A Soft Decoding for High Rate Reed-Solomon codes

Sergey Egorov

Southwest State University, Kursk, Russia
sie58@mail.ru

Abstract. A novel algorithm is proposed for soft decoding of Reed-Solomon codes. This algorithm is based on list decoding algorithm allowed to correct errors beyond half the minimum distance. The coding gain of the proposed algorithm is shown for some high rate codes. A block diagram of new Reed-Solomon soft decoder is given.

Keywords: Reed-Solomon codes, soft-decision decoding, list decoding.

1 Introduction

A Reed-Solomon (RS) code [1] is described as an (n, k) code, where the codeword consists of n symbols from a Galois Field (GF) of q elements, k of which are information symbols, with $r = (n-k)$ check symbols. $\alpha^{m_0}, \alpha^{m_0+1}, ..., \alpha^{m_0+r-1}$ are the roots of any RS codeword, where α is a primitive element of the field and m_0 is an integer. Define the minimum distance, $d = r + 1$ and $t_C = \lfloor (d-1)/2 \rfloor$, the maximum number of error symbols that can be always corrected.

It is known that RS codes can correct any pattern of t errors or less iff $2t+1 \leq d$ (or $t \leq t_C$). Several efficient RS decoding algorithms are developed for correcting up to t_C errors. Two frequently used are the Berlekamp-Massey (BM) algorithm and the algorithm by Sugiyama et *al*. [1, 2].

A procedure providing t_C+1 error correction for RS codes was developed by Blahut [3] based on the BM algorithm. This procedure was improved by Egorov and Markarian [4, 5]. Afterwards in [6] this procedure was expanded for multiple extra error correction.

In [7] Berlekamp proposed bounded distance + 1 soft decision Reed-Solomon decoding based on the Welch-Berlekamp algorithm.

Guruswami and Sudan proposed an algorithm providing $t_C + \tau$ ($\tau \geq 1$) error correction for RS codes (GS-algorithm) [8]. The soft decision version of GS-algorithm was introduced by Koetter and Vardy [9]. VLSI architectures for soft RS decoder on the base of this algorithm were developed in [10].

In this paper a novel algorithm for soft decoding of Reed-Solomon codes is proposed. This algorithm is based on list decoding algorithm allowed to correct errors beyond half the minimum distance [6]. The simple RS soft decoder can be constructed on the base of the proposed algorithm.

2 Basic List Decoding Procedure

The proposed soft decoding algorithm is based on the list decoding algorithm. In this section this list decoding algorithm is described following [6].

Basic list decoding algorithm belongs to the class of decoding algorithms known as "syndrome decoding algorithms" and is based on the Berlekamp-Massey algorithm [11].

The main idea of this algorithm is to continue analytically the Berlekamp-Massey algorithm through 2τ more iterations and to search for values of error position, such that the corresponding error locator polynomial of degree t_C+τ has exactly t_C+τ legitimate roots.

The basic list decoding algorithm consists of the following steps:

1) Calculate the syndrome polynomial $S(x)$.

2) Calculate the locator polynomial $\Lambda^{(2t_C)}(x)$, auxiliary polynomial $B^{(2t_C)}(x)$ and formal degree of the locator polynomial L_{2t_C} using $2t_C$ iterations of the Berlekamp-Massey algorithm.

3) Calculate the Fourier transforms of the polynomials $\Lambda^{(2t_C)}(x)$ and $B^{(2t_C)}(x)$.

4) Search for unknown discrepancies $\Delta_{2t_C+1}, \Delta_{2t_C+2}, \ldots, \Delta_{2t_C+2\tau}$, such as that $\Lambda^{(2t_C+2\tau)}(\alpha^{-i}) = 0$ for exactly t_C+τ values of i. The values of i locate error positions.

5) Compute error magnitudes and correct errors.

The algorithm described above fulfils a list decoding procedure, it finds all the codewords that lie within the decoding sphere of radius $t = t_C + \tau$ drawn about the received codeword.

Computational complexity of the step 4 of the algorithm is bounded by polynomial in n for constant τ. In particular, for τ = 1 the complexity is described as $O(n^2)$, for τ = 2 as $O(n^4)$. Total complexity of the algorithm is less than one of Guruswami-Sudan algorithm for small τ.

In detail this list algorithm was presented in [6].

3 Soft Decoding Algorithm

The proposed soft decoding algorithm is based on list decoding algorithm described above. The main step 4 of the list decoding algorithm is altered as following:

Calculate the sequences $S_{L[i_1],L[i_2],\ldots,L[i_{2\tau-1}]}$ of possible values of the discrepancy $\Delta^{L[i_1],L[i_2],\ldots,L[i_{2\tau}]}$ for all permitted sets of indices $i_1, i_2, \ldots, i_{2\tau-1}$ ($i_{2\tau} = i_{2\tau-1}+1, \ldots, n_C-1$), and search for discrepancy values Δ which are happened exactly w times in any sequences $S_{L[i_1],L[i_2],\ldots,L[i_{2\tau-1}]}$:

$$S_{L[i_1],L[i_2],\ldots,L[i_{2\tau-1}]} = \{\Delta^{L[i_1],L[i_2],\ldots,L[i_{2\tau}]} =$$

$$= \frac{F^{o1}(L[i_1], L[i_2],..., L[i_{2\tau}], R_{L[i_1]}, R_{L[i_2]},..., R_{L[i_{2\tau}]})}{F^{o2}(L[i_1], L[i_2],..., L[i_{2\tau}], R_{L[i_1]}, R_{L[i_2]},..., R_{L[i_{2\tau}]})};\ i_{2\tau} = \overline{i_{2\tau-1}+1, n_C-1}\}. \quad (1)$$

where:

$$F^{o}(L[i_1], L[i_2],..., L[i_{2\tau}], R_{L[i_1]}, R_{L[i_2]},..., R_{L[i_{2\tau}]}) =$$

$$= \sum_{\substack{\forall\{j_1,j_2,...,j_o\}=J \\ j_1<j_2<..<j_o \\ j_1,j_2,...,j_o\in\{1,2,...,2\tau\}}} \left[\prod_{\substack{\forall\{k1,k2\} \\ k1<k2 \\ k1,k2\in J}} (\alpha^{L[i_{k1}]} + \alpha^{L[i_{k2}]}) \prod_{\substack{\forall\{k1,k2\} \\ k1<k2 \\ k1,k2\in\{1,2,...,l\}\setminus J}} (\alpha^{L[i_{k1}]} + \alpha^{L[i_{k2}]}) \prod_{k=1}^{o} R_{L[i_{j_k}]} \right],$$

$$R_i = \begin{cases} \alpha^{(2s+1)i} B^{(2t_C)}(\alpha^{-i}) / \Lambda^{(2t_C)}(\alpha^{-i}) & \text{if } s \geq 0, \\ \alpha^{(-2s-1)i} \Lambda^{(2t_C)}(\alpha^{-i}) / B^{(2t_C)}(\alpha^{-i}) & \text{in other case} \end{cases},$$

$$s = t_C - L_{2t_C},\ o1 = \tau - |s+1|,\ o2 = \tau - |s|.$$

If there exists a discrepancy value Δ which is happened exactly w times in some sequence $S_{L[i_1],L[i_2],...,L[i_{2\tau-1}]}$ then error positions are given by index set $L[i_1]$, $L[i_2]$,…, $L[i_{2\tau-1}]$ of this sequence and by set of $L[i_{2\tau}]$ values corresponding the discrepancy value Δ in this sequence.

The error position search is fulfilled in order of ascending total symbol reliability measure. A symbol reliability measure may be estimated on the base of soft decisions about symbol bits. The sequence of symbol position ordered by symbol reliability measures is stored in the table $L[\ ]$. The search range is bounded by $n_C \leq n$ and contains the least reliable symbol position of the received codeword.

The soft decoding algorithm is many orders lower in complexity compared to the basic list decoding algorithm.

The proposed soft decoding algorithm consists of the following steps:

1) Calculate the syndrome polynomial $S(x)$. If terms of $S(x)$ are all zero, then go to step 14 (there are no errors in the received codeword).

2) Calculate the locator polynomial $\Lambda^{(2t_C)}(x)$, auxiliary polynomial $B^{(2t_C)}(x)$ and formal degree of the locator polynomial L_{2t_C} using $2t_C$ iterations of the Berlekamp-Massey algorithm.

3) If $L_{2t_C} \leq t_C$, then roots of the polynomial $\Lambda^{(2t_C)}(x)$ are searched. If the number of legitimate roots is equal to L_{2t_C}, then their inverses are considered as error locators. Error magnitudes are computed using Forney's formula [11]. A false error pattern is rejected, true one is introduced in the error list.

4) Set s (shift): $s = t_C - L_{2t_C}$. If $s \geq \tau$ or $s < -\tau$, go to step 13.

5) Calculate the Fourier transforms of the polynomials $\Lambda^{(2t_C)}(x)$ and $B^{(2t_C)}(x)$.

6) Calculate the set of fractions $R_i = \alpha^{(2s+1)i} B^{(2t_C)}(\alpha^{-i}) / \Lambda^{(2t_C)}(\alpha^{-i})$ when $s \geq 0$ or $R_i = \alpha^{(-2s-1)i} \Lambda^{(2t_C)}(\alpha^{-i}) / B^{(2t_C)}(\alpha^{-i})$ otherwise, i=0,…,n-1.

7) Set v = 1 (v is number of extra errors for correction).

8) Calculate the auxiliary variables: $l = 2v$, $o1 = v - |s+1|$, $o2 = v - |s|$, $w = t_C + 1 - v$.

9) Set sc = 1 (sc is counter of equation set).

10) Calculate the sequences $S_{L[i_1],L[i_2],\ldots,L[i_{l-1}]}$ of possible values of the discrepancy $\Delta^{L[i_1],L[i_2],\ldots,L[i_l]}$ for most probable sets of indices $i_1, i_2,\ldots,i_{l-1}$ ($i_l = i_{l-1}+1,\ldots,n-1$), and search for discrepancy values Δ which are happened exactly w times in any sequences $S_{L[i_1],L[i_2],\ldots,L[i_{l-1}]}$ (1).

If there exists a discrepancy value Δ which is happened exactly w times in some sequence $S_{L[i_1],L[i_2],\ldots,L[i_{l-1}]}$ then error positions are given by index set $L[i_1],L[i_2],\ldots,L[i_{l-1}]$ of this sequence and by set of $L[i_l]$ values corresponding the discrepancy value Δ in this sequence.

Compute error magnitudes using Forney's formula. A false error pattern is rejected, true one is introduced in the list.

If error pattern is introduced in the list and (*exit* = 1 or *exit* = 2), then search is finished. If *exit* = 1 then go to step 13. If *exit* = 2 then go to step 11.

11) If $v = -s$, then go to step 12, else $sc = sc + 1$, $l = l - 1$, $o1 = o1 - 1$, $o2 = o2 - 1$, $w = w + 1$.

If $sc \leq (v-|s|)$, then go to step 10.

12) $v=v+1$. If $v \leq \tau$, then go to step 8.

13) If the error list is empty, then decoding failure. If error list contains one error pattern, then this error pattern is corrected. If error list contains more error patterns, then error pattern is corrected such as closest to the received codeword.

14) End.

The following notations are used for the algorithm description:

- L is lookup table contained symbol position ordered by reliability.
- *exit* is exit mode of the algorithm, $exit \notin \{0,1,2\}$, 0 – usual exit, 1 - urgent exit after detecting first error pattern, 2 – exit after detecting first error patterns for all system sets.

The soft decoding algorithm was simulated for some practical high rate RS codes with d=17. The modulation used was BPSK and the channel model was AWGN. The minimum of symbol bit LLRs is taken as the reliability measure of the symbol.

Fig. 1, 2 show the performance of the new algorithm for these codes. Fig. 3 shows the average complexity of discrepancy computation for RS code (120,104). On the all figures: **1** denotes the conventional decoding algorithm correcting up to t_C errors (t=8), **2** – the proposed soft decoding algorithm with τ =1 (t=9), **3** – this algorithm with τ =2 (t=10), **4** – this algorithm with τ =3 (t=11), **5** – Koetter-Vardy algorithm (multiplicity 2) [12]. On the fig. 3: **A** denotes the list decoding algorithm without soft decisions (see section II) with τ =1 (t=9), **B** – this algorithm with τ =2 (t=10).

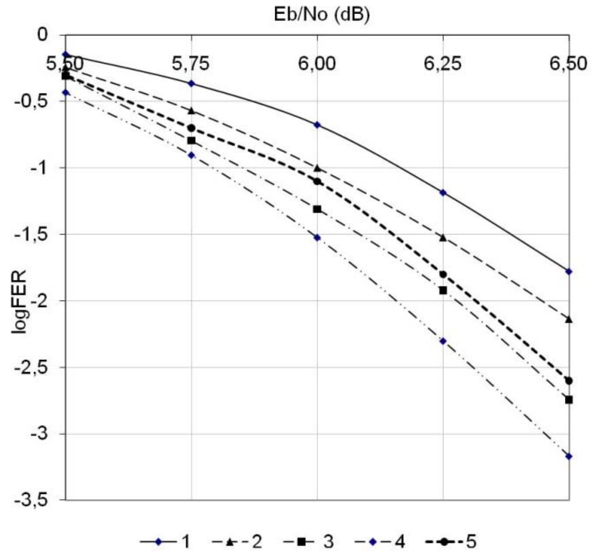

Fig. 1. FER (Frame Error Ratio) for decoding of the (255,239) RS code

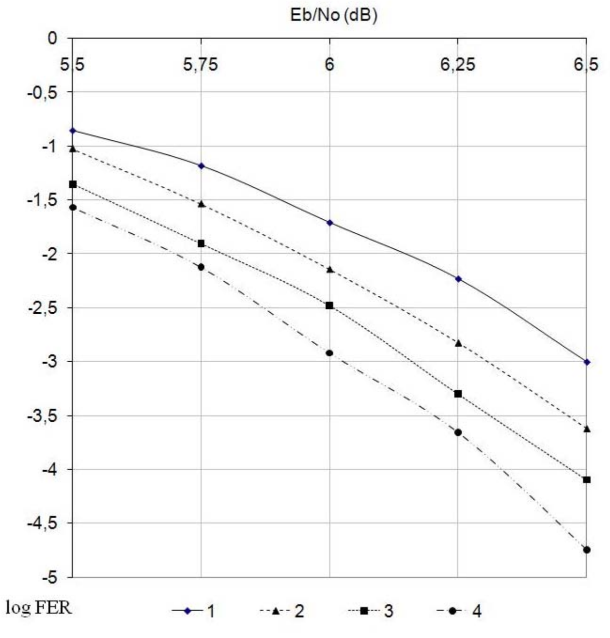

Fig. 2. FER (Frame Error Ratio) for decoding of the (120,104) RS code

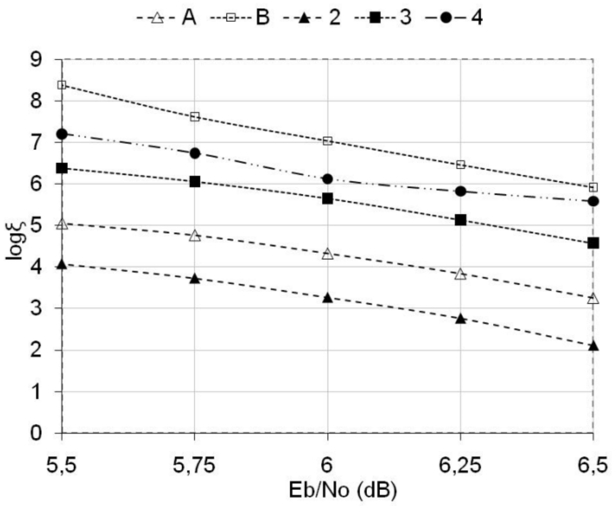

Fig. 3. Average complexity of discrepancy computation for RS code (120,104)

A (255,239) RS code is used for long hall optical transmission. A coding gain of 0.42 dB (FER of 10^{-2}) over the conventional algorithm is achieved with τ =3 (t=11) (see fig. 1). This coding gain is more than one of Koetter-Vardy algorithms with small multiplicities.

The simulation results for the (120,104) RS code used in WORM optical disks are shown in the figure 2. We see that a coding gain of 0.48 dB (FER of 10^{-3}) over the conventional decoding algorithm is achieved by the new algorithm with τ =3 (t=11).

In the fig. 3 ξ denotes an average number of GF arithmetical operations needed for calculation of discrepancies for one received codeword. It is assumed that one multiplication is equaled to m additions, and one division is equaled to m^2 additions, (m – number of bits in the RS-code symbol). The adding of soft decisions into the basic list decoding algorithm decreases in computational complexity by decimal order and more.

4 Implementation of the Soft Decoding Algorithm

The proposed algorithm can be implemented with simple hardware. A block diagram of a decoder is shown in fig. 4. The decoder consists of a data buffer, a syndrome calculator, a sorting circuit, a Galois processor, a discrete Fourier transform (DFT) circuit, an error position searcher and an error value calculator.

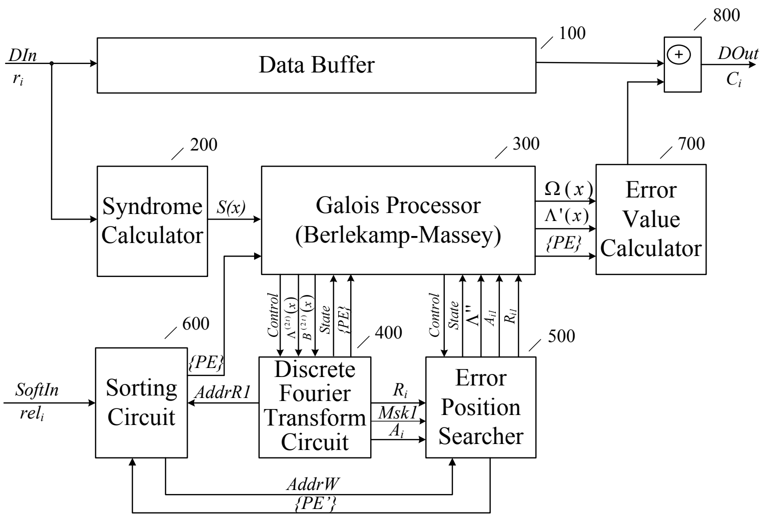

Fig. 4. A Block Diagram of the Reed-Solomon Soft Decoder

The decoder operates as a pipeline. Its units handle simultaneously the varied sequenced received codeword.

The data buffer, the syndrome calculator, the Galois processor and the error value calculator function as usual. Additionally, the Galois processor calculates $\Lambda^{(2t_C+2\tau)}(x)$, when there are $t_C+\tau$ error positions found.

The sorting circuit generates a sequence of codeword symbol position ordered by symbol reliability measures and stores it in the table $L[\ ]$.

The DFT circuit calculates Fourier transforms of the polynomials $\Lambda^{(2t_C)}(x)$ and $B^{(2t_C)}(x)$ and calculates coefficients R_i. Additionally, the DFT circuit calculates the inverse of the roots of the polynomial $\Lambda^{(2t_C)}(x)$ and checks that the number of roots is not equal to L_{2t_C} when $L_{2t_C} \leq t_C$.

The error position searcher fulfils steps 7-10 of the proposed algorithm. The $t_C+\tau$ error positions $\{PE'\}$ and the corresponding values Δ are found by the searcher. The error position searcher performs most operations in the present decoder.

5 Conclusion

The proposed soft decoding algorithm increases the coding gain of RS codes in telecommunication and storage systems without any modification of the existing standards.

Using symbol reliability measures reduces greatly the computational complexity of the algorithm comparing to basic list decoding algorithm.

References

1. Berlekamp, E.R.: Algebraic Coding Theory. McGraw-Hill, New York (1968).
2. Sugiyama, Y., Kasahara, M., Hirasawa, S., Namekawa, T.: A method for solving key equation for decoding Goppa codes. Inf. And Contr., v. 27, 87-99 (1975).
3. Blahut, K R.E.: Transform Techniques for Error Control Codes. IBM J. Res. Develop., vol.23, No.3, 299-315, (1979).
4. Egorov, S., Markarian, G.: An Algorithm for *t*+1 Error Correction in Reed-Solomon Codes. In Proc. ICC'04: 2004 IEEE International Conference on Communications, Paris, France, vol.2, pp. 651-655 (2004).
5. Egorov, S., Markarian, G.: Error Correction beyond the Conventional Error Bound for Reed-Solomon Codes. Journal of Electrical Engineering, vol. 54. No 11-12, 305-310 (2003).
6. Egorov, S.: A List Decoding Algorithm for Practical Reed-Solomon codes. In Proc. EWDTS 2013: 11th IEEE East-West Design & Test Symposium, Rostov-on-Don, Russia, pp. 275-278 (2013).
7. Berlekamp, E. R.: Bounded distance+1 soft-decision Reed-Solomon decoding. IEEE Trans. Info. Theory, vol. IT-42, 704 – 720 (1996).
8. Guruswami, V., Sudan, M.: Improved Decoding of Reed-Solomon and Algebraic-Geometry Codes. IEEE Trans. Inform. Theory, vol. 45, No. 6, 1757-1767 (1999).
9. Koetter, R., Vardy, A.: Algebraic soft-decision decoding of Reed-Solomon codes. IEEE Trans. Inform. Theory, vol. 49, No. 6, 2809-2825 (2003).
10. Gross, W.J. Kschischang, F.R. Koetter, R. Gulak, P.G.: Towards VLSI architecture for interpolation-based soft-decision Reed-Solomon decoders. J. VLSI Signal Processing, vol.39, No.1-2, 93-111 (2005).
11. Clark, G.C. Cain, J.B. Error-Correction Coding for Digital Communications. New York: Plenum Press (1982).
12. Gross, W.J. Kschischang, F.R. Koetter, R. Gulak, P.G.: Simulation Results for Algebraic Soft-Decision Decoding of Reed-Solomon Codes. In Proc. of the 21st Biennial Symposium on Communications, Queen's University, Kingston, Canada, pp. 356-360 (2002).