

Resilience of Reed-Solomon Codes Against Single-Frequency Electromagnetic Disturbances: Fault Elimination Through Encoder Tuning

Pejman Memar

Department of Computer Science
KU Leuven, Bruges Campus
Bruges, Belgium
pejman.memar@kuleuven.be

Jens Vankeirsbilck

Department of Computer Science
KU Leuven, Bruges Campus
Bruges, Belgium
jens.vankeirsbilck@kuleuven.be

Dries Vanoost

Department of Electrical Engineering
KU Leuven, Bruges Campus
Bruges, Belgium
dries.vanoost@kuleuven.be

Tom Holvoet

Department of Computer Science
KU Leuven
Leuven, Belgium
tom.holvoet@kuleuven.be

Jeroen Boydens

Department of Computer Science
KU Leuven, Bruges Campus
Bruges, Belgium
jeroen.boydens@kuleuven.be

Abstract—In increasingly electromagnetic-polluted environments, communication networks are becoming more vulnerable. Even networks equipped with error control techniques suffer from this problem. Electromagnetic disturbances can result in corrupted data which are undetectable by error control techniques. Such scenarios are extremely dangerous as the system is unaware of the corruption. This could lead to critical failures. Thus, protecting communication networks against this type of undetected corrupted data is of the utmost importance. In this regard, this paper presents an effective fault elimination approach through encoder tuning. This technique enhances the resiliency of a well-known forward error correction code, known as primitive Reed-Solomon Codes, against steady-state single-frequency electromagnetic disturbances. It is found that this approach outperforms the previously proposed multi-layer inversion-based fault elimination approach in mitigating undetected corrupted data. Furthermore, it is shown that encoder tuning has two main implementation advantages over our previous approach. First, it does not require an extra layer to perform fault elimination. Second, it eliminates the overhead of performing double syndrome calculation at the consumer side.

Index Terms—Communication channel, electromagnetic disturbance, Reed-Solomon codes, resilience, encoder tuning.

I. INTRODUCTION

THE Internet of Things (IoT) will be the backbone of future intelligent societies and industries. It is estimated that there will be around 32 IoT devices per person by 2030 [1]. This indicates that its market size could grow at an annual rate of 20%, reaching \$11.6 Trillion by 2030. Undoubtedly, safety-critical and mission-critical technologies such as autonomous vehicles, surgical robots, and smart factories, will have a

The research leading to these results has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie Grant Agreement No 812.790 (MSCA-ETN PETER). This publication reflects only the authors' view, exempting the European Union from any liability. Project website: <http://etn-peter.eu/>.

significant share in this huge market.

This fast-growing market is a true indication of rising electrical, electronic and programmable electronic (E/E/PE) devices. Furthermore, certain design characteristics of these modern E/E/PE devices, including smaller feature size and voltage levels, make them more vulnerable to electromagnetic disturbances (EMD) [2]. Consequently, these E/E/PE devices could lead to more extreme electromagnetic (EM) environments. Correspondingly, communication networks in these E/E/PE devices could be adversely affected. In an EM-polluted environment, EMD induces additional voltages onto the communication channel which can result in bit-flips in the transmitted data.

Error Control Techniques (ECT) have been developed and employed to protect communication networks against a limited number of random bit-flips [3]. Forward Error Correction (FEC) is a subgroup of ECT that is commonly used in a lower-layer communication. FEC can recover data without asking for retransmission when a limited number of errors are introduced. Correspondingly, FEC adds extra information to the data words at the data producer side and generates a dictionary of valid code words. This information is later used at the time of decoding to detect and correct the corrupted data.

FEC, however, has a major vulnerability that results in undetected corrupted data (UCD). This happens when a code word gets corrupted in such a way that it turns into another valid code word. In such a scenario, the FEC is unaware of the corruption and the decoder assumes that the received data is correct. From the safety viewpoint, this could lead to critical failures and in extreme cases, harm to users, bystanders, and the environments. Therefore, it is vital to reduce the number of UCD to a level as low as reasonably practicable in safety

or mission critical applications.

In our previous studies, the effectiveness of a well-known FEC, known as primitive Reed-Solomon Codes (RS Codes), was investigated against steady-state single-frequency EMD by means of our in-house simulation framework [4], [5]. It was found that the majority of UCD are caused by one-symbol-value code words (i.e. all symbols in a code word are identical). In accordance with the favorable impact of the over-voltage detection (OVD) mechanism on the Hamming and the Triplication codes [6], the impact of this mechanism was assessed on primitive RS Codes [5]. Our simulations showed that by choosing an appropriate voltage range, OVD could substantially decrease the number of UCD. However, this is obtained at a cost: decreasing the availability of data transmission. To alleviate this trade-off, a multi-layer inversion-based fault elimination technique was introduced. Compared to the OVD mechanism, this technique provides a better resiliency and availability against single-frequency EMD. However, this extra layer increases the overhead of the considered networks. This study, therefore, utilizes previous findings to mitigate this overhead and to improve the EM-resiliency of primitive RS Codes even further through encoder tuning. This study proves that by choosing a specific initial root for the generator polynomial, RS Codes would generate dictionaries of code words which are more resilient against the single-frequency EMD.

The remainder of this paper is organized as follows. Section II explains the theory behind the primitive RS Codes. The experimental setup is covered in Section III. Section IV details the enhanced implementation of multi-layer structure, and describes an effective fault elimination technique through encoder tuning. Finally, conclusions are stated in Section V.

II. REED SOLOMON CODES

Reed-Solomon codes (RS Codes) are linear block-based FEC and are a subclass of Bose–Chaudhuri–Hocquenghem (BCH) codes. A primitive error correction RS Code (n, k) with a block length of n symbols and a data word length of k symbols is defined over a finite field with $q = P^r$ symbols, where $n = q - 1$. According to the finite field definitions, P is always a prime number and r , which represents the symbol size, is a positive integer (\mathbb{Z}^+) [7], [8]. Fig. 1 presents the structure of the generated code word by RS Codes. It should be noted that finite fields with base 2 (i.e. $P = 2$) will be considered within this paper as the code words are transmitted in the form of binary sequences.

Primitive error correction RS Codes are forward error correction codes which have the capability to correct s symbol errors, and to detect up to $2s$ symbol errors, where $n - k = 2s$. In this regard, RS Codes add $2s$ symbols as parity to the data word and generate a code word. Through this step, RS Codes generate a dictionary of q^k valid code words with the Hamming distance of $n - k + 1$ bits. Correspondingly, RS Codes take advantage of these parity symbols to detect and correct the possibly corrupted code words.

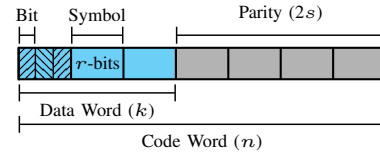


Figure 1: The structure of the RS Codes' code word

A. Reed Solomon Codes Encoder

At this step, the generator polynomial $g(x)$ adds the parity information to the input data words and generates a dictionary. As indicated in Equation (1), the generator polynomial $g(x)$ is constructed based on the cyclic characteristic of the multiplicative group of finite field elements [8]. Within this paper, α is used as the primitive element of this field. This indicates that each non-zero element of this finite field is describable in the form of α^i with $i \in \mathbb{Z}^+$.

$$g(x) = \prod_{i=f_{cr}}^{f_{cr}+2s-1} (x - \alpha^i) \quad (1)$$

Here, f_{cr} indicates the power of the first consecutive root of the considered finite field. As shown in Equation (2), systematic coding is employed to encode the data words. This coding scheme is used because it embeds the data word within the generated code word. Therefore, in case RS Codes detect that the received code word is uncorrupted, the data word can be easily stripped from the code word without further unnecessary decoding steps which normally happen in the non-systematic coding approach.

$$c(x) = x^{2s} \cdot m(x) - [x^{2s} \cdot m(x) \text{ mod } g(x)] \quad (2)$$

Here, $m(x)$ and $c(x)$ present the data word and the code word polynomials, respectively. Additionally, the factor x^{2s} is used to prevent the overlapping between the data word and the parity information by shifting the data word to a higher order.









B. Reed Solomon Codes Decoder

Upon receiving the code word at the consumer side, the RS Codes decoder calculates the syndrome components as the first step. This detection step determines whether the received code word is valid.

In case of a zero syndrome, the received code word is valid, and no further step is required. However, this scenario could happen either when the original data is received or when the data consumer receives a code word that is corrupted, but still valid. This happens when the corruption is in such a way that it turns a code word to another valid code word. Such a scenario is undetectable and is detrimental to the overall system safety. In both cases, the data word is eventually stripped from the code word.

A nonzero syndrome, however, implies the presence of faults, and it demands further steps to correct the code word. Note that this paper follows the same correction steps employed in [4].

Table I: An overview of the considered categories.

Category	Channel Status	Data Status	Detector Status	Label	
Data True Positive (DTP)	Data In Control	Uncorrupted	No Warning	Green	
Data True Negative (DTN)	Data In Control	Corrupted	Warning	Shade of Orange	
Data False Positive (DFP)	Data In Control	Uncorrupted	Warning	Shade of Orange	
Data False Negative (DFN)	Data In Control	Corrupted	No Warning	Shade of Red	
Channel True Positive (CTP)	Channel In Control	Uncorrupted	No Warning	Shade of Red	
Channel True Negative (CTN)	Channel In Control	Corrupted	Warning	Shade of Orange	
Channel False Positive (CFP)	Channel In Control	Uncorrupted	Warning	Shade of Orange	
Channel False Negative (CFN)	Channel In Control	Corrupted	No Warning	Shade of Red	

III. EXPERIMENTAL SETUP

This paper has employed the same fault model and experimental setups which were previously used in [4], [5]. Additionally, the proposed condition assessment definitions by Claeys et al. are employed within this paper to analyze the effectiveness of RS Codes under the considered setup [9]. Based upon this category system, categories are generated from the following three fundamental questions:

- 1) Is the output data word correct? *Positive or Negative*
- 2) Is the detection outcome correct after considering the output of the previous question? *True or False*
- 3) Is the data in control or the channel in control? *Data or Channel*

In accordance with the different outputs of the aforesaid questions, eight distinct categories are considered. These categories are presented in Table I. As can be seen, all categories in shades of orange produce a warning, while categories in shades of red receive no warning. The latter is the focus of this paper as the system is unaware of the corruption. Consequently, such scenarios could lead to critical failures. Hence, it is essential to mitigate these specific categories to a level that is as low as reasonably practicable. Note that based upon the initial assumption (i.e., the data producer and the data consumer are assumed to be protected from any EMD or internal hacking), all categories except DFN could happen under the considered setup.

IV. FAULT ELIMINATION

As it was found in [4], the main vulnerability of RS Codes to CTP and CFN is due to the occurrence of one-symbol-value code words at specific frequencies. It is shown that the disturbance frequencies which lead to these undetected corrupted data (UCD) are directly dependent to the symbol size (r) and the channel bit-rate (f_{bit}) as follows:

$$f_{\text{UCD}} = \frac{j}{r} \cdot f_{\text{bit}}, \quad j \in \mathbb{N}^+ \quad (3)$$

Mitigating this specific type of vulnerability would substantially enhance the resiliency of RS Codes against single-frequency EMD. The previous study showed that an inversion layer, through inverting one symbol in each code word, effectively reduces the occurrence of CTP and CFN. For further information, the reader is referred to [10].

The aforesaid fault elimination technique can be implemented via two approaches. The first approach is to store a dictionary at the consumer side to check whether the received code

word is valid at the inversion layer. Despite its simplicity, this approach is inefficient and almost impractical due to the following limitations:

- 1) Storing the generated dictionary by RS Codes at the consumer side could become challenging or sometimes even impossible as the number of code words (i.e., q^k) exponentially increases for larger r and k values. The required amount of memory can be calculated as follows. Given that each symbol consists of r bits and each code word has n symbols, then each code word contains $n \times r$ bits of information. In this regard, for a dictionary with q^k code words, the required amount of memory is $(n \times r) \times q^k$ bits. For instance, for an RS Code with $r = 8$ (i.e., 1 byte of information per symbol) and $k = 3$, the required space to store the dictionary is about 4GB.
- 2) The inversion layer also results in a lower transmission rate for big dictionaries as each received code word must be checked against the whole dictionary at the consumer side prior to the decoding step.

The subsequent section provides a more efficient implementation to alleviate these limitations.

A. Inversion Layer with Double Syndrome Calculation

As mentioned in Section II-B, the syndrome components can turn to zero under two scenarios, i.e., when the transmitted code word is uncorrupted or when the transmitted code word gets corrupted but in a way that it turns into another valid code word. In other words, when the decoder encounters a valid code word based on the original dictionary, the calculated syndrome will be zero. Therefore, rather than comparing the received code words at the consumer side with a stored dictionary, it is possible to only calculate the syndrome at this stage, i.e., only the detection step of the RS Codes decoder. In case of a zero syndrome, the inversion layer must switch to a minimum risk state as it is not possible to receive a valid code word after the inversion process. A non-zero syndrome, however, indicates that the received code word is not valid and the inversion must be reverted. Following this step, the code word is transmitted to the decoder for both error detection and correction to obtain the output data word. As can be seen, through this approach, RS Codes calculate the syndrome two times which increases the overhead of the decoding process. This adverse impact, however, can be reduced by employing a transmission buffer at the consumer side. The block diagram of the proposed implementation is presented in Fig. 2.

In the following section, an effective approach through encoder tuning is proposed. This approach eliminates the need of employing an extra layer for fault elimination. Furthermore, it eliminates the mentioned overhead of performing double syndrome calculation at the consumer side.

B. Encoder Tuning

As mentioned earlier, the generator polynomial, $g(x)$, is responsible for generating a dictionary by encoding the data words. Different initial roots of $g(x)$, which is determined by

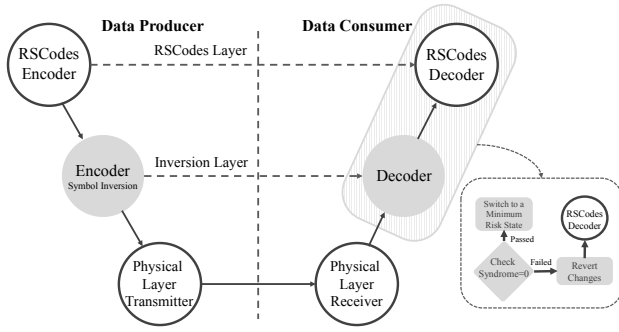


Figure 2: An overview of the proposed implementation of the inversion layer with double syndrome calculations.

the power of the first consecutive root (i.e., $pfcr$), can result in different dictionaries. Using the contradiction approach, it is possible to indirectly prove that when $pfcr = 1$ (i.e., the first root is α), RS Codes generate a dictionary containing all possible one-symbol-value code words, and when $pfcr = 0$ (i.e., the first root is $\alpha^0 = 1$), RS Codes produce a dictionary free of one-symbol-value code words except for the all 0s code word. The latter would reduce the ratio of CTP and CFN significantly as the generated one-symbol-value code words get limited to all 0s code word. Correspondingly, the subsequent lemmas are required as the foundation of these proofs.

Lemma IV-B.1. *All roots of $g(x)$ must be valid roots for $c(x)$ as every $c(x)$ is a multiple of $g(x)$.*

Proof: According to Section II-A, code words can be systematically generated as follows:

$$g(x) = \prod_{i=pfcr}^{pfcr+2s-1} (x - \alpha^i)$$

$$c(x) = x^{2s} \cdot m(x) - [x^{2s} \cdot m(x) \bmod g(x)]$$

Let,

$$R(x) = x^{2s} \cdot m(x) \bmod g(x)$$

Where $R(x)$ is the remainder of $\frac{x^{2s} \cdot m(x)}{g(x)}$, and,

$$x^{2s} \cdot m(x) = Q(x) \cdot g(x) + R(x)$$

Where $Q(x)$ is the quotient of $\frac{x^{2s} \cdot m(x)}{g(x)}$. Accordingly,

$$c(x) = x^{2s} \cdot m(x) - R(x)$$

$$= [Q(x) \cdot g(x) + R(x)] - R(x)$$

$$= Q(x) \cdot g(x)$$

Therefore, $c(x)$ is a multiple of $g(x)$. \square

Lemma IV-B.2. *The outputs of Exclusive OR (XOR) operation on a finite sequence of integers starting from 1 up to N are as follows:*

$$f(N) = 0 \oplus 1 \oplus 2 \oplus \dots \oplus (N-2) \oplus (N-1) \oplus N$$

$$f(N) = \begin{cases} N & \text{If } (N \bmod 4) = 0 \\ 1 & \text{If } (N \bmod 4) = 1 \\ N+1 & \text{If } (N \bmod 4) = 2 \\ 0 & \text{If } (N \bmod 4) = 3 \end{cases}$$

Proof: Lemma IV-B.2 can be proved through induction:

For $N = 0$, $f(0) = 0$. Assume this is also true for all integers, $0 \leq N \leq 4K$, where $K \in \mathbb{Z}^+$. In this regard, it is only required to demonstrate that this assumption asserts the conjecture is valid for $4K+1$, $4K+2$, $4K+3$, and $4K+4$.

For $N = 4K+1$:

$$f(N) = f(N-1) \oplus N \rightarrow f(N) = f(4K) \oplus (4K+1)$$

Here, it is assumed that $f(4K) = 4K$. In addition, as $4K$ is even, its least significant bit is zero, thus, $N = 4K+1 = 4K \oplus 1$. Accordingly,

$$f(N) = 4K \oplus (4K+1) = 4K \oplus 4K \oplus 1 = 1$$

For $N = 4K+2$:

$$f(N) = f(N-1) \oplus N \rightarrow f(N) = f(4K+1) \oplus (4K+2)$$

The least significant bit of $4K+2$ is also zero as it is even, thus,

$$f(N) = 1 \oplus (4K+2) = 4K+3 = N+1$$

For $N = 4K+3$:

$$f(N-1) = f(4K+2) = N+1 = 4K+2+1 = 4K+3$$

$$f(N) = f(N-1) \oplus N \rightarrow (4K+3) \oplus (4K+3) = 0$$

For $N = 4K+4$:

$$f(N-1) = f(4K+3) = 0 \rightarrow$$

$$f(N) = f(N-1) \oplus N \rightarrow f(4K+3) \oplus (4K+4)$$

$$f(N) = 0 \oplus (4K+4) = 4K+4 = N$$

\square

Lemma IV-B.3. *The remainder of n divided by 4 is always 3.*

Proof: According to the primitive RS Codes definition, $n = 2^r - 1$, where $r \in \mathbb{N}_1$. However, for a functional RS Code which has a detection and correction capability, r must be greater than 1.

To prove this lemma, it is required to demonstrate that for each r , there is always a non-negative integer (\mathbb{Z}^{0+}) quotient available for $(2^r - 1)/4$, which results in a remainder of 3. In this regard, let assume that 3 and Q are the remainder and quotient of $(2^r - 1)/4$, respectively. Thus,

$$Q \cdot 4 + 3 = 2^r - 1 \rightarrow Q \cdot 4 + 4 = 2^r$$

$$2^2 \cdot (Q+1) = 2^r \rightarrow Q+1 = 2^{r-2}$$

As it is evident, for $r \geq 2$, 2^{r-2} is in \mathbb{Z}^+ and, therefore, it can be concluded that Q is always in \mathbb{Z}^{0+} . \square

In what follows, it is proven that when $pfcr = 1$, RS Codes generate a dictionary which contains all one-symbol-value code words. Additionally, for $pfcr = 0$, it is demonstrated that only the all 0s code word is valid in the generated dictionary, and other one-symbol-value code words could never be generated.

As the main step in the contradiction approach, it is assumed that the generated dictionary contains one-symbol-value code words. An one-symbol-value code word can be presented in the form of Equation (4).

$$c(x) = M \cdot (x^n + x^{n-1} + \dots + 1) \quad (4)$$

Where M is the symbol value, and $0 \leq M \leq n$.

Theorem IV-B.1. *For $pfcr = 1$, RS Codes generate a dictionary in which all one-symbol-value code words are valid.*

Proof: Based on Lemma IV-B.1, all roots of $g(x)$ must be valid roots for $c(x)$.

$$\begin{aligned} pfcr = 1 : g(x) &= \prod_{i=1}^{2s} (x - \alpha^i) = (x - \alpha) \dots (x - \alpha^{2s}) \\ \text{Roots of } g(x) &= \alpha, \alpha^2, \dots, \alpha^{2s} \\ \text{If } c(x) &= M \cdot (x^n + x^{n-1} + \dots + 1) \\ \text{for } x = \alpha^j, 1 \leq j \leq 2s : \\ c(\alpha^j) &= M \cdot \alpha^{j-1} \cdot (\alpha^n + \alpha^{n-1} + \dots + 1) \end{aligned}$$

$f(n) = \alpha^n + \alpha^{n-1} + \dots + 1$ consists of all the field elements. Considering the cyclic nature of finite field elements, $f(n)$ can be presented as a sequence of numbers as follows:

$$f(n) = \alpha^n + \alpha^{n-1} + \dots + 1 = n \oplus (n-1) \oplus (n-2) \oplus \dots \oplus 1$$

Note that in a finite field with base 2, the multiplication and summation operations become AND (\wedge) and XOR (\oplus) operations, respectively. Furthermore, since 0 is considered as the identity element for XOR, $f(n)$ can also be represented as follows:

$$f(n) = f'(n) = n \oplus (n-1) \oplus (n-2) \oplus \dots \oplus 1 \oplus 0$$

Thereby,

$$c(1) = M \wedge \alpha^{j-1} \wedge f'(n)$$

Based upon Lemmas IV-B.2 and IV-B.3:

$$\begin{aligned} n = 2^r - 1 \bmod 4 = 3 \rightarrow f'(n) &= 0 \\ c(1) = M \wedge \alpha^{j-1} \wedge 0 &= 0 \end{aligned}$$

This proves that all roots are valid in $c(x)$, which indicates that all one-symbol-value code words are valid when $pfcr = 1$. \square

Theorem IV-B.2. *For $pfcr = 0$, the only one-symbol-value code word that could be generated is all 0s.*

Proof: Similarly:

$$\begin{aligned} pfcr = 0 : g(x) &= \prod_{i=0}^{2s-1} (x - \alpha^i) = (x - 1) \dots (x - \alpha^{2s-1}) \\ \text{Roots of } g(x) &= 1, \alpha, \dots, \alpha^{2s-1} \\ \text{If } c(x) &= M \cdot (x^n + x^{n-1} + \dots + 1) \\ \text{for } x = 1 : c(1) &= M \wedge (1 \oplus 1 \oplus \dots \oplus 1) \end{aligned}$$

Since n is always odd (i.e., $2^r - 1$), then, $c(1) = M \wedge 1$, as,

$$1 \oplus 1 \oplus \dots \oplus 1 = \begin{cases} 1 & n \text{ is odd} \\ 0 & n \text{ is even} \end{cases}$$

Therefore, $c(1) \neq 0$ for $M \neq 0$.

As evidenced, the first root of $g(x)$ (i.e., 1) is only valid in $c(x)$ if the symbol value, M , is equal to zero, which indicates that except for the all 0s code word, the rest of one-symbol-value code words could never be generated when $pfcr = 0$. \square

C. Performance Evaluation

Figs. 3 (a-d) and (e-h) present the fault category distribution of the original RS Codes as a baseline (i.e., $pfcr = 1$) and the impact of the inversion layer on it. In addition, Figs. 3 (i-l) depict the effectiveness of the encoder tuning on the resiliency of RS Codes (i.e., $pfcr = 0$). As can be seen, most major peaks of CTP and CFN are eliminated at the considered frequencies (i.e., f_{UCD}), except the peaks at the frequencies of harmonic (i.e., 200MHz and 400MHz). Compared to the baseline, these ratios are reduced to half. This is due the fact that at harmonics, EMD tends to turn all bits of all symbols into 0 or 1 by a 50% probability. Since the all 0s code word is the only valid one-symbol-value code word in this setup, these ratios are reduced to half.

One solution to eliminate these remaining peaks is to exclude the all 0s code word from the transmission. This would provide a better resiliency at a cost of losing one code word of a dictionary. Figs. 3 (m-p) demonstrate this improvement. As can be seen, the rates of CTP and CFN have dropped to an absolute zero in Figs. 3-m, 3-o, and 3-p, as was the objective of this paper. Nevertheless, there is an extremely small drop in the ratio of DTP (i.e., the green category) due to the exclusion of all 0s code word which becomes smaller for larger dictionaries. Note that, it is also possible to use the inversion layer to save the majority of uncorrupted data from exclusion. However, this creates an unnecessary overhead which is against the objective of this approach. Thus, this paper recommends the first solution to arm the RS Codes against the single-frequency EMD.

V. CONCLUSION

This paper first proposed an efficient implementation of the inversion layer through double syndrome calculation. However, due to the imposed overhead of calculating the syndrome two times, the transmission rate could affect adversely. Correspondingly, to alleviate this limitation, this paper presented an effective fault elimination approach through encoder tuning to arm the primitive RS Codes against single-frequency EMD. The performance of this approach was assessed by means of

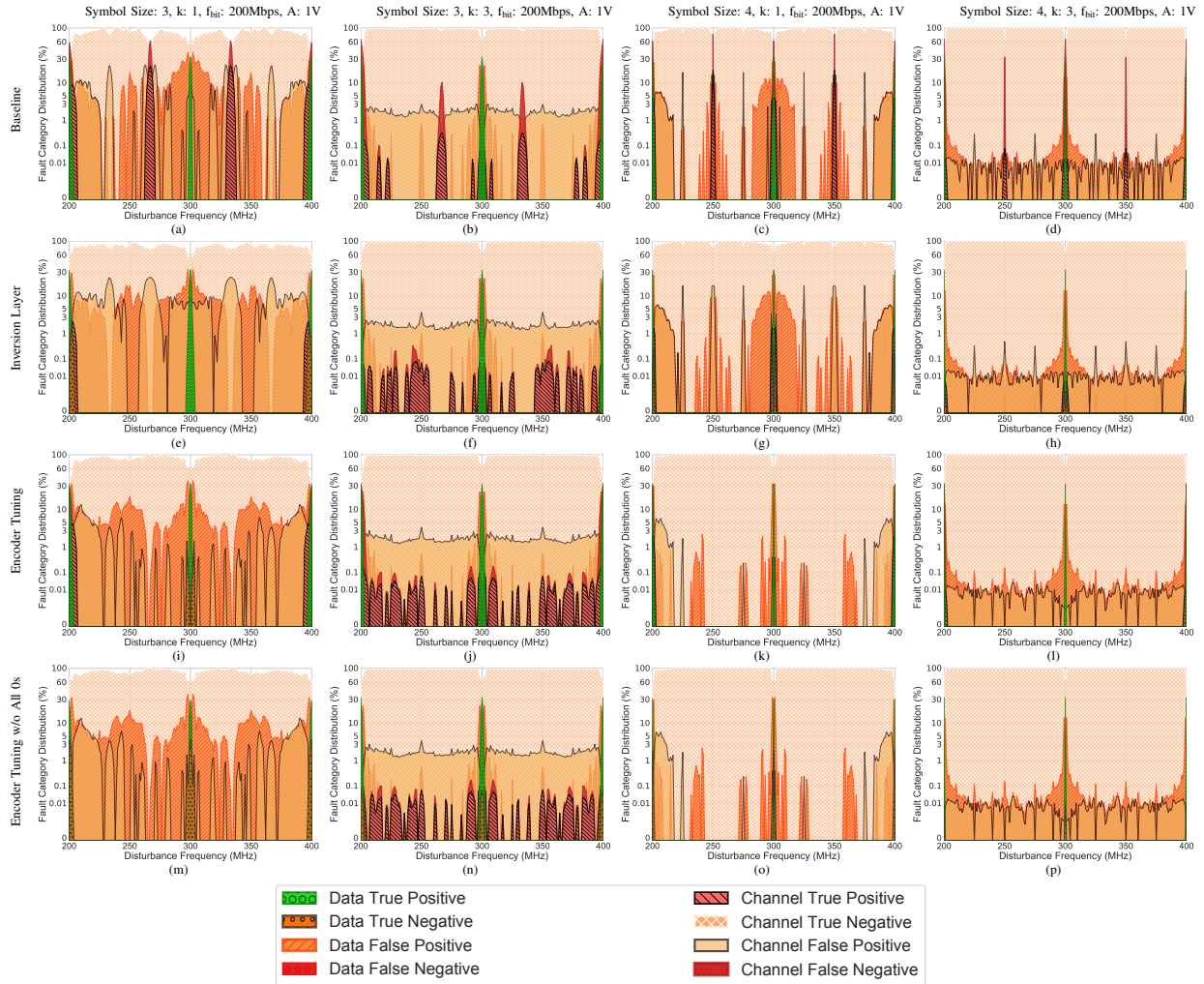


Figure 3: The impact of different fault elimination techniques on the resiliency of RSCodes against single-frequency EMD.

our in-house simulation framework. It was shown that the encoder tuning approach outperforms the multi-layer inversion-based technique. Furthermore, apart from its enhanced performance, encoder tuning has two main implementation advantages over the multi-layer structure. First, it does not require an extra layer to make the RS Codes more resilient against the single-frequency EMD. Second, this approach eliminates the overhead of performing double syndrome calculation at the consumer side.

Accordingly, compared to the state-of-the-art, it can be concluded that encoder tuning is the most suitable approach to substantially limit the impact of single-frequency EMD on primitive RS Codes in safety-critical or mission-critical applications.

REFERENCES

- [1] S. Al-Sarawi, M. Anbar, R. Abdullah, and A. B. Al Hawari, "Internet of things market analysis forecasts, 2020–2030," pp. 449–453, 2020.
- [2] D. Pissoort and K. Armstrong, "Why is the IEEE developing a standard on managing risks due to EM disturbances?" pp. 78–83, 2016.
- [3] R. W. Hamming, "Error detecting and error correcting codes," *The Bell system technical journal*, vol. 29, no. 2, pp. 147–160, 1950.
- [4] P. Memar, J. Vankeirsbilck, D. Vanoost, T. Claeys, D. Pissoort, and J. Boydens, "Resilience of Reed-Solomon codes against single-frequency electromagnetic disturbances: Fault mechanisms and fault elimination through symbol inversion," *Electronics*, vol. 11, no. 9, 2022. [Online]. Available: <https://www.mdpi.com/2079-9292/11/9/1292>
- [5] P. Memar, J. Vankeirsbilck, D. Vanoost, T. Holvoet, and J. Boydens, "Resilience of Reed-Solomon codes against harsh electromagnetic disturbances: Influence of over-voltage detection," in *2021 IEEE International Joint EMC/SI/PI and EMC Europe Symposium*, 2021, pp. 868–873.
- [6] J. Van Waes, D. Vanoost, J. Vankeirsbilck, J. Lannoo, D. Pissoort, and J. Boydens, "Resilience of error correction codes against harsh electromagnetic disturbances: Fault elimination for triplication-based error correction codes," *IEEE Transactions on Electromagnetic Compatibility*, vol. 62, no. 5, pp. 1929–1938, 2020.
- [7] I. S. Reed and G. Solomon, "Polynomial codes over certain finite fields," *Journal of the society for industrial and applied mathematics*, vol. 8, no. 2, pp. 300–304, 1960.
- [8] W. A. Geisel, "Tutorial on reed-solomon error correction coding," 1990.
- [9] T. Claeys, H. Tirmizi, H. Habib, D. Vanoost, D. Pissoort *et al.*, "A system's perspective on the use of EMI detection and correction methods in safety critical systems," pp. 905–910, 2021.
- [10] P. Memar, J. Vankeirsbilck, D. Vanoost, T. Claeys, D. Pissoort, T. Holvoet, and J. Boydens, "Resilience of Reed-Solomon codes against single-frequency electromagnetic disturbances," Submitted to *IEEE Transactions on Reliability*.