On the decoding modes of C1 and C2

Suayb S. Arslan

QTM Irvine Team
July 2, 2013

1 Introduction

1.1 RS decoder

In this note, we assume a bounded distance decoder for C1 and C2 Reed-Solomon (RS) codes defined over the field $GF(q)$, where q is a prime power. The extension field $q = 2^m$ is usually assumed for hardware friendly implementation. For any of these RS codes, let the minimum distance of the code be d. The decoder geometrically assumes spheres of radius $\lfloor \frac{d}{2} \rfloor$ centered around in every valid codeword. If the received word falls in one of these spheres, the decoding is successful and the decoded codeword is the one associated with that sphere. Otherwise one of the two things may happen: (1) the received word can fall in the interstitial space between spheres in which case a decoding failure is declared or (2) it may fall in one of the other spheres in which case the error cannot be detected and the codeword is miscorrected.

1.2 Modes of operation

Consider the product code constituted of C1 and C2, two RS codes encoding a 2-D array of user data. Let us assume that C1 RS code with parameters (n_{C1}, k_{C1}) defined over $GF(q)$, is set to correct byte errors only. Then, C2 RS code defined over the same field with parameters (n_{C2}, k_{C2}) can operate at $\lfloor \frac{n_{C2}-k_{C2}}{2} \rfloor + 1$ different modes. For example the decoder can be set to correct 0 errors and $n_{C2} - k_{C2}$ erasures. Similarly, it can also be set to correct 1 byte error and $n_{C2} - k_{C2} - 2$ erasures, so on so forth. The list of modes are illustrated in the Table below. A similar table can be generated for C1 code decoding as well.

MODES of C2	Byte Errors	Byte Erasures
Mode 1	0	$n_{C2} - k_{C2}$
Mode 2	1	$n_{C2} - k_{C2} - 2$
Mode 3	2	$n_{C2} - k_{C2} - 4$
$\vdots$	$\vdots$	
Mode $\lfloor \frac{n_{C2}-k_{C2}}{2} \rfloor + 1$	$\lfloor \frac{n_{C2}-k_{C2}}{2} \rfloor$	0

Table 1: Modes of operation

Note that the decoder, set to operate in Mode i, can correct up to $i-1$ byte errors and $n_{C2} - k_{C2} - 2(i-1)$ erasures. A C2 decoder operating at Mode i will successfully decode the received codeword if the number of byte errors τ and erasures s satisfy $\tau \leq i-1$ and $s \leq n_{C2} - k_{C2} - 2(i-1)$ at the same time. Otherwise a failure message will be generated. In an unlikely event of no reported erasures, the decoder may even miscorrect the received word if enough number of byte errors corrupt the codeword.

Note that one could have set $C2$ to operate in any mode based on the the number of byte errors and erasures, so that one can have more flexible decoding algorithm. In other words, $C2$ will be able to decode any codeword as long as $2\tau + s \leq n_{C2} - k_{C2}$. In a way, this corresponds to an operation using the union of all modes (Note that this constraint is a looser constraint than is the previous one). The problem with this approach is that when the C2 decoder operates at Mode 1, all-erasure mode, C2 will either fail or miscorrect (decoding error) every so often the C1 decoder leads to miscorrections (and C1 miscorrections are passed to C2 decoder unnoticed). In particular, if the byte errors happen not to be in one of the erased locations, the C2 decoder may fail. In the worst case, C2 decoder can even miss the erroneous bytes.

The C2 decoding error is something quite undesirable from a customer point of view. Therefore, it might be good idea to operate C2 decoder in all modes except Mode 1. Although switching from one mode to another might be an extra complexity task, such a flexibility can be shown to improve performance. On the other hand, one can fix C2 mode during subsequent decodings to one of the possibilities shown in Table 1.

Traditionally, since byte errors are only due to C1 miscorrections and the number of miscorrections are limited to one with high probability, C2 decoder is set to operate at Mode 2. However, the choice of the decoder mode should be a function of error statistics and target C2 failure rates. In otherwords, given the byte error statistics and a target failure rate, C2 decoder operating at Mode i' may be preferable over C2 decoder operating at Mode i'' where $i', i'' \in \{$ Mode 1,..., Mode $\lfloor \frac{n_{C2}-k_{C2}}{2} \rfloor\}$. In this short note we will explore some particular cases to support this claim.

1.3 Random and Independent Byte Errors

The straightforward model for byte errors is given by a binomial distribution in which the byte errors are assumed to be completely random and independent of eachother. We consider two modes of operation of particular interest for C1 and C2 in the next two subsections.

1.3.1 CASE 1: C1 Mode $\lfloor \frac{n_{C1}-k_{C1}}{2} \rfloor + 1$, C2 Mode 2

Since C2 is operating at Mode 2, let us assume no performance degradation due to C1 miscorrections as it is extremely low probability to have two or more C1 miscorrections per product code (under the binomial model) that will violate this assumption. Let p_{byte} be the byte error rate at the input of C1 decoder. Probability that C1 declares a decoding failure is therefore given by the binomial cumulative distribution function as follows

$$p_{C1,f} = \sum_{i=\lfloor \frac{n_{C1}-k_{C1}}{2} \rfloor+1}^{n_{C1}} \binom{n_{C1}}{i} p_{byte}^{i}(1-p_{byte})^{n_{C1}-i} \quad (1)$$

Here we can show that the probability of having two C1 decoding errors can be upper bounded by

$$Pr\{\text{Violation of Assumption}\} < \frac{p_{C1,f}}{\lfloor \frac{n_{C1}-k_{C1}}{2} \rfloor !} n_{C1}(n_{C1}-1) \quad (2)$$

which is way lower than $p_{C1,f}$ for small p_{byte} and large $n_{C1} - k_{C1}$.

After C1 decoding, the decoder may adapt a strategy to pass erasure information to C2. Here we assume a simple methodology that once a C1 decoding fails, all the bytes in the C1 codeword are labeled as "erasures" with probability one and passed to C2 decoder (This strategy is not necessarily the optimal way as we shall see later different approaches can improve the C2 decoding performance. We do not explore this any further in this note).

Since $C2$ operate at Mode 2, C2 failure probability can be approximated by

$$p_{C2,f} = \sum_{i=n_{C2}-k_{C2}-1}^{n_{C2}} \binom{n_{C2}}{i} p_{C1,f}^{i}(1-p_{C1,f})^{n_{C2}-i} \quad (3)$$

Here, the worst case C2 decoder error can be upper bounded by $< p_{C2,f}/(n_{C2} - k_{C2} - 2)!$ [1]. Thus, the above approximation is quite accurate. Also, the binomial distribution in this expression is not due to our original random and independent error modeling. It is rather due to C1 codewords scattered across different locations on the recording medium so that the errors effecting those codewords, which happen to belong to the same product code, are almost random and independent.

1.3.2 CASE 2: C1 Mode $\lfloor \frac{n_{C1}-k_{C1}}{2} \rfloor + 1$, C2 Mode $\lfloor \frac{n_{C2}-k_{C2}}{2} \rfloor + 1$

In another scenario, we could use C2 the way we use C1, i.e., all error detecting and correcting mode. In this scenario there is no information passing between decoders i.e., they work independent of eachother. So here we investigate whether information exchange between decoders is of any value from a performance point of view.

When C1 decoder fails, due to the decoding spheres argument of bounded distance decoding, there must be at least $\lfloor \frac{n_{C2}-k_{C2}}{2} \rfloor + 1$ byte errors. If there is a miscorrection by C1, the most likely decoded codewords would be the neighboring codewords. Thus, the decoded wrong codeword will lead to at most $n_{C2} - k_{C2} - \lfloor \frac{n_{C2}-k_{C2}}{2} \rfloor$ byte errors which is less than or equal to the number of byte errors if the C1 decoder fails. We will consider number of byte errors due to C1 failures as our reference point in case of miscorrections as this will provide the worst case scenario.

Since we are talking about average failure probabilities, we should consider the average number of byte errors due to C1 decoding failure. Let ρ_{C1} be the expected number of byte errors whenever C1 decoder fails or miscorrects. Using 2-D array product code structure, one can anticipate that the probability of any byte of a C2 codeword is in error is given by ρ_{C1}/n_{C1} from "throwing balls to bins" argument. Assuming that there are m C1 decoder failures, probability that $k \leq m$ byte positions of a C2 codeword are actual byte errors is given by

$$\binom{m}{k} \left(\frac{\rho_{C1}}{n_{C1}}\right)^k \left(1 - \frac{\rho_{C1}}{n_{C1}}\right)^{m-k} \tag{4}$$

In order for C2 failure to occur, we must have $m \geq k \geq \lfloor \frac{n_{C2}-k_{C2}}{2} \rfloor + 1$, other wise C2 will result in successful decoding. By averaging over the C1 failure probabilities, we obtain

$$\begin{aligned} p_{C2,f} &= \sum_{m=\lfloor \frac{n_{C2}-k_{C2}}{2} \rfloor+1}^{n_{C2}} \sum_{k=\lfloor \frac{n_{C2}-k_{C2}}{2} \rfloor+1}^{m} \binom{m}{k} \left(\frac{\rho_{C1}}{n_{C1}}\right)^k \left(1 - \frac{\rho_{C1}}{n_{C1}}\right)^{m-k} \binom{n_{C2}}{m} p_{C1,f}^m (1 - p_{C1,f})^{n_{C2}-m} \\ &= \sum_{m=\lfloor \frac{n_{C2}-k_{C2}}{2} \rfloor+1}^{n_{C2}} \sum_{k=\lfloor \frac{n_{C2}-k_{C2}}{2} \rfloor+1}^{m} \binom{n_{C2}}{k} \binom{n_{C2}-k}{m-k} \left(\frac{\rho_{C1}}{n_{C1}}\right)^k \left(1 - \frac{\rho_{C1}}{n_{C1}}\right)^{m-k} p_{C1,f}^m (1 - p_{C1,f})^{n_{C2}-m} \end{aligned} \tag{5}$$

One can notice that for small enough p_{byte}, we have $\rho_{C1} \approx \lfloor \frac{n_{C1}-k_{C1}}{2} \rfloor + 1$ because other possibilities are quite unlikely under random and independent byte error assumption. We plot the C2 failure probabilities for both cases covered so far in Fig. 1 for (240,230) C1 and (96,84) C2 RS codes. As can be seen, for a target failure rate of 1e-17, Case 2 – all decoders operating at error correcting mode – is preferable.

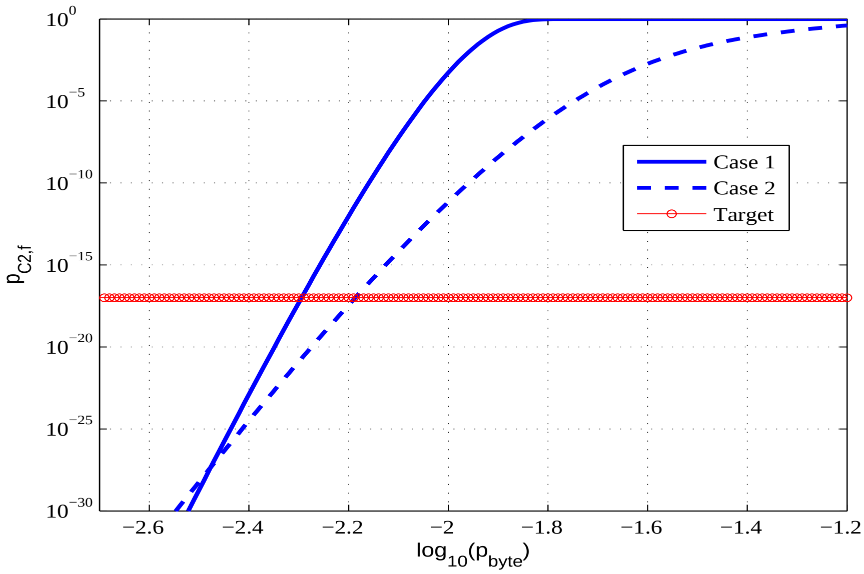

Figure 1: Comparison of the performance due to Case 1 and Case 2 under random and independent byte errors

1.4 Totally Correlated Byte Errors

In this section, we assume the other extreme that once the C1 decoder fails, all the bytes are in error. In otherwords, if a failure occurs, the whole codeword is assumed to be full of actual byte errors. The probability of C1 failure is still assumed to be given by the binomial distribution. Of course, real data modeling and noise/burst characteristics may render this assumption wrong.

1.4.1 CASE 1: C1 Mode $\lfloor \frac{n_{C1}-k_{C1}}{2} \rfloor + 1$, C2 Mode 2

Since in this case, whenever C1 decoder declares a failure, the whole codeword bytes are flagged as erasure, the performance expressions do not change.

1.4.2 CASE 2: C1 Mode $\lfloor \frac{n_{C1}-k_{C1}}{2} \rfloor + 1$, C2 Mode $\lfloor \frac{n_{C2}-k_{C2}}{2} \rfloor + 1$

When C1 decoder fails, the number of bytes in error is n_{C1} with probability one. Likewise, if C1 decoder miscorrects, the wrongly decoded codeword causes number of byte errors assumed to be close to n_{C1}. This assumption simplifies our expressions that are derived earlier.

Let us represent the number of failed C1 codewords by m. In order for C2 failure to occur, $m \geq \lfloor \frac{n_{C2}-k_{C2}}{2} \rfloor + 1$. This probability can be approximated by

$$p_{C2,f} = \sum_{i=\lfloor \frac{n_{C2}-k_{C2}}{2} \rfloor+1}^{n_{C2}} \binom{n_{C2}}{i} p_{C1,f}^{i} (1-p_{C1,f})^{n_{C2}-i} \qquad (6)$$

Let us plot this result on top of our previous derived performance results for (240,230) C1 and (96,84) C2 RS codes.

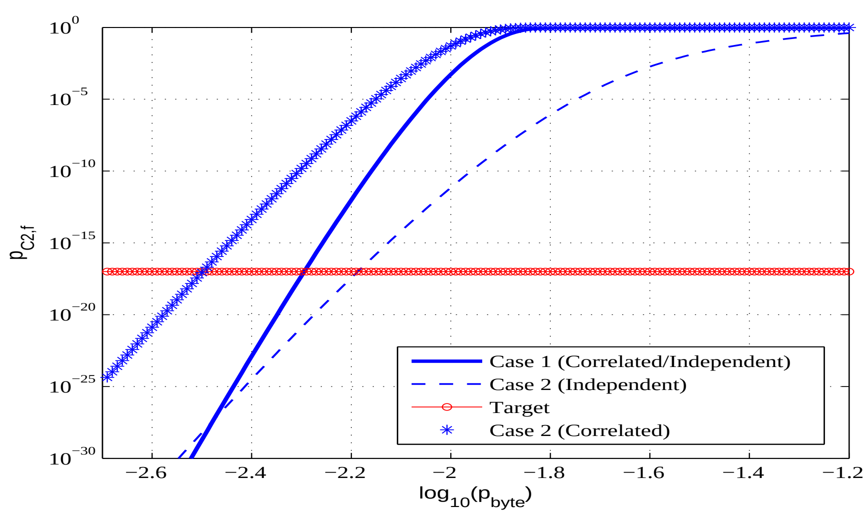

Figure 2: Comparison of the performance due to Case 1 and Case 2 under random and independent byte errors

Let us assume a target failure rate of 1e-17. As can be seen from the figure, if the byte errors were completely random and independent, Case 2 would be able to get our performance down to the target rate. However, when a complete correlation is assumed between C1 codeword byte errors, even though we use binomial model for this scenario, Case 2 suffers a lot compared to Case 1 and no longer is preferable. Assuming a binomial model, our performance will be somewhere in between of these performance curves.

Another comment is for Case 1. As can be seen in both "correlated" and "independent" scenarios, the decoding performance shows the same values. In fact, in "correlated" case, the decoder uses its full potential for erasure decoding and computes error values for each of these erasure locations. Whereas in "independent" case, erasure decoding mostly computes 0s for those byte locations in which there found to be no byte errors. Those locations were simply flagged by C1 decoder as erasures erroneously. Therefore, this implies that if the decoder was able to use its full potential in "independent" case, the decoding performance would be a lot better. The latter advices us that better decoding alternatives might be possible for "independent" case.

2 Conclusions and Future Work

As this short note shows, the modes of C1 and C2 must be selected based on the error statistics and parameters of C1 and C2 RS codes. As the real drive lies between "random and independent" and "total correlation" scenarios, one mode might be preferable over the other. For example if the drive is known to suffer due to major burst type of errors it might be preferable to have C2 work in Mode 2.

These results also show that a more flexible decoding procedure can be devised for C2 decoder that can handle byte errors other than miscorrections due to C1, while taking advantage of more flexible schemes such as the one that can choose among the possible modes of operation while decoding subsequent codewords.

References

[1] R. J. McEliece and L. Swanson, "On the Decoder Error Probability for Reed–Solomon Codes," The Telecommunications and Data Acquisition Progress Report 42-84, October–December 1985, Jet Propulsion Laboratory, Pasadena, California, pp. 66–72, Feb. 15, 1986.