Trap-decoding of Reed-Solomon codes and its burst-error-correcting performance evaluation

Dunwei Xue, Liuguo Yin

Abstract—The communication and storage systems may be corrupted by bursts of noise. These bursts may be long in duration, resulting in a significant degradation for the system performance. Reed-Solomon (RS) codes are proven to be very effective in correcting burst errors. According to the Singleton bound, the maximum length of burst errors that can be corrected by an (n, k) RS code is (n-k)/2 symbols. However, it turns out that, if correlation between erroneous symbols within bursts is considered and well used, variables for burst locations will be decreased and, accordingly, decoding capability may be enhanced with increased length of correctable bursts. As such, we propose a new trap-decoding algorithm for RS codes in this paper. It is shown that, for (n, k) RS codes, this algorithm can correct continuous burst errors with length that approaches to n-k symbols with fairly low miscorrection probability, achieving a good performance over long-burst channels. Moreover, we will further show that the complexity of the proposed algorithm is fairly lower, and the decoding delay is much less than those of existing burst-error-correcting algorithms.

Keywords—Reed-Solomon (RS) code, burst errors, trap decoding algorithms, burst-error-correcting capability.

I. Introduction

THE data transmission and storage systems may be corrupted by burst errors, which may be caused by linear noise, synchronization errors in demodulation, or fading effects in wireless systems, and so on. These bursts may be random with different pattern and quite long in duration. Usually, much redundancy should be introduced to eliminate the effects of these long burst errors, resulting in a significant reduction in transmission rate and storage capacity [1-4].

It is well known that RS codes are quite effective in correcting burst errors, and the maximum length of a burst that can be corrected by an (n, k) RS code is symbols according to the Singleton bound [1]. Recently, studies have shown that, when the correlation within burst errors are properly incorporated, the error correcting capability of RS codes can be increased beyond the Singleton bond with a very small miscorrection probability. A such burst-error-correcting algorithm was firstly presented by J. Chen and P. Owsely [2], and then modified by E. Dawson and A. Khodkar [3]. In these decoding algorithms, the syndrome polynomial is obtained by the use of the Fourier transform of the received polynomial in Galois field. Then, a search procedure is implemented to find all possible burst locations. After that, every burst pattern that corresponds to a burst location is computed with inverse Fourier transform. Finally, burst errors in the received polynomial are corrected with the burst pattern that has the shortest burst length. On the other hand, it is noted that these decoding algorithms are rather too complex and likely result in a long decoding delay.

In this paper, we propose a new decoding algorithm for burst-error-correction with RS codes. Theoretical analysis is given to show that the proposed algorithm can effectively correct long burst errors with a very small miscorrection probability even when the burst length approaches to $n - k$ symbols. By Comparison we further shows that, as far as VLSI implementation is concerned, the complexity of the proposed algorithm is fairly lower, and the decoding delay is much less than those of existing burst-error-correcting algorithms.

The rest of this paper is organized as follows. In section II, we will present the proposed new decoding algorithm. Then the miscorrection probability with the proposed algorithm is analyzed in section III. Section IV provides the simulation results for the proposed algorithm over a bursty channel that presented in [5]. After that, section V provides a comparison between the proposed algorithm and the existing algorithms in terms of computational complexity. Finally, conclusion is drawn in section VI.

II. The Proposed Trap-Decoding Algorithm

It is noted that Reed-Solomon code is a set of cyclic code. Hence, with a cyclical shift, a RS code polynomial will become another code polynomial. Assume that a burst pattern has a format of the following polynomial [2]:

$$E(x) = e_l x^{i+l-1} + e_{l-1} x^{i+\nu-2} + \cdots + e_1 x^i \tag{1}$$

where l is the length of the burst, e_r is the *r-th* error value, and i is the initial burst location. Let the code polynomial of an (n, k) RS code be *C(x)* and the received polynomial be *R(x)*. Then,

$$\begin{aligned} R(x) &= C(x) + E(x) \\ &= c_{n-1}x^{n-1} + \cdots + (c_{i+l} + e_l)x^{i+l-1} \\ &\quad + \cdots + (c_i + e_1)x^i + \cdots + c_0 \end{aligned} \tag{2}$$

Manuscript received August 3, 2011: Revised version received October 30, 2011. This work was supported in part by National Natural Science Foundation of China (No. 61021001, 61101072) and Program for New Century Excellent Talents in University (NCET).

Dunwei Xue is with School of Electronic and Information Engineering, Beihang University, Beijing, 100191, CHINA (Telephone: +8610-6277-3410; fax: +8610-6277-0528; e-mail: xuedunw@yahoo.com.cn).

Liuguo Yin is with School of Aerospace, Tsinghua University, Beijing, 100084, CHIAN (e-mail: lgyin@wmc.ee.tsinghua.edu.cn).

According to cyclic property of RS codes, the initial location of the burst in the received polynomial becomes x^0 if we cyclically shift left the received polynomial by n-i symbols. As a result, the received polynomial after the shift is:

$$\begin{aligned} R'(x) &= R(x)\cdot x^{n-i} \quad mod \quad x^n - 1 \\ &= C(x)\cdot x^{n-i} + E(x)\cdot x^{n-i} \quad mod \quad x^n - 1 \\ &= c_{i-1}x^{n-1} + \cdots + c_0 x^{n-i} + c_{n-1}x^{n-i-1} + \cdots \\ &\quad + (c_{i+l} + e_l)x^{l-1} + \cdots + (c_i + e_1) \\ &= C'(x) + E'(x) \end{aligned} \tag{3}$$

Likewise, let the generator polynomial be

$$G(x) = x^{2t} + g_{2t-1}x^{2t-1} + \cdots + g_1 x + g_0 \tag{4}$$

where t is the number of random errors that can be corrected by the RS code generated with this polynomial. Then, we compute the syndrome polynomial by dividing the received polynomial with the generator polynomial. By doing so, we obtain:

$$S(x) = R(x) \quad mod \quad G(x) \tag{5}$$

Based on formula (4)-(5), the relationship between the burst polynomial and the syndrome polynomial can be written as:

$$\begin{aligned} S'(x) &= R'(x) \quad mod \quad G(x) \\ &= C'(x) + E'(x) \quad mod \quad G(x) \\ &= E'(x) \end{aligned} \tag{6}$$

It turns out from formula (6) that the burst polynomial may be obtained through the syndrome polynomial of the cyclically shifted received polynomial. Consequently, we propose the new trap-decoding algorithm as follows:

Step 1) Initialize the previous minimum burst length to be $2t$, and the corresponding burst error values to be zeros;

Step 2) Compute the syndrome polynomial with the received polynomial, and at the same time computing the burst length of the syndrome polynomial. If the new burst length is not less than the previous minimum burst length, go to step 3); Else, substitute the burst length and the burst error values with the currently computed results, and at the same time count down the current shift times that corresponds to the lowest error location;

Step 3) If the total shift length is less than the code length, shift the received polynomial left by a symbol length, then go to step 2); or go to step 4) else;

Step 4) Correct burst errors with the current shortest error pattern and the lowest burst location that obtained from above steps;

Step 5) If this is not the end of the decoding procedure, go to step 1); or exit else.

It is noted that the received polynomial after the decoding procedure becomes a code polynomial no matter which $E'(x)$ from formula (6) is selected. Therefore, the decoding procedure is in fact to select a code polynomial according to the received polynomial on condition that the corresponding burst has the shortest length. Since the minimum weight of an (n, k) RS codes is $2t+1$, for a burst with a length of l, and $l \leq t$, no miscorrection occurs with the proposed decoding algorithm. But, as $l > t$, a miscorrection may occurs if a code polynomial $C_0(x)$ exists which meets the condition that the burst polynomial $C_0(x)+E(x)$ has a length that is less than l. In particular, when the burst length of $C_0(x)+E(x)$ is equal to l, the decoder may randomly take either $E(x)$ or $C_0(x)+E(x)$ as the maximum likelihood burst polynomial, resulting in a miscorrection probability of 1/2. In the next Section, we will prove that, with the proposed algorithm, the probability of miscorrection is very small even when the length of a burst approaches to $n - k$.

III. Miscorrection Probability of The Proposed Decoding Algorithm

From the above discussion, we know that any burst that may lead to a miscorrection should be a piece of a code polynomial, with a length that is greater than t. To be convenient, we call this kind of polynomials to be mis-correcting polynomials. Moreover, a mis-correcting polynomial may have two parts: one is the burst polynomial $\varepsilon_l(x)$ that actually occurs in the received polynomial; the other is the error polynomial, $(c_0(x) + \varepsilon_l(x))$, which is resulted from the decoding procedure. Then, we define the pattern distance as the distance from the lowest non-zero element of $\varepsilon_l(x)$ to the lowest non-zero element of $(c_0(x) + \varepsilon_l(x))$ in a mis-correcting polynomial. It turns out that the pattern distance won't be changed during a cyclic shift of a mis-correcting polynomial. As such, two mis-correcting polynomials may be viewed as containing two distinct bursts if there is:

1) Difference in pattern distances;
2) Difference in the beginning location of a burst;
3) Difference in the length of a burst pattern;
4) Difference in the length of an error pattern;
5) Difference in the locations of zero-value elements in a burst pattern;
6) Difference in the locations of zero-value elements in an error pattern.

Let $\{C_e(i, k)\}$ be the set of mis-correcting polynomials. We further assume that the non-zero elements in burst patterns distribute in the information field of a code polynomial with locations of $a_1, a_2, \cdots, a_i$; and the non-zero elements in the corresponding error pattern distribute in the parity check field with locations of $c_1, c_2, \cdots, c_{2t-k}$. Likewise, define the number of the elements in $\{C_e(i, k)\}$ to be $N_0(i, k)$, where i is the weight of a burst pattern, k is the weight of an error pattern. We then deduct in the following a useful formula for computing $N_0(i, k)$, which is a key to the computation of miscorrection probability.

Select $(g_{a_1}, g_{a_2}, \cdots, g_{a_i})$ from a set of independent basis code vectors of the (n, k) RS code vector space, that is:

$$\begin{cases} g_1 = (\alpha^0 \; 0 \; 0 \cdots 0 \; \vdots \; g_{11} \; g_{12} \cdots g_{12t}) \\ g_2 = (0 \; \alpha^0 \; 0 \cdots 0 \; \vdots \; g_{21} \; g_{22} \cdots g_{22t}) \\ \vdots \\ g_k = (0 \; 0 \; 0 \cdots \alpha^0 \; \vdots \; g_{k1} \; g_{k2} \cdots g_{k2t}) \end{cases} \tag{7}$$

Then, all the coefficient vectors of the mis-correcting polynomials in *$\{C_e(i, k)\}$* should be the linear combinations of $(g_{a_1}, g_{a_2}, \cdots, g_{a_i})$. Now, we construct matrix G_i as the below:

$$G_i = \begin{pmatrix} g_{a_1} \\ g_{a_2} \\ \vdots \\ g_{a_i} \end{pmatrix} = \begin{pmatrix} 0, \cdots, \alpha^0, \cdots, 0, \cdots, 0, \cdots, 0 \vdots g_{a_1 1}, g_{a_1 2}, \cdots, g_{a_1 2t} \\ 0, \cdots, 0, , \cdots \alpha^0, \cdots, 0, \cdots, 0 \vdots g_{a_2 1}, g_{a_2 2}, \cdots, g_{a_2 2t} \\ \vdots \quad \vdots \qquad \vdots \\ 0, \cdots, 0, \cdots, 0, \cdots, \alpha^0, \cdots, 0 \vdots g_{a_i 1}, g_{a_i 2}, \cdots, g_{a_i 2t} \end{pmatrix} \tag{8}$$

Adding the first row of matrix G_i to the second row, the third row, ……, the *i-th* row after respectively multiplying with coefficients $g_{a_2 c_1} / g_{a_1 c_1}$, $g_{a_3 c_1} / g_{a_1 c_1}$, ……, $g_{a_i c_1} / g_{a_1 c_1}$, a new matrix $G_i^{(1)}$ is generated as given by,

$$G_i^{(1)} = \begin{pmatrix} g_{a_1} \\ g_{a_2}^{(1)} \\ \vdots \\ g_{a_i}^{(1)} \end{pmatrix} = \begin{pmatrix} 0, \cdots, \quad \alpha^0 \quad, \cdots, 0, \cdots, 0, \cdots, 0 \vdots g_{a_1 1}, g_{a_1 2}, \cdots, g_{a_1 c_1}, \cdots, g_{a_1 2t} \\ 0, \cdots, g_{a_2 c_1} / g_{a_1 c_1}, \cdots, \alpha^0, \cdots, 0, \cdots, 0 \vdots g_{a_2 1}^{(1)}, g_{a_2 2}^{(1)}, \cdots, 0 \quad, \cdots, g_{a_2 2t}^{(1)} \\ \vdots \quad \vdots \qquad \vdots \\ 0, \cdots, g_{a_2 c_1} / g_{a_1 c_1}, \cdots, 0, \cdots, \alpha^0, \cdots, 0 \vdots g_{a_i 1}^{(1)}, g_{a_i 2}^{(1)}, \cdots, 0 \quad, \cdots, g_{a_i 2t}^{(1)} \end{pmatrix} \tag{9}$$

It is observed that in matrix $G_i^{(1)}$, elements located in the column headed by $g_{a_1 c_1}$ are zeros (except $g_{a_1 c_1}$ itself), while other parity check symbols are non-zero elements because the weight of any code polynomial should not less than $2t+1$, and the weight of the information symbols is two. Similarly, adding the second row to the third row, the fourth row, ……, the *i-th* row of matrix $G_i^{(1)}$ after multiplying with coefficients $g_{a_3 c_2}^{(1)} / g_{a_2 c_2}^{(1)}$, $g_{a_4 c_2}^{(1)} / g_{a_2 c_2}^{(1)}$, …, $g_{a_i c_2}^{(1)} / g_{a_2 c_2}^{(1)}$, respectively, we get matrix $G_i^{(2)}$, in which elements in the columns including $g_{a_1 c_1}, g_{a_1 c_2}$, from the third row to the *i-th* row, are zeros. Repeat this procedure until matrix $G_i^{(2t-k)}$ is obtained, in which code symbols in the columns including $g_{a_1 c_1}, g_{a_1 c_2}, \cdots, g_{a_1 c_{2t-k}}$ from the *(2t-k+1)-th* row to the *i-th* row are zeros. Obviously, the lower $i+k-2t$ row vectors of matrix $G_i^{(2t-k)}$ are linearly uncorrelated vectors and all the coefficient vectors of the mis-correcting polynomials in set *$\{C_e(i, k)\}$* may be expressed as the linear combinations of these $i + k - 2t$ mis-correcting vectors. If $i+k-2t>1$, j ($j \le i+k-2t-1$) locations may be randomly selected from non-zero parity check locations to continue above produce. By doing so, matrix $G_i^{(2t-k-j)}$ is obtained. All the error patterns of the lower $i+(k-j)-2t$ row vectors weights $k - j$ in $G_i^{(2t-k-j)}$. As a result, the set of mis-correcting polynomials whose coefficients are obtained from all linear combinations of the lower $i+k-2t$ rows of matrix $G_i^{(2t-k)}$ includes mis-correcting polynomials that belong to sets *$\{C_e(i, k-1)\}$, $\{C_e(i, k-2)\}$,…, $\{C_e(i, 2t - (k-1))\}$*, as well as, all the polynomials that belong to *$\{C_e(i, k)\}$*. The relationship between the numbers of these different mis-correcting polynomial sets is given by

$$N_0(i,k) = (q-1)^{i+k-2t} - \sum_{j=1}^{i+k-2t-1} \binom{k}{j} N_0(i, k-j) \tag{10}$$

where

$$\binom{k}{j} = \frac{j!}{k!(j-k)!} \tag{11}$$

and recall that q is the number of field elements of *GF(q)*. By the use of (10)-(11), the number of elements in $\{C_e(i,k)\}$ may be computed iteratively.

Consequently, let the number of mis-correcting polynomials be $N(l,i,k)$, in which the locations of non-zero elements within burst $\varepsilon_l(x)$ are predefined, i is the weight of $\varepsilon_l(x)$, and k is the weight of the error pattern $(c_0(x) + \varepsilon_l(x))$. Then, considering the above-mentioned six conditions for distinguishing different mis-correcting polynomials, we can conclude that,

1) The pattern distance of these polynomials may be $0, 1, \cdots, q-l-l_e$, where l_e is the length of an error pattern within $(c_0(x)+\varepsilon_l(x))$, and polynomials with pattern distances less than zero are ignored;
2) The beginning locations of a burst may be $\alpha^0, \alpha^1, \cdots, \alpha^{q-2}$;
3) The length of the burst is l;
4) The length of a burst within $(c_0(x)+\varepsilon_l(x))$ is less than or equal to l;
5) The weight of the burst is i and the locations of the non-zero elements are predetermined;
6) The weight of an error pattern within $(c_0(x)+\varepsilon_l(x))$ is k, and the number of the distributions of non-zero elements in the error pattern is $\binom{l_e-2}{k-2}$.

From the above deduction procedure of (10), it follows that the number of elements in any sub-set generated from above classifications is *$N_0(i, k)$*. It is also noted that when the length of a burst is equal to that of the error pattern in a mis-correcting polynomial, the probability of miscorrection is 1/2. Therefore, the formula for computing $N(l,i,k)$ is given by

$$N(l,i,k) = N_0(i,k) \cdot \left[\frac{1}{2} \cdot \binom{l-2}{k-2} \cdot (q-2l) + \sum_{l_e=k}^{l-1} \binom{l_e-2}{k-2} \cdot (q-l-l_e) \right] \cdot (q-1) \tag{12}$$

Now, consider the number of these mis-correcting polynomials whose burst length is l $(t+1 \le l \le 2t-1)$, and let the number of these polynomials be $M(l)$. Since the weight of the burst pattern may be $i = 2t-l+1, 2t-l+2, \cdots, l$, the number for possible distributions of non-zero elements in a burst is $\binom{l-2}{i-2}$, and the weight of the error pattern may be $j = 2t-(i-1),\ 2t-(i-1)+1,\ \cdots,\ i$, the number of the mis-correcting polynomials is:

$$M(l) = \sum_{i=2t-l+1}^{l} \left(\binom{l-2}{i-2} \sum_{j=2t-(i-1)}^{i} N(l,i,j) \right) \tag{13}$$

Moreover, for a burst with length of l, the number of possible pattern values is *$(q-1) \cdot q^{l-2} \cdot (q-1)$* with possible locations of *$(q-1)$*.

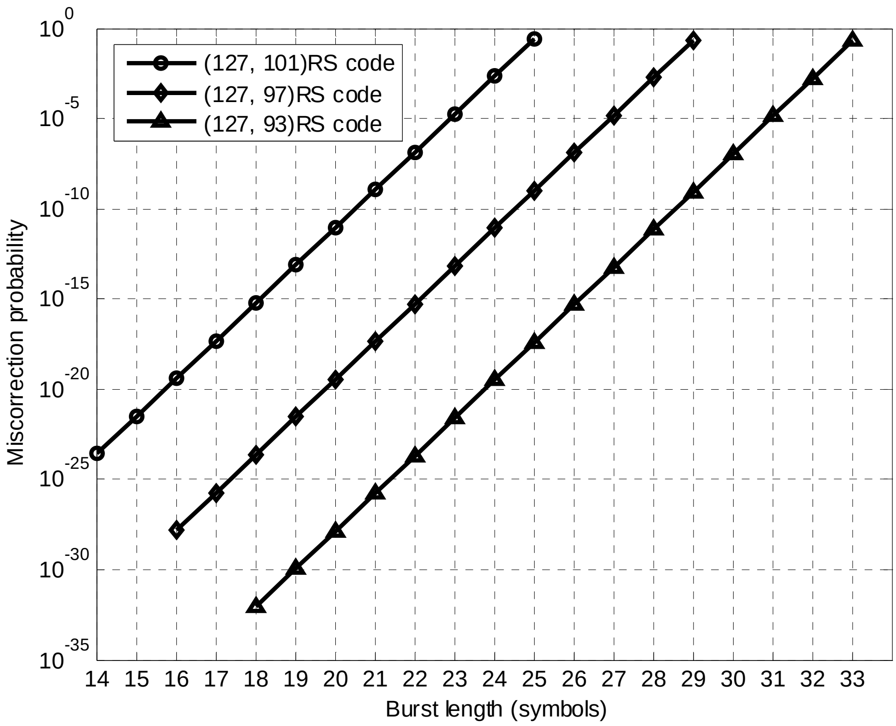

Fig. 1 Miscorrection probability with the proposed algorithm

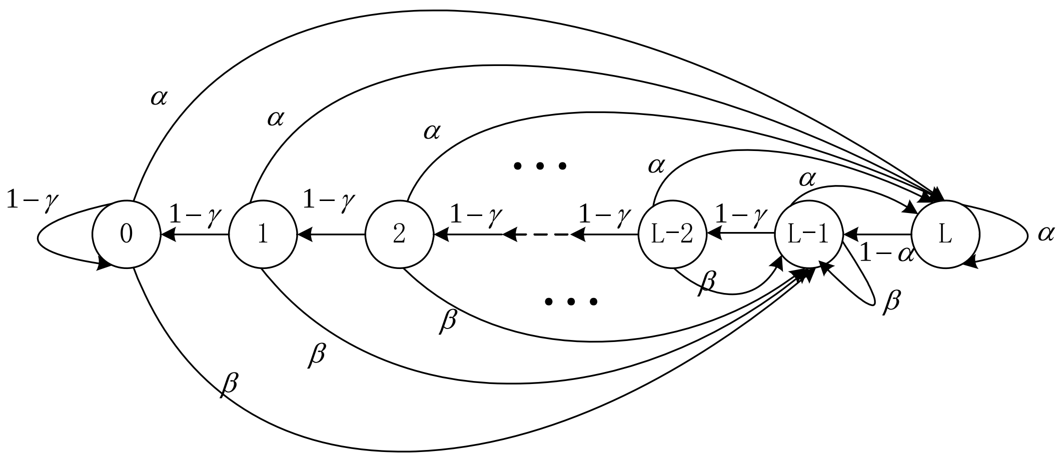

Fig. 2 The proposed L-state bursty channel model

As such, the total pattern number for a burst with length of l is $(q-1)^3 \cdot q^{l-2}$. Accordingly, the probability of miscorrection for a given burst length l is:

$$p_e(l) = \frac{M(l)}{(q-1)^3 q^{l-2}} \quad (t+1 \le l \le 2t-1) \tag{14}$$

From (10)-(14), the probabilities of miscorrection for any given burst lengths may be calculated. As an example, some numerical results of the miscorrection probabilities with the proposed algorithm for some RS codes in GF(2^7) are listed in Fig. 1. It is convinced that the proposed algorithm can effectively correct burst errors even when burst lengths approach to n-k symbols.

IV. Performance of the Proposed Algorithm over a Bursty Channel

In this section, we consider one specific case of an (255, 223) RS code and a bursty channel, and determine the performance improvement that can be achieve by the proposed algorithm instead of the existing Berlekamp- Massey decoding algorithms. The bursty channel under consideration is described in [5], as shown in Fig. 2. This channel is modeled as a binary symmetric channel. It makes the assumption that the bursts of errors arrive with a Poisson distribution, and all burst

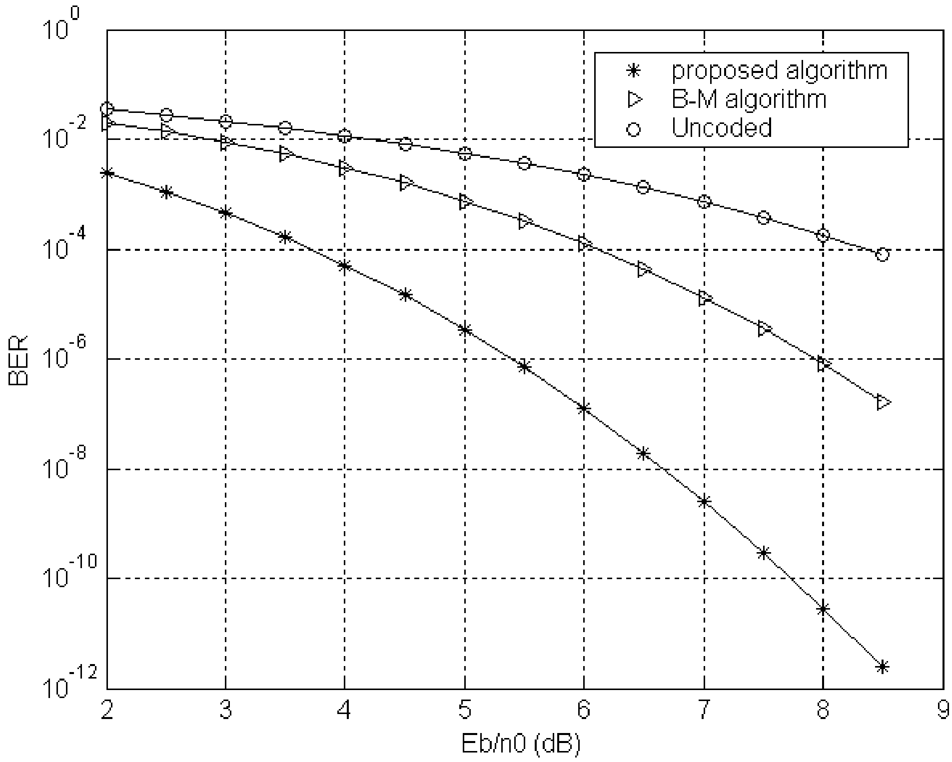

Fig. 3 Performance of two decoding algorithms over the bursty Channel

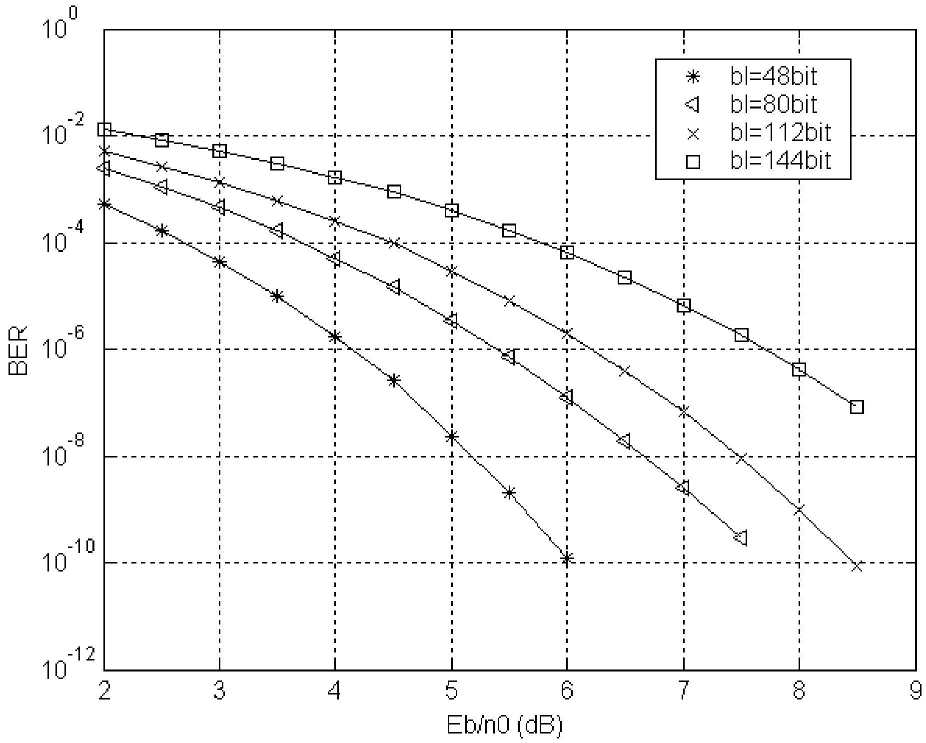

Fig. 4 Performance of the proposed algorithm over burst channels with different typical burst lengths

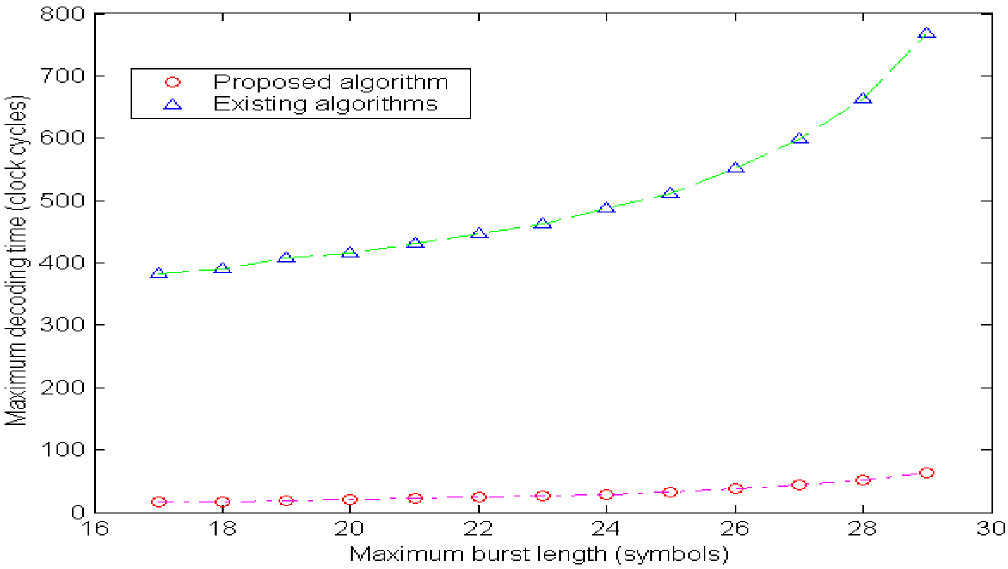

Fig. 5 Maximum decoding time with the proposed algorithm and the existing algorithms

hits are equally severe. Moreover, we further assume that the probability for a bit in bursty state equals to the error probability for a bit with specified E_b/N_0 over a BIAWGN channel. Based on these assumptions, a simulation result for the proposed algorithm and the Berlekamp-Massey algorithm over the channel is plotted in figure 3. Noted that, under high E_b/N_0, the proposed algorithm performs much better than that of Berlekamp-Massey algorithm.

Figure 4 plots the BER vs. E_b/N_0 for channels with different typical burst lengths after the decoding with the proposed algorithm. For a channel with typical burst length equals to 112 bits, a BER of 10^{-8} may be obtained with 7.5 dB in E_b/N_0.

V. Complexity Evaluation and Hardware Implementation of the Proposed Algorithm

In the proposed decoding algorithm, the error values are obtained directly from the syndrome polynomial, while the computational complexity for the syndrome with a polynomial division is the same as that with the Fourier transform. Compared with existing algorithms, the search procedure for finding the location of the burst and the complicated computations for the corresponding burst values are exempted with the proposed decoding algorithm, resulting in a fairly low computational complexity. In particular, with the proposed decoding algorithm, the whole procedure for finding the maximum likelihood burst pattern only takes n clock cycles as far as a VLSI implementation is concerned. While with existing algorithms, a search procedure takes n clock cycles, and to every possible X_1, the computation of S_j/X_1 (S_j is the syndrome component, and X_1 is the initial location of the burst) needs m clock cycles, where m is the length in bits of a code symbol. Furthermore, the maximum number of possible X_1, according to the search procedure, may be $\left\lfloor \frac{n+2t-v_{\max}}{2t-v_{\max}+1} \right\rfloor$, where $\lfloor x \rfloor$ is the largest integer that is less than or equal to x. As a result, the maximum decoding time should be $n+m\times\left\lfloor \frac{n+2t-v_{\max}}{2t-v_{\max}+1} \right\rfloor$ clock cycles with the existing algorithms. Therefore, the decoding delay of the proposed algorithm is much less than that of existing algorithms.

Based on above discussions a decoder structure for the proposed algorithm is shown in Figure 3. In which a "select" signal is used to control the modified syndrome circuit to compute the syndromes or to compute the corresponding syndromes of the received vector that are cyclically shifted left i symbol length in every decoding step. The circuits for multiplying the decoding matrix A_{2t}^{-1} with the shifted syndrome are implemented by combinational logic circuits and the output of which are transmitted into two circuits, one is circuits for computing the current burst length and then comparing it with the former minimum results; if the current length is less than the former result, then the output of which enables the other circuits to storage the current error pattern and at the same time enables the location counter loading the corresponding lowest burst location. Finally the error pattern is serially outputted to correct the burst error during the outputting of the information estimators.

VI. Conclusion

A new burst-error-correcting algorithm for RS codes has been proposed in this paper. Analytical result shows that the miscorrection probability of the proposed algorithm is very small even as the length of the burst approaches to n-k. It is verified that the computational complexity of the proposed

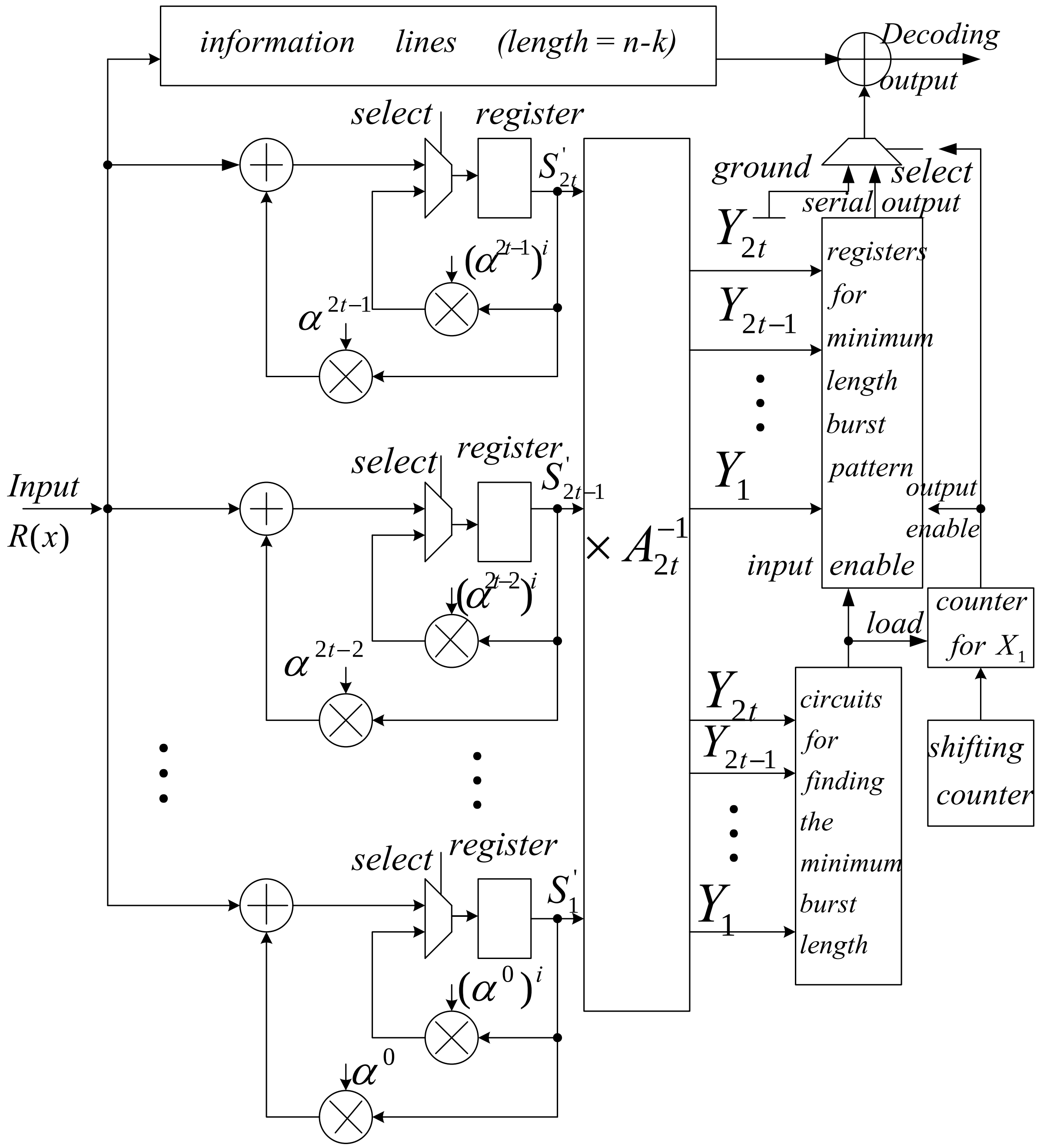

Fig. 6 VLSI hardware decoder structure for the proposed burst-error-correcting algorithm

decoding algorithm is fairly lower than existing burst-error-correcting algorithms, and the decoding delay is much less than that of existing techniques. We conclude by noting that the proposed scheme is very suitable for implementation with VLSI, and will find wide applications in information transmission and storage systems.

Acknowledgment

The authors would like to thank the anonymous reviewers for very insightful comments which greatly improved the quality of this manuscript.

References

[1] R. E. Blahut, Theory and practice of error control codes. Reading, MA: Addison-Wesley, 1984.

[2] J. Chen and P. Owsely, "A burst-error-correcting algorithm for Reed-Solomon codes", IEEE Trans. Inform. Theory, vol. IT-38, (6), pp. 1807-1872, Nov. 1992.

[3] E. Dawson and A. Khodkar, "Burst error-correcting algorithm for Reed-Solomon codes", Electronic Papers, vol. 31, No. 11, pp. 848-849, 25th May, 1995.

[4] G. D. Forney, "On decoding BCH codes", IEEE Trans. Inform. Theory, vol. IT-11, pp. 549-557, 1975.

[5] J. Hagenauer, “Sequential decoding for burst-error-channels”, in Communication systems and Random Process Theory, J. K. Skwirzynski, Ed. Alphen am Rhein, The Netherlands: Sijthoff & Noordhoff, 1978, pp. 249-265.

[6] S. Lin and D. J. Costello, Error control coding: fundamentals and applications, pp. 211 Prentice-Hall, Englewood Cliffs, NJ, 1983.

[7] P. G. Farrell and S. J. Hopkins, “Burst-error-correcting array codes”, Radio Elec. Eng., vol. 52, no. 4, pp. 188-192, Apr. 1982.

[8] E. R. Berlekamp, “Bit-Serial Reed-Solomon encoders”, IEEE Trans. Inform. Theory, vol. IT-28, (6), pp. 869-874, 1982.

[9] 1. Cipra, Barry A., "The Ubiquitous Reed–Solomon Codes", SIAM News, 1993, vol. 26, no. 1, http://www.eccpage.com/reed_solomon_codes.html.

[10] Berlekamp, Elwyn R., Nonbinary BCH decoding, International Symposium on Information Theory, San Remo, 1967, Italy.

[11] Berlekamp, Elwyn R., Algebraic Coding Theory, Laguna Hills, CA: Aegean Park Press, ISBN 0894120638, 1984.

[12] K. Zhang, L. Yin, and J. Lu, “New Code Construction Method and High-Speed VLSI Codec Architecture for Repeat-Accumulate Codes,” WSEAS Transactions on Communications, vol. 8, no. 7, pp. 608-617, 2009.

[13] Forney, Jr., G., "On Decoding BCH Codes", IEEE Transactions on Information Theory 11 (4): 549–557, doi:10.1109/TIT.1965.1053825, October 1965.

[14] M.A. Armand, "Solving the Welch-Berlekamp key equation over a Galois ring," WSEAS Transactions on Mathematics, vol. 4, no. 1, Jan 2005.

[15] Gill, John, EE387 Notes #7, Handout #28, Stanford University, http://www.stanford.edu/class/ee387/handouts/notes7.pdf, retrieved April 21, 2010.

[16] Hong, Jonathan; Vetterli, Martin (August 1995), "Simple Algorithms for BCH Decoding", IEEE Transactions on Communications 43 (8): 2324–2333.

[17] Koetter, Ralf, Reed–Solomon Codes, MIT Lecture Notes 6.451 (Video), http://ocw.mit.edu/OcwWeb/Electrical-Engineering-and-Computer-Science/6-451Spring-2005/LectureNotes/detail/embed10.htm.

[18] MacWilliams, F. J.; Sloane, N. J. A., The Theory of Error Correcting Codes, New York, NY: North-Holland Publishing Company,1977.

[19] Massey, J. L., "Shift-register synthesis and BCH decoding", IEEE Transactions on Information Theory, 1969, IT-15 (1): 122–127, http://crypto.stanford.edu/~mironov/cs359/massey.pdf.

[20] Peterson, Wesley W., "Encoding and Error Correction Procedures for the Bose-Chaudhuri Codes", IRE Transactions on Information Theory (Institute of Radio Engineers), 1960, IT-6: 459–470.

[21] Ta-Hsiang Hu, Ming-Hua Chang, "Decoding Shortened Reed Solomon Codes at Bit Level," WSEAS TRANSACTIONS on COMMUNICATIONS, Vol. 9, no. 11, pp. 695-707, 2010.

[22] Reed, Irving S.; Chen, Xuemin, Error-Control Coding for Data Networks, Boston, MA: Kluwer Academic Publishers,1999.

[23] Reed, Irving S.; Solomon, Gustave, "Polynomial Codes over Certain Finite Fields", Journal of the Society for Industrial and Applied Mathematics (SIAM), 1960, vol. 8, no. 2, pp. 300–304, doi:10.1137/0108018.

[24] Welch, L. R., The Original View of Reed–Solomon Codes, Lecture Notes, 1997, http://csi.usc.edu/PDF/RSoriginal.pdf.

[25] Liuguo Yin, Jianhua Lu, K B Letaief, and Youshou Wu. Burst-error-correcting algorithm for Reed-Solomon codes. Electronics Letters. 2001, 37(11): 695-697.

[26] Liuguo Yin, Jianhua Lu, K B Letaief, and Youshou Wu. A fast decoding algorithm for Reed-Solomon codes with enhanced burst correcting capability. IEEE VTS 54th Vehicular Technology Conference. 2001, 3: 1477-1481.

[27] Liuguo Yin, Jianhua Lu, K B Letaief, and Youshou Wu. Burst-Error-Correcting Algorithm for RS Codes and Its Performance over A Bursty Channel. Proceeding of IEEE International Conference on Communications, Circuits And Systems (ICCCAS 2002), 2002, 1: 77-81.

[28] J. Xie, L. Yin, N. Ge, and J. Lu, “Fast Convergence Algorithm for Decoding of Low Density Parity Check Codes,” WSEAS Transactions on Communications, vol. 8, no. 7, pp. 598-607, 2009.

[29] M.A. Armand, "Efficient decoding of Reed-Solomon codes over Zq based on remainder polynomials," WSEAS Transactions on Communications, vol. 1, no. 1, pp. 116-121, 2002.