Interleaved Reed - Solomon Codes in Concatenated Code Designs

Kwame Ackah Bohulu

November 13, 2019

1 Abstract

Interleaved Reed–Solomon codes allow the correction of errors beyond half the minimum code distance if the errors are not distributed independently in the received signal but occur in bursts. Therefore, these codes are mainly considered for applications in channels, that cause correlated error patterns, i.e., error bursts. However, they can also be quite interesting for memoryless channels causing independent random errors, if they are applied in concatenated code designs. We present such concatenated codes with several outer Reed–Solomon codewords and demonstrate the gain, that can be obtained by interleaved Reed-Solomon decoding in comparison to independently decoding the several words of the underlying Reed–Solomon codes.

2 Introduction

Interleaved Reed-Solomon(IRS) have been widely researched many different focuses. [1] focused on the decoding of certain burst errors whiles [2] research was based on some artificial symmetric model. In [3],[4] the use of IRS decoding for concatenated scheme outer code decoding was suggested. This is useful because the error caused by inner code decoding is usually of the burst error type.

The main idea of the IRS is the combination of l of RS codeword in the form depicted in figure x

In the case an error occurs as depicted in the figure, the chosen l codewords have symbol errors at the same position and this joint error location makes it possible to correct t errors even if $t > \frac{\lfloor N-K \rfloor}{2}$.

In[2] and [5] the decoding algorithm is based on the Welch-Berlekamp approach which finds the error location by solving a linear system of equations. It is possible for a unique solution to exist in the case where the number of linear equations in the system of equations (the contraints) is greater than the number of unknowns. For the case of a single RS codeword, this is possible only if $t \leq \frac{\lfloor N-K \rfloor}{2}$. However, for the case of the IRS the unknowns remain the same (since we know that the errors ocuur at he same positions) while different constraints are applied to it and this makes it possible to decode $t \leq \frac{\lfloor N-K \rfloor}{2}$ provided that the constraints obtained from the different codewords are linearly independent.

The disadvantage to using the Welch-Berlekamp approach is that as the IRS codwords get larger, the computational efficiency decreases. As an alternative, an iterative algorithm for Multisequence Shift-Register Synthesis (MSRS) is used. They may be seen as a generalization of the Berlekamp-Massey algorithm to multiple sequences an the complexity is the same as applying the Berlekamp-Massey algorithm l times to an individual RS codeword.

In this research, a simple concatenated scheme consisting of several codewords of a Reed–Solomon code of length N, dimension$K$, and minimum distance $D = N - K + 1$ interleaved and mapped to some codewords of a binary inner

code. Also bounds are derived to demonstrate what gain can be obtained by interleaved Reed- Solomon decoding in comparison to independently decoding the l Reed–Solomon codewords.

3 Concatenated codes with outer Interleaved Reed-Solomon Codes

Simple concatenated codes are considered in this research paper to make it easier to investigate the gains which are possible if IRS codes are applied to concatenated codes. The following definitions are necessary.

Definition 1 (Fourier Transform in $\mathbb{F}_q$) Let $p(x) = p_0 + p_1 x + \cdots + p_{N-1}x^{N-1}$be a polynomial of degree $< N$ with coefficients from the field $\mathbb{F}_q$, and let α be some element of $\mathbb{F}_q$ of order N. Then, the polynomial

$$P(x) = \mathscr{F}(p(x)) = P_0 + P_1 X + \cdots + P_{N-1}X^{N-1}$$

is called the Fourier Transform of $p(x)$ whereas the coefficients
$P_0, ..., P_{N-1}$ are calculated by $P_i = p(\alpha^i)$
Given $P(X)$ of degree $< N$, the inverse Fourier Transform

$$p(x) = \mathscr{F}^{-1}(P(x)) = p_0 + p_1 x + \cdots + p_{N-1}x^{N-1}$$

where $p_i = N^{-1}P(\alpha^{-i}), i = 0, 1, ..., N-1$ In the terminology of the Fourier Transform we formally call $p(x)$ a time domain polynomial and $P(X)$ frequency domain polynomial.

Definition 2 (RS code) Let

$$\{C(X)\} = \Big\{ \sum_{i=0}^{K-1} C_i X^i, C_i \in \mathbb{F}_q \Big\}$$

be the set of all polynomials of degree $< K$ with coefficients C_i from the field $\mathbb{F}_q$, and let α be some element of $\mathbb{F}_q$ of order N.

Then, a RS code $C_{RS} = \mathcal{RS}(q; N, K, D)$ of length N, dimension K and minimum Hamming Distance $D = N - K + 1$ can be defined as the set of polynomials

$$C_{RS} = c(x) = \{\mathscr{F}^{-1}C(X) | C(X) \in \{C(X)\}\}$$

Concatenated Code Construction

- assuming $q = 2^m$, we take l RS codewords $a^{(l)}(x) = a_0^{(l)} + a_1^{(l)}x + \cdots + a_{N-1}^{(l)}x^{N-1}, l = 1, ..., l$, form vectors from their coefficients and arrange

the vectors row-wise into a matrix.

$$\mathbf{A} = \begin{bmatrix} a^{(1)} \\ a^{(2)} \\ \vdots \\ a^{(l)} \end{bmatrix} = \left(\mathbf{a}_0, ..., \mathbf{a}_{N-1} \right)$$

- We then group the elements of $\mathbf{A}$ into the column vectors $\mathbf{a}_j$

$$\mathbf{a} = \left(a_j^{(1)}, \cdots, a_j^{(l)} \right)^T, j = 0, ..., N-1$$

- Next, a linear binary code is selected $\mathcal{B}$ of length n, dimension $k = l \cdot m$ and minimum distance d and encode the column vectors $\mathbf{a}_j$ to obtain the matrix

$$C = \left(g(\mathbf{a}_0), ..., g(\mathbf{a}_{N-1}) \right) = \left(\mathbf{b}_0^T, ..., \mathbf{b}_{N-1}^T \right) \tag{1}$$

  where all the columns of C are codewords of the inner code $\mathcal{B}$ and $g(a)$ defines the mapping $g(a) : \mathbf{a} \in \mathbb{F}_{2^m}^l \rightarrow \mathbf{b} \in \mathcal{B}$. We define the set$\{C\}$ of all matrices obtainable in this way to be the concatenated code $\mathcal{C}$

Decoding of concatenated code

- Assuming a codeword from $\mathcal{C}$ is transmitted over a memoryless noisy channel and the received word at the channel output $\mathbf{Y} = \left(\mathbf{y}_0^T, ..., \mathbf{y}_{N-1}^T \right)$
- To decode $\mathbf{Y}$ a maximum likelihood (ML) decoder is used on the inner code $\mathcal{B}$ to find the ML estimates for all columns of $\mathbf{Y}$.
- The ML outputs are converted to symbols from $\mathbb{F}_{2^m}^l$ to obtain the matrix

$$\mathbf{R} = \left(g^{-1}(\hat{\mathbf{b}}_0^T), ..., g^{-1}(\hat{\mathbf{b}}_{N-1}^T) \right) = \begin{bmatrix} r^{(1)} \\ r^{(2)} \\ \vdots \\ r^{(l)} \end{bmatrix} \tag{2}$$

Any row of $\mathbf{R}$ corresponds to a polynomial $r^{(l)}(x) = r_0^{(l)} + r_1^{(l)} x + \cdots + r_{N-1}^{(l)} x^{N-1}$, which is a potentially corrupted word of a RS code which can either be decoded independently by a conventional Reed–Solomon decoder or interpret all l words together as input word for an IRS decoder.

4 MSRS decoding of IRS codes

To explain how the MSRS decoding can be applied to interleaved Reed–Solomon codes, we first consider the classical case of a single Reed–Solomon code $\mathcal{A} = \mathcal{RS}(q; N, K, D)$. Let $a(x) \in \mathcal{A}$ be some codeword of the RS codeword. We then assume that $a(x)$ is transmitted over a noisy channel, which adds an error polynomial $e(x)$ of degree smaller than N with coefficients from $\mathbb{F}_q$. We can then rewrite $r(x) = a(x) + e(x)$. In the case where $e(x) \notin \mathcal{A}$ some of the coefficients $R_K, ..., R_{N-1}$ will be non-zero, where $R(X) = \mathscr{F}(r(x))$. We may intepret them as syndrome coefficients and denote them by $S_j = R_{K+j}, j = 0, ..., N-K-1$

To decode t errors, the standard approach for algebraic Reed–Solomon decoding is to define a polynomial $\lambda(X)$, such that the coefficient λ_j is zero whenever the corresponding coefficient e_j of $e(x)$ is not equal to zero, and non-zero, whenevee$_j = 0$. Consequently, we have $\lambda_j \cdot e_j = 0 \ \forall j = 0, ..., N-1$.

Due to the properties of the Fourier Transform, this relation is transformed into

$$\Lambda(x) \cdot E(x) \equiv \mod X^N - 1 \tag{3}$$

$\Lambda(x) = \Lambda_0 + \Lambda_1 X + \cdots + \Lambda_t X^t$ is called error locator polynomial. Since the roots of$\Lambda(x)$ are not modified by a multiplicative constant factor we set $\Lambda_0 = 1$ Equation (3) forms a linear system of equations, which contains t equations only dependent on the known syndrome coefficients S_t and the unknown coefficients $\Lambda_1, ..., \Lambda_t$. With this t equations we can write the matrix equation

$$\begin{bmatrix} S_0 & S_1 & \dots & S_{t-1} \\ S_1 & S_2 & \dots & S_t \\ & & & \vdots \\ S_{M-t-1} & S_{M-t} & \dots & S_{M-2} \end{bmatrix} \begin{bmatrix} \Lambda_t \\ \Lambda_{t-1} \\ \vdots \\ \Lambda_1 \end{bmatrix} = \begin{bmatrix} -S_t \\ -S_{t+1} \\ \vdots \\ -S_{M-1} \end{bmatrix} \tag{4}$$

$$\mathbf{S\Lambda} = \mathbf{T}$$

Where $M = N-K$ By solving Equation(4), we get the error locator polynomial and also the locations of the errors. Since the matrix $\mathbf{S}$ consists of $N-K-t$ rows and t columns, a unique solution never exists if $t > \frac{N-K}{2}$ and we aren't able to correct more than $\frac{\lfloor N-K \rfloor}{2}$.

We can rewrite Equation(4) as

$$S_k = -\sum_{j=1}^{t} \Lambda_j S_{k-j}, k = t, ..., N-K-1 \tag{5}$$

Now, the problem of calculating $\Lambda(x)$ is transformed to the problem of finding the connection weights $(\Lambda_1, ..., \Lambda_t)$ for the smallest possible t (or equivalently the shortest shift-register), which recursively generates the syndrome sequence $(S_0, ..., S_{N-K-1})$. This linear recursion synthesis problem is exactly what the Berlekamp–Massey algorithm is able to solve in a very efficient way.

In the case of an interleaved Reed–Solomon code consisting of l codewords. Then we have l received vectors $r^(l) = a^(l)(x) + e^(l)(x)$ at the output of the channel with l different error polynomials $e^(l)(x)$ non-zero coefficients at the same positions. From these received vectors we calculate the syndrome sequences $(S_0^{(l)}, ..., S_{N-K-1}^{(l)}), l = 1, ..., l$, and use them to state the linear system of equations

$$\begin{bmatrix} S^{(1)} \\ S^{(2)} \\ \vdots \\ S^{(l)} \end{bmatrix} \begin{bmatrix} \Lambda_t \\ \Lambda_{t-1} \\ \vdots \\ \Lambda_1 \end{bmatrix} = \begin{bmatrix} -T^{(1)} \\ -T^{(2)} \\ \vdots \\ -T^{(l)} \end{bmatrix} \tag{6}$$
$$\mathbf{S}_l \mathbf{\Lambda} = \mathbf{T}_l$$

The linear system of Equations (6) consists of t unknowns and $l \cdot (N - K - t)$ equations. Hence, it may have a unique solution, provided that $t \leq \left\lfloor \frac{l}{l+1}(N-K) \right\rfloor$.

we can state Equation (6) in the form of the linear recursions

$$S_k^{(l)} = -\sum_{j=1}^{t} \Lambda_j S_{k-j}^{(l)}, k = t, ..., N-K-1, (l) = 1, ...l \tag{7}$$

However, we now have to find the connection weights $(\Lambda_1, ..., \Lambda_t)$ for the minimum t, which simultaneously generate the l different syndrome sequences $(S_0^{(l)}, ..., S_{N-K-1}^{(l)}), l = 1, ..., l$.

Algorithm 1 gives a description of the MSRS algorithm in pseudo code, suitable for locating errors in an interleaved RS code. For $l = 1$ Algorithm 1 reduces to the classical Berlekamp-Massey algorithm. For $l > 1$ the inner for-loop is repeated l times. Hence, the complexity of Algorithm 1 is approximately l times the complexity of the Berlekamp-Massey algorithm.

After calculating $\Lambda(x)$, we know where the errors are located. However, to complete the decoding, we still have to evaluate the error values. This can be done in the time domain by calculating the coefficients of the polynomials e^l with the well known Forney algorithm or whatever means that seems to be more suitable for a specific application.

Note that if the degree of $\Lambda(X) > \left\lfloor \frac{l}{l+1}(N-K) \right\rfloor$ after applying Algorithm 1, the linear system of equations (4) cannot have a unique solution and results in a decoding failure. However, if $\left\lfloor \frac{l}{l+1}(N-K) \right\rfloor < deg(\Lambda(X)) \leq \left\lfloor \frac{l}{l+1}(N-K) \right\rfloor$ Equation (7) will only have a unique solution, if rank $(\mathbf{S}_l) = t$ and if rank $(\mathbf{S}_l) < t$,the MSRS algorithm cannot detect that there does not exist a unique solution and will just create an error locator polynomial out of several possible solutions. This can be seen as a disadvantage of the MSRS algorithm .

5 Bounds on Concatenated Decoding Performance

Even though it is possible to decode beyond half the minimum distance of the RS code using IRS codes, the probabilistic nature of the decoder makes it difficult to guarantee that all error patterns of weight t can be corrected even if t is in the range $\lfloor \frac{N=K}{2} \rfloor < t \leq \left\lfloor \frac{l}{l+1}(N-K) \right\rfloor$ This is because some error patters create syndromes with rank less than t and therefore such error patterns can be decoded with some probability $P_f(t) < 1$

To analyze the probability $P_f(t)$ we assume that for any position of the interleaved Reed–Solomon code any non-zero error pattern $\mathbf{e}_j = (e_j^1, ..., e_j^l)$ occurs with the same probability. Under this assumption, upper bounds for P_f have been derived in [2] and [10]. However since these bounds are not very tight or only tight for the maximum error correction radius $t = \left\lfloor \frac{l}{l+1}(N-K) \right\rfloor$ they are generally not suited very well for estimating the decoding performance of our concatenated codes. In [1], the probability $P_f(t)$ has been upper bounded dependent on t for the case of folded Reed–Solomon codes. Using similar but simpler techniques of proving, we are able to derive the following theorem for interleaved RS codes.

Theorem 1 Let a codeword $(C) \in C$ be corrupted by some errors $e^{(l)}(x) = e_0^{(l)} + e_1^{(l)}x + \cdots + e_{N-1}^{(l)}x^{N-1}$ and let $\mathbf{E} = (\mathbf{e}_0, ..., \mathbf{e}_{N-1}), e_j = (e_j^{(1)}, ..., e_j^{(l)})^T \neq 0$ be an $(l \times n)$ matrix with t non-zero columns, whereas $\lfloor \frac{N-K}{2} \rfloor < t \leq \left\lfloor \frac{l}{l+1}(N-K) \right\rfloor$. Further assume that all non-zero error patterns $\mathbf{e}_j \neq 0$ occur equiprobable. Then, the probability for a decoding failure is upper bounded by

$$P_f(t) \leq (\frac{q^l - \frac{l}{q}}{q^l - 1}) \cdot \frac{q^{-\delta(t)}}{q-1}$$

where $\delta(t) = l \cdot (N-K-t) - t$ is the difference between the number of equations and the number of unknowns in Equation(7)

Since the proof for this theorem is too complex it was left out of the research paper.

Next, we assume that the concatenated code C is transmitted ove the AWGN channel with BPSK modulation and transmitted element-wise. At the output we observe the matrix $\mathbf{Y} = (y_0^T, ..., y_{N-1}^T) = (y_{i,j} = x_{i,j} + w_{i,j})$ The variables$w_{i,j}$) are statistically independent Gaussian random variables with zero mean and variance $\sigma^2 = N_0/2$. Next ML decoding is done on the columns of $\mathbf{Y}$. The word error probability p_w at the output of this inner decoder will be the input symbol error probability for the interleaved Reed–Solomon code, i.e., the probability that a column of $\mathbf{R}$ will be erroneous. The probability p_w can be overbounded by a Union Bound, if the weight distribution of the code $\mathcal{B}$ is known. Let m_w be the number of codewords in $\mathcal{B}$, which have Hamming weight w. Then p_w can be overbounded by

$$p_w \leq \frac{1}{2} \sum_{w=d}^{n} m_w \cdot erfc\Big(\sqrt{\frac{w}{2\sigma^2}}\Big) = \hat{p}_w \tag{8}$$

If we decode each row of $\mathbf{R}$ independently by a standard Reed–Solomon decoder we know, that we can correct up to $\lfloor \frac{N-K}{2} \rfloor$ errors in the l words. Therefore, the word error probability p_w after outer decoding is

$$P_W^{RS} = \sum_{\lfloor \frac{N-K}{2}+1 \rfloor}^{N} \binom{N}{i} \cdot p_w^i \cdot (1-p_w)^{N-i} \tag{9}$$

To overbound this expression, we define the function $f(p_w) = (p_w^i \cdot (1 - p_w)^{N-1})$ and calculate the derivative. From this we see that $f(p_w)$ is a monotonically nondecreasing function, if $p_w \leq \frac{i}{N}$. Therefore, we can overbound P_W^{RS} by

$$P_W^{RS} = \sum_{\lfloor \frac{N-K}{2}+! \rfloor}^{N} \binom{N}{i} \cdot \hat{p}_w^i \cdot (1-\hat{p}_w)^{(N-i)\cdot\epsilon(i)} \tag{10}$$

where $\epsilon(i) = 1$ if $i \geq \hat{p}_w \cdot N$ and $\epsilon(i) = 0$ otherwise.

If we use an interleaved Reed–Solomon decoder to decode $\mathbf{R}$ we are able to correct $\lfloor \frac{N-K}{2} \rfloor < t \leq \left\lfloor \frac{l}{l+1}(N-K) \right\rfloor$ errors with the probability $P_f(t)$. Theorem 1 gives us an upper bound on $P_f(t)$ under the assumption of equiprobable distributed errors. Unfortunately, this assumption is anything but fulfilled after decoding the inner code. Due to the characteristics of ML decoding, low-weight error patterns will occur more frequently than high-weight patterns. To be able to apply Theorem 1 to our concatenated design, we have to modify it slightly to randomize the error patterns after the inner decoding. For this purpose, let $\mathcal{N}_1$ be the set of all $l \times l$matrices with elements from the field $\mathbb{F}_q$ and let $\mathcal{M}_l \subset \mathcal{N}_l$ be the subset of all nonsingular matrices. Now we modify the encoding rule given by Equation (1) into

$$C = \Big(g(\mathbf{M}_0\mathbf{a}_0), ..., g(\mathbf{M}_{N-1}\mathbf{a}_{N-1})\Big) = \Big(\mathbf{b}_0^T, ..., \mathbf{b}_{N-1}^T\Big)$$

whereas the matrices $\mathbf{M}_i$ are statistically independent random matrices, uniformly distributed in M_l The reverse mapping after inner decoding described by Equation (3) is modified to

$$\mathbf{R} = \Big(\mathbf{M}_0^{-1} \cdot g^{-1}(\hat{\mathcal{B}}_0^T), ..., \mathbf{M}_{N-1} \cdot g^{-1}(\hat{\mathcal{B}}_{N-1}^T)\Big)$$

By this randomization procedure, we do not influence the correct columns of $\mathbf{R}$, but we ensure that the error patterns after inner decoding are transformed to uniformly distributed error patterns. Since the number of erroneous columns is not changed by randomization, we do not expect a negative impact on the concatenated decoding performance. After randomizing, we can utilize Theorem 1 to derive an upper bound on the word error probability P_w^{IRS} w after outer

interleaved RS decoding. To do this, we use the same technique as for Equation (10), but weight the i-th summand by the factor

$$\hat{P}_f(i) = \begin{cases} (\frac{q^l - \frac{l}{q}}{q^l - 1}) \cdot \frac{q^{-\delta(t)}}{q-1}, & i \leq \left\lfloor \frac{l}{l+1}(N-K) \right\rfloor \\ 1, & \text{otherwise} \end{cases}$$

obtained from Theorem 1. In this way, we overbound P_W^{IRS} by

$$P_W^{RS} = \sum_{\lfloor \frac{N-K}{2}+1 \rfloor}^{N} \binom{N}{i} \cdot \hat{P}_f(i) \cdot \hat{p}_w^i \cdot (1-\hat{p}_w)^{(N-i)\cdot\epsilon(i)} \tag{11}$$