ISSN 2959-6157

Enhancing Erasure Resilience in Reed-Solomon Codes: Theory and Applications in Data Storage Systems

Yuepeng Zhang

College of Mathematics and Physics, Shanghai University of Electric Power, Shanghai, China
Email: 18917970682@mail.shiep.edu.cn

Abstract:

Reed-Solomon (RS) codes are widely utilized in data storage and communication systems due to their robust error detection and correction capabilities. However, traditional RS codes encounter challenges in addressing the increasing data corruption rates demanded by modern storage systems. This paper aims to overcome these limitations by modifying and extending the structure of traditional RS codes, particularly enhancing their recovery capabilities in the face of significant data loss. We begin by reviewing the theoretical foundations of RS codes and existing extension methods and discussing emerging technologies integrated into RS codes. We then present detailed methodologies for RS code encoding, erasure recovery mechanisms, soft decision decoding, and interleaving coding techniques, followed by a series of experiments designed to test the proposed methods. The experimental results indicate that soft-decision decoding outperforms traditional hard-decision decoding under high signal-to-noise ratio conditions, albeit with increased computational complexity as the list of candidate codewords grows. Interleaved Reed-Solomon (IRS) coding offers improved performance under low signal-to-noise ratio conditions but may introduce additional system complexity due to the interleaving and deinterleaving processes. We conclude with a summary of the research findings, a discussion of the study’s limitations, and suggestions for future research directions.

Keywords: Reed-Solomon code; erasure recovery; data storage; extension.

1. Introduction

Reed-Solomon (RS) codes are a class of error-correcting codes that are widely used in data storage and communication systems. The fundamental concept behind RS codes involves representing data as polynomials over a finite field, from which parity symbols are generated. These parity symbols enable detecting and correcting errors that occur during data transmission or storage. RS codes are particularly effective in recovering erasures and correcting errors, making them invaluable in scenarios where portions of data may be lost or corrupted. Because of their reliability and efficiency, RS codes play a critical role in ensuring data integrity in modern digital communication systems, including applications such as optical disc storage, satellite communications, and QR codes. Due to their versatility, RS codes are also widely employed in the medical field for the reliable transmission and processing of medical images and biological data, as well as in the financial sector for secure financial data exchange and ensuring the integrity of blockchain information.

Despite their widespread use, traditional Reed-Solomon codes face challenges in meeting the increasing demands of modern data storage systems, especially under conditions of higher data corruption rates. As data volumes grow and storage requirements become more stringent, the limitations of RS codes in terms of error correction and recovery become more pronounced. Specifically, their recovery capabilities may be insufficient when faced with severe data loss, leading to potential data integrity issues. This research aims to address these limitations by modifying and extending the traditional RS code structure. The primary objective is to enhance the error correction capabilities of RS codes, ensuring reliable data recovery even under conditions of significant data loss. By exploring new methods and techniques, this study seeks to improve the resilience and effectiveness of RS codes in contemporary data storage environments.

This paper is organized into five main sections. Following this introduction, the Literature Review provides an overview of the theoretical foundations of RS codes, examines existing extension methods, and discusses emerging technologies that are being integrated into RS codes. The Methodology and Technical Model section presents the technical details of RS code encoding, erasure recovery mechanisms, soft decision decoding, and interleaving coding techniques. In the Experiments and Model Evaluation section, the design and results of experiments conducted to test the proposed methods are detailed. Finally, the Conclusion and Future Work section summarizes the findings, discusses the research limitations, and suggests potential directions for future research.

2. Literature Review

2.1 Theoretical Basis of Reed-Solomon Code

Reed-Solomon codes (RS) have been used in computers and other storage devices, and a lot of related research has been accumulated. RS codes, based on finite fields, use polynomial representation and generator matrices for effective error detection and correction in data transmission or storage [1]. The generator matrix for RS codes is constructed by evaluating a set of polynomials, typically over a Galois field, at distinct points corresponding to the codeword positions, ensuring that each row of the matrix represents a unique polynomial evaluated at those points, which forms the basis for encoding data into codewords [2]. Reed-Solomon encoding involves representing the input data as coefficients of a polynomial over a finite field, then evaluating this polynomial at different points within the field to generate a codeword, which consists of both the original data symbols and additional parity symbols that provide redundancy for error detection and correction during transmission or storage [3]. The basic principle of Reed-Solomon erasure recovery involves using the redundant parity symbols added during encoding to reconstruct the original data by identifying and correcting errors based on discrepancies in the received codeword [4]. This simple but efficient coding logic makes RS code easy to use and expand.

2.2 Existing Extension Methods of Reed-Solomon Code

To address the limitations of traditional RS codes, researchers have explored various methods to extend their error correction capabilities.

One common approach is to adjust the parameters (code length n and data symbol k) to add redundancy, thereby increasing the code's ability to correct more errors and erasures. For instance, in distributed storage systems like Google File System and Facebook's storage solutions (as shown in Table.1), RS codes are often customized to balance storage overhead with error correction strength [5]. Increasing redundancy in Reed-Solomon codes enhances error correction and erasure recovery capabilities but comes at the cost of increased storage or bandwidth requirements, higher computational complexity, and potential delays in real-time systems.

Table 1. RS code of different companies[5]

Storage system	Erasure code	# tolerable failures
Google File System II (Colossus)	RS(9,6)	Three
Quantcast File System	RS(9,6)	Three
Hadoop Distributed File System	RS(9,6)	Three
Yahoo Cloud Object Store	RS(11,8)	Three
Backblaze Vaults	RS(20,17)	Three
Facebook f4 Storage System	RS(14,10)	Four
Baidu Atlas Cloud Storage	RS(12,8)	Four

Other techniques include the use of hybrid coding schemes, where RS codes are combined with other error-correcting codes or integrated into more complex systems like RAID configurations [6]. This kind of concatenated coding can also achieve the effect of improving error recovery, but it also makes the coding process more complex and difficult to design and maintain.

Multilevel coding is also a great extension. It combines multiple coding schemes to enhance error correction by applying different codes to different data layers. For example, using LDPC as the outer code and Reed-Solomon as the inner code enhances error correction capabilities in challenging communication environments. This approach improves the system's resistance to both burst and random errors, making it suitable for high-reliability applications like UAV video transmissions [7]. It offers high error resilience and flexibility but can increase complexity and computational cost.

These extensions have shown promise in enhancing the reliability of RS codes in large-scale storage environments.

2.3 Application of emerging technologies in Reed-Solomon Code

In recent years, new technologies have been applied to RS codes to further improve their error correction capabilities.

One example is the locally repairable code (LRC). When LRC are combined with RS codes, they incorporate local repair constraints. These constraints allow some redundancy data to be used for repairing specific failed symbols by accessing only a small number of other symbols, rather than requiring access to the entire dataset [8]. This design is typically achieved through optimization algorithms that ensure minimal storage overhead while increasing redundancy. This combination enhances the system's robustness and efficiency, particularly in large-scale data storage systems, making it effective in handling multiple failures.

Another technique is soft decision decoding. Traditional RS codes use hard decision decoding, where each received symbol is simply classified as correct or incorrect. However, soft decision decoding leverages additional information from the received signal, such as signal-to-noise ratio or symbol confidence, to improve error correction performance [9]. Koetter-Vardy algorithm [10] and neural network-assisted decoding are the two most commonly used soft decision decoding in RS codes. It enhances error correction by utilizing additional information like confidence, leading to better performance in noisy environments.

Interleaving coding rearranges data sequences during encoding to reduce the impact of burst errors on the system. This is particularly useful in communication and data storage, especially when dealing with correlated failure. This technique, combined with RS codes, allows for more flexible management of different levels of error protection, which is crucial in high-speed data transmission and storage systems [11]. Interleaving coding in RS codes effectively mitigates the impact of burst errors but increases complexity and may require additional processing time and memory to handle the reordering and decoding processes.

These methods offer significant potential for improving the robustness of RS codes, particularly in environments where data loss occurs in clusters or bursts.

3. Methodology and Technical Model

3.1 Encoding and Erasing Recovery Mechanism

Reed-Solomon codes are block codes that are constructed by encoding data as polynomials over a finite field, typically $GF\left(2^m\right)$ or $GF\left(n\right)$, where m is an integer and n is a prime number. The generator matrix G is used to produce codewords by multiplying it with data vectors. The codeword, which includes both the original data and parity symbols, is transmitted or stored. The decoding process involves identifying and correcting errors or erasures by leveraging the algebraic structure of the code. The error correction capability of RS codes is determined by the minimum distance $d_{min}=n-k+1$, where n is the code length and k is the number of data symbols. RS codes are particularly effective in correcting both random errors and erasures, making them ideal for use in environments where data integrity is critical.

As shown in Fig.1, the diagram illustrates the basic process of Reed-Solomon encoding and decoding in a communication or storage system. The two blocks are abbreviated representations of the RS encoding and decoding operators. This flowchart runs from left to right. First, the data is transmitted from the data source to the Reed-Solomon encoder, where it is encoded with additional redundant information. Then, the encoded data passes through a communication channel or storage device, during which it may be affected by noise or errors. Next, the Reed-Solomon decoder receives the data and uses the redundancy to detect and correct any errors. Finally, the corrected data is sent to the data sink.

Fig 1. RS code recovery process[12]

Here is an example of using a (5,3)RS code in GF(7) to recover from two erasures. In general, we construct the generator matrix like this($a_1, a_2 \ldots a_n$ are distinct elements of your Galois field):

$$G = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ a_1 & a_2 & \cdots & a_n \\ a_1^2 & a_2^2 & \cdots & a_n^2 \\ \cdots & \cdots & \cdots & \cdots \\ a_1^{k-1} & a_2^{k-1} & \cdots & a_n^{k-1} \end{bmatrix} \tag{1}$$

As shown in Fig.2, the process of using the generator matrix to recover erasures (missing data) in a Reed-Solomon code. Here's a breakdown of the steps:

Step 1. Generator matrix G^*:

The generator matrix G^* is shown, where each element is calculated as powers of integers. Then it is simplified to the actual generator matrix using modulo 7 arithmetic.

Step 2. Message encoding:

A message vector m is multiplied by the generator matrix G to produce the codeword c. This codeword is what gets transmitted or stored.

Step 3. Received message with erasures:

Data corruption occurs during transmission or storage. The received message contains two erasures(missing symbols). To recover the missing data, a submatrix of G corresponding to the received positions is formed, and its inverse matrix is used.

Step 4. Recovery of the original input message:

The received vector is then multiplied by this inverse to recover the original message vector m. That's a successful recovery.

$$G^* = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 & 5 \\ 1^2 & 2^2 & 3^2 & 4^2 & 5^2 \end{bmatrix} mod7 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 & 5 \\ 1 & 4 & 2 & 2 & 4 \end{bmatrix}$$

example: $\mathrm{m} = [2 \quad 1 \quad 3]$ $\qquad$ $\mathrm{c} = \mathrm{m} \cdot \mathrm{G} = [6 \ 2 \ 4 \ 5 \ 5]$

received message: $[6 \ ? \ 4 \ ? \ 5]$

$$G = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 3 & 5 \\ 1 & 2 & 4 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & 6 & 1 \\ 4 & 5 & 5 \\ 3 & 3 & 1 \end{bmatrix}$$

$$m = cG^{-1} = [6 \quad 4 \quad 5] \begin{bmatrix} 1 & 1 & 1 \\ 1 & 3 & 5 \\ 1 & 2 & 4 \end{bmatrix}^{-1} mod7 = [2 \quad 1 \quad 3]$$

Fig 2. RS code erasure recovery

3.2 Soft Decision Decoding using Koetter-Vardy algorithm

The Koetter-Vardy(KV) algorithm is a significant advancement in the field of error correction, specifically for Reed-Solomon codes. It is a soft-decision decoding algorithm that improves upon the traditional hard-decision decoding methods.

Unlike hard-decision decoding, which uses binary decisions (correct or incorrect) based on received symbols, soft-decision decoding considers the reliability of the received symbols. This means it takes into account additional information like confidence levels or probabilities. The Koetter-Vardy algorithm seeks to maximize the likelihood of the received sequence being a valid codeword. It does this by considering the probability of each symbol and adjusting the decoding process accordingly. For example, one bit erasure happened in our codeword and we are in GF(7). Now we need to know every possibility of that position(in this case, it's an integer between 0 and 6) and calculate their probability respectively. It can provide us with decision basis, performance evaluation, and uncertainty management.

As shown in Fig.3, the process of the KV soft decision begins with assigning multiplicity values to each symbol based on its reliability, giving more weight to more reliable symbols(reliability information). The algorithm then constructs a bivariate polynomial through interpolation, which considers these multiplicities and the structure of the RS code. After finding the roots of this polynomial, the algorithm generates a list of candidate codewords and selects the most likely one based on the reliability information. This selected codeword is then output as the decoded message, offering improved accuracy in decoding.

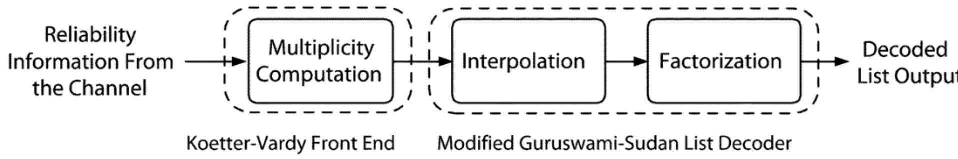

Fig 3. Block diagram of the KV soft-decision decoder[13]

The Koetter-Vardy algorithm is particularly useful in scenarios with high noise levels, such as satellite communication, deep-space communication, and certain storage systems where data reliability is critical. By integrating machine learning or data analysis with the KV algorithm for Reed-Solomon codes, we can enhance the accuracy of data recovery. Machine learning models can predict the likelihood of erasure positions based on communication channel features. These predictions, represented as probabilities, are then used to adjust the soft-decision inputs and weights in the Koetter-Vardy algorithm, leading to improved decoding performance and more accurate data recovery in challenging environments.

3.3 Interleave Coding

Interleaving coding is used in Reed-Solomon (RS) codes to combat burst errors by rearranging the order of data symbols before encoding. This process involves taking multiple blocks of data and interleaving them so that consecutive symbols in the output stream come from different blocks. If a burst error affects the transmitted data, the errors are spread out across multiple blocks when deinterleaving, making them easier for the RS decoder to correct. We know that it's impossible for a (5,3)RS code to recover from three erasures. However, if we put two 5-bit codewords together to form a new 10-bit codeword, it can recover from some specific combinations of three erasures and the recovery rate is about 83%. Here is an example, as shown in Fig.4, two 5-bit codewords c_1 and c_2 are merged to a new codeword. It is followed by two possible situations of three erasures, one is recoverable and the other is not.

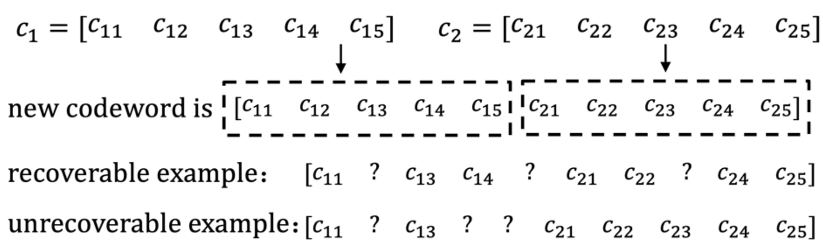

Fig 4. Merge codewords for better recovery capability

Now, as shown in Fig.5, if the two original codewords are no longer simply placed end to end, but in a new interlaced order like the following image(one from the first codeword, one from the second codeword…), the recovery rate can be further improved, especially when erasures occur in bursts. This is the basic concept of interleave coding.

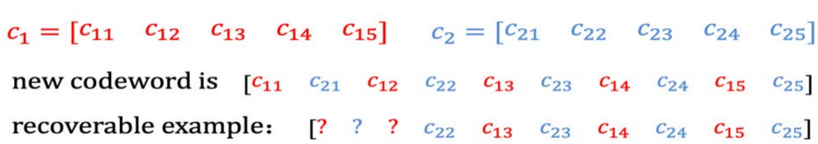

Fig 5. Interleave coding

As shown in Fig.6, the diagram illustrates an Interleaved Reed-Solomon(IRS) Code. "N" is The total number of symbols after encoding. "Information symbols" are the original data after encoding. "Redundancy symbols" are additional symbols for error detection and correction. "K(1), K(2), K(3), K(l)" are different levels of redundancy or encoding steps, where K indicates the degree of the generator polynomial. Burst error is a type of error where multiple errors occur consecutively, which the code is designed to handle effectively.

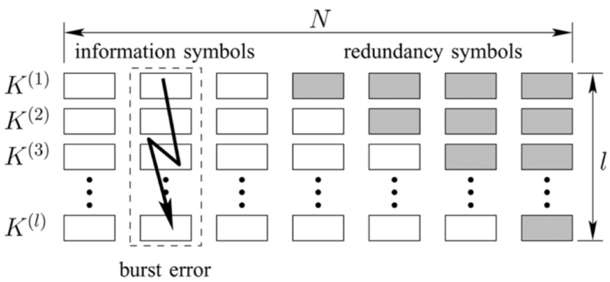

Fig 6. Interleaved Reed-Solomon code[11]

Now, the decoder can correct up to $1-\frac{2k+n}{3n}$ [14]fraction of random errors[15]. This technique enhances the error correction performance of RS codes, particularly in channels where errors tend to occur in bursts.

4. Experiments and Model Evaluation

4.1 Experiment with constant n and increasing t

First, we can simulate the effect on bit error rate (BER) performance with increasing error correction capability t ($t=\frac{n-k}{2}$) for a fixed value of n.

As shown in Fig.7, the graph presents the results of the experiment investigating the BER performance of RS codes over an AWGN (additive white Gaussian noise) channel using 32-ary FSK(Frequency-Shift Keying) modulation with a constant block length ($n=31$) and varying error correction capabilities t. The horizontal axis represents the signal-to-noise ratio(SNR), E_b/N_0, measured in decibels (dB). Here, E_b stands for the energy per bit, and N_0 represents the noise power spectral density. Moving from left to right on the x-axis indicates an increasing SNR, meaning the signal strength relative to the noise is improving as E_b/N_0 increases. The vertical axis represents the BER, which is the proportion of bits received with errors out of the total number of bits transmitted. The y-axis is displayed on a logarithmic scale, ranging from 10^{-8} to 10^{0}. A lower BER indicates better system performance with fewer bit errors. In summary, each curve in the graph shows how the BER varies with changes in E_b/N_0 for different configurations of RS codes.

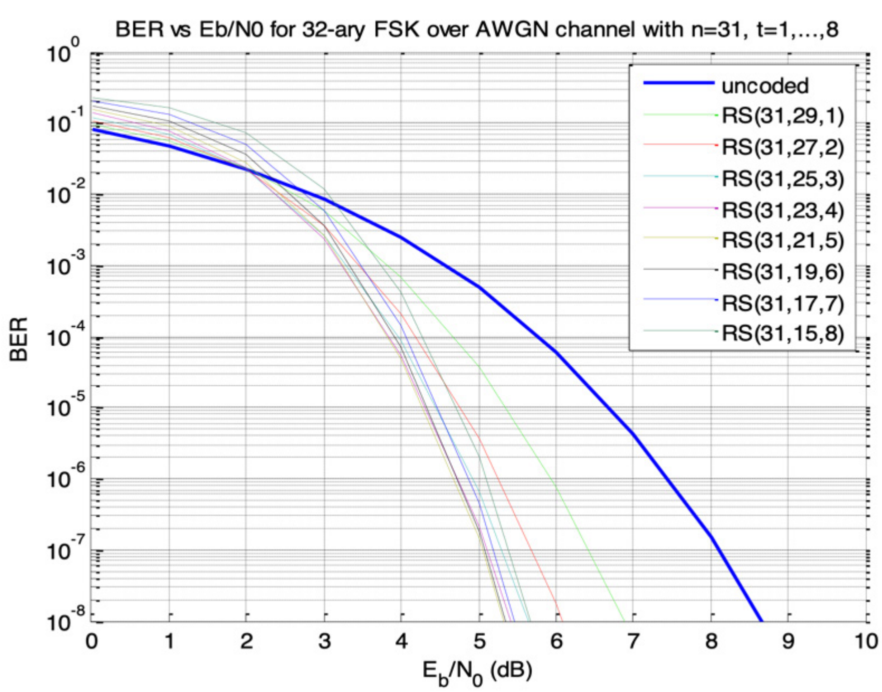

Fig 7. RS error performance for MFSK over AWGN channel with constant *n* and increasing *t* [16]

It was observed that increasing t from 1 to 5 improves BER performance, but further increases in t degrade performance due to the reduced code rate and higher computational complexity. RS(31, 21) with $t=5$ and a code rate of 0.68 was identified as the optimal configuration, offering a coding gain of approximately 3.3 dB. This balance between error correction capability and computational power is crucial for optimal performance.

4.2 Experiment with constant k and increasing n - k

Then we can design an experiment with constant message length k and increasing *n¨Ck*.

As shown in Fig.8, the graph illustrates the impact of increasing redundancy (*n¨Ck*) while keeping the number of data symbols constant ($k = 29$) on the BER performance of RS codes with 32-FSK modulation over an AWGN channel. This graph uses the same horizontal and vertical coordinates as the previous one. The results indicate that as the codeword length (n) increases, the error correction capability of RS coding is enhanced, achieving a lower BER at the same SNR. However, the code rate (k / n) decreases with increased redundancy, which may affect the system's throughput.

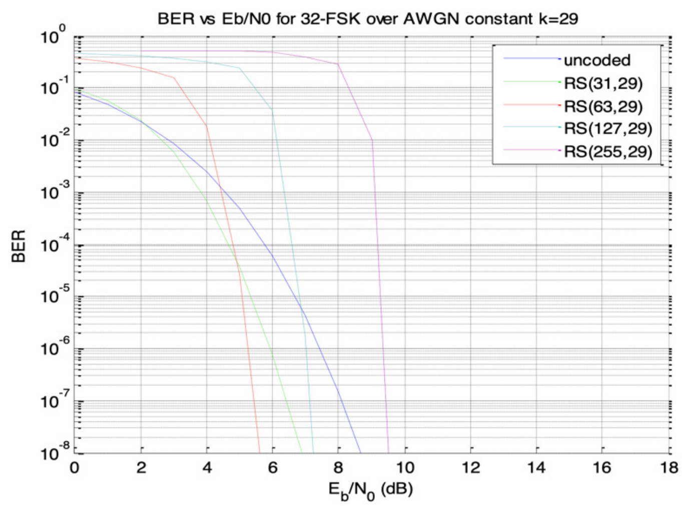

Fig 8. RS error performance for MFSK over AWGN channel with constant k, increasing $n - k$ [16]

4.3 Experiment of soft decision decoding in RS code

As shown in Fig.9, the figure illustrates the codeword error rate (CER) versus signal-to-noise ratio (SNR) for different decoding algorithms applied to a (225, 144 122) Reed-Solomon code with 256-QAM(quadrature amplitude modulation) over an AWGN channel. The horizontal axis represents the SNR, and the vertical axis represents the CER. L denotes the list size in soft-decision decoding, indicating the number of candidate codewords considered during decoding(When L is small, the decoder considers only a few candidate codewords, which may lead to a higher error rate because the range of possibilities is too narrow. As L increases, the decoder can consider more candidate codewords, making it more likely to find the correct codeword, and the error rate decreases accordingly. When $L = \infty$, the decoder theoretically considers all possible candidate codewords. In this case, the soft-decision decoder performs the best, achieving the lowest error rate.). The experimental results show that as L increases, the performance of soft-decision decoding gradually improves, especially at higher SNRs, significantly reducing the error rate and outperforming traditional hard-decision decoding algorithms like Berlekamp-Welch and Guruswami-Sudan.

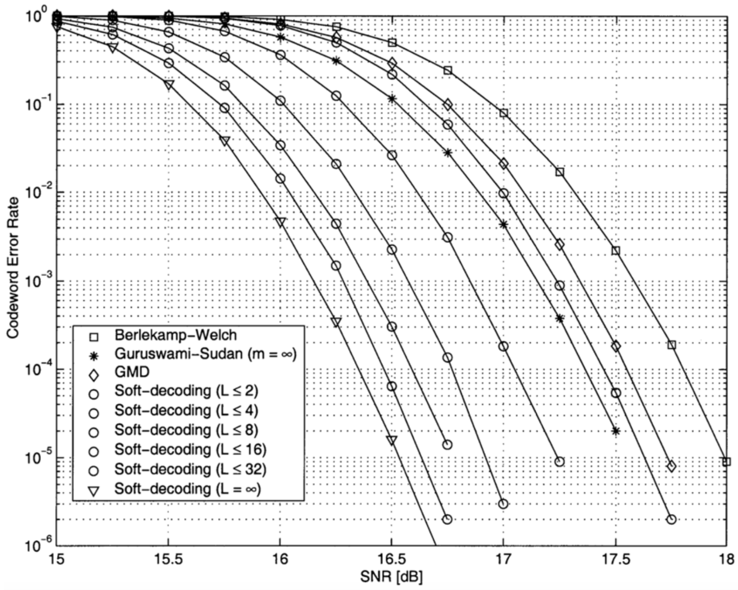

Fig 9. Performance of algebraic soft-decision decoding for the (255, 144, 112) Reed–Solomon code with 256-QAM modulation on an AWGN channel [10]

4.4 Experiment of Interleaved Reed-Solomon(IRS) code

As shown in Fig.10, the experiment displays the decoding performance of the concatenated code on an additive white Gaussian noise (AWGN) channel using binary phase shift keying (BPSK) modulation. It is a concatenated code composed of an outer RS code and an inner rate 1/2 memory 6 tail-biting convolutional code. The following graph compares the performance of independent RS decoding and collaborative interleaved Reed-Solomon(IRS) decoding. The horizontal axis represents the signal-to-noise ratio(SNR), E_b / N_0. The vertical axis represents the word error rate (WER), which is the probability of decoding errors occurring during the process, shown in logarithmic form. From the graph, we can see the vertical axis ranges from 10^0 (100% error rate) to 10^{-6} (an error rate of one in a million). The upper bound curves provide an expectation of performance in an ideal situation, while the actual decoding performance curves show what can be achieved in real decoding processes. The graph shows that collaborative IRS decoding outperforms independent RS decoding, indicating that decoding the entire IRS code as one unit can improve decoding performance. Randomized collaborative IRS decoding may provide performance improvements or simplify the decoding process in certain situations compared to non-randomized decoding.

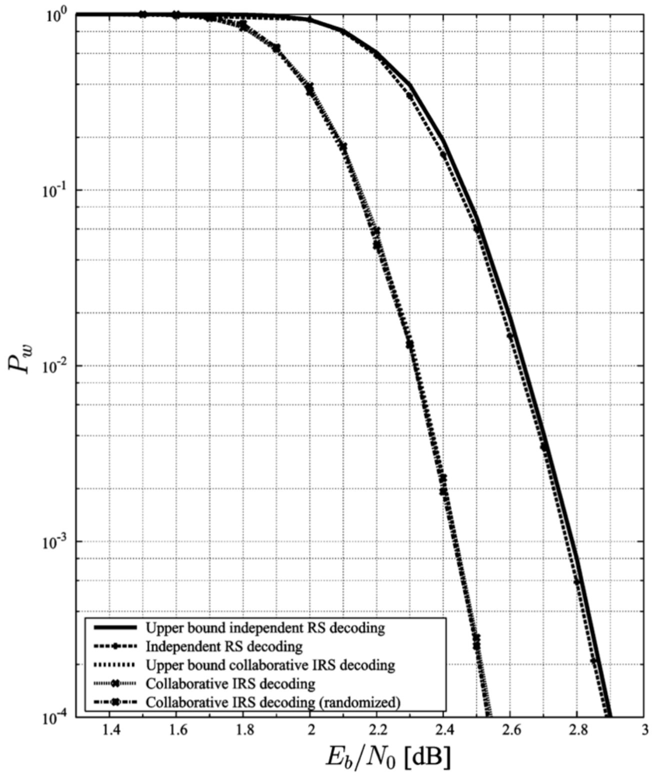

Fig 10. Simulated decoding performance of the concatenated code composed of three codewords from the outer code ($2^8;255,233,33$), and an inner rate 1/2 memory 6 tail-biting convolutional code, AWGN channel, BPSK modulation[11]

5. Conclusion

5.1 Summary

In the series of experiments conducted in the paper, we analyzed the error-correcting capability of RS codes in data storage systems and enhanced its erasure resilience through various extension methods. Soft-decision decoding demonstrated better performance than traditional hard-decision decoding under high SNR conditions, but the computational complexity increases with the enlargement of the candidate codeword list. IRS coding provided better performance under low SNR conditions but may increase system complexity due to the interleaving and deinterleaving processes. Considering both the recovery rate and computational complexity, soft-decision decoding is suitable for high SNR environments, while IRS coding is more effective under low SNR conditions. The optimal method should be chosen based on the specific application scenario and system requirements.

5.2 Limitation

The research presented in this paper has made strides in exploring the enhancement of erasure resilience in Reed-Solomon codes for data storage systems, yet it faces several challenges and limitations. Firstly, while soft-decision decoding demonstrates superior performance under high SNR conditions, this improvement comes at the cost of increased computational complexity. As the list of candidate codewords expands, the demand for computational resources and time grows, potentially impacting the application in real-time systems. Secondly, IRS coding, though effective in improving the recovery rate under low SNR conditions, adds complexity to the system due to the interleaving and deinterleaving processes, which may lead to processing delays and affect the overall throughput of the system.

Moreover, the experiments were primarily conducted in a simulated environment and based on theoretical models, which may not fully account for the complexities encountered in practical applications, such as hardware limitations, characteristics of transmission media, and interference in multi-user environments. Therefore, the generalizability of the experimental results and their effectiveness in real-world applications require further validation in future studies.

5.3 Future Prospects

The future research directions stemming from this study are poised to address and transcend the current limitations by embracing innovative approaches. A pivotal area of focus will be the integration of artificial intelligence to enhance erasure prediction and recovery mechanisms, tailoring the response to the nuanced patterns of data loss. Additionally, there's a drive towards the invention of novel coding structures that are adept at countering the specific challenges present in various data loss scenarios, potentially incorporating principles from graph theory or quantum error correction. The integration of these techniques with existing RS codes could pave the way for the next generation of error-correcting codes.

Furthermore, the translation of these theoretical advancements into tangible outcomes is emphasized through rigorous testing within real-world contexts, ensuring the proposed solutions are robust across diverse storage media and network conditions. There is also a recognition of the need to meticulously balance the computational complexity with the performance gains, to ensure that the decoding processes remain efficient without being prohibitively resource-intensive. Lastly, fostering interdisciplinary collaboration will be key to harnessing a diverse array of expertise, paving the way for breakthroughs that can significantly bolster the reliability and integrity of data storage systems.

References

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

[2] W. A. Geisel, "A tutorial on Reed-Solomon error correction coding," NASA Technical Memorandum 102162, Aug. 1990. [Online]. Available: https://ntrs.nasa.gov/citations/19900019023.

[3] Stephen B. Wicker; Vijay K. Bhargava, "An Introduction to Reed-Solomon Codes," in Reed-Solomon Codes and Their Applications, IEEE, 1994, pp.1-16, doi: 10.1109/9780470546345.ch1.

[4] R. Con, A. Shpilka and I. Tamo, "Optimal Two-Dimensional Reed–Solomon Codes Correcting Insertions and Deletions," in *IEEE Transactions on Information Theory*, vol. 70, no. 7, pp. 5012-5016, July 2024, doi: 10.1109/TIT.2024.3387848.

[5] H. Dau, I. M. Duursma, H. M. Kiah and O. Milenkovic, "Repairing Reed-Solomon Codes With Multiple Erasures," in *IEEE Transactions on Information Theory*, vol. 64, no. 10, pp. 6567-6582, Oct. 2018, doi: 10.1109/TIT.2018.2827942.

[6] S. -J. Lin, A. Alloum and T. Y. Al-Naffouri, "RAID-6 reed-solomon codes with asymptotically optimal arithmetic complexities," *2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, Valencia, Spain, 2016, pp. 1-5, doi: 10.1109/PIMRC.2016.7794681.

[7] Dong, P.; Xiang, X.; Liang, Y.; Wang, P. A Block-Based Concatenated LDPC-RS Code for UAV-to-Ground SC-FDE Communication Systems. Electronics 2023, 12, 3143. https://doi.org/10.3390/electronics12143143

[8] Lin-Zhi SHEN, Yu-Jie WANG, Optimal (r, δ)-Locally Repairable Codes from Reed-Solomon Codes, IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, Article ID 2023EAL2026, Advance online publication May 30, 2023, Online ISSN 1745-1337, Print ISSN 0916-8508, https://doi.org/10.1587/transfun.2023EAL2026

[9] J. Justesen, "Soft-decision decoding of RS codes," *Proceedings. International Symposium on Information Theory, 2005. ISIT 2005.*, Adelaide, SA, Australia, 2005, pp. 1183-1185, doi: 10.1109/ISIT.2005.1523528.

[10] R. Koetter and A. Vardy, "Algebraic soft-decision decoding of Reed-Solomon codes," in *IEEE Transactions on Information Theory*, vol. 49, no. 11, pp. 2809-2825, Nov. 2003, doi: 10.1109/TIT.2003.819332.

[11] G. Schmidt, V. R. Sidorenko and M. Bossert, "Collaborative Decoding of Interleaved Reed–Solomon Codes and Concatenated Code Designs," in *IEEE Transactions on Information Theory*, vol. 55, no. 7, pp. 2991-3012, July 2009, doi: 10.1109/TIT.2009.2021308.

[12] Martyn Riley and Iain Richardson, "An introduction to Reed-Solomon codes: principles, architecture and implementation", http://www.cs.cmu.edu/~guyb/realworld/reedsolomon/reed_solomon_codes.html.

[13] X. Zhang, "Reduced Complexity Interpolation Architecture for Soft-Decision Reed–Solomon Decoding," in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 14, no. 10, pp. 1156-1161, Oct. 2006, doi: 10.1109/TVLSI.2006.884177.

[14] D. Bleichenbacher, A. Kiayias, and M. Yung, "Decoding interleaved Reed–Solomon codes over noisy channels", Theoretical Computer Science 379, 348 (2007) https://doi.org/10.1016/j.tcs.2007.02.043.

[15] "Interleaved RS (IRS) code", The Error Correction Zoo (V. V. Albert & P. Faist, eds.), 2022. https://errorcorrectionzoo.org/c/interleaved_reed_solomon.

[16] Ma, Ruiping & Xing, Liudong & Wang, Yujie. (2019). Performance Analysis of Reed-Solomon Codes for Effective Use in Survivable Wireless Sensor Networks. International Journal of Mathematical, Engineering and Management Sciences. 5. 13-28. 10.33889/IJMEMS.2020.5.1.002.