An Iterative Decoding Scheme on Random Burst Error Correction with Reed Solomon Codes

Xin Gao

Department of Electrical and Computer Engineering, The University of Arizona, Tucson, AZ, USA

Email: tzgaoxin2016@hotmail.com

Abstract: *We investigate the decoding scheme of conventional Reed-Solomon (R-S) codes, and propose a systematic model to achieve better decoding results on random burst error corrections. We follow the iterative decoding procedure and verify this decoding model by implementing a typical R-S (15, 9) code, then present the complexity analysis of this decoding model on the improvements for burst error correction. Simulations on several examples of R-S codes display the validity of this decoding scheme.*

Keywords: *Reed-Solomon (R-S) code, burst error correction, decoding*

Reference to this paper must be made as follows: Gao, X. (2018) ‘An iterative decoding scheme on random burst error correction with Reed Solomon codes’, *Int. J. Inform. Coding Theory*, Vol. x, No. x, pp. xxx-xxx.

Biographical notes: Xin Gao is a Ph.D. candidate in Department of Electrical and Computer Engineering, The University of Arizona, Tucson, AZ, USA. He received a Bachelor’s degree with high distinctions majored with Information Engineering from Nanjing University of Aeronautics and Astronautics (NUAA), Jiangsu, China in June 2008, an M.S.E.E. degree in Electrical Engineering from University of Minnesota-Twin Cities, Minneapolis in July 2011, and a Master’s degree in Electrical and Computer Engineering from The University of Arizona, Tucson in December 2016. Mr. Gao had been advocated to Ph.D. candidacy (All-But-Dissertation status) since August 2013. From October 2007 to August 2009, he was a research assistant in Department of Electronic Engineering, NUAA. From January 2010 to April 2013, he served as teaching assistants for nine courses in Department of Electrical Engineering, University of Minnesota and in Department of Electrical and Computer Engineering, University of Arizona. From April 2013 to October 2016, he served as a graduate research assistant in Signal and Image Lab (SaIL), Department of Electrical and Computer Engineering, The University of Arizona, Tucson. His research interests mainly include communications and coding theory, statistical signal processing, image analysis algorithms on vehicle tracking and detection, motion object detection, data (signal, image and video, etc.) compression in intelligent transportation systems, biomedical signal processing and imaging devices, and dynamic quantitative fiscal modelling.

1 Introduction

Widely employed in a variety of communication systems and data storage devices for the capacity of correcting burst errors, Reed-Solomon (R-S) codes have been recognized

in many applications for their powerful use of correcting large numbers of random errors [1], [2]. The maximal capability of error correction by R-S codes meets the singleton bound [3]: a typical R-S (n, k) code corrects random errors of which the length is no more than $(n - k)/2$, or corrects burst errors of continuous length no more than $(n - k)/2$. However, the probability of erroneous correction becomes relatively trivial when the continuous length exceeds $(n - k)/2$ [2], [4]. Several other recent soft decoding schemes [9]-[11] for Reed-Solomon codes have been proposed to improve decoding performance and reduce the computational costs. While the correlation of error positions within code elements has been fully exploited, some R-S (n, k) code can even correct burst errors with continuous length, which is close to $(n - k)$. As a result, enhancing the capability of error correction without increasing the code length or distance, comes into reality for the appearance of continuous burst errors within suitable communication channels.

Conventional algorithms for R-S burst error correction and decoding generally adopt the Berlekamp [1] iteration scheme to find the error position, then reach the error pattern by solving the corresponding equation groups, and perform error corrections in the last step. It is only possible to correct a maximum length of t burst errors [3], [5], which explicitly suggests necessary improvement on capability limitations.

The remainder of this paper is organized as follows. In Section 2, we concisely present the realization of traditional burst-error correction model and a typical decoding example of R-S $(15, 9)$ code. The analysis on symbol-error probability and the complexity contrast of R-S decoding schemes, have been illuminated in Section 3. In Section 4, we discuss the improvement of iterative searching scheme, illustrate the corresponding adoption via simulations on random channels [5], exhibit robustness and simplified calculations of the suggested improvement in our decoding scheme for correcting burst-errors in contrast to several R-S coding examples. The last Section 5 summarizes our conclusions.

2 Sample Model for R-S decoding

2.1. Traditional R-S Error Correction Model

Let us suppose some common error occurs in a code symbol during the process of bit transmission or data storage. The error pattern can be expressed as [6]:

$$E(x) = e_{i_1} x^{l_1} + e_{i_2} x^{l_2} + \dots + e_{i_v} x^{l_v} \quad (1)$$

where v represents the actual number of errors, which is no more than t ; e_{i_r} and i_r denotes the value of r -th error and the position of r -th error, respectively. Hence, the received polynomial $R(x)$ can be denoted as [6]

$$R(x) = C(x) + E(x) \quad (2)$$

where $C(x) = c_{n-1}x^{n-1} + c_{n-2}x^{n-2} + \dots + c_1x + c_0$ stands for the code polynomial after R-S (n, k) coding.

With respect to $R(x)$, the adjoint polynomial S_j [2], [6] can be obtained via

$$S_j = R(\beta^j) = C(\beta^j) + E(\beta^j) = E(\beta^j), j = 1, 2, \dots, 2t; \quad (3)$$

where β denotes a primitive root in the finite field $\mathbf{F}_q$, indicating q fits the smallest positive power of β for $\beta^{q-1} = 1$ [8]; all the roots β^j ($1 \leq j \leq 2t - 1$) are distinct from each other.

Title

For the convenience of computation, the error value is defined as $Y_r = e_{i_r}, r = 1, \dots, v$; in similar cases, $X_r = \beta_{i_r}, r = 1, \dots, v$ denotes the positions of errors.

It follows to construct the error pattern in form of $E(\beta^j)$ as

$$\begin{aligned} E(\beta^j) &= e_{i_1} (\beta^j)^{i_1} + e_{i_2} (\beta^j)^{i_2} + \dots + e_{i_v} (\beta^j)^{i_v} \\ &= e_{i_1} (\beta^{i_1})^j + e_{i_2} (\beta^{i_2})^j + \dots + e_{i_v} (\beta^{i_v})^j \\ &= Y_1 X_1^j + Y_2 X_2^j + \dots + Y_v X_v^j \end{aligned} \quad (4)$$

Substituting the expanded form of Equation (Eq.) (4) into Eq. (3) for each j , a sum of $2t$ equations [6] in matrix form can be structured via

$$\begin{cases} S_1 = Y_1 X_1 + Y_2 X_2 + \dots + Y_v X_v \\ S_2 = Y_1 X_1^2 + Y_2 X_2^2 + \dots + Y_v X_v^2 \\ \vdots \\ S_{2t} = Y_1 X_1^{2t} + Y_2 X_2^{2t} + \dots + Y_v X_v^{2t} \end{cases} \quad (0 \leq v \leq t) \quad (5)$$

where the coefficients of each β^j in $X_r, r = 1, \dots, v$ in Eq. (5), will build up an error-check matrix for the R-S cyclic codes.

2.2. Proposal for Burst Error Correction via R-S Decoding [6]

Consider the error polynomial $E(x)$ with burst errors

$$E(x) = e_{i_1} x^{i_1} + e_{i_2} x^{i_1+1} + \dots + e_{i_v} x^{i_1+v-1} \quad (6)$$

where e_{i_r} represents the r -th erroneous value, i_1 denotes the lowest position of long burst errors in the code vector, and v stands for the length of burst errors. Hence, the adjoint polynomial S_j in Eq. (6) can be rewritten as

$$S_j = E(\beta^j) = e_{i_1} (\beta^{i_1})^j + e_{i_2} (\beta^{i_1+1})^j + \dots + e_{i_v} (\beta^{i_1+v-1})^j, \quad j = 1, \dots, 2t; \quad (7)$$

Now that the equation group in Eq. (5) yields

$$\begin{cases} S_1 = Y_1 X_1 + \beta Y_2 X_1 + \dots + \beta^{(v-1)} Y_{2t} X_1 \\ S_2 = Y_1 X_1^2 + \beta^2 Y_2 X_1^2 + \dots + \beta^{2(v-1)} Y_{2t} X_1^2 \\ \vdots \\ S_{2t} = Y_1 X_1^{2t} + \beta^{2t} Y_2 X_1^{2t} + \dots + \beta^{2t(v-1)} Y_{2t} X_1^{2t} \end{cases} \quad (0 \leq v \leq t) \quad (8)$$

For the simplicity of observations, dividing both side of each equation in Eq. (8) by j -th power of X_1 ($1 \leq j \leq 2t$), the error check matrix form of Eq. (8) is now expressed as

$$\begin{bmatrix} S_1 / X_1 \\ S_2 / X_1^2 \\ \vdots \\ S_{2t} / X_1^{2t} \end{bmatrix} = \begin{bmatrix} 1 & \beta & \dots & \beta^{(v-1)} \\ 1 & \beta^2 & \dots & \beta^{2(v-1)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \beta^{2t} & \dots & \beta^{2t(v-1)} \end{bmatrix} \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_{2t} \end{bmatrix}; \quad (9)$$

Taking the first n rows to structure matrix β_v , we denote the inverse of β_v as β , where

Author

the matrix inverse of β_v is written as

$$[\beta] = [\beta_v]^{-1} = \begin{bmatrix} r_{11} & r_{12} & L & r_{1v} \\ r_{21} & r_{22} & L & r_{2v} \\ M & M & O & M \\ r_{v1} & r_{v2} & L & r_{vv} \end{bmatrix}; \quad (10)$$

Let also Eq. (9) be separated into two parts: the first part (containing a number of v equations) form the equations for the solution to errors:

$$\begin{bmatrix} Y_1 \\ Y_2 \\ M \\ Y_v \end{bmatrix} = \begin{bmatrix} 1 & \beta & L & \beta^{(v-1)} \\ 1 & \beta^2 & L & \beta^{2(v-1)} \\ 1 & M & O & M \\ 1 & \beta^v & L & \beta^{v(v-1)} \end{bmatrix}^{-1} \begin{bmatrix} S_1 / X_1 \\ S_2 / X_1^2 \\ M \\ S_v / X_1^v \end{bmatrix} \quad (11)$$

And the remainder of Eq. (9) constructs the key equations which find solutions to the starting point of error positions:

$$\begin{aligned} \begin{bmatrix} S_{(v+1)} / X_1^{v+1} \\ S_{(v+2)} / X_1^{v+2} \\ M \\ S_{2t} / X_1^{2t} \end{bmatrix} &= \begin{bmatrix} 1 & \beta^{v+1} & L & \beta^{(v+1)(v-1)} \\ 1 & \beta^{v+2} & L & \beta^{(v+2)(v-1)} \\ 1 & M & O & M \\ 1 & \beta^{2t} & L & \beta^{2t(v-1)} \end{bmatrix} \begin{bmatrix} Y_{v+1} \\ Y_{v+2} \\ M \\ Y_{2t} \end{bmatrix} \\ &= \begin{bmatrix} b_{(v+1)1} & b_{(v+1)2} & L & b_{(v+1)v} \\ b_{(v+2)1} & b_{(v+2)2} & L & b_{(v+2)v} \\ M & M & O & M \\ b_{(2t)1} & b_{(2t)2} & L & b_{(2t)v} \end{bmatrix} \begin{bmatrix} S_1 / X_1 \\ S_2 / X_1^2 \\ M \\ S_v / X_1^v \end{bmatrix}; \quad (12) \end{aligned}$$

An alternative view of Eq. (12) can be structured as

$$\begin{cases} S_{(v+1)} = b_{(v+1)1} S_1 X_1^v + b_{(v+1)2} S_2 X_1^{v-1} + L + b_{(v+1)v} S_v X_1 \\ S_{(v+2)} = b_{(v+2)1} S_1 X_1^{v+1} + b_{(v+2)2} S_2 X_1^v + L + b_{(v+2)v} S_v X_1^2; \\ M \\ S_{2t} = b_{(2t)1} S_1 X_1^{2t-1} + b_{(2t)2} S_2 X_1^{2t-2} + L + b_{(2t)v} S_v X_1^{2t-v} \end{cases} \quad (13)$$

which is explicitly having the similar expression as Eq. (8).

2.3. Realization of Decoding through the R-S(15, 9) code [5]

Construct $F_{2^4} = F_2 / (x^4 + x^3 + 1)$, where $x^4 + x^3 + 1$ is irreducible in the Galois Field $GF(2^4)$ in which the root $\beta^{15} = \beta^0 = 1$ [8]. Take the decoding scheme for the R-S (15, 9) code as an example. Since R-S (15, 9) has the capacity of correcting $t = (n - k)/2 = (15 - 9)/2 = 3$ random errors, the maximum number of burst error that can be corrected must be $v_{\max} = 2t - 2 = 4$. Thereafter, the proposed steps of decoding R-S (15, 9) code [5], [6] in $GF(2^4)$ with our iterative scheme can be achieved as below:

Title

Step 1. Search out the solution to the coefficients of error equations by implementing the matrix inverse:

$$[\beta] = [\beta_v]^{-1} = \begin{bmatrix} 1 & \beta & \beta^2 & \beta^3 \\ 1 & \beta^2 & \beta^4 & \beta^6 \\ 1 & \beta^3 & \beta^6 & \beta^9 \\ 1 & \beta^4 & \beta^8 & \beta^{12} \end{bmatrix}^{-1} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & r_{14} \\ r_{21} & r_{22} & r_{23} & r_{24} \\ r_{31} & r_{32} & r_{33} & r_{34} \\ r_{41} & r_{42} & r_{43} & r_{44} \end{bmatrix} = \begin{bmatrix} \beta^{11} & 1 & \beta^{13} & \beta^5 \\ \beta^{14} & \beta & \beta^5 & \beta^9 \\ \beta^{11} & \beta^4 & \beta^{12} & \beta^7 \\ \beta^2 & \beta^7 & \beta^6 & \beta^{14} \end{bmatrix} \quad (14)$$

Step 2. Obtain the matrix coefficients of Eq. (12) so as to solve the critical equations:

$$\begin{bmatrix} b_{31} & b_{32} & b_{33} & b_{34} \\ b_{61} & b_{62} & b_{63} & b_{64} \end{bmatrix} = \begin{bmatrix} 1 & \beta^5 & \beta^{10} & 1 \\ 1 & \beta^6 & \beta^{12} & \beta^3 \end{bmatrix} \cdot \begin{bmatrix} \beta^{11} & 1 & \beta^{13} & \beta^5 \\ \beta^{14} & \beta & \beta^5 & \beta^9 \\ \beta^{11} & \beta^4 & \beta^{12} & \beta^7 \\ \beta^2 & \beta^7 & \beta^6 & \beta^{14} \end{bmatrix} = \begin{bmatrix} \beta^6 & \beta^9 & \beta^2 & \beta^6 \\ \beta^{12} & \beta^8 & \beta^5 & \beta^7 \end{bmatrix} \quad (15)$$

Step 3. Cyclic decoding. This procedure consists of three sub-asks. Take the assumption on the error pattern of a codeword: $\beta^3 x^{14} + \beta^2 x^{13} + \beta^1 x^{12} + \beta^0 x^{11} = 0$ as an example. Let us settle the correction for maximum number of burst errors by the following sub-steps:

i) The adjoint polynomial can be calculated as

$$\begin{aligned} S_1 &= R(\beta^1) = C(\beta^1) + E(\beta^1) = E(\beta^1) = \beta^8, \\ S_2 &= R(\beta^2) = C(\beta^2) + E(\beta^2) = E(\beta^2) = \beta^4, \\ S_6 &= R(\beta^6) = C(\beta^6) + E(\beta^6) = E(\beta^6) = \beta^0 = 1. \end{aligned}$$

Hence, we have

$$[S_1 \ S_2 \ S_3 \ S_4 \ S_5 \ S_6]^T = [\beta^8 \ \beta^4 \ \beta^{12} \ \beta^{14} \ \beta^4 \ 1]^T \quad (16)$$

ii) With respect to Eq. (13), the critical equation generated by Eq. (15) and Eq. (16) can be formulated as

$$\beta^{14} X_1^4 + \beta^{13} X_1^3 + \beta^{14} X_1^2 + \beta^5 X_1 + \beta^4 = 0 \quad (17)$$

$$\beta^5 X_1^4 + \beta^{12} X_1^3 + \beta^2 X_1^2 + \beta^6 X_1 + 1 = 0 \quad (18)$$

iii) Find out the common root(s) of Eq. (17) and Eq. (18) via the Chien-search algorithm [6], [7].

By Chien-search [6], [7] and verification of trials, the roots are β^4 , β^9 and β^{11} for Eq. (17), while β^{11} is the unique root for Eq. (18). Hence, the common root β^{11} of Eq. (17) and Eq. (18) has been obtained as the location when burst error starts.

Step 4. Finally, obtain the error values via Eq. (11):

$$\begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \\ Y_4 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & r_{14} \\ r_{21} & r_{22} & r_{23} & r_{24} \\ r_{31} & r_{32} & r_{33} & r_{34} \\ r_{41} & r_{42} & r_{43} & r_{44} \end{bmatrix} \cdot \begin{bmatrix} S_1 / X_1 \\ S_2 / X_1^2 \\ S_3 / X_1^3 \\ S_4 / X_1^4 \end{bmatrix} = \begin{bmatrix} \beta^{11} & 1 & \beta^{13} & \beta^5 \\ \beta^{14} & \beta & \beta^5 & \beta^9 \\ \beta^{11} & \beta^4 & \beta^{12} & \beta^7 \\ \beta^2 & \beta^7 & \beta^6 & \beta^{14} \end{bmatrix} \cdot \begin{bmatrix} \beta^{12} \\ \beta^{12} \\ \beta^9 \\ 1 \end{bmatrix} = \begin{bmatrix} \beta^3 \\ \beta^2 \\ \beta^1 \\ 1 \end{bmatrix} \quad (19)$$

Author

From Eq. (19), the error pattern in polynomial term is now denoted as:

$$\beta^3 x^{14} + \beta^2 x^{13} + \beta^1 x^{12} + x^{11} = 0 \quad (20)$$

which displays exactly the same error pattern as previously proposed in Step 3.

Similarly, the characteristic polynomial of x^4+x+1 can be constructed in a Galois Field $\mathbf{F}_{2^4} = \mathbf{F}_2 / (x^4 + x + 1)$. Meanwhile, take another pattern of random error for example, let β^6 begins with the codeword, the error pattern $\beta^6 x^{14} + \beta^4 x^{13} + \beta^2 x^{12} + \beta^1 x^{11} = 0$, also yields the deserved error-checking result.

3 Error Analysis for Checking Codewords

3.1. Error Analysis for R-S Codes [7]

In Section 2, we have presented a specific iterative scheme for an accurate R-S (15, 9) decoding. With respect to the aforementioned discussions in the introduction section, it is known to have $(2t - v_{\max})$ critical equations for X_1 , of which the common root(s) will display the primary position of burst errors. However, when $v_{\max} > t$, those common roots cannot be used may have some possibility to come out; on the other hand, the length of correctable error is stretched with the increment of $v_{\max}$, which also results into the probability level-up of unusable common roots [6].

In order to lower down the probability of erroneous corrections, we need to carefully select a proper set of $v_{\max}$. When we have a unique solution to the critical equations of X_1 with length of error shorter than $v_{\max}$, the common roots are continuous, as a result, accurate decoding can be achievable; otherwise, according to the maximum likelihood (ML) principle, the common root containing a shortest length of burst errors, must be chosen as the error pattern of any codeword.

Let P_e denote the R-S decoded symbol-error probability in terms of the channel symbol-error probability p_0 , Referring to P. Owsley and J. P. Odenwalder's work [4], the formula for calculating P_e via binomial expressions yields

$$P_e \approx \frac{1}{2^m - 1} \sum_{j=t+1}^{2^m-1} j \binom{2^m-1}{j} p_0^j (1-p_0)^{2^m-1-j} \quad (21)$$

where t stands for the capability for R-S code error-correction and m represents the number of bits per symbol.

Justifying from the components of adjoint polynomials, the following lemmas state the error analysis for R-S code:

Lemma 1 [4],[6]: For the R-S code, when the number of errors does not exceed the error-correction capability for t errors, it is justified that no error exists in the code as long as $S_1, S_2, \dots, S_t$ are fully zeros.

Lemma 2 [5]: For the R-S code, when the number of errors does not exceed the error-correction capability for t errors, if $S_1, S_2, \dots, S_t$ satisfy $S_{i+1} / S_i = S_i / S_{i-1} \neq 0, i = 2, 3, \dots, t$, it is justified that only a single error come out in this codeword.

Notably, erroneous error correction occurs in condition that the burst-error has the

exact length of the error pattern; if the critical equations share the same multiple common roots, such kind of errors will be justified as not correctable.

3.2. Complexity Comparison on R-S Decoding Schemes [7]

With respect to the steps of classical R-S decoding scheme, Horner algorithm [1] can be adopted for the first t adjoint polynomials when only one single error occurs. By the Binary Merging (BM) iterations and Chien-Searching method [6], the erroneous position polynomial and the error position are achieved; while the error value is calculated by the Forney algorithm [7]. Within the necessary conditions shown above, in comparison to that of traditional Berlekamp iterations, computational cost of the proposed R-S decoding scheme are estimated as follows [7]:

- (i) For the absence of error occurrence, traditional method computes the total parts of $2t$ equations adjoint polynomials which requires $2t*(n-1)$ times of both multiplications and additions. Our R-S decoding scheme consumes $t*(n-1)$ times of both operations since only the first t components ($S_1, \dots, S_t$) require computation.
- (ii) When one single error happens, traditional method urges to compute $2t$ equations of adjoint polynomials for BM iteration and Chien-search, which requires $(2t+1)*n$ times of both additions and multiplications. However, our scheme only needs to compute the first $(t+1)$ components ($S_1, \dots, S_{t+1}$) and judge their proportionality; therefore, the consumption of multiplications and additions times are correspondingly counted as $(t+1)*n$ and $(t+1)*(n-1)$, representing a reduced cost on calculations.
- (iii) Assume that more than one error have occurred, the proposed scheme also needs to calculate $2t$ equations of adjoint polynomials within the several steps, and it has even induced the supplemental judgment for cyclic proportion. However, the judgment is often terminated via quite few repetitions; hence, the increment of operations can be neglected. The proposed scheme needs approximately $2t*(n-1) + (2t+n)*e + 2e^2$ ($e \geq 2$) times of both multiplications and additions, where e represents the number of multiple errors, which still calls for further simplification.

4 Simulation Results

Since the random bursting errors are usually invoked by the deep fading or multipath propagation effects from the fast-moving targets in wireless communication systems, they have random occurrence and often exhibits long duration of time [1], [6]. As a result, the information rate has been restricted under a certain systematic bandwidth and Signal-to-Noise Ratio (SNR). When the burst errors went beyond the capacity of ordinary decoding methods, the originally received codeword comes through the refreshment of decoding by means of adopting this decoding scheme on burst-error correction. Hence, regarding to general communication systems, the capacity of correcting frequently-hopped burst errors can be greatly enhanced when implementing the proposed scheme.

The simulations for R-S (15, 11) code with burst noise in a random communication channel [4] has been depicted in Figure 1, which depicts the record of error detection and comparison of message and decoded signals. Note that RS-DECO represents an R-S decoder from the MATLAB communication toolbox. For the sequential signal of $L_1 = 2^8 = 256$ bits, 5% of stochastic noise has been added (where $\text{SNR} \approx 13\text{dB}$ in that condition).

Author

It is clear to justify that the erroneous bits are quite rare; our scheme has displayed meritorious performance from sending message to the decoded signal via this R-S code example.

Figure 1. Performance of R-S (15, 11) code for error correction via burst-error decoding scheme ($L = 256$, 5% noise)

Let us make a bit more aggregation of the noise, and carry out longer signal sequence to testify the robustness of the proposed method. The percentage of noise is set as 9.1% ($\text{SNR} \approx 10\text{dB}$ in that condition) to the extended sequential signal $L_2 = 2^{10} = 1024$ bits, then we plot the corresponding record and decoding performance of burst error detection, which have been exhibited in Figure 2. From Figure 2, it is indicated that the proportion of errors to sequence length does not change much as the noise environment receives more interference, and the comparison shows explicit the robustness of decoded signal in contrast to original message. The implementation of our proposed scheme suggests good capacity maintenance for burst error correction over comparatively long communication sequence. Since the message output of an R-S decoder is regularly divided into packets, we make combinations of the message and compare the counted erroneous bits with proportion to the source symbols [5].

Title

Figure 2. Performance of R-S (15, 11) code for error correction via burst-error decoding scheme ($L = 1024$, 9.1% noise)

Without loss of generality, we exploit a simple case of R-S (7, 3) code with 8-PSK for symbol mapping and generate 10^5-1 symbols. The collected data from our decoding scheme via MATLAB Monte-Carlo trials, has been tabulated in Table I. Meanwhile, regarding to bit error rate (BER) of our decoding scheme for R-S (7, 3) code, the plots of BER (P_e) versus the SNR ranging from 1~10 dB has been displayed in Figure 3, where the curve on its probability of wrong-bit occurrence starts in condition of approximately 10% error. As SNR increases, the performance of BER (P_e) suggests downgrading, which become extremely small when SNR comes close to 10 dB.

Table I. Bit error rate versus SNR for R-S (7, 3) code

Noise power	SNR	Wrong Bits	BER (P_e)
0.03	14.137	0	0
0.05	11.919	8	$8.0 \cdot 10^{-5}$
0.10	8.9086	41	$41 \cdot 10^{-4}$
0.12	8.1167	92	$9.2 \cdot 10^{-4}$
0.14	7.4473	189	$1.89 \cdot 10^{-3}$
0.16	6.8674	432	$4.32 \cdot 10^{-3}$
0.20	5.8983	803	$8.03 \cdot 10^{-3}$
0.24	5.1064	1889	$1.89 \cdot 10^{-2}$
0.28	4.4370	4178	$4.18 \cdot 10^{-2}$
0.40	2.8880	10450	0.1045

Figure 3. Bit-error rate (BER) versus SNR [1: 10 dB] for R-S (7, 3) decoding

Simulation for common R-S (255, 223) decoding over $GF(2^8)$ with similar scheme on that of R-S(7, 3) decoding has been illustrated in Figure 4. The more complicated part lies on the implementation of bit-interleaving scheme for the arranged information frames as well as the Viterbi decoding with soft decision [4]. It is justified from our experiments that in condition of lower SNR (less than 5dB), the symbol error rate (SER) got boomed and hence interrupts the burst-error channel, while the SER reduced below 10^{-2} in condition of a 3dB gain. It is also suggested that R-S (255, 223) code works well under higher SNR (greater than 8 dB) via our decoding scheme. Better effects have also been achieved for longer sequence of R-S (255, 223) code on burst-error or mixed-error channels [5] after the adoption of both bit-interleaving and our decoding scheme.

Figure 5 displays the performance of frame-error rate (FER) in contrast to BER via the proposed iterative R-S decoding scheme. The random input information sequence was generated by integers from [0, 15], where the code length was fixed at 11. Iteration times was set as 500, initial energy power = 1.25, and SNR was plotted from -3 dB to 12 dB with a step size of 1 dB. This test suggest that for random data sequence with white Gaussian noise, as SNR increases, the FER drops faster than BER; when SNR reaches higher than 10 dB, BER reduced to the range of 10^{-2} ; while SNR exceeds 12 dB, both FER and BER reduced to zero, which can be regarded as nearly perfect data transmission with no error.

In sum, the proposed iterative decoding scheme has provided an algorithmic model for typical binary R-S codes. This method is capable of easily locating random burst errors in condition of long communication sequences, displays relative low error rate on two R-S decoding examples when SNR is lower than 10 dB, achieves error-free transmission for random data sequence with fixed code length when SNR is higher than 12 dB.

Title

Figure 4. Symbol-error rate (SER) versus SNR [5: 0.5: 8 dB] for R-S (255, 223) decoding

Figure 5. Proposed iterative R-S decoding: FER and BER versus SNR [-3: 1: 12 dB] for random input information, integer sequence in [0, 15], finite code length = 11

5 Conclusions

In this paper, we have presented the comparison of conventional Reed-Solomon (R-S) decoding models with experimental design for the iterative decoding scheme with lower computational costs. The decoding procedure associated with sequential decoding for R-S (7, 3) and R-S (15, 11) codes were exhibited through error-checking experiments on multiple length of communication sequence. Implementations of burst error corrections for Reed-Solomon codes have also been testified throughout the R-S (7, 3), R-S (255, 223) codes and an example of random data sequence with finite length. The proposed iterative scheme with typical R-S code examples indicates good quantitative results for detecting random burst errors, exhibits relatively low error rate in condition of matched channel environments, displays robustness in accordance with the complexity analysis on error-correcting operations.

6 References

- [1] J. Chen and P. Owsley, "A burst error correcting algorithm for Reed Solomon Codes," *IEEE Trans. Inform Theory*, vol. 38, no. 6, pp. 1807-1812, Nov. 1992.
- [2] G. Dubney, "DECODING ERROR CORRECTION CODES UTILIZING BIT ERROR PROBABILITY ESTIMATES," Ph.D. Dissertation, Department of Electrical Engineering, University of Southern California, Jan. 2006.
- [3] J. Jiang and K. R. Narayanan, "Algebraic soft-decision decoding of Reed-Solomon codes using bit-level Soft information," *IEEE Trans. Inform Theory*, vol. 54, no. 9, pp. 3907-3928, Sep. 2008.
- [4] D. J. Costello, J. Hagenauer, H. Imai, and S. B. Wicker, "Applications of error-control coding," *IEEE Trans. Inform Theory*, vol. 44, no. 6, pp. 2531-2560, Oct. 1998.
- [5] K. Phil, "Reed-Solomon Coding/Decoding Package (V1.0) [EB/OL]. (1996-09-12)," [Online] available: <http://www.piclist.com/techref/method/error/rs-gp-pk-uoh-199609>.
- [6] Z. Yang, Y. Liu, Y. Wang, and D. Tang, "Study and implementation of burst-error correcting algorithm for Reed-Solomon codes," *J. Inform. Eng. University*, vol. 5, no. 4, pp. 77-79, Dec. 2004.
- [7] Y. Hao and J. Jiang, "Improved time-domain decoding algorithm of R-S code," *Comput. Eng.*, vol. 34, no. 14, pp. 104-106, Jul. 2008.
- [8] P. Garrett, *The Mathematics of Coding Theory*. Prentice Hall, 2004.
- [9] L. Chen, S. Tang, and X. Ma, "Progressive algebraic soft-decision decoding of Reed-Solomon codes," *IEEE Trans. Commun.*, vol. 61, no. 2, pp. 433-442, Feb. 2013.
- [10] L. Chen and M. Bossert, "Algebraic Chase decoding of Reed-Solomon codes using module minimisation," in *Proc. Int. Symp. Inform. Theory & Appl. (ISITA)*, Oct. 30 - Nov. 02, 2016, Monterey, USA, pp. 305-309.
- [11] S. Lin, K. Abdel-Ghaffar, J. Li, and K.-K. Liu, "Iterative soft-decision decoding of Reed-Solomon codes of prime lengths," in *IEEE Int'l Symp. Inform. Theory (ISIT)*, Jun. 25-30, 2017, Aachen, Germany, pp. 341-345.