Buletinul Ştiinţific al Universităţii "Politehnica" din Timişoara

Seria ELECTRONICĂ şi TELECOMUNICAŢII
TRANSACTIONS on ELECTRONICS and COMMUNICATIONS

Tom 49(63), Fascicola 2, 2004

Performances of the Reed-Solomon Codes Decoded with the Guruswami-Sudan Algorithm

Zsolt Polgar, Ana Năstase, Vasile Bota [1]

Abstract – The Guruswami-Sudan (GS) decoding algorithm is a list-type decoding algorithm that corrects more errors than the "declared" capability, for certain coding rates of the Reed-Solomon (RS) codes. Using computer simulations, the paper presents a comparison between the correction capability and the processing time of the GS and Berlekamp-Massey (BM) algorithms. The simulations are employed to establish the optimum values of the GS parameters that ensure the maximum performance/processing time ratio. Some methods of changing the GS parameters, in terms of the packet-error length, which provide shorter decoding times, are also presented.

Keywords: RS codes, Guruswami-Sudan decoding algorithm, Berlekamp-Massey decoding algorithm, correction capability, decoding time.

I. INTRODUCTION

Considering an RS-type code C, defined over the Galois field F_q, with the parameters n – codeword length, k – information word length, d – Hamming distance, there are three possible definitions for such a code [1] [2], namely:

- Cyclic codes: if a code word $c \in C$ and τ is the cyclic shift operator, then $\tau(c) \in C$. The code word can be expressed as:

$$RS_C(k) = \left\{ (c_0, c_1, \ldots, c_{n-1}) : \sum_{j=0}^{n-1} c_j x^j = 0 \text{ for } \alpha, \alpha^2, \ldots, \alpha^{n-k} \right\} \quad (1)$$

- Evaluation codes: the code word is obtained by evaluating a polynomial f(x), defined by (2), associated to the information word v (v_0, v_1, . v_{k-1}), over the elements of F_q, as shown in (3):

$$f(x) = \sum_{j=0}^{k-1} v_j \cdot x^j \quad (2)$$

$$RS_G(k) = \left\{ \left(f(\alpha^0), f(\alpha^1), \ldots, f(\alpha^{n-1}) \right), \deg f < k \right\};\ f \in F_q[x] \quad (3)$$

- Codes dual to the evaluation codes.

Defining the RS codes as evaluation codes, leads to the possibility of employing list-type algorithms for their decoding, algorithms that provide higher performances than the classical ones, represented in this paper by the BM algorithm.

The list-type decoding algorithms [3] [5] operate with a decoding radius higher than $(d_{min}-1)/2$, delivering a list of possible code words. If the distances between code words are distributed in such a manner that the decoding list contains, in most cases, a single word, then this algorithm-type ensures a higher correction capability than the classical algorithms, such as BM.

II. THE GS ALGORITHM. MAIN ASPECTS

The operation steps of list-type GS decoding algorithm for the RS codes are [3] [4] [5]:

- let $(\alpha^0, \alpha^1, \ldots, \alpha^{n-1})$ be the elements of F_q, f(x) the polynomial corresponding to the information word (2) and $(\beta_0, \beta_1, \ldots, \beta_{n-1})$ the received code word. If a code word is correctly received, then relations (4) hold true:

$$\beta_i = f(\alpha^i);\ i \in [0, n-1] \quad (4)$$

- a two-variable interpolation polynomial Q(x,y), which has an m-order multiplicity zero in every point (α^i, β_i), is built.
- the polynomial Q(x,y) is decomposed in (y-f(x))-type (factorization), whit deg $f(x) \le k$; the polynomials f(x) obtained represent the code words from the decoding list.

A two variable polynomial, Q(x,y), is an ordered structure of two-variable monomials, expressed as:

$$Q(x,y) = \sum_{i,j \ge 0} a_{i,j} \cdot x^i \cdot y^j = \sum_{j=0}^{J} a_j \cdot \phi_j(x,y) \quad ; \quad (5)$$
$$1 = \phi_0(x,y) < \phi_1(x,y) < \phi_2(x,y) < \ldots < \phi_J(x,y);$$

J denotes the rank of the Q(x,y) polynomial and $\phi_J(x,y)$ is the leading monomial.

The monomials $\phi(x,y)$ are ordered according to their weighted degree, defined by:

$$\deg_w x^i y^j = u \cdot i + v \cdot j;\ w = (u, v) \quad (6)$$

There are two possible ordering rules, namely direct ordering (lex order) and reversed ordering (revlex order), defined by:

$$\text{lex order}: x^{i1} y^{j1} < x^{i2} y^{j2} \text{ if } ui_1 + vj_1 < ui_2 + vj_2 \text{ or}$$
$$ui_1 + vj_1 = ui_2 + vj_2 \text{ and } i_1 < i_2 \quad (7)$$
$$\text{revlex order}: \text{the same order but for } i_1 > i_2$$

[1] Technical University of Cluj Napoca, Communications Department, 26-28 G. Baritiu Str., 400027 Cluj Napoca, e-mail: Zsolt.Polgar@com.utcluj.ro

A significant theorem that, together with other theorems, secures the existence of an interpolation polynomial is [3]:

Theorem 1: Let $\{m(\alpha,\beta):(\alpha,\beta)\in F^2\}$ be the multiplicity function of the zeros of $Q(x,y)$ and $\phi_0<\phi_1<....$ an arbitrary monomial order. There always exists a polynomial $Q(x,y)$:

$$Q(x,y)=\sum_{i=0}^{C} a_i \cdot \phi_i(x,y) \qquad (8)$$

In (8), C is expressed by:

$$C=\sum_{\alpha,\beta}\binom{m(\alpha,\beta)+1}{2} \qquad (9)$$

The complete proof of the existence of the interpolation polynomial is to be found in [4] and [5].

The existence of a polynomial that could be decomposed in (y-f(x)) factors is secured by theorem 2 [3]:

Definition 1: For $Q(x,y)\in F[x,y]$ and $f(x)\in F[x]$ the Q-score of f(x) is defined as:

$$S_Q(f)=\sum_{\alpha} \text{ord zero}(Q:\alpha, f(\alpha)) \qquad (10)$$

Theorem 2:

If $f(x)\in F_k[x]$, $Q(x,y)\in F[x,y]$ and $S_Q(f) > \deg_{1,v} Q$ (11)

then y-f(x) is a factor of Q(x,y) ; v=k-1.

A thorough analysis of the factorization step is presented in [4] and [5].

One of the most efficient interpolation algorithms is the Koetter algorithm [3], which is defined by the pseudo-code below:

Koetter interpolation algorithm

- input data: L – number of code words in the list, $(\alpha_i, \beta_j)_{i=1}^{n}$ – interpolation points, $(m_i)_{i=1}^{n}$ – zero's multiplicity order, (1,k-1) – monomials weighted degree.

```
1.  FOR j=0 to L
    g_j=y^j
2.  FOR i=1 to n DO
2.      FOR (r,s)=(0,0) to (m_i-1,0) DO /*lex order
4.          FOR j=0 to L DO
5.          Δ_j=D_{r,s}g_j(α_i,β_j)
6.      J={j: Δ_j ≠0}
7.      IF J≠Φ
8.          j*=min_rank {g_j:j∈J}
9.          f=g_{j*} ; Δ=Δ_{j*}
10.         FOR j∈J DO
11.             IF (j≠j*)
12.                 g_j=Δ·g_j+Δ_j·f
13.             ELSE IF (j=j*)
14.                 g_j=Δ·(x+α_i) ·f
15. Q_0(x,y)=min_rank{g_j(x,y)}  /* the interpolation polynomial
```

One of the best factorization algorithms, the Roth-Ruckenstein [3], was used in the present analysis.

The bounded values of two significant parameters of the GS algorithm, the number of code words in the decoding list, L, and the decoding radius, r_d, are given by [3]:

$$L_{max}=\left\lfloor\sqrt{\frac{n}{k-1}\cdot m\cdot(m+1)+\left(\frac{k+1}{2k-2}\right)^2}-\frac{k+1}{2k-2}\right\rfloor<(m+0.5)\cdot\sqrt{\frac{n}{k-1}} \qquad (12)$$

$$n-\left\lfloor\sqrt{(k-1)\cdot\frac{m+1}{m}}\right\rfloor\le r_d\le n-1-\left\lfloor\sqrt{(k-1)\cdot\frac{m+1}{m}}-\frac{k-1}{m}\right\rfloor \qquad (13)$$

III. ANALYSIS OF THE SIMULATION RESULTS

The main goal of this paper is to compare, by computer simulations, the correction capability and processing time of the GS and BM (representative for the classical algorithms) RS decoding algorithms. The analysis is intended to establish the optimum values of the parameters of the GS algorithm, for which a maximum ratio correction capability/decoding time is accomplished and to elaborate some "thumb rules" for adapting these parameters, so that shorter decoding times could be attained.

The software simulator, that can operate in the Galois fields $GF(2^3)$, $GF(2^4)$, $GF(2^5)$, $GF(2^6)$ and $GF(2^8)$, performs the following functions:

- generation of a symbol-sequence represented on the number of bits corresponding to the employed Galois field.
- RS encoding (cyclic code for the BM or evaluation code for the GS), depending on the decoding algorithm employed.
- serialization of the coded bits, generation of the packet-errors and their insertion in the coded bits.
- GS or BM decoding and computation of the parameters of the simulated transmission, namely: bit and symbol error rates, the ratio of the correction capability of the GS algorithm versus the correction capability of the BM algorithm, the numbers of words in the decoding list and erasures, both for the GS algorithm.

The generation of the packet-errors, which simulates the transmission channel, is performed according to the impulse noise models employed for the xDSL transmissions [6]. This model was adopted, with several simplifications, since it is a representative one for transmission systems employing RS codes as outer codes. The main features of algorithm that generates the packet-errors are:

- the distance in symbols between two packet-errors has a Poisson distribution, with a modifiable average value λ. In the simulations performed, the value of λ equaled the number of symbols of two code words, for each GF.
- the packet-error length, in bits, has a gaussian distribution, defined by the average value τ and variance σ. The value of τ equaled $t_b \cdot q$, t_b denoting the number of error-symbols that could be corrected by the classical decoding algorithms (e.g. BM) and q denoting the number of bits/character of the GF employed. The value of σ was set according to the estimated correction capability of the GS algorithm.
- the positions of the errors inside the packet are random, being distributed according to a uniform law.

A. *Decoding Capability of the GS Algorithm*

The performances of the GS algorithm were evaluated for RS codes with the coding rate $R_c \in [0.3, 0.65]$. The parameters that indicate the correction capability are the minimum and the maximum decoding radius, computed using (12), and the correction rate R_d (obtained by simulations). The R_d parameter is defined as the ratio of the number of error words after the GS decoding and the number of error words that have a number of error symbols higher than t_b (the decoding radius of the classical algorithms), before the decoding. The codes with $R_c < 0.3$ were not considered, since they are of low practical importance. As for the codes with $R_c > 0.5 - 0.6$ (depending on the employed GF), the correction capability of the GS algorithm is the same with the one of the BM algorithm.

Note: In figs. 1-5 m denotes the multiplicity order of the zeros in the GS algorithm; m = 0 is actually equivalent to the BM algorithm; for this algorithm:

$$r_{min} = r_{max} = t_b = n(1-R_c)/2; \quad R_d = 1; \qquad (14)$$

The variance σ of the packet-errors, depicted in figs. 1.b, 2.b, 3.b, 4.b, for each R_c and GF, was set in all simulations to a value that provides packet lengths close to the r_{max} of the GS algorithm.

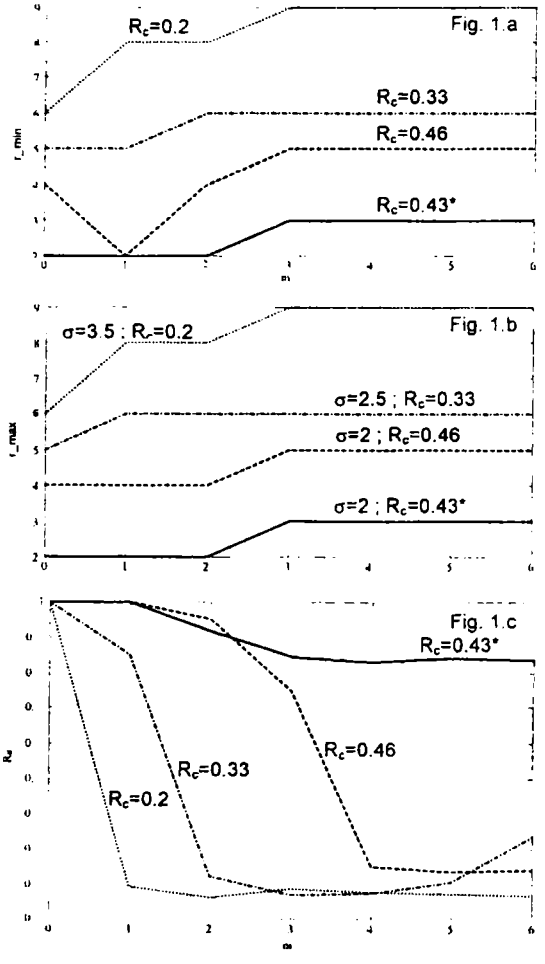


Fig.1 Minimum, r_{min} (1.a), maximum decoding radius r_{max} (1.b), correction rate R_d (1.c) in terms of m ; RS codes in Galois GF(2^3) and GF(2^4) ; * denotes codes defined in GF(2^3);

Fig.1.c shows that for RS codes defined in GF(2^3) and GF(2^4), m has to be set to 3 or 4, for R_c close to 0.5, and to 1 or 2 for R_c close (or smaller) than 0.3. The increase of m above a certain limit does not bring a performance improvement, but it might lead to a decrease of performances (see R_c = 0.33). A more complete evaluation of the GS decoder requires the consideration of the r_{min} and r_{max}, as well; good decoding performances should be accomplished when the two parameters take equal or close values. The optimum values of m can not be established by considering only the r_{min} and r_{max} parameters of the code, as shown by R_c = 0.46.

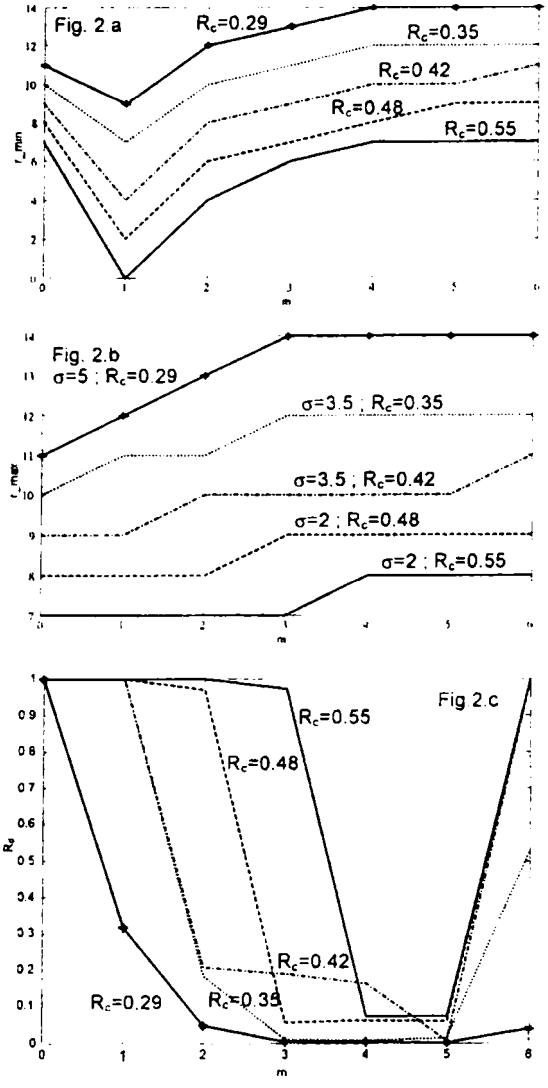


Fig.2 Minimum, r_{min} (2.a), maximum decoding radius r_{max} (2.b), correction rate R_d (2.c) in terms of m ; RS codes in Galois GF(2^5);

The values of R_d, see fig.2.c, indicate that for RS codes defined over GF(2^5) the optimum values of m are m = 3 - 4 for R_c close to 0.5, m = 2 - 3 for R_c around 0.3 and m = 4 - 5 for R_c around 0.4. There should be noticed that for m=6, the performances of the GS decoder exhibit a significant decrease, especially for high values of the coding rate R_c.

For RS codes defined in GF(2^5) having the mentioned R_c and for the optimum values of m, the

decoding radius of the GS lies between $r_{min} \geq t_b$, and $r_{max} = r_{min}+1$ or $r_{min}+2$. The values of σ parameter of the error-packet for the considered rates are given in fig.2.b.

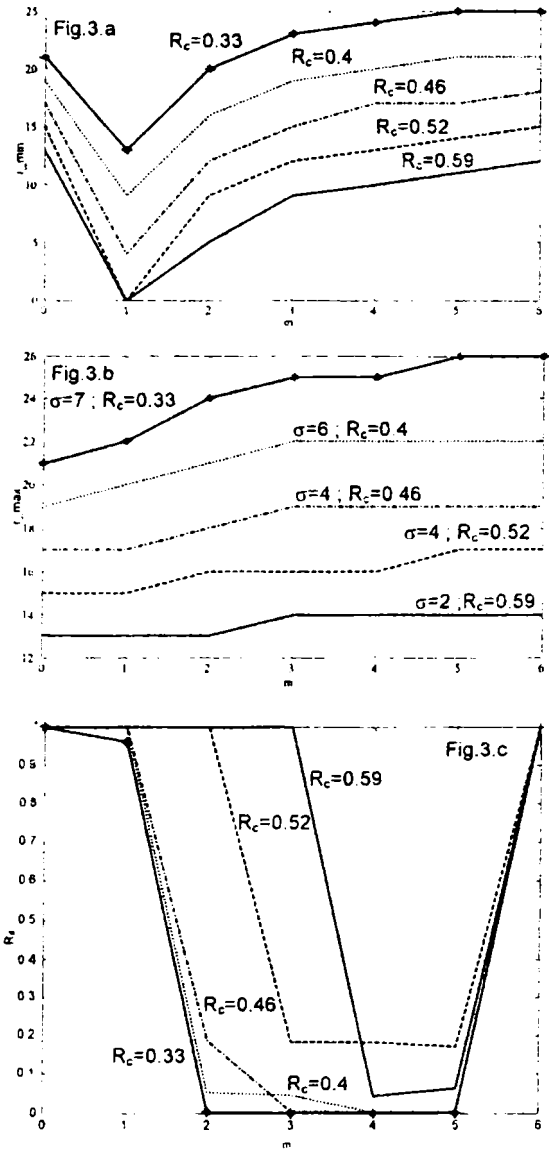


Fig.3 Minimum, r_{min} (3.a), maximum decoding radius r_{max} (3.b), correction rate R_d (3.c) in terms of m ; RS codes in Galois GF(2^6).

Considering the RS codes defined in GF(2^6), see figs. 3, they exhibit a clear separation of the optimum values of m, in terms of the coding rate R_c. For $R_c \geq 0.5$, optimum m equals 3 or 4, but for $R_c \leq 0.45$, optimum m equals 2 or 3. Sometimes, see $R_c = 0.4$, m = 4 provides better performances at the expense of a longer decoding time.

The codes defined in GF(2^6) exhibit the same decrease of performance for higher values of m (e.g. m = 6), as the ones defined in GF(2^5): for the considered values of R_c, the performances secured by the GS become equal to the ones of the BM. The values of σ parameter of the error-packet for the considered rates are given in fig.3.b.

The performance loss exhibited by the GS algorithm for high values of m, regardless the coding rate, could be explained by the incomplete factorization, see (11), the requirements for the interpolation being ensured by a proper choice of the number of words within the decoding st.

A primary analysis of the interpolation algorithm presented in Section II and of the properties of the two-variable polynomials [3] leads to the following:

- the number of iterations, n_{it}, performed by the interpolation algorithm for n-symbol code words and multiplicity order of zeros equaling m, is :

$$n_{it} = n \cdot \frac{m \cdot (m+1)}{2} \tag{15}$$

- the initial polynomials of Koetter interpolation algorithm, for maximum L words in the final decoding list, are:

$$p_0(x,y) = 1, p_1(x,y) = y, p_2(x,y) = y^2, \ldots, p_2(x,y) = y^L \tag{16}$$

- supposing that the values of Δ, computed within the Koetter algorithm, never equal zero (supposition that does not always hold true), then after $L \cdot (L+1)/2 \cdot (k-1)$ iterations all polynomials will have the same degree $L \cdot (k-1)$. The leading monomials of these polynomials are:

$$lp_0(x,y) = x^{L \cdot (k-1)}, lp_1(x,y) = x^{(L-1)(k-1)} \cdot y,$$
$$lp_2(x,y) = x^{(L-2)(k-1)} \cdot y^2, \ldots, lp_L(x,y) = y^L \tag{17}$$

- taking into account that each iteration increases the degree of the polynomial with the minimum rank and that a polynomial with a higher degree also has a higher rank, then we may assert that the increase of the degree of each polynomial will require L+1 iterations. By the end of the algorithm the degree, deg_{min}, of the minimum-degree polynomial would be:

$$deg_{min} = \left| \frac{n_{it} - L \cdot (L+1)/2 \cdot (k-1)}{L+1} + L \cdot (k-1) \right| \tag{18}$$

- the minimum value of the S_Q parameter (10) of an inter olation ol nomial associated to a n-s mbol code word and to a multiplicity order of zeros equaling m and to a decoding radius r, is:

$$S_{Q\,min} = (n-r) \cdot m \tag{19}$$

- from the factorization requirements we have:

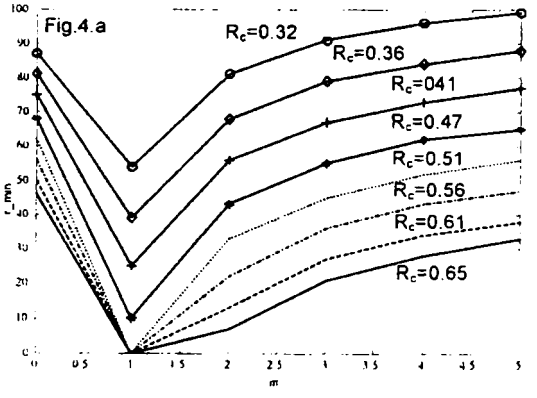
$$\frac{S_{Q\,min}}{deg_{min}} \cong \frac{\left(1 - \frac{r}{n}\right) \cdot \frac{(L+1)}{2}}{(m+1) + \frac{L \cdot (L+1) \cdot R}{m}} > 1 \tag{20}$$

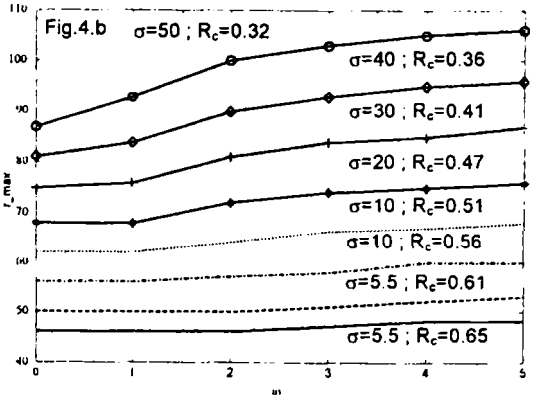
The values of the ratio defined in (20), for the codes of figs. 2 and 3 and for various values of m, are smaller than 1 (approximately equal, but smaller). There should be noted that the considerations above are not complete, since it did not considered that the evolution of polynomials degrees within the interpolation algorithm would be different, mostly because of the fact that Δ might equal zero quite often, changing the evolution of the polynomials degrees

(see the interpolation algorithm in Section II), and decreasing significantly the values of deg_{min}. Also, the value of S_Q might be higher than the value computed by (19). Nevertheless, the considerations above show that, for different coding rates and various values of m, there is a possibility that the GS algorithm would not be effective, even for a high decoding radius. The suppression of this limitation of the value of m may be accomplished by using different values of m for every interpolation point, values chosen depending of the channel characteristics [5]. Obviously, this approach would complicate the implementation of the decoding GS algorithm.

Unlike the previous cases, for optimal values of m, the codes defined in GF(2^6) have $r_{min} < t_b$, but the difference $r_{max} - r_{min}$ takes values between 3 and 6. The difference $r_{max} - t_b$ takes values between 0 and 3. So, for the codes defined in GF(2^6) the optimum values of m cannot be evaluated only be considering the limit values of the decoding radius, r_{min} and r_{max}.

Note: the relation $r_{min} < t_b$ does not imply that the GS algorithm could not correct t_b symbol-errors (the "declared" correction capability of the code), so practically one should consider that $r_{min} \geq t_b$. The values of r_{min} and r_{max} provided by (13) evaluate the possibility of the GS algorithm to correct more errors than the classical algorithms. As for the RS codes defined in GF(2^8), see figs. 4, the considerations regarding r_{min} and r_{max}, presented above, are still valid. There should be mentioned that, for the optimal values of m, $r_{min} \leq t_b$, and the difference $r_{max} - r_{min}$ takes values higher or equal than 20, and the difference $r_{max} - t_b$ lies between 2 and 13.





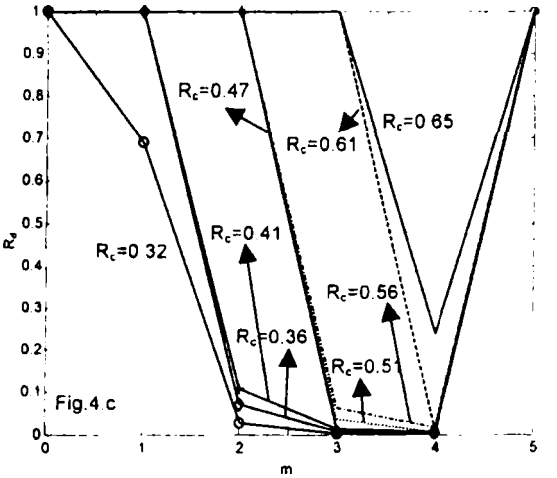


Fig.4 Minimum, r_{min} (4.a), maximum decoding radius r_{max} (4.b), correction rate R_d (4.c) in terms of m ; RS codes in Galois GF(2^8).

Fig. 4.c shows three optimum values of m, depending of the coding rate R_c, for the codes defined in GF(2^8). For coding rates higher or equal to 0.6 the optimum value of m is 4, for $R_c \in (0.6, 0.45)$ the optimum value of m is 3, and for coding rates ranging between 0.3 and 0.45, the optimal m equals 2. The figure also shows that, similar to the codes defined in GF(2^6), the maximum limit of m falls to 5, for the coding rates considered.

The comparison of the results presented in figs. 1.c - 4.c show that the coding rate for which the GS decoding algorithm provides better performances than the classical decoding algorithms increases with the increase of dimension of the Galois field in which the RS codes are defined.

Regarding the number of words in the decoding list, the simulations performed by the authors show that for the RS codes defined in GF(2^5) and in the higher fields, the number of the words in the list equals 1, with very few exceptions, when then list contains more than one code word. As for the codes defined in GF(2^3) and GF(2^4), there are more cases when the decoding list contains more than one code word, but their percentage is still small, about 1%. As a general conclusion, if the GS decoding algorithm can not correct a code word, this fact is owned to an unsuccessful interpolation or factorization and, quite seldom, to the presence of more than one code words in the decoding list.

B. *Evaluation of the GS algorithm decoding time*

The references [3] [4] [5] present some considerations regarding the number of operations performed by the GS algorithm, which affect significantly the decoding time, but these considerations do not include a comparison to the decoding time required by the classical RS decoding algorithms. The software simulator implemented by the authors includes a RS decoder based on an optimized version of the Berlekamp-Massey (BM), described in [1]. For comparison, the simulations using the BM decoding algorithm were performed in the same conditions as the ones using the GS algorithm.

The evaluation of the decoding time implied the measurement, for a certain number of code words, of the simulation time t_{sim}, and of the time required for encoding and error-pattern insertion t_{aux} ; for the measurement of t_{aux} the decoding procedures were removed from the simulation program. There should be noted that the time required to decode a correct code word differs from the time required to decode an error code word for both algorithms, especially for the GS algorithm. The ratio between the average decoding times, t_{dec}, of the two algorithms is expressed by:

$$t_d = \frac{t_{decGS}}{t_{decBM}} = \frac{t_{simGS} - t_{auxGS}}{t_{simBM} - t_{auxBM}} \approx \frac{t_{simGS}}{t_{simBM}} \quad (21)$$

Fig. 5 presents the variation of the ratio t_d (expressed on a logarithmic scale) between the average decoding times of the GS and BM algorithms in terms of m, for various coding rates and for codes defined in several Galois fields.

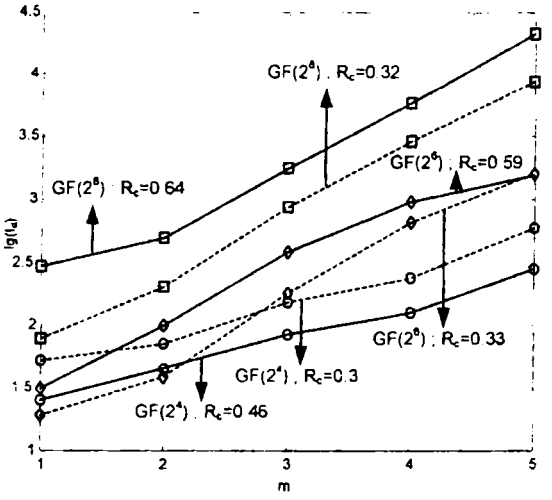


Fig.5 $\lg(t_d)$ ratio between the average decoding time of the GS and BM algorithms, in terms of m, for various coding rates and for RS codes defined in several Galois fields

The results presented in fig. 5 show that the decoding time required by the GS algorithm is much larger than the one required by the BM algorithm. The t_d ratio increases significantly with the increase of m and with the increase of dimension of the Galois field employed. The increase of the coding rate for codes defined over GF higher than $GF(2^4)$ also increases the value of t_d. By changing the coding rate from 0.3 to 0.6 for these codes, involves an increase of the ratio t_d by a factor ranging from 2 to 3. There should be mentioned that the implementations of the two algorithms were optimized to the best knowledge of the authors.

The results displayed in fig. 5 underline the importance of establishing optimal values for the parameter m and the necessity of finding some variants of the GS algorithms (decoding strategies), which should require a decoding time as small as possible. The authors have considered three possible variants to accomplish the GS decoding, namely:

- employing the same value of m for the decoding of every code word; this variant would require a very large average decoding time, even larger than the ones presented in fig. 5, because even the correctly received code words would be decoded in a very long time.
- the successive increase of the value of m, from 1 to a maximum optimal value. The decoding is stopped when the decoding list contains at least one code word; this approach would require a smaller average decoding time for packet-errors with relatively small lengths, compared to the maximum packet length for which a successful GS decoding is accomplished.
- the employment of two values for parameter m, namely 1 and an optimum value m_{opt}. The correct code words and the ones affected by a small number of errors (equal or higher than t_b) would be decoded using m=1, and the code words with more errors would be decoded with $m = m_{opt}$; this last option should be employed if the decoding with m=1 generates no code word in the decoding list. This variant of employing the GS algorithm provides a smaller average decoding time for long error-packets, compared the maximum packet length for which a successful GS decoding is accomplished. The results displayed in fig. 5 were obtained using this decoding varian .

IV. CONCLUSIONS

The computer simulations performed by the authors showed that the GS decoding of RS codes, defined in the $GF(2^4)$ $GF(2^5)$ $GF(2^6)$ and $GF(2^8)$, provides a significantly greater correction capability than the BM decoding, for coding rates ranging between 0.3 and 0.6. The improvement becomes more obvious as the coding rate decreases and the dimension of the Galois field increases. The optimum values of the factor m (zeroes multiplicity order) for which a maximum correction capability/decoding time ratio is accomplished, are also presented in the paper. As for the decoding time, the simulations performed showed that the time required by the GS is significantly longer than the one required by the BM algorithm. The paper presents some decoding strategies for the GS that lead to a significant decrease of its decoding time for various lengths of the packet-errors.

REFERENCES

[1] M.E.O'Sullivan, *Coding Theory*, http://www-roham.sdsu.edu/~mosulliv/Courses/coding02.html

[2] J.I.Hall, *Notes on Coding Theory*, http://www.mth.msu.edu/~jhall/classes/codenotes/coding-notes.html.

[3] R.J. McEliece, "The Guruswami-Sudan Decoding Algorithm for Reed-Solomon Codes", *IPN Progress Report 42-153*, 15 May, 2003.

[4] M Sudan, "Decoding of Reed Solomon codes beyond the error-correcting bound", *Journal of Complexity*, vol. 13, 1997

[5] V. Guruswami, M Sudan, "Improved Decoding of Reed-Solomon Codes and Algebraic Geometry Codes", *IEEE Trans. Inform. Theory*, vol. 45, no 6, September 1999

[5] W. Gross, Fr. Kschischang, R. Koetter, P.G. Gulak, "Applications of Soft-Decision Decoding of Reed-Solomon Codes", submitted to *IEEE Trans. Comm.* July 2003, http://www.macs.ece.mcgill.ca/~wjgross/papers/gkkg_tc.pdf.

[6] W. Henkel, T. Kessler, "A wideband impulsive noise survey in the German telephone network-Statistical description and modeling", AEU, vol. 48, no. 6, Nov./Dec. 1994.