^{q}\sum_{j=0}^{n} a_{i,j}\sigma_3^i\sigma_2^i = 0, \text{ where } a_{i,j} \in \mathrm{GF}(2). \tag{6.14}$$

We substitute all possible $\sigma_3 \in \mathcal{G}$ into (6.14) and solve for σ_2 from the resultant univariate polynomial equation. This requires $n(q+1) + q^2$ multiplications at most. Then σ_1 can be solved from (6.13). The valid $\sigma(x)$ should be products of distinct linear factors and checking the $\sigma(x)$ obtained requires $nq \times 3n = 3Ln^3$ multiplications at most. The overall complexity is at most $\mathcal{O}(n(n(q+1)+q^2) + 3Ln^3)$ in terms of multiplications, which is $\mathcal{O}((L^2+4L)n^3)$.

If $S_3 = 0$,

$$S_1\sigma_3 + S_2\sigma_2 + S_4 = 0. \tag{6.15}$$

the possible solution for (σ_3, σ_2) can be found as in the previous case. Since the linear constraint in (6.15) is for σ_3 and σ_2, there are at most q solutions for (σ_3, σ_2). However, since σ_1 is not involved in (6.15), each element in $\mathrm{GF}(q)$ may be a solution for σ_1. Hence, there are at most q^2 possible solutions for $\sigma(x)$ and their validity are checked with $3q^2n$ multiplications. The overall complexity is at most $\mathcal{O}(n(n(q+1)+q^2) + 3q^2n)$ which is $\mathcal{O}((4L^2+L)n^3)$.

Hence, the decoding complexity of an $(n, n-4)$ RS code is upper bounded by $\mathcal{O}((4L^2+L)n^3)$.

The strategy described above can be applied to any $(n, n-4)$ RS codes. Example 6.3 illustrates the algorithm as describe above. For simplicity, we show the decoding of a $(7, 3)$ RS code over $\mathrm{GF}(8)$.

Example 6.3 *Let $\mathcal{C}$ be a $(7, 3)$ RS codes over $\mathrm{GF}(8)$ and with zeros $\alpha, \alpha^2, \alpha^3, \alpha^4$. A noisy received vector of $\mathcal{C}$ is $\mathbf{r} = (0, \alpha, 0, \alpha^3, 1, 0, 0)$. The corresponding*

syndrome sequence is $\alpha^5, \alpha^6, \alpha^4, 1$. *By the BMA, an error locator polynomial* $\alpha^3 x^2 + x + 1$ *for two errors is found and the error locations are* $(0, 2)$. *We next assume there are three errors in* $\mathbf{r}$. *Let the error locator polynomial be* $\sigma_3 x^3 + \sigma_2 x^2 + \sigma_1 x + 1$. *From (6.2),*

$$\sigma_1 = \alpha^2 \sigma_2 + \alpha \sigma_3 + \alpha^3. \tag{6.16}$$

From Example 6.1, $f_0(\sigma_3) = \sigma_3^7 + 1 = 0$ *and all the nonzero elements in* GF(8) *are solutions for* σ_3. *Substitute each* σ_3 *in* $f_1(\sigma_3, \sigma_2) = \sigma_2^5 + \sigma_2^4 \sigma_3^3 + \sigma_2^3 \sigma_3^6 + \sigma_2 \sigma_3^5$ *and solve for* σ_2. *Then solutions for* σ_1 *is obtained by (6.16) from the solutions of* (σ_3, σ_2). *The results is as shown in the table follows.*

Table 6.1: Results for decoding $\mathbf{r} = (0, \alpha, 0, \alpha^3, 1, 0, 0)$**.**

σ_3	σ_2	σ_1	valid $(\sigma_3, \sigma_2, \sigma_1)$
1	$(0, 1, \alpha, \alpha^2, \alpha^4)$	$(1, \alpha^6, \alpha, \alpha^5, \alpha^2)$	$(1, 0, 1), (1, \alpha, \alpha)$
α	$(0, 1, \alpha^3, \alpha^5, \alpha^6)$	$(\alpha^5, \alpha^3, 0, \alpha^4, \alpha^6)$	$(\alpha, 0, \alpha^5), (\alpha, \alpha^3, 0)$
α^2	$(0, 1, \alpha, \alpha^3, \alpha^6)$	$(0, \alpha^2, \alpha^3, \alpha^5, \alpha)$	–
α^3	$(0, \alpha^2, \alpha^3, \alpha^4, \alpha^6)$	$(\alpha^6, \alpha^3, \alpha, 0, \alpha^5)$	$(\alpha^3, \alpha^6, \alpha^5)$
α^4	$(0, 1, \alpha^2, \alpha^5, \alpha^6)$	$(\alpha^2, 0, \alpha, \alpha^6, \alpha^4)$	–
α^5	$(0, \alpha, \alpha^2, \alpha^3, \alpha^5)$	$(\alpha^4, \alpha^6, 0, 1, \alpha^5)$	$(\alpha^5, 0, \alpha^4)$
α^6	$(0, \alpha, \alpha^4, \alpha^5, \alpha^6)$	$(\alpha, 1, \alpha^5, \alpha^3, 0)$	$(\alpha^6, \alpha^5, \alpha^3)$

In the table, the entry "valid $(\sigma_3, \sigma_2, \sigma_1)$*" means the corresponding* $\sigma(x)$ *is products of distinct linear factors over* GF(8). *From the table, we can see that we have* 7 *solutions for error locator polynomials of degree* 3 *and their corresponding error locations are* $(3, 5, 6)$, $(0, 2, 5)$, $(1, 3, 4)$, $(0, 2, 6)$, $(0, 1, 2)$, $(0, 3, 2)$, $(0, 2, 4)$. *But some of the error location sets do not satisfy Lemma* 5.2, *such as* $(0, 1, 2)$ *and* $(0, 3, 2)$. *Actually, the error locator polynomials, corresponding to the error location sets* $(0, 2)$, $(0, 2, 5)$, $(0, 2, 6)$, $(0, 1, 2)$, $(0, 3, 2)$, $(0, 2, 4)$, *decode the received*

vector to the same codeword $\mathbf{c} = (\alpha^3, \alpha, 1, \alpha^3, 1, 0, 0)$. *Hence, only three error locator polynomials,* $\alpha^2 x^2 + x + 1$, $x^3 + x + 1$, $\alpha x^3 + \alpha^5 x + 1$, *correct* $\mathbf{r}$ *to three different codewords,* $(\alpha^3, \alpha, 1, \alpha^3, 1, 0, 0)$, $(0, \alpha, 0, 1, 1, \alpha, \alpha^3)$ *and* $(0, 0, 0, 0, 0, 0, 0)$, *respectively.*

This strategy can also be used to decode (n, k) RS codes with up to $t = \frac{n-k}{2} + 1$ errors when $n-k$ is even. From (6.2), there are $n-k-\frac{n-k}{2}-1 = \frac{n-k}{2}-1$ linear equations for σ_i. All σ_i can be expressed as linear combination of σ_t and σ_{t-1}. One of these linear equations only involves $\sigma_t, \sigma_{t-1}, \sigma_{t-2}$. Combining these linear equations and $f_1(\sigma_t, \sigma_{t-1}) = 0$, we can solve for $\sigma(x)$ with complexity at most $\mathcal{O}(((t+1)L^2 + L)n^3)$ in terms of multiplications.

6.3.4 Combining with Erasures

It is possible that some positions in the received vector are identified as erasures. Given an (n, k) RS codes, when the number of erasures $s \leq n-k-1$, $\lfloor \frac{n-k-1-s}{2} \rfloor$ errors can be corrected by classical decoding algorithm and $\lfloor n - s - \sqrt{(n-s)(k-1)} \rfloor$ errors can be corrected by GS algorithm. Making use of the triangular form of $\mathcal{F}$, up to $\lfloor \frac{n-k-1-s}{2} \rfloor + 1$ errors can be corrected.

Let $\psi(x)$ be the erasure locator polynomial, whose inverse zeros indicates the erasure positions. Let $\sigma'(x) = \psi(x)\sigma(x)$ be the modified error locator polynomial. Since the coefficients of $\sigma'(x)$ instead of $\sigma(x)$ are the unknowns in the basis of $\mathcal{F}$, some process is required to solve for the coefficients of $\sigma(x)$. Since $\psi(x)$ is known, each σ'_i, for $1 \leq i \leq s+t$, can be represented as linear functions of σ_j, $1 \leq j \leq t$. Only the first t σ'_i are independent and σ_j can be expressed as the linear function of these first t σ'_i. Let this relation be

$$\sigma_j = u_j(\sigma'_1, \sigma'_1, \ldots, \sigma'_t). \tag{6.17}$$

Assume the syndrome sequence is $S_1, S_2, \ldots, S_{n-k}$. The erasure polynomial

modified the syndrome sequence to $S'_1, S'_2, \ldots, S'_{n-k-s}$ by $S'_j = \sum_{i=1}^{s+1} S_i \psi_{s+j-i}$, $1 \leq j \leq n-k-s$. This modified syndrome sequence and coefficients of $\sigma(x)$ satisfy (6.1). After substituting (6.17) to this equations, linear relations for $\sigma'_1, \sigma'_2, \ldots, \sigma'_t$ similar to (6.2) can be derived. After solving for the $\sigma'_1, \ldots, \sigma'_{2t-(n-k-s)}$ from the basis of $\mathcal{F}$, valid $\sigma'_1, \sigma'_2, \ldots, \sigma'_t$ and valid $\sigma_1, \sigma_2, \ldots, \sigma_t$ can be found.

6.4 Decoding IRS Codes

Let $\mathcal{C}$ be an (n, k) RS code over GF(q) with zeros $\alpha, \alpha^2, \ldots, \alpha^{n-k}$. An $r' \times n$ array can be constructed by arranging one codeword of $\mathcal{C}$ in each row of this array. All such arrays are called IRS code. It is assumed that an array is transmitted column by column in a burst error channel and that a burst error vector has length r'. It is also assumed all the error values are independent. IRS codes can be used to combat the burst errors which occur frequently in storage channel and wireless fading channels. In [8], it is shown an $r' \times n$ IRS code, where row codes are (n, k) RS codes over GF(q), can correct $t \leq \lfloor \frac{(n-k)r'}{r'+1} \rfloor$ burst errors with high probability by simultaneous polynomial reconstruction when $r' \geq \frac{t}{n-k-t}$. The probability of correcting t burst errors is $1 - \frac{t}{q}$.

Since all the row codes in an IRS code share the same error locator polynomial, the IRS code can also be decoded via error locator polynomial. Let $\mathbf{c}_i \in \mathcal{C}$ and $r' \geq 1$. A code array for an IRS code can be $\mathbf{C} = (\mathbf{c}_1^T, \mathbf{c}_2^T, \ldots, \mathbf{c}_r^T)^T$. Assume $t \leq \lfloor \frac{(n-k)r'}{r'+1} \rfloor$ burst errors occur. Let the error vector in the ith row be $(e_{i,0}, e_{i,1}, \ldots, e_{i,n-1})$ for $1 \leq i \leq r'$. The jth syndrome for the ith row code is

$S_{i,j} = \sum_{v=0}^{n-1} e_{i,v}\alpha^{vj}$. Let

$$\mathbf{S}_t = \begin{pmatrix} S_{1,1} & S_{1,2} & \cdots & S_{1,t} \\ \vdots & \vdots & \vdots & \vdots \\ S_{1,n-k-1-t} & S_{1,n-k-t} & \cdots & S_{1,n-k-1} \\ \vdots & \vdots & \vdots & \vdots \\ S_{r',1} & S_{r',2} & \cdots & S_{r',t} \\ \vdots & \vdots & \vdots & \vdots \\ S_{r',n-k-1-t} & S_{r',n-k-t} & \cdots & S_{r',n-k-1} \end{pmatrix}.$$

By the (6.2),

$$\mathbf{S}_t \begin{pmatrix} \sigma_t \\ \sigma_{t-1} \\ \vdots \\ \sigma_1 \end{pmatrix} = \begin{pmatrix} S_{1,t+1} \\ \vdots \\ S_{1,n-k} \\ \vdots \\ S_{r',t+1} \\ \vdots \\ S_{r',n-k} \end{pmatrix} \tag{6.18}$$

When $t \leq \lfloor \frac{n-k}{2} \rfloor$, $\mathbf{S}_t$ is full rank. A unique solution for the error locator polynomial can be found from (6.18). When $\lfloor \frac{n-k}{2} \rfloor < t \leq \lfloor \frac{(n-k)r'}{r'+1} \rfloor$, $\mathbf{S}_t$ may not be full rank. Since each entry in $\mathbf{S}_t$ is linear combination of the error values, the rank of $\mathbf{S}_t$ depends on the error values. All the error values are random variables and the Lemma 6.1 shows the probability of $\mathbf{S}_t$ being full rank is at least $1 - \frac{t}{q}$.

Lemma 6.1 *The matrix $\mathbf{S}_t$ is full rank with probability at least $1 - \frac{t}{q}$.*

Proof: Let $\bar{\mathbf{S}}_t$ be any $t \times t$ submatrix of $\mathbf{S}_t$. The matrix $\mathbf{S}_t$ is full rank if $\bar{\mathbf{S}}_t$ is full rank. Let the rows in $\mathbf{S}_t$ is indexed by integers $1, 2, \ldots, r'(n-k-t)$. Let $\mathcal{B}$ be the set of indexes of rows in $\bar{\mathbf{S}}_t$. Then $S_{i,j}$ is in $\bar{\mathbf{S}}_t$ for any $i \in \mathcal{B}$ and $0 \leq j \leq n-1$.

Since $S_{i,j}$ is a linear combination of the entries in $\mathbf{e}_i$, the determinant $\det(\bar{\mathbf{S}}_t)$ is a non-trivial multivariate polynomial of $e_{i,j}$ for $i \in \mathcal{B}$ and $0 \leq j \leq n-1$. The total degree of this polynomial is t. Since $e_{i,j}$ are random variables over GF(q), by [77], the probability of this polynomial equal to zero is at most $\frac{t}{q}$. Hence, $\mathbf{S}_t$ is full rank with probability at least $1 - \frac{t}{q}$. ■

From Lemma 6.1, the probability of successful decoding t burst errors is $1 - \frac{t}{q}$. When the rank of $\mathbf{S}_t$ is less than t, it is still possible to decode the received array by making use of (6.18) and the equations in the triangular form of $\mathcal{F}$. Due to complexity, we only consider the cases when rank of $\mathbf{S}_t$ are $t-1$ and $t-2$. If $\text{rank}(\mathbf{S}_t) = t-1$, all the σ_i, $1 \leq i \leq t$, can be expressed as linear functions of σ_t from (6.18). This is similar as decoding an (n,k) RS codes up to $\lfloor \frac{n-k}{2} \rfloor + 1$ errors when $n-k$ is odd and it has the same complexity. If $\text{rank}(\mathbf{S}_t) = t-2$, it is similar as decoding an (n,k) RS codes up to $\lfloor \frac{n-k}{2} \rfloor + 1$ errors when $n-k$ is even and it has the same complexity. Hence, the decoding failure is with probability at most $\frac{t-2}{q}$. When cardinality of noise is small, this improvement is significant. Hence, this probabilistic decoding algorithm can correct up to $n-k-1$ burst errors with probability at least $1 - \frac{t-2}{q}$ if there are $n-k-1$ rows in an IRS code array. Since $k < n$ and $r' \geq \frac{t}{n-k-t}$, $t \leq \lfloor \frac{(n-k)r'}{r'+1} \rfloor \leq n-k-1$.

6.5 Decoding Codes of Length 7

In this section, decoding RS codes of length 7 over GF(8) is considered. Since these short RS codes are applied in high speed communications and the real-time communications over wireless channel [87, 88], it is interesting to consider these short codes with mediate and high rate.

The decoding algorithm is based on the decomposition of the conjugate syndrome sequence of S_1, S_2, S_4. Before we present the algorithm for this class of

RS codes, a method for fining the decomposition of these syndromes is needed.

6.5.1 Decomposition of S_1, S_2, S_4 and Decoding $(7, 3)$ RS Codes over GF(8)

Assume a received vector $\mathbf{r} = \mathbf{c} + \mathbf{e}$, where $\mathbf{c}$ is a codeword of an (n, k) RS code over GF(2^m) and the error vector $\mathbf{e} = (e_0, e_1, \ldots, e_{n-1})$. Assume the syndromes $S_j = \sum_{i=0}^{n-1} r_i \alpha^{ij}$, where $1 \leq j \leq n-k$ and $\mathrm{ord}(\alpha) = n$, are known. The finite field GF(2^m) is isomorphic to the residue class GF(2)$[x]/p(x)$, where $p(x)$ is a primitive polynomial in GF(2) and $\deg(p(x)) = m$. Then $e_i \in$ GF(2^m) can be represented by its binary image $(e_{i,0}, e_{i,1}, \ldots, e_{i,m-1}) \in$ GF$(2)^m$ or $e_i = \sum_{u=0}^{m-1} e_{i,u}\alpha^u$. Let $S_j^{(u)} = \sum_{i=0}^{n-1} e_{i,u}\alpha^{ij}$. The syndromes of $\mathbf{r}$ can be decomposed as

$$\begin{aligned} S_j &= \sum_{i=0}^{n-1} r_i \alpha^{ij} = \sum_{i=0}^{n-1} c_i \alpha^{ij} + e_i \alpha^{ij} = \sum_{i=0}^{n-1} \sum_{u=0}^{m-1} e_{i,u} \alpha^u \alpha^{ij} \\ &= \sum_{u=0}^{m-1} \alpha^u \sum_{i=0}^{n-1} e_{i,u} \alpha^{ij} = \sum_{u=0}^{m-1} \alpha^u S_j^{(u)}. \end{aligned} \tag{6.19}$$

The decomposition of $S_1, S_2, S_4, \ldots, S_{2^{m-1}}$ can be obtained by Theorem 6.2.

Theorem 6.2 *Given the syndromes $S_1, S_2, S_4, \ldots, S_{2^{m-1}}$ for a received word $\mathbf{r} \in$ GF$(2^m)^n$, their decomposition can be obtained with $\mathcal{O}(m^2)$ multiplications.*

Proof: Recall that if $x, y \in$ GF(p^m) and p is a prime integer, $(x+y)^{p^l} = x^{p^l} + y^{p^l}$ [91] and if $x \in$ GF(2) and $y \in$ GF(2^m), $(xy)^l = xy^l$ for $l \in \mathbb{Z}^+$. From (6.19) and $j = 2^{l'}$ for $l' \in [0, m-1]$, we have

$$\begin{aligned} S_j^{2^l} &= (\sum_{u=0}^{m-1} \alpha^u \sum_{i=0}^{n-1} e_{i,u} \alpha^{ij})^{2^l} = \sum_{u=0}^{m-1} \alpha^{u2^l} \sum_{i=0}^{n-1} e_{i,u} \alpha^{ij2^l} \\ &= \sum_{u=0}^{m-1} \alpha^{u2^l} \sum_{i=0}^{n-1} S^{(u)}_{j2^l \,(\mathrm{mod}\, 2^m-1)} = \sum_{u=0}^{m-1} \alpha^{u2^l} \sum_{i=0}^{n-1} S^{(u)}_{2^{(l+l')} \,(\mathrm{mod}\, m)}. \end{aligned} \tag{6.20}$$

Since $S_j^{2^m} = S_j$, there are m equations from (6.20) for all $l \in [0, m-1]$ and given j. Among these m equations, there is one and only one linear equation

involving $S_{2^{l'}}^{(u)}$ for each $u \in [0, m-1]$, given $l' \in [0, m-1]$. Hence, there are m^2 equations from (6.20). Among them, m are linear equations involving $S_{2^l}^{(u)}$ for given l and u. Denoting $\langle l - l' \rangle = l - l' \pmod m$, these m equations are as follows.

$$\begin{aligned}
S_1^{2^l} &= \sum_{u=0}^{m-1} \alpha^{u2^l} S_{2^l}^{(u)} \\
S_2^{2^{\langle l-1 \rangle}} &= \sum_{u=0}^{m-1} \alpha^{u2^{\langle l-1 \rangle}} S_{2^l}^{(u)} \\
S_{2^2}^{2^{\langle l-2 \rangle}} &= \sum_{u=0}^{m-1} \alpha^{u2^{\langle l-2 \rangle}} S_{2^l}^{(u)} \\
&\vdots \\
S_{2^{m-1}}^{2^{\langle l-(m-1) \rangle}} &= \sum_{u=0}^{m-1} \alpha^{u2^{\langle l-(m-1) \rangle}} S_{2^l}^{(u)}.
\end{aligned} \tag{6.21}$$

These equations in matrix form are

$$\begin{aligned}
\begin{pmatrix} S_1^{2^l} \\ S_2^{2^{\langle l-2 \rangle}} \\ \vdots \\ S_{2^{m-1}}^{\langle l-(m-1) \rangle} \end{pmatrix} &= \begin{pmatrix} 1 & \alpha^{2^l} & \cdots & \alpha^{(m-1)2^l} \\ 1 & \alpha^{2^{\langle l-1 \rangle}} & \cdots & \alpha^{(m-1)2^{\langle l-1 \rangle}} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{2^{\langle l-(m-1) \rangle}} & \cdots & \alpha^{(m-1)2^{\langle l-(m-1) \rangle}} \end{pmatrix} \begin{pmatrix} S_{2^l,0} \\ S_{2^l,1} \\ \vdots \\ S_{2^l,m-1} \end{pmatrix} \\
&= \mathbf{A} \times \begin{pmatrix} S_{2^l,0} \\ S_{2^l,1} \\ \vdots \\ S_{2^l,m-1} \end{pmatrix},
\end{aligned} \tag{6.22}$$

where

$$\mathbf{A} = \begin{pmatrix} 1 & \alpha^{2^l} & \cdots & \alpha^{(m-1)2^l} \\ 1 & \alpha^{2^{\langle l-1 \rangle}} & \cdots & \alpha^{(m-1)2^{\langle l-1 \rangle}} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{2^{\langle l-(m-1) \rangle}} & \cdots & \alpha^{(m-1)2^{\langle l-(m-1) \rangle}} \end{pmatrix}.$$

Since $0 \leq l \leq m-1$, $\alpha^{2^l}, \alpha^{2^{\langle l-1 \rangle}}, \ldots, \alpha^{2^{\langle l-(m-1) \rangle}}$ are distinct nonzero elements in GF(2^m), the Vandermonde matrix $\mathbf{A}$ is invertible. So the decomposition of the syndromes $S_1, S_2, S_4, \ldots, S_{2^{m-1}}$ is decided by

$$\begin{pmatrix} S_{2^l,0} \\ S_{2^l,1} \\ \vdots \\ S_{2^l,m-1} \end{pmatrix} = \mathbf{A}^{-1} \times \begin{pmatrix} S_1^{2^l} \\ S_2^{2^{\langle l-2 \rangle}} \\ \vdots \\ S_{2^{m-1}}^{\langle l-(m-1) \rangle} \end{pmatrix}, \text{ for } 0 \leq l \leq m-1. \tag{6.23}$$

It is sufficient to solve for $S_{1,0}, S_{1,1}, \ldots, S_{1,m-1}$ to find the decomposition because

$$S_{2^l}^{(u)} = \sum_{i=0}^{n-1} e_{i,u} \alpha^{i2^l} = (\sum_{i=0}^{n-1} e_{i,u} \alpha^i)^{2^l} = (S_1^{(u)})^{2^l}, \text{ for } 0 \leq l \leq m-1.$$

Thus, the m^2 multiplications over GF(2^m) in computing $S_{1,0}, S_{1,1}, \ldots, S_{1,m-1}$ is the major computation required since the matrix $\mathbf{A}^{-1}$ can be computed in advance. ■

We next consider decoding $(7,3)$ RS codes over GF(2^3) with up to 3 errors to show the application of this decomposition. Let the reciprocals of the zeros of a polynomial $\sigma^{(u)}(x) = \sum_{i=1}^{t_u} \sigma_i^{(u)} x^i + 1$ indicate the positions of the nonzero coordinates in $(e_{0,u}, e_{1,u}, \ldots, e_{n-1,u})$. Since the $(e_{0,u}, e_{1,u}, \ldots, e_{n-1,u})$ is binary, $S_j^{(u)}$ for $1 \leq j \leq n-k$ and $\sigma^{(u)}(x)$ satisfies the GNI and $\sigma_1^{(u)} = S_1^{(u)}$. For example, when $t_u = 3$, according to the GNI,

$$\begin{aligned} S_1^{(u)} &= \sigma_1^{(u)} \\ S_2^{(u)} &= (\sigma_1^{(u)})^2 \\ S_3^{(u)} &= \sigma_1^{(u)} S_2^{(u)} + \sigma_2^{(u)} S_1^{(u)} + \sigma_3^{(u)} = (S_1^{(u)})^3 + \sigma_2^{(u)} S_1^{(u)} + \sigma_3^{(u)} \\ S_4^{(u)} &= (\sigma_1^{(u)})^4 \end{aligned} \tag{6.24}$$

For $t = 1, 2$, the received vector of the $(7,3)$ RS code can be decoded by the classical decoding algorithms. For $t = 3$, we make use of the above decomposition.

For each u, there are 4 possible cases for $(e_{0,u}, e_{1,u}, \ldots, e_{n-1,u})$, which are $t_u = 0, 1, 2, 3$. If $t_u = 0$, $S_1^{(u)} = S_2^{(u)} = S_3^{(u)} = S_4^{(u)} = 0$. If $t_u = 1$, from (6.24), $S_3^{(u)} = (S_1^{(u)})^3$ since $\sigma_2^{(u)} = \sigma_3^{(u)} = 0$. If $t_u = 2$, $S_3^{(u)} = (S_1^{(u)})^3 + \sigma_2^{(u)} S_1^{(u)}$. Since $\sigma_1^{(u)}$ is known, the possible $\sigma_2^{(u)}$ can be solved from $f_1(\sigma_1^{(u)}, \sigma_2^{(u)}) = 0$. If $t_u = 3$, $\sigma^{(u)}(x) = \sigma(x)$ and $\sigma_1 = S_1^{(u)}$. Then $S_4 = S_3\sigma_1 + S_2\sigma_2 + S_1\sigma_3 = S_3 S_1^{(u)} + S_2\sigma_2 + S_1\sigma_3$. If $S_1 \neq 0$,

$$\sigma_3 = \frac{S_4 + S_3 S_1^{(u)} + S_2\sigma_2}{S_1}. \tag{6.25}$$

substituting (6.25) into $f_1(\sigma_2, \sigma_3) = 0$, the σ_3 can be solved. If $S_1 = 0$, then σ_2 can be solve directly and substitute into $f_1(\sigma_2, \sigma_3) = 0$ to solve for σ_3. For $0 \leq u \leq 2$, the errors in $(e_{0,u}, e_{1,u}, \ldots, e_{n-1,u})$ share 3 error locations. Then the possible error location combinations can be constructed. Moreover, For each valid combination of error locations, $S_3 = \sum_{i=0}^{2} \alpha^u S_3^{(u)}$. This is shown in the example follows.

Example 6.4 *Let* $\mathbf{r} = (0, \alpha, \alpha^2, 0, 0, \alpha^4, 0)$ *be a received vector of a* $(7, 3)$ *RS code. The syndromes are* $S_1 = \alpha^4, S_2 = \alpha^5, S_3 = \alpha^3, S_4 = \alpha^5$. *If* $t = 2$, *by classical decoding algorithm, an error locator polynomial is* $\sigma(x) = \alpha^3 x^2 + \alpha^3 x + 1$, *which indicates two errors in locations* 4 *and* 6, *respectively. The received vector is decoded as* $(0, \alpha, \alpha^2, 0, \alpha^2, \alpha^4, \alpha^4)$. *If* $t = 3$, *the decomposition of* (S_1, S_2, S_4) *are* $(S_1^{(0)}, S_2^{(0)}, S_4^{(0)}) = (0, 0, 0)$, $(S_1^{(1)}, S_2^{(1)}, S_4^{(1)}) = (\alpha^6, \alpha^5, \alpha^3)$ *and* $(S_1^{(2)}, S_2^{(2)}, S_4^{(2)}) = (\alpha^3, \alpha^6, \alpha^5)$. *For* $u = 0, 1, 2$, *the results are as Table 6.2, 6.3 and 6.4, respectively. (When* $t_u = 2$, $f_1(\sigma_1^{(2)}, \sigma_2^{(2)}) = \sigma_1^6 + \sigma_2\sigma_1^4 + \sigma_2^3$. *When* $t_u = 3$, $f_1(\sigma_2, \sigma_3) = \sigma_2^5 + \sigma_2^4\sigma_3^3 + \sigma_2^3\sigma_3^6 + \sigma_2\sigma_3^5$, $f_2(\sigma_1, \sigma_2, \sigma_3) = \sigma_3 + \sigma_2^3\sigma_3^3 + \sigma_3^5$.*)* *When* $t = 3$, $(3, 4, 6)$, $(2, 5, 6)$ *and* $(0, 1, 6)$ *are possible error location combinations. However, by Lemma* 5.2, *they are plausible since* $(4, 6)$ *are error position set for* $t = 2$. *The remaining possible error position combinations are as the first column in Table 6.5 and validity of them are checked by comparing the corresponding* S_3 *and the given true value of* $S_3 = \alpha^3$. *Only one possible error position set,* $(1, 2, 5)$, *is valid when* $t = 3$. *The corresponding error locator polynomial is* $x^3 + x^2 + \alpha^5 x + \alpha$ *and the received vector is decoded as* $(0, 0, 0, 0, 0, 0, 0)$.

Table 6.2: Result for $u = 0$.

t_0	$S_3^{(0)}$	$\sigma^{(0)}(x)$	error locations
0	0	-	-
1	-	-	-
2	-	-	-
3	-	-	-

Table 6.3: Result for $u = 1$.

t_1	$S_3^{(1)}$	$\sigma^{(1)}(x)$	error locations
0	-	-	-
1	α^4	$\alpha^6 x + 1$	6
2	α^3	$x^2 + \alpha^6 x + 1$	$3, 4$
	α^2	$\alpha^2 x^2 + \alpha^6 x + 1$	$0, 2$
	1	$\alpha^6 x^2 + \alpha^6 x + 1$	$1, 5$
3	-	-	-

If the code is a $(7, 4)$ cyclic code over GF(8) with zeros $\alpha, \alpha^2, \alpha^4$, S_3 is not available. Correcting up to 3 errors is still possible in this case. When $t = 1$, the error locator polynomial can be solved from S_1, S_2. When $t = 2$, since $S_3 = \sigma_1 S_2 + \sigma_2 S_1$ and $S_4 = \sigma_1 S_3 + \sigma_2 S_2$, we have $\sigma_2 = \frac{S_4 + \sigma_1^2 S_2}{\sigma_1 S_1 + S_2}$. Then the possible solutions for σ_1 and σ_2 can be solved from $f_1(\sigma_2, \sigma_3) = 0$. When $t = 3$, the decomposition of S_1, S_2, S_4 can be computed and used to solve for $\sigma^{(u)}(x)$ for each u. But the checking of the error position combinations is not needed, since S_3 is not known. Hence, all the combinations with 3 error positions are valid solutions.

Table 6.4: Result for $u = 2$.

t_2	$S_3^{(2)}$	$\sigma^{(2)}(x)$	error locations
0	-	-	-
1	α^2	$\alpha^3 x + 1$	3
2	α^5	$x^2 + \alpha^3 x + 1$	2, 5
	α	$\alpha x^2 + \alpha^3 x + 1$	0, 1
	1	$\alpha^3 x^2 + \alpha^3 x + 1$	4, 6
3	-	-	-

Table 6.5: Possible error position combinations.

error positions	S_3	validity
$(0, 2, 3)$	α^6	-
$(0, 2, 5)$	α	-
$(0, 1, 2)$	0	-
$(1, 3, 5)$	α^2	-
$(1, 2, 5)$	α^3	$\surd$
$(0, 1, 5)$	1	-

6.5.2 Decoding RS Codes over GF(8) with Restricted Error Value

We consider case that the error values are a subset of the RS code symbol alphabet. This is possible in concatenated coding scheme when the RS code over GF(q) is the outer code, because the inner code may introduce error symbols only in a subset of GF(q) due to the error propagation. With this limited error value set, more errors can be corrected.

If $\beta \in \mathrm{GF}(q)$ and $q = p^m$, the trace of this element over $\mathrm{GF}(p)$ is defined as $\mathrm{Tr}(\beta) = \sum_{u=0}^{m-1} \beta^{p^u}$. The trace operation has been shown as a linear transformation from $\mathrm{GF}(q)$ to $\mathrm{GF}(p)$ in [45, Theorem 2.23]. The following Theorem 6.3 also gives a property of the trace operation.

Theorem 6.3 *[45, Theorem 2.25] Let $\beta \in \mathrm{GF}(q)$ and $q = p^m$. If and only if $\beta = \gamma^p - \gamma$ for some $\gamma \in \mathrm{GF}(q)$, $\mathrm{Tr}(\beta) = 0$.*

When $p = 2$, there are only two elements, 0 and 1, in $\mathrm{GF}(2)$. Then $\mathrm{Tr}(\gamma)$ is either 1 or 0. We consider the number of elements in $\mathrm{GF}(2^m)$ mapped to 0 and 1 by trace operation, respectively.

Theorem 6.4 *There are 2^{m-1} elements in $\mathrm{GF}(2^m)$ mapped to $0, 1 \in \mathrm{GF}(2)$ by trace operation, respectively.*

Proof: According to Theorem 6.3, if $\mathrm{Tr}(\beta) = 0$, the equation $y^2 + y = \beta$ must have solutions over $\mathrm{GF}(2^m)$. If y_1 and y_2 are solutions of this equation over $\mathrm{GF}(2^m)$, $y_1 \neq y_2$. This is because $y_1 + y_2 = 1$ from the coefficients of this equation. In addition, for $\beta_1 \neq \beta_2$ and $\mathrm{Tr}(\beta_1) = \mathrm{Tr}(\beta_2) = 0$, the sets of solutions for $y^2 + y = \beta_1$ and $y^2 + y = \beta_2$ are disjointed.

On the other hand, $y^2 + y$ maps each $y \in \mathrm{GF}(2^m)$ to an element in $\mathrm{GF}(2^m)$. Here $y^2 + y$ is a surjective mapping and exact 2 elements y_1 and y_2 satisfying $y_1 + y_2 = 1$ in the domain are mapped to an element in the codomain. Let the codomain have x elements. We have $2x = 2^m$ and $x = 2^{m-1}$. This means there are 2^{m-1} elements in $\mathrm{GF}(2^m)$ can be represented as $y^2 + y$ for some $y \in \mathrm{GF}(2^m)$. Hence, 2^{m-1} elements in $\mathrm{GF}(2^m)$ are mapped to 0 and the remaining 2^{m-1} are mapped to 1 by the trace operation. ■

Denote the sets $\mathcal{S}_0, \mathcal{S}_1 \subset \mathrm{GF}(2^m)$ as the subsets of elements mapped to $0, 1 \in \mathrm{GF}(2)$ by the trace operation, respectively. Then $|\mathcal{S}_0| = |\mathcal{S}_1| = 2^{m-1}$.

We next describe the decoding of the received vector with restricted error values. Two cases are considered, where the error values are in $\mathcal{S}_0$ and $\mathcal{S}_1$, respectively.

$e_i \in \mathcal{S}_0$

When the error value are in $\mathcal{S}_0$, $(7,4)$ and $(7,5)$ RS code over GF(8) can correct up to 3 errors by the decomposition of the syndromes S_1, S_2, S_4. The decoding algorithm makes use of the following Theorem 6.5.

Theorem 6.5 *For RS codes over* GF(2^m)*, if all the error values are in* $\mathcal{S}_0$*,* $S_{2^{m-1}} = \sum_{i=0}^{m-2} S_{2^i}^{2^{m-1-i}}$.

Proof: Since $S_{2^i} = \sum_{e_j \neq 0} e_j \alpha^{j2^i}$, we have $S_{2^i}^{2^{m-1-i}} = \sum_{e_j \neq 0} e_j^{2^{m-1-i}} \alpha^{j2^{m-1}}$. From $e_j \in \mathcal{S}_0$, we have $\mathrm{Tr}(e_j) = 0$ and

$$\begin{aligned}\sum_{i=0}^{m-2} S_{2^i}^{2^{m-1-i}} &= \sum_{i=0}^{m-2} \sum_{e_j \neq 0} e_j^{2^{m-1-i}} \alpha^{j2^{m-1}} = \sum_{e_j \neq 0} \alpha^{j2^{m-1}} \sum_{i=0}^{m-2} e_j^{2^{m-1-i}} \\ &= \sum_{e_j \neq 0} \alpha^{j2^{m-1}} (\mathrm{Tr}(e_j) + e_j) = \sum_{e_j \neq 0} e_j \alpha^{j2^{m-1}} = S_{2^{m-1}}.\end{aligned}$$

■

If all $e_j \in \mathcal{S}_0$ and S_{2^i} for $0 \leq i \leq m-2$ are known, $S_{2^{m-1}}$ can be computed according to Theorem 6.5. The syndromes S_1, S_2 and S_3 can be obtained from the received vector of a $(7,4)$ RS code over GF(8). Since the error values are in $\mathcal{S}_0$, S_4 can be computed from S_1, S_2. Then the decoding procedure is similar as decoding a $(7,3)$ RS code over GF(8) in the previous subsection. But the error values are in $\mathcal{S}_0$.

Further, the syndromes $S_1$and S_2 can be obtained from a received vector of a $(7,5)$ RS code. If the error values are in $\mathcal{S}_0$, S_4 can also be obtained. Then the decoding is similar as decoding $(7,3)$ cyclic codes over GF(8) with zeros $\alpha, \alpha^2, \alpha^4$. Moreover, when $e_i \in \mathcal{S}_0$ of GF(8), $e_{i,0} = 0$ for $0 \leq i \leq 7$. So, there is no need to find the possible $\sigma^{(0)}(x)$.

$e_i \in \mathcal{S}_1$

Assume $\mathbf{r}$ is a received vector of a $(7, 3)$ RS code over GF(8). When the error values in $\mathbf{r}$ are from $\mathcal{S}_1$, the complexity of decoding $\mathbf{r}$ with up to 3 errors can be reduced based on the Theorem 6.6 follows.

Theorem 6.6 *For RS codes over* GF(2^m)*, if all the error values are in* $\mathcal{S}_1$*,* $\sigma_1^{2^{m-1}} = \sum_{i=0}^{m-1} S_{2^i}^{2^{m-1-i}}$.

Proof: From $S_{2^i} = \sum_{e_j \neq 0} e_j \alpha^{j2^i}$, we have $S_{2^i}^{2^{m-1-i}} = \sum_{e_j \neq 0} e_j^{2^{m-1-i}} \alpha^{j2^{m-1}}$. Also, $\sigma_1 = \sum_{e_j \neq 0} \alpha^j$. Since $e_j \in \mathcal{S}_1$, we have $\mathrm{Tr}(e_j) = 1$ and

$$\begin{aligned}\sum_{i=0}^{m-1} S_{2^i}^{2^{m-1-i}} &= \sum_{i=0}^{m-1} \sum_{e_j \neq 0} e_j^{2^{m-1-i}} \alpha^{j2^{m-1}} = \sum_{e_j \neq 0} \alpha^{j2^{m-1}} \sum_{i=0}^{m-1} e_j^{2^{m-1-i}} \\ &= \sum_{e_j \neq 0} \alpha^{j2^{m-1}} \mathrm{Tr}(e_j) = \sum_{e_j \neq 0} \alpha^{j2^{m-1}} = (\sum_{e_j \neq 0} \alpha^j)^{2^{m-1}} = \sigma_1^{2^{m-1}}.\end{aligned}$$

■

The syndromes S_1, S_2, S_3, S_4 are available for a received vector of a $(7, 3)$ RS code. When $t \leq 2$, the possible error locator polynomial can be found by the equation from (6.1). When $t = 3$, assume the error locator polynomial is $\sigma_3 x^3 + \sigma_2 x^2 + \sigma_1 x + 1$, where σ_1 can be computed according to Theorem 6.6. From (6.1), $S_4 = \sigma_1 S_3 + \sigma_2 S_2 + \sigma_3 S_1$. Combined with $f_2(\sigma_1, \sigma_2, \sigma_3) = 0$, all the possible σ_1 and σ_2 can be solved.

For a $(7, 4)$ cyclic code over GF(q) with zeros $\alpha, \alpha^2, \alpha^4$, if the error values are in $\mathcal{S}_1$, it is possible correcting up to 3 errors in the received vector. Let the syndromes for a received vector be S_1, S_2, S_4. If $t = 1$, σ_1 can be computed according to Theorem 6.6. If $t = 2$, assume the error locator polynomial is $\sigma_2 x^2 + \sigma_1 x + 1$. From (6.1),

$$S_4 = \sigma_1 S_3 + \sigma_2 S_2,$$

$$S_3 = \sigma_1 S_2 + \sigma_2 S_1.$$

Then $S_4 = \sigma_1^2 S_2 + \sigma_1\sigma_2 S_1 + \sigma_2 S_2$. Since σ_1 is known, σ_2 can be solved. If $t = 3$, there are only one equations from (6.1) and there are three unknowns, S_3, σ_2, σ_3, in this equation. We can solve σ_2 from $f_2(\sigma_1, \sigma_2, \sigma_3) = 0$ for each possible σ_3. About $7^2 \times 5$ multiplications are involved to solve for all the possible σ_2 and σ_3.

6.6 Summary

A decoding strategy for RS codes based on the GNI and the orthogonal relation are proposed in this chapter. It is a list-type decoding method with improved error-correction capability. For $(n, n-3)$ RS codes, 2 errors can be corrected with lower complexity than the GSA. For $(n, n-4)$ RS codes, 3 errors can be corrected when $n \geq 9$. The error-correction capability is better than that of the classical algorithms and the GSA in this case. The algorithm can be applied in decoding mediate and high rate RS codes, BCH codes, IRS codes and FRS codes. The application of this technique in decoding cyclic codes over GF(8) with restricted error values is also studied.

Chapter 7

Conclusion and Proposals for Future Work

In this chapter, we draw the conclusion for the research work conducted in this thesis. Possible future research topics are also proposed and applications are suggested.

7.1 Conclusion

In this thesis, we have shown that FRS codes could be constructed from any RS code with codelength a composite number, which generalizes the construction of FRS codes in [44]. Instead of studying the syndromes of the row codes in the resulting code array, we analyze the zeros of the code polynomials of these row codes. We show that the zeros of these row codes can be obtained by distributing the zeros of the original RS code. In addition, these row codes are identified as GRS codes. Also, the syndromes of the row codes can be obtained by distributing the syndromes of the original RS code. FRS codes have an interleaved structure due to their construction. They are advantageous in correcting burst errors when transmitted column by column in burst error channels. Moreover, to detect burst

errors effectively, TFSRS codes and a decoding algorithm based on the GSA are proposed. We also derive estimations of the probability of successful decoding, decoder error and decoding failure of our algorithm.

An FRS code can be viewed as an IRS code if the column transformation is performed before the transmission. Thus each row code can be encoded independently. However, if these row codes are not encoded via the evaluation mapping, the output list of the interpolation-based list decoders is a coset of the most possible candidate messages. So we need retrieve these most possible candidate messages from the output list of such a decoder. In this thesis, we interpret the evaluation map as the GFFT of the extended message vectors and derive a decomposition of the extended generator matrix. We then establish a relationship between codewords resulting from the generator-matrix-based encoding, and codewords obtained via the evaluation map. We further derive from this relationship, a transformation for recovering the generator-matrix-based coded message under the interpolation-based list decoder. The transformation matrix can be computed in advance. To retrieve the message data, an average computational overhead of $\mathcal{O}(k^2)$ is required for an (n, k) RS code. In addition, to improve the performance of systems employing RS codes, incorporating the interpolation-based list decoder in existing systems employing RS codes is obvious a good choice. But most of these systems encode RS codes by the generator polynomial. The technique proposed in this thesis can be a way to solve this problem.

Moreover, we show that FGRS codes can be constructed from GRS codes and that all the row codes of the resulting FGRS code array can be modified as GRS codes with the zeros from the same support set. The syndromes of the row codes in the resulting FGRS code array may not be consecutive. To decode such FGRS codes, we proposed a method for the synthesis of multisequences with

unknown elements in the middle. Based on this method, we present a decoding algorithm for FGRS codes. When an FGRS code array is transmitted column by column in burst error channels, this algorithm can exploit the fact that all the rows in the code array share the same error pattern. From the construction of FGRS codes, we can see that folded codes can also been constructed from BCH codes. The proposed algorithm can be applied to decode the resulting folded codes.

Further, it is shown by the results of the algebraic list decoding that RS codes are highly non-perfect codes. Their error-correction capability can be improved by the list decoding technique. Given a Hamming sphere with radius significant larger than the classical error-correction capability, there are a few codewords in this sphere in most of the cases. We expect decoding row codes of an FRS codes by the list decoding to be advantageous. Especially, when all the row codes in an FRS code array shared the same error pattern, the decoding of successive rows can make use of the error locations found in the previous row codes. Hence, the error-correcting performance can be improved. Based on these, we propose two list decoding algorithms for RS codes.

First, we present a search-based list decoding algorithm capable of correcting up to $n - k - 1$ errors, given an (n, k) RS code. Its error-correction capability exceeds that of the GSA for a wide range of code parameters, although with increased decoding complexity. Nevertheless, we have demonstrated that for short, high rate codes, it is possible that the average complexity of the proposed search procedure is less than n^2 at WERs of practical interests. This algorithm can be applied to decode FRS code, where the rows of the array are short and high rate RS codes. An appropriate choice of dimension for this array will thus permit the proposed algorithm to be applied with reasonable complexity at practical WERs. Moreover, although we describe our decoding algorithm in the context of

RS codes, it is clear that our decoding method is in fact applicable to some GRS codes and its subfield subcodes which have consecutive syndrome sequences.

Next, we study the list decoding algorithm based on the combination of the GNI and the GB method. For an (n, k) RS code over GF(q), the GB is for the equations from the relation of $x^{\deg(\sigma(x))}\sigma(x^{-1})h(x) = x^n - 1$, where $\sigma(x)$ is the error locator polynomial and the $h(x)$ can be factorized as products of $\deg(h(x))$ distinct linear factors over GF(q). Moreover, the group of linear equations from the GNI for a received vector are combined with the equations obtained from the GB. The solutions give a list of the most possible error locator polynomials for a received vector. For $(n, n-3)$ RS codes, 2 errors can be corrected with lower complexity than that of the GSA. For $(n, n-4)$ RS codes, 3 errors can be corrected when $n \geq 9$. In this case, the error-correction capability is more than those of the classical algorithms and the GSA. This method can be applied to decode FRS/IRS codes with rows codes being mediate and high rate RS codes. In addition, we apply this method to decode some cyclic codes over GF(8) and with restricted error values.

7.2 Future Work

The decoding of folded codes in this thesis is only a unidirectional corporation method. The performance of the folded codes studied in this thesis may be further improved by using an iterative decoding technique. In this technique, the erasure information may be used in an iterative fashion.

Codes constructed from expander graphs in [82] are asymptotical good codes. They can also be encoded and decoded in linear time. In addition, linear time encodable and decodable NMDS codes based on expander graphs are studied in [75]. These NMDS codes have RS codes as constituent codes and achieve a

good tradeoff between code rate and minimum distance. FRS codes discussed in this thesis have the same rate as the original RS codes. Also, row codes of an FRS code are GRS codes. Because of these features, it will be interesting to use FRS codes as constituent codes in expander codes.

Long burst errors frequently occur in wireless communications due to deep fading and other interferences in wireless channels. Also, burst errors occur in the storage channel because of the error propagation or dust and scratches on the media surface. The folded codes studied in this thesis can effectively correct long burst errors and therefore can be applied in such systems.

Bibliography

[1] A. Ahmed, R. Koetter, and N. R. Shanbhag, "VLSI architectures for soft-decision decoding of Reed-Solomon codes," In *ICC2004, Chicago, USA*, pp. 2584–2590, 2004.

[2] N. Alon, J. Bruck, J. Naor, M. Naor, and R. M. Roth, "Construction of asymptotically good low-rate error-correcting codes through pseudorandom graphs," *IEEE Trans. Inform. Theory*, vol. 38, pp. 509–516, Mar. 1992.

[3] Marc A. Armand and Jianwen Zhang, "Nearly MDS expander codes with reduced alphabet size," *submitted to IEEE Trans. Inform. Theory*, 2007.

[4] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory*, vol. 20, pp. 284–287, Mar. 1974.

[5] A. Barg and G. Zemor, "Error exponents of expander codes," *IEEE Trans. Inf. Theory*, vol 48, pp. 1725–1729, Jun. 2002.

[6] Claude Berrou, Alain Glavieux, and Punya Thitimajshima, "Near shannon limit error-correcting coding and decoding: Turbo-codes," In *Proceedings of IEEE International Communications Conference*, 1993.

[7] Richard. E. Blahut, *Algebraic Codes for Data Transmission*, Cambridge University Press, 2003.

[8] D. Bleichenbacher, A. Kiyayias, and M. Yung, "Decoding of interleaved Reed-Solomon codes over noisy data," In *Proceedings of ICALP 2003*, pp. 97–108, 2003.

[9] Andrew Brown, Lorenz Minder, and Amin Shokrollahi, "Probabilistic decoding of Interleaved RS-Codes on the Q-ary symmetric channel," In *ISIT2004, Chicago, USA*, pp. 326, 2004.

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

[11] Chen, Reed, Helleseth, and Truong, "General principles for the algebraic decoding of cyclic codes," *IEEE Trans. Inform. Theory*, vol. 40, pp. 1661–1663, Sep. 1994.

[12] Chen, Reed, Helleseth, and Truong, "Use of grobner bases to decode binary cyclic codes up to the true minimum distance," *IEEE Trans. Inform. Theory*, vol. 40, pp. 1654–1661, Sep. 1994.

[13] Kar Ming Cheng, "More on the decoder error probability for Reed-Solomon codes," *IEEE Trans. Inform. Theory*, vol. 35, pp. 895–900, Jul. 1989.

[14] Michael K. Cheng, Jorge Campello, and Paul H. Siegel, "Soft-decision Reed-Solomon decoding on partial response channels," *Global Telecommunications Conference, 2002*, vol. 2, pp. 1026–1030.

[15] Sae Young Chung, G. David Forney, Thomas J. Richardson, and Rdiger Urbanke, "On the design of low-density parity-check codes within 0.0045 db of the shannon limit," *IEEE Communications Letters*, vol. 5, pp. 58–60, Feb. 2001.

[16] M. C. Davey and D. MacKay, "Low-density parity check codes over GF(q)," *IEEE Communications Letters*, vol. 2, pp. 165–167, Jun. 1998.

[17] Ivana Djurdjevic, Jun Xu, Khaled Abdel-Ghaffar, and Shu Lin, "A class of low-density parity-check codes constructed based on Reed-Solomon codes with two information symbols," *IEEE Communications Letters*, vol. 7, pp. 317–319, Jul. 2003.

[18] M. El-Khamy, R. J. McEliece, and J. Harel, "Performance enhancements for algebraic soft decision decoding of Reed-Solomon codes," In *International Symposium on Information Theory*, pp. 419–419, 2004.

[19] Mostafa El-Khamy and Robert J. McEliece, "Iterative algebraic soft-decision list decoding of Reed-Solomon codes," *IEEE Journal on Selected Areas in Communications*, vol. 24, pp. 481–490, Mar. 2006.

[20] P. Elias, "Coding for noisy channels," *IRE Conv. Record part 4*, pp. 37–46, 1955.

[21] P. Elias, "Error-correcting codes for list decoding," *IEEE Trans. Inform. Theory*, vol. 37, pp. 5–12, Jan. 1991.

[22] Gui-Liang Feng and Kenneth K. Tzeng, "Decoding cyclic and BCH codes up to actual minimum distance using nonrecurrent syndrome dependence relations," *IEEE Trans. Inform. Theory*, vol. 37, pp. 1716–1723, Nov. 1991.

[23] Gui-Liang Feng and Kenneth K. Tzeng, "A generalization of the Berlekamp-Massey algorithm for multisequence shift-register synthesis with applications to decoding cyclic codes ," *IEEE Trans. Inform. Theory*, vol. 37, pp. 1274–1287, Sep. 1991.

[24] J. Fitzgerald and R. F. Lax, "Decoding affine variety codes using Gröbner Bases," *Designs, Codes and Cryptography*, vol. 13, issue 2, pp. 147–158,Feb. 1998.

[25] P. Fitzpatrick, "On the key equation," *IEEE Trans. Inform. Theory*, vol 41, pp. 1290–1302, Sep. 1995.

[26] G. D. Forney, "On decoding bch codes," *IEEE Trans. Inform. Theory*, vol. 11, pp. 549–557, Oct. 1965.

[27] G. D. Forney, "The Viterbi algorithm," *Proceedings of the IEEE*, vol 61, pp. 268–278, 1973.

[28] Jr. G. D. Forney, "Generalized minimum distance decoding," *IEEE Trans. Inform. Theory*, vol. 12, pp. 125–131, Apr. 1966.

[29] Robert G. Gallager, *Low-Density Parity-Check coding*. Ph.d thesis, 1963, MIT.

[30] W. J. Gross, F. R. Kschischang, and P. G. Gulak, "Architecture and implementation of an interpolation processor for soft-decision Reed-Solomon decoding," *IEEE Trans. on very Large Scale Integration (VLSI) Systems*, vol. 15, pp. 309–318, Mar. 2007.

[31] W. J. Gross, Frank R. Kschischang, Ralf Koetter, and P. Glenn Gulak, "Applications of algebraic soft-decision decoding of Reed-Solomon codes," *IEEE Trans. on Comms.*, vol. 54, pp. 1224–1234, Jul. 2006.

[32] V. Guruswami and P. Indyk, "Near-optimal linear-time codes for unique decoding and new list-decodable codes over smaller alphabets," In *Proc. 34th Annu. ACM Symp. Theory of Computing (STOC), Montreal, QC, Canada*, pp. 812–821, May 2002.

[33] V. Guruswami and M. Sudan, "Improved decoding of Reed-Solomon and algebraic-geometric codes," *IEEE Trans. Inform. Theory*, vol. 45, pp. 1757–1767, Sep. 1999.

[34] Venkatesan Guruswami and Alexander Vardy, "Maximum-likelihood decoding of Reed-Solomon codes is NP-hard," *IEEE Trans. Inform. Theory*, vol. 51, pp. 2249–2256, Jul. 2005.

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

[36] M. Hall Jr. *Combinatorial theory*. A Wiley-Interscience publication, 1986.

[37] R. W. Hamming, "Error detecting and error correcting codes," *The Bell System Technical Journal*, 29:147–160, 1950.

[38] C. R. P. Hartmann and K. K. Tzeng, "Generalizations of BCH bound," *Inform. Contr.*, vol. 20, pp. 489–498, Jun. 1972.

[39] Jing Jiang and Krishna R. Narayanan, "Iterative soft decision decoding of Reed Solomon codes based on adaptive parity check matrices," In *International Symposium on Information Theory*, 2005.

[40] Jing Jiang and Krishna R. Narayanan, "Iterative soft-input-soft-output decoding of Reed-Solomon codes by adapting the parity check matrix," *IEEE Trans. Inform. Theory*, vol. 52 , pp. 3746–3756, Aug. 2006.

[41] Sarah J. Johnson and Steven R. Weller, "Codes for iterative decoding from partial geometries," *IEEE Trans. on Comms*, vol. 52, pp. 236–243, Feb. 2004.

[42] P. Y. Kam. Lecture notes of digital communications. 2004.

[43] R. Koetter and A. Vardy, "Algebraic soft-decision decoding of Reed-Solomon codes," *IEEE Trans. Inform. Theory*, vol. 46, pp. 809–2825, Nov. 2003.

[44] Victor Y. Krachkovsky, "Reed-Solomon codes for correcting phased error bursts," *IEEE Trans. Inform. Theory*, vol. 49, pp. 2975–2984, Nov. 2003.

[45] Rudolf Lidl and Harald Niederreiter, *Introduction to finite fields and their applications*. Cambridge University Press, revised edition.

[46] Shu Lin and Daniel J. Costello, *Error Control Coding: Fundamentals and Applications*. Pearson Prentice Hall, 2 edition.

[47] Philippe Loustaunau and Eric V. York, On the decoding of cyclic codes using Gröbner Bases. *Applicable Algebra in Engineering, Communication and Computing*, vol. 8, issue 6, pp. 469–483, Dec. 1997.

[48] A. Lubotsky, R. Philips, and P. Sarnak, Ramanujan graphs. *Combinatorica*, vol. 8, pp. 261–277, 1988.

[49] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," In *International Symposium on Information Theory*, pp. 113, 1997.

[50] David J. C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Trans. Inform. Theory*, vol. 45, pp. 399–431, Mar. 1999.

[51] David J. C. MacKay and Radford M. Neal, "Near shannon limit performance of Low Density Parity Check codes," *Electronics Letters*, vol. 32, pp. 1645–1646, Aug. 1996.

[52] David J. C. MacKay and Radford M. Neal, Near shannon limit performance of Low Density Parity Check codes. *Electronics Letters*, vol. 33, pp. 457–458, Mar. 1997.

[53] F. J. MacWilliams and N. J. A. Sloane, *The theory of error correcting codes*. North-Holland, Amsterdam, 1977.

[54] G. A. Margulis, "Explicit group theoretical constructions of combinatorial schemes and their applications to the design of expanders and concentrators," *Probl. Inform. Transm.*, vol. 24, pp. 39–46, 1988.

[55] R. J. McEliece, "The Guruswam-Sudan decoding algorithm for Reed-Solomon codes," report in Caltech, Apr. 2003.

[56] R. J. McEliece, "On the average list size for the Guruswami-Sudan decoder," *7th. International Symposium on Communication Theory and Applications*, pp. 2–6, 2003.

[57] R. J. McEliece, D. J. C. MacKay, and J. F. Cheng, "Turbo decoding as an instance of pearls belief propagation algorithm," *IEEE Journal on Selected Areas in Comm.*, vol. 16, pp. 140–152, Feb. 1998.

[58] Robert J. McEliece, *Finite fields for computer scientists and engineers*. Kluwer Academic Publishers, Boston, 1987.

[59] D. J. Muder, "Minimal trellises for block codes," *IEEE Trans. Inform. Theory*, vol. 34, pp. 1049–1053, Sep. 1988.

[60] R. R. Nielsen, *Decoding AG-codes Beyond Half the Minimum Distance*. Ph.d thesis, Danmarks Tekniske Universitet, Aug. 1998.

[61] Henry O'Keeffe and Patrick Fitzpatrick, "Gröbner basis solutions of constrained interpolation problems," *Linear algebra and its Applications*, vol. 351, pp. 533–551, 2002.

[62] Emmanuela Orsinia and Massimiliano Sala, "Correcting errors and erasures via the syndrome variety," *Journal of Pure and Applied Algebra*, vol. 200, pp. 191–226, Feb. 2005.

[63] J. Pearl, "Reverend bayes on inference engines: A distributed hierarchical approach," In *Proc. Conf. Nat. Conf. AI, Pittsburgh, PA*, pp. 133–136, 1982.

[64] J. Pearl, "Fusion, propagation, and structuring in belief networks," *Artif. Intell.*, vol. 29, pp. 241–288, Sep. 1986.

[65] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo, CA: Morgan Kaufmann, 1988.

[66] Vishakan Ponnampalam and Branka Vucetic, "Soft decision decoding of Reed-Solomon codes," *IEEE Trans. on Comms.*, vol. 50, pp. 5–12, Nov. 2002.

[67] John G. Proakis, *Digital Communications, Fourth Edition*. McGraw-Hill.

[68] Ramesh Mahendra Pyndiah, "Near-optimum decoding of product codes: Block turbo codes," *IEEE Trans. on Comms.*, vol. 46, pp. 1003–1010, Aug. 1998.

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

[70] I. S. Reed and G. Solomon, "Polynomial codes over certain finite fields," *SIAM Journal on Applied Mathematics*, vol. 4, pp. 300–304, 1960.

[71] T. Richardson and R. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Trans. Inform. Theory*, vol. 47, pp. 599–618, Feb. 2001.

[72] Thomas J. Richardson, M. Amin Shokrollahi, and Rdiger L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. Inform. Theory*, vol. 47, pp. 619–637, Feb. 2001.

[73] Roth M. Ron, *Introduction to coding theory*. Cambridge, UK ; New York : Cambridge University Press, 2006.

[74] Ron M. Roth and Gitit Ruckenstein, "Efficient decoding of Reed-Solomon codes beyond half the minimum distance," *IEEE Trans. Inform. Theory*, vol. 49, pp. 246–257, Jan. 2000.

[75] Ron M. Roth and Vitaly Skachek, "Improved nearly-MDS expander codes," *IEEE Trans. Inform. Theory*, vol. 52, pp. 3650–3661, Aug. 2006.

[76] Georg Schmidt, Vladimir R. Sidorenko, and Martin Bossert, "Collaborative decoding of interleaved Reed-Solomon codes and concatenated code designs," Oct. 2006, available at http://arxiv.org/abs/cs/0610074.

[77] Jacob T. Schwartz, "Probabilistic algorithms for verification of polynomial identities (invited)," In *EUROSAM '79: Proceedings of the International Symposiumon on Symbolic and Algebraic Computation*, pp. 200–215, London, UK, 1979. Springer-Verlag.

[78] Claude E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol.27, pp. 379–423 and 623–656, Jul. and Oct. 1948.

[79] S. K. Shin and P. Sweeney, "Soft decision decoding of Reed-Solomon codes using trellis methods," *Electron. Lett.*, vol. 14, pp. 303–308, Oct. 1994.

[80] M. Sipser and D. A. Spielman, "Expander codes," *IEEE Trans. Inform. Theory*, vol. 42, pp. 1710–1722, Nov. 1996.

[81] V. Skachek and R. M. Roth, "Generalized minimum distance iterative decoding of expander codes," In *Proc. IEEE Information Theory Workshop (ITW)*, pp. 245–248, Mar. 2003.

[82] D. A. Spielman, "Linear-time encodable and decodable error-correcting codes," *IEEE Trans. Inform. Theory*, vol. 42, pp. 1723–1731, Nov. 1996.

[83] M. Sudan, "Decoding of Reed Solomon codes beyond the error-correction bound," *J. Complexity*, vol. 13, pp. 180–193, Mar. 1997.

[84] R. M. Tanner, "A recursive approach to low-complexity codes," *IEEE Trans. Inform. Theory*, vol. 27, pp. 533–547, Sep. 1981.

[85] Stephan ten Brink, "Convergence of iterative decoding," *Electronics Letters*, vol. 35, pp. 806–808, May 1999.

[86] Jean-Pierre Tignol, *Galois' Theory of Algebraic Equations*. World Scientific Publishing Company, 2001.

[87] E. Uhlemann, P.-A. Wiberg, T.M. Aulin, and L.R. Rasmussen, "Deadline dependent coding-a framework for wireless real-time communication,"" In *Real-Time Computing Systems and Applications, 2000. Proceedings. Seventh International Conference on*, pp. 135–142.

[88] E. Uhlemann, P.-A. Wiberg, T.M. Aulin, and L.R. Rasmussen, "Concatenated hybrid ARQ - a flexible scheme for wireless real-time communication," In *Real-Time and Embedded Technology and Applications Symposium, 2002. Proceedings. Eighth IEEE*, 2002.

[89] A. Vardy and Yair Be'ery, "Bit-Level Soft-Decision Decoding of Reed-Solomon codes," *IEEE Trans on Comms*, vol. 39, pp. 440–444, Mar. 1991.

[90] A. J. Viterbi, "Error bound for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Trans. Inform. Theory*, vol. 13, pp. 260–269, Apr. 1967.

[91] Stephen B. Wicker, *Error Control Systems for Digital Communication and Storage*. Prentice Hall, 1995.

[92] J. K. Wolf, "Efficient maximum likelihood decoding of linear block codes using a trellis," *IEEE Trans. Inform. Theory*, vol. 24, pp. 76–80, Jan. 1978.

[93] J. M. Wozencraft and B. Reiffen, *Sequential Decoding*. Cambridge, MA, MIT Press.

[94] Haitao Xia and J. R. Cruz, "Application of soft-decision Reed-Solomon decoding to magnetic recording channels," *IEEE Trans. on Magn.*, vol. 40, pp. 3419–3430, Sep. 2004.

[95] G. Zemor, "On expander codes," *IEEE Trans. Inform. Theory*, vol. 47, pp. 835–837, Feb. 2001.

[96] Jianwen Zhang and Marc A. Armand. On transformed folded shortened Reed-Solomon codes for the correction of phased bursts. In *The Fifth International Conference on Information, Communications and Signal Processing*, 2005.

List of Publications

Journal Papers (under review)

1. M. A. Armand and J. Zhang, "Nearly MDS expander codes with reduced alphabet size," *submitted to IEEE Transactions on Information Theory*.

Conference Papers (published)

1. J. Zhang and M. Armand, "Synthesis of multi-sequence having unknown elements in the middle with decoding applications," in *PIMRC2006, Sep. 11-14, Helsinki, Finland*.

2. J. Zhang and M. A. Armand, "On transformed folded shortened Reed-Solomon codes for the correction of phased bursts," in *The Fifth International Conference on Information, Communications and Signal Processing*, 2005.