A New Soft Decision Decoding Algorithm for Reed-Solomon Codes

Edelmar Urba
Faculdade Dom Bosco - Engenharia de Computação
Av. Manoel Ribas, 2181
CEP 80810-000 Curitiba - PR - BRAZIL
engcomp-faculdade@dombosco.com.br

Walter Godoy Júnior
Centro Federal de Educação Tecnológica do Paraná - CEFET PR - CPGEI
Av. 7 de Setembro, 3165
CEP 80230-901 Curitiba - PR - BRAZIL
godoy@cpgei.cefetpr.br

Abstract

We propose an efficient soft-decision decoding algorithm for Reed-Solomon codes, called symbol substitution decoding (SSD), that supplies the algebraic decoder with a set of candidate sequences. The algebraic decoder generates a set of candidate codewords and the most likely codeword is chosen as the transmitted codeword. SSD outperforms successive erasures decoding (SED) and generalized minimum distance decoding (GMD). SSD can approach the performance of maximum-likelihood decoding (MLD) when the number of candidate sequences increases.

1 Introduction

The central problem with soft-decision for Reed-Solomon at the symbol level is to find an efficient technique that can generate a set of candidate codewords that will contain the codeword that is most likely with high probability.

In this paper, we develop an efficient soft-decision Reed-Solomon decoding algorithm, called symbol substitution decoding (SSD), that supplies the algebraic decoder with a set of candidate sequences. The algebraic decoder generates a set of candidate codewords and the most likely codeword among those is chosen as the transmitted codeword [1].

In Section 2, we intend to introduce our communication system model and the algorithm. In Section 3, SSD associated with Bounded Minimum Distance Decoding is shown. In Section 4, we intend to discuss the algorithm complexity. In Section 5, the simulation results are shown and Section 6 is an conclusion.

2 Communication System Model and the Algorithm

The system block diagram is shown in Fig. 1.

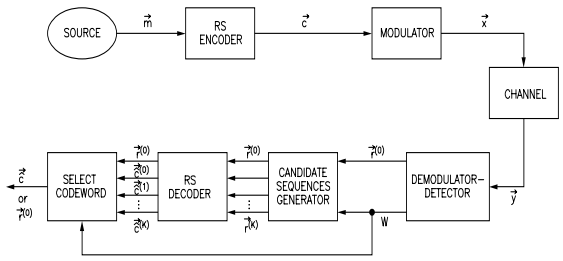

Figure 1: Communication System Model.

A sequence $\vec{m} = (m_j)$ of k message symbols, $j = 0, 1, \cdots, k-1$, $m_j \in \mathrm{GF}(2^l)$, is encoded into a block of $n = 2^l - 1$ RS code symbols $\vec{c} = (c_j)$, $j = 0, 1, \cdots, n-1$, $c_j \in \mathrm{GF}(2^l)$, by a (n,k) RS code.

The index j is the position of symbol c_j in the codeword and l is the number of bits per symbol.

Each code symbol c_j is expanded into its representation in GF(2), generating the sequence of $n \cdot l$ binary digits b_q, $b_q \in \{0,1\}$, where $q = 0,1,\ldots,(n \cdot l) - 1$. The index $q = (l \cdot j) + f$, where $f = 0,1,\cdots,l-1$, denotes the position of a binary digit within one RS codeword in GF(2) representation and $j = 0,1,\cdots,n-1$ denotes the position of symbol c_j in the codeword in GF(2^l) representation. These binary digits b_q are fed into a BPSK modulator generating an antipodal signal $x_q = \sqrt{E_s}(1-2b_q)$, $x_q \in \{-\sqrt{E_s}, \sqrt{E_s}\} \subset \mathbb{R}$. E_s is the energy per digit of the signal.

In the channel, the signal x_q is corrupted by additive noise z_q and is received as $y_q = x_q + z_q$, where the components of z_q are statistically independent Gaussian random variables with mean $\mu = 0$ and variance $\sigma^2 = N_0/2$.

The Gaussian channel output $\vec{y}$ is the input to the demodulator-detector, which produces as output the matrix $\mathrm{W} = (w_{ij})$, $w_{ij} \in \mathbb{R}$, $i = 0,1,\cdots,2^l-1$ and $j = 0,1,\cdots,n-1$. W is the matrix of reliability values which provides likelihood information $w_{ij} = -\ln p(\langle a_v\rangle|\langle y_q\rangle)$, where $\langle a_v\rangle$, $v = (l \cdot i) + f$, $f = 0,1,\cdots,l-1$, $i = 0,1,\cdots,2^l-1$, is obtained by the expantion of all GF(2^l) symbols into their GF(2) representation and converted with the help of a linear mapping to a real number 1 or -1. The following convention is adopted: the $1 \in$ GF(2) corresponds to $-1 \in \mathbb{R}$ and the $0 \in$ GF(2) corresponds to $1 \in \mathbb{R}$. The index i denotes the elements of GF(2^l) (see Wicker [6]). $p(\langle a_v\rangle|\langle y_q\rangle)$ is the conditional probability of the sequence $\langle a_v\rangle$ given the occurrence of the sequence $\langle y_q\rangle$ and is calculated symbol-by-symbol as [2]

$$p(\langle a_v\rangle|\langle y_q\rangle) = \prod_{f=0}^{l-1} \frac{1}{1 + e^{\frac{-2a_{v=(l\cdot i)+f} \cdot y_{q=(l\cdot j)+f}}{\sigma^2}}}, \quad (1)$$

where the variance $\sigma^2 = N_0/2$.

The value w_{ij} is known as the soft weight of the received sequence $\langle y_q\rangle$ with respect to the field element $\langle a_v\rangle$ [3, 4, 5]. A large value of w_{ij} corresponds to a symbol $\langle a_v\rangle$ with a smaller likelihood of being the transmitted symbol. On the other hand, a small value of w_{ij} corresponds to symbol $\langle a_v\rangle$ being the transmitted symbol with higher likelihood. The minimum value per column of W corresponds to the hard-decision symbol.

The matrix of reliability values W is the input of the block candidate sequences generator, which produces a set of vectors called candidate sequences, denoted as $\vec{r}^{(c)} = (r_j^{(c)})$, $c = 0,1,\cdots,c_{max}-1$; $j = 0,1,\cdots,n-1$; $r_j^{(c)} \in$ GF(2^l). c_{max} is chosen as a power of 2 in order to use all possible combinations of a number p of unreliable positions in the codeword. One of the vectors is the sequence $\vec{r}^{(0)} = (r_j^{(0)})$, $j = 0,1,\cdots,n-1$; $r_j^{(0)} \in$ GF(2^l), which is the hard-decision sequence. Further candidates sequences are assembled by substituting the p least reliable symbols of $\vec{r}^{(0)}$ as follows.

$\vec{r}^{(1)}$ is assembled by excluding the least reliable symbol of $\vec{r}^{(0)}$, also called generator sequence, and including in this position the second most reliable symbol (or the second smallest value) of the respective column of W.

$\vec{r}^{(2)}$ is assembled by excluding the second least reliable symbol of $\vec{r}^{(0)}$ and including in this position the second most reliable symbol of the respective column of W.

$\vec{r}^{(3)}$ is assembled by excluding both the two least reliable symbols of $\vec{r}_0$ and including in these positions the second most reliable symbols of the respective columns of W.

$\vec{r}^{(4)}, \vec{r}^{(5)}, \cdots, \vec{r}^{(c_{max}-1)}$ are assembled in a similar form.

The candidate sequence generator outputs $\vec{r}^{(0)}$ and $\vec{r}^{(c)}$ are the inputs of the block RS-decoder, which produces as output a set of vectors called candidate codewords $\hat{\vec{c}}^{(c)} = (\hat{c}_j^{(c)})$, $c = 0,1,\cdots,c_{max}-1$; $j = 0,1,\cdots,n-1$. The candidate codewords $\hat{\vec{c}}^{(c)}$ and the matrix of reliability values $\mathrm{W} = (w_{ij})$ are the inputs of the codeword selection producing a final estimate codeword $\hat{\vec{c}}$ as output. The matrix of reliability values W is used to calculate the reliability $z^{(c)}$ of each estimate codeword $\hat{\vec{c}}^{(c)}$ as

$$z^{(c)} = \sum_{j=0}^{n-1} w_{ij},$$

where $i = \hat{c}_j^{(c)}$. The estimate codeword with minimum $z^{(c)}$ is chosen as the final estimate codeword. When the decoder does not find a codeword, i.e., a decoding failure, the hard-decision sequence $\vec{r}^{(0)}$ is the final output. We call this strategy scheme 1.

3 Symbol Substitution Decoding Associated with Bounded Minimum Distance Decoding

For RS-codes with code length 63 or larger, a simplification is possible without loss in performance compared to scheme 1. Begin the decoding algorithm by decoding the hard-decision sequence $\vec{r}^{(0)}$ using an errors-only decoding algorithm. If a codeword is found take the result to be the final estimate codeword $\vec{c}$ and the process is complete.

If the decoding attempt fails to find a codeword, compute the matrix of reliability values, generate the set of candidate sequences, decode using an errors-only decoding algorithm and choose the codeword with minimum $z^{(c)}$.

If again no codeword results, the sequence $\vec{r}^{(0)}$ is delivered.

For RS-codes with code length 31 or smaller this simplification induces a loss in performance, compared to scheme 1, because the probability of decoder error is high [7].

We call this strategy scheme 2.

4 Symbol Substitution Decoding Complexity

The decoding complexity of SSD is determined by 3 factors:

1. The complexity to calculate the reliability matrix W.
2. The complexity of the selection algorithm in the candidate sequences generator block.
3. The complexity of the RS decoder.

The complexity of calculating the reliability matrix W is given by the number of operations with real numbers in the equation 1.

The complexity of the selection algorithm depends on the basic algorithm of selecting the M*th* largest elements in a vector. The fastest general algorithm for this basic task uses the *partitioning* method. Selecting a random partition element, one marches through the vector, forcing smaller elements to the left, larger elements to the right. For selection, it can be ignored a subset and attend only to the one that contains the desired M*th* element. Selection by partitioning counts scales as N, where N is the vector length. In this case, N is equal to the number of elements in GF(2^l). For more information about this selection algorithm see Press [10].

It is possible to show that every linear code can be decoded by a machine with computational complexity proportional to n^2, where n is the codeword length [11]. Therefore, it is possible to attain an RS-decoder complexity proportional to n^2. In this case, the SSD complexity is proportional to the number of candidate sequences times n^2. This is an upper estimate.

Justesen [12] proposes that certain q-ary RS codes can be decoded by an algorithm requiring only $q \log^2 q$ in terms of additions and multiplications in GF(q). Therefore, a realistic estimation of the RS decoding process in SSD is the number of candidate sequences times $q \log^2 q$.

5 Simulation Results

In this section computer simulations is used in order to compare the performance of the SSD with scheme 1 and scheme 2 for binary antipodal signals (binary phase-shift keying, BPSK) over the Gaussian channel. In these figures the acronyms have the following meaning. HDD means hard-decision decoding; Scheme 1 - XC means symbol substitution decoding with scheme 1 and with X candidates, where X can be 4, 8, 16, 32, 64, 128, 256, 512, 1024 and 2048; Scheme 2 - XC means symbol substitution decoding with scheme 2 and with X candidates; TMLD means trellis maximum-likelihood decoding.

In order to give an idea of the accuracy of this simulation, we would like to point out that 10^6 codewords could be transmitted. The simulation finishes if the decoding process yields 10^3 word errors or if all codewords were transmitted.

Fig. 2 shows that a bit error rate (BER) of 10^{-5}, SSD with 128 candidate sequences applied to a (7,3) RS code offers a 2.5 dB improvement in power efficiency compared to HDD. Another SSD characteristic that can be observed, is that the gain increases with the number of candidate sequences.

In Fig. 3, scheme 1 for a (15,11) RS code is compared to the trellis maximum-likelihood decoding (TMLD) algorithm proposed by Vucetic and Vuckovic [13]. This TMLD algorithm is implemented by searching a trellis. The main idea lies in sorting of all possible n-tuples from GF(q) with respect to their distances from the received sequence. Start-

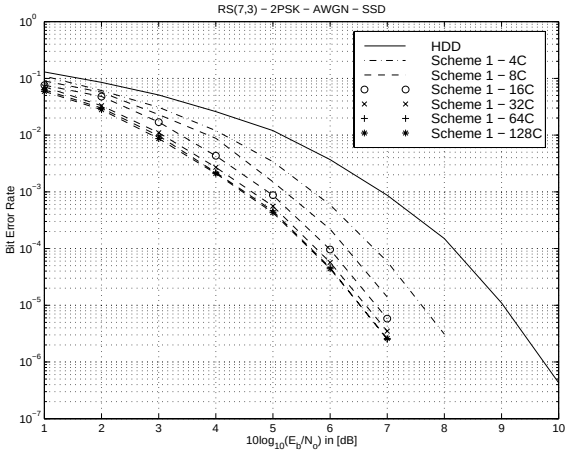

Figure 2: SSD, scheme 1, applied to a (7,3) RS code.

ing from the closest one, one by one is tested, while increasing the distance from the received sequence successively, until the first codeword is found [13]. The non-binary error patterns, which belong to the same level, are represented by a trellis. The trellises can be constructed prior to the decoding process. The redecoding process starts by searching the paths in the trellis for the lowest level, and stops when added to the received sequence results in a codeword [13]. From this figure, it can be concluded that SSD can achieve maximum-likelihood decoding (MLD) if the number of candidate sequences increases.

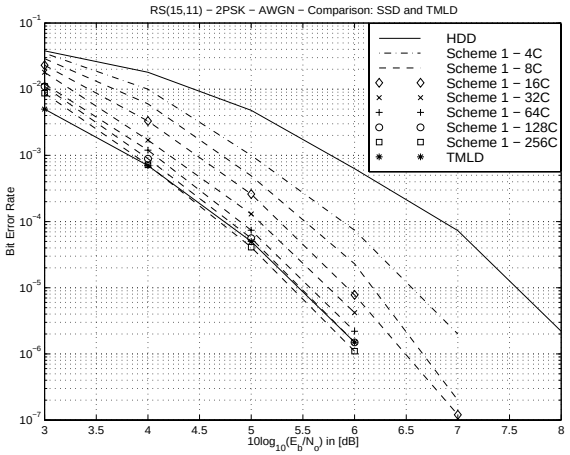

Figure 3: Comparison between SSD and MLD when applied to a (15,11) RS code.

Fig. 4 shows the results of scheme 1 with a (31,23) RS code. Note that the gain at BER = 10^{-5} and scheme 1 with 128 candidate sequences is approximately 1.4 dB. In comparison with scheme 1 applied to a (7,3) RS code the gain decreased. Hence, the gain decreases with codeword length for an equal number of candidate sequences.

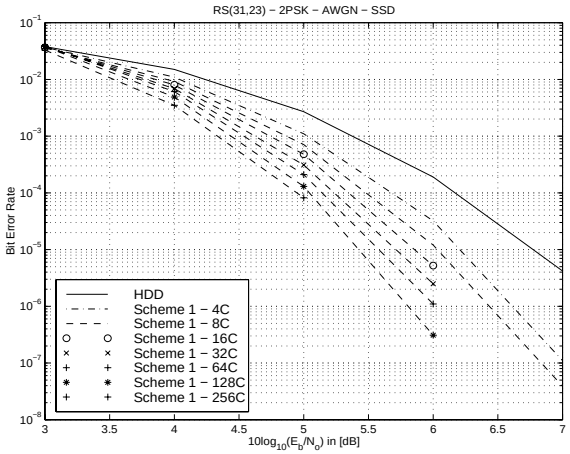

Figure 4: SSD, scheme 1, applied to a (31,23) RS code.

Fig. 5 shows that SSD is superior to SED [9] with maximal 4 and 8 erasures and GMD decoding, while the number of hard-decision decoding procedures remains constant.

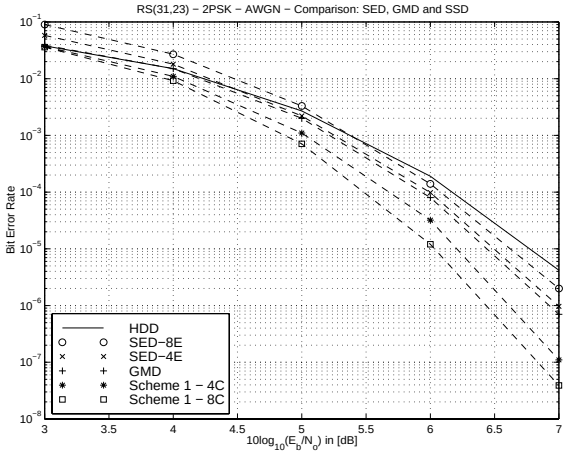

Figure 5: Comparison between SED, GMD, SSD and scheme 1, when applied to a (31,23) RS code.

Finally, in Fig. 6 the performance of scheme 2 when applied to a (255,223) RS code is shown.

Results for SSD associated with errors-and-erasures decoding and SSD associated with SED can be found in [1].

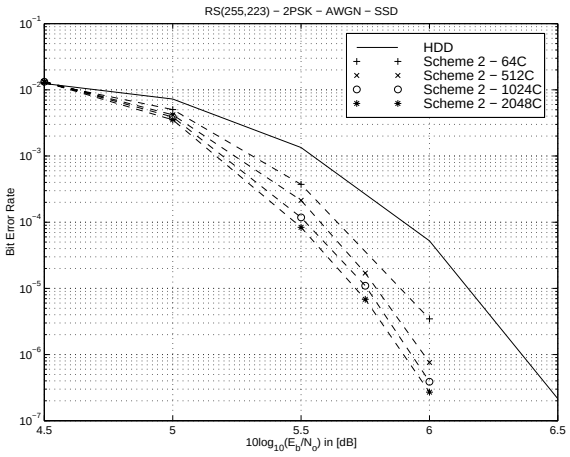

Figure 6: SSD, scheme 2, applied to a (255,223) RS Code.

6 Conclusions

The new soft decision algorithm proposed in this paper, called symbol substitution decoding (SSD), belongs to the class of decoding methods in which an algebraic decoder is used to generate a number of candidate codewords. The innovation is that SSD supplies the algebraic decoder with a set of candidate sequences generated so that the information contained in the received sequence is processed with more efficiency.

SSD has the following main characteristics:

- The gain increases with the number of candidate sequences.
- SSD can achieve MLD if the number of candidate sequences increases.
- The gain decreases with code length for equal number of candidate sequences.

SSD outperforms SED [9] and GMD [8] by far.

The decoding complexity of SSD is governed by the complexity of calculating the reliability matrix W, by the complexity of the selection algorithm in the candidate sequences generator block, which is proportional to the number of candidate sequences times q, where q is the number of elements in GF; and by the complexity of the RS-decoder block, which is proportional to the number of candidate sequence times $q \log^2 q$.

Acknowledgments

One of the authors (Urba, E.) would like to thank Prof. Dr.-Ing. Johannes Huber for his valuable contribution.

This work is supported by Conselho Nacional de Pesquisa e Desenvolvimento - CNPq, Brazil, and Deutsch Akademische Austauschdienst - DAAD, Germany.

References

[1] Urba, Edelmar, *Soft-Decision for Reed-Solomon Codes with Applications*. Ph.D. Thesis by Friedrich-Alexander-Universität Erlangen-Nürnberg, Lerhstuhl für Nachrichtentechnik II, Germany, to be presented in May, 1999.

[2] Johannes Huber: *Codierung für gedächtnisbehaftete Kanäle*. Ph.D. Thesis by Hochschule der Bundeswehr München, pp. 232, Mar. 1982.

[3] A. Brinton Cooper III: *Soft-Decision Decoding of Reed-Solomon Codes* in Reed-Solomon Codes and Their Applications, Edited by Stephen B. Wicker; Vijay K. Bhargava, IEEE Press, pp. 321, New York, 1994.

[4] F. Taleb and P. G. Farrel: *Minimum Weight Decoding of Reed-Solomon Codes*, in Cryptography and Coding II, C. Mitchel (ed.), Oxford: Claredon Press, 1992.

[5] M. Rice, D. J. Tait, and P. G. Farrel: *A* Soft-Decision Reed-Solomon Decoder, 1988 IEEE International Symposium on Information Theory, Kobe, Japan.

[6] Stephen B. Wicker:Error Control Systems for Digital Communication and Storage. Prentice Hall, Englewood Cliffs, p. 512, New Jersey, 1995.

[7] R. J. McEliece and L. Swanson: *On the Decoder Error Probabilty for Reed-Solomon Codes*. IEEE Transaction on Information Theory. Vol. IT-32, pp. 701-703, September 1991.

[8] G. David Forney, Jr.: Generalized Minimum Distance Decoding. IEEE Transaction on Information Theory, Vol. IT-12, pp. 125-131, April 1966.

[9] Gőran Einarsson; Carl Eric Sundberg: *A Note on Soft Decoding with Successive Erasures*. IEEE Transaction on Information Theory, pp. 88-96, January 1976.

[10] William H. Press; Saul A. Teukolsky; Willian T. Vetterling; and Brian P. Flannery: Numerical Recipes in C. The Art of Scientific Computing. Cambridge University Press, - 2nd ed., Cambridge, 1994.

[11] J. E. Savage: *Three Measures of Decoder Complexity*. IBM Journal Res. and Dev., pp. 417-425, July 1970.

[12] Jørn Justesen: *On the complexity of Decoding Reed-Solomon Codes*. IEEE Transaction on Information Theory, Vol. IT-32, No. 5, pp. 709-714, september 1986.

[13] Jelena S. Vuckovic; Branca S. Vucetic: *Maximum-Likelihood Decoding of Reed-Solomon Codes*. ISIT 1997, p. 400, Ulm, Germany, June 29- July4, 1997