Bit-Level Soft-Decision Decoding of Reed–Solomon Codes

Alexander Vardy and Yair Be'ery, *Member, IEEE*

Abstract—**In this paper, we present a Reed–Solomon decoder that makes use of bit-level soft-decision information. A Reed–Solomon generator matrix which possesses a certain inherent structure in GF(2) is derived. This structure enables representation of the code as a union of cosets, each coset being an interleaver of several binary BCH codes. Such partition into cosets provides a clue for efficient bit-level soft-decision decoding. Two decoding algorithms are derived. In the development of the first algorithm we assume a memoryless channel, which makes the value of this algorithm more conceptual than practical. The second algorithm, which is obtained as a modification of the first, does account for channel memory and thus accommodates a bursty channel. Both decoding algorithms are in many cases orders of magnitude more efficient than conventional techniques.**

I. Introduction

THE practical importance of Reed–Solomon codes is well established (see [1]–[7]). The application of Reed–Solomon codes spans from deep-space communication standard [2]–[4] and over the air teletext broadcast [5] to frequency-hop spread-spectrum systems [6] and optical communication [7]. Hard-decision decoders for RS codes are readily available using algebraic decoding algorithms. Such decoders have been implemented and operate at rates above 120 Mb/s. Soft-decision decoding of RS codes is, however, an entirely different matter. The advantage of soft decision over hard-decision decoding is adequately established in many works [8]–[10] (for graphs of soft-decision coding gain versus SNR for various RS codes see, for example, [10]). Yet even though the decoder can be often supplied with reliable soft-decision data relatively easily [1], the high complexity of optimal soft decoding makes full utilization of such data prohibitive. In fact, the available soft-decoding algorithms, such as Forney's generalized minimum distance decoding [11] and others [12], [13], make use of soft-decision information only on the byte level. Namely, the confidence values of the received bits are processed in one way or another [12] to generate the average confidence value of the symbol, which is then transferred to the decoder. Thus, the bit-level soft-decision information is lost. In this context Berlekamp *et al.* [1] state that "the major drawback with RS codes (for satellite use) is that the present generation of decoders do not make full use of bit-based soft-decision information." In this paper, we develop Reed–Solomon decoder that makes use of bit soft-decision information. The proposed decoding algorithms are in many cases several orders of magnitude more efficient than the existing techniques (say, Viterbi decoding based on the conventional Wolf's trellis [14]). The reduced complexity of our algorithms is due to a certain symmetric structure of the RS generator matrix over GF(2) which is derived in Section II. The bit-level soft-decision decoders utilizing this structure are presented in Section III.

Paper approved by the Editor for Satellite Communications and Coding of the IEEE Communications Society. Manuscript received April 20, 1989; revised December 10, 1989 and April 25, 1990. This paper was presented in part at the 16th Conference of Electrical and Electronic Engineering in Israel, Tel-Aviv, Israel, 1989, and also in part at the IEEE International Symposium on Information Theory, San Diego, CA, January 1990.

The authors are with the Department of Eletrical Engineering, Tel-Aviv University, Ramat Aviv, 69978 Tel-Aviv, Israel.

IEEE Log Number 9042388.

II. Structure of the Generator Matrix

Let $\mathcal{R}(N, K)$ be the Reed–Solomon code over GF(2^m) of length $N = 2^m - 1$ and dimension K. We assume that $\mathcal{R}$ is used on a binary channel. Hence, the encoder must employ some fixed linear mapping ϕ: GF$(2^m)^N \to$ GF$(2)^{mN}$ to convert a sequence of N elements of GF(2^m) into a string of mN binary digits.[1] Namely, a codeword $\boldsymbol{c} = (c_0, c_1, \cdots c_{N-1}) \in \mathcal{R}$, $c_i \in$ GF(2^m) is transmitted as

$$\phi(\boldsymbol{c}) = \left(c_0^1, c_0^2, \cdots c_0^m, c_1^1, c_1^2, \cdots c_1^m, \cdots c_{N-1}^1, c_{N-1}^2, \cdots c_{N-1}^m\right)$$

where $c_i^j \in$ GF(2). Now let α be a primitive element of GF(2^m) and let $\alpha^s, \alpha^{s+1} \cdots \alpha^{s+N-K-1}$ be the set of zeros of $\mathcal{R}$. Denote by $\mathcal{B}$ the binary BCH code of length $N = 2^m - 1$ with zeros at $\alpha^s, \alpha^{s+1} \cdots \alpha^{s+N-K-1}$ and their cyclotomic conjugates over GF(2). Let $\gamma_1, \gamma_2, \cdots \gamma_m$ be the basis of GF(2^m) over GF(2) employed for the linear mapping ϕ. We define the codes $\mathcal{B}_1, \mathcal{B}_2, \cdots, \mathcal{B}_m$ as

$$\mathcal{B}_j = \left\{\left(\gamma_j b_0, \gamma_j b_1, \cdots, \gamma_j b_{N-1}\right) \mid \boldsymbol{b} = (b_0, b_1, \cdots, b_{N-1}) \in \mathcal{B}\right\} \quad \text{for } j = 1, 2, \cdots, m \tag{1}$$

where $b_i \in$ GF(2) and the product $\gamma_j b_i$ is in GF(2^m). It is well known [15] that $\mathcal{B}$ is a subfield subcode of $\mathcal{R}$ and, hence, the m codes defined in (1) are also subcodes of $\mathcal{R}$. Therefore if $\{\boldsymbol{v}_1^j, \boldsymbol{v}_2^j, \cdots \boldsymbol{v}_k^j\}$ is a set of k generators for $\mathcal{B}_j$ we may use the set

$$\bigcup_{j=1}^{m} \left\{\phi(\boldsymbol{v}_1^j), \phi(\boldsymbol{v}_2^j), \cdots \phi(\boldsymbol{v}_k^j)\right\} \tag{2}$$

as the first mk rows of a binary generator matrix for $\mathcal{R}$. By rearranging the columns the structure of Fig. 1 is obtained. This proves our basic theorem.

Theorem 1: Let $\boldsymbol{B} = [b_{ij}]$ be a generator matrix of the binary BCH code of length $N = 2^m - 1$, dimension $k < K$ and designed distance $d \geq N - K + 1$. Then there exists a binary generator matrix of $\mathcal{R}$, $\boldsymbol{G} = [g_{ij}]$, $0 \leq i \leq mN - 1$, $0 \leq j \leq$

[1] Alternatively, one could start with binary data and discuss grouping to form the Reed–Solomon symbols over GF(2^m).

0090-6778/91/0300-0440$01.00

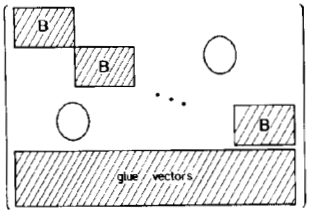

Fig. 1. Structure of the Reed-Solomon generator matrix over GF(2); B is a generator matrix of $\mathscr{R}$.

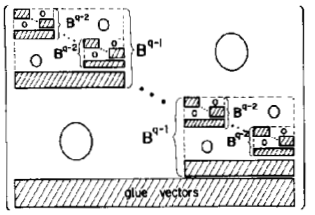

Fig. 2. Recursive structure of the Reed-Solomon generator matrix; B^j is the generator matrix of $\mathscr{R}^{(j)}$.

$mK - 1$, such that $0 \le j \le mk - 1$

$$1)\ g_{ij} = 0 \qquad \text{if } \lfloor j/k \rfloor \ne \lfloor i/N \rfloor \tag{3a}$$

$$2)\ g_{ij} = b_{\bar{i}\bar{j}} \qquad \text{if } \lfloor j/k \rfloor = \lfloor i/N \rfloor \tag{3b}$$

where $\bar{i} \equiv i \pmod N$ and $\bar{j} \equiv j \pmod k$.

Using the foregoing theorem $\mathscr{R}$ may be written as a union of cosets

$$\mathscr{R} = \bigcup_{l=0}^{2^{\Delta}-1} \mathbb{G}_l \tag{4}$$

such that $\Delta = m(K-k)$ and $\mathbb{G}_l = \{\boldsymbol{r}^l + \boldsymbol{c} \mid \boldsymbol{c} \in \mathbb{G}\}$ where

$$\mathbb{G} = \mathscr{B}_1 \oplus \mathscr{B}_2 \oplus \cdots \oplus \mathscr{B}_m$$

is a direct sum of the m codes defined in (1) and the vectors $\boldsymbol{r}^l$, $l = 0, 1, \cdots, 2^{\Delta} - 1$ are coset representatives for $\mathbb{G}$ in $\mathscr{R}$. Note that the conditions (3a) and (3b) arrange the columns of $\boldsymbol{G}$ in such a way that the m bits of each channel symbol are distributed evenly among the m BCH codes. This property of the generator matrix offers an alternative insight into the burst error correction capability of Reed-Solomon codes, since each given coset of $\mathscr{R}$ may be viewed as an *interleaver*. Assume for simplicity that some message vector $\boldsymbol{s}$ is mapped into the coset corresponding to $\boldsymbol{r} = \boldsymbol{0}$. Then for this message $\boldsymbol{s}$ employing $\mathscr{R}(N, K)$ on a binary channel is equivalent to interleaving m t-error correcting binary BCH codes of length $2^m - 1$ where $2t + 1 = N - K + 1$. A coset corresponding to $\boldsymbol{r} \ne \boldsymbol{0}$ is also an interleaver which interleaves m translates of the binary $(N, k, 2t + 1)$ BCH code. Any channel burst of less than $m(t-1) + 2$ bits will be distributed evenly among the m codes, at most t bits to a code, and therefore will be successfully corrected by the hard-decision decoder. This conception of the RS code as a union of cosets, each coset being an interleaver, will be extensively used in the next section.

It is noteworthy that Theorem 1 may be straightforwardly generalized so as to apply to any linear code that contains a subfield subcode. If m is composite we can employ such generalization of Theorem 1 to obtain additional structure for the binary generator matrix of $\mathscr{R}$. Let $m = p_1 p_2 \cdots p_q$ where $q \ge 2$ and $p_1, p_2, \cdots, p_q$ are (not necessarily distinct) primes. Then $\mathrm{GF}(2) \subset \mathrm{GF}(2^{p_1}) \subset \cdots \subset \mathrm{GF}(2^m)$. Let $\mathscr{B}^{(j)}$ be the BCH code of length $2^m - 1$ and designed distance $N - K + 1$ over $\mathrm{GF}(2^{p_1 p_2 \cdots p_j})$. Evidently $\mathscr{B}^{(j_1)}$ is a subfield subcode of $\mathscr{B}^{(j_2)}$, provided that $j_2 > j_1$. Hence, instead of going directly from $\mathscr{R}$ to $\mathscr{B}$ we may use the structure of the whole chain of nested BCH codes over nested fields

$$\mathscr{B} \subset \mathscr{B}^{(1)} \subset \cdots \subset \mathscr{B}^{(q)} = \mathscr{R}. \tag{5}$$

Applying Theorem 1 successively q times we obtain the Reed-Solomon generator matrix given in Fig. 2. For a specific example see the binary generator matrix of the (15, 9) RS code supplied in Fig. 3.

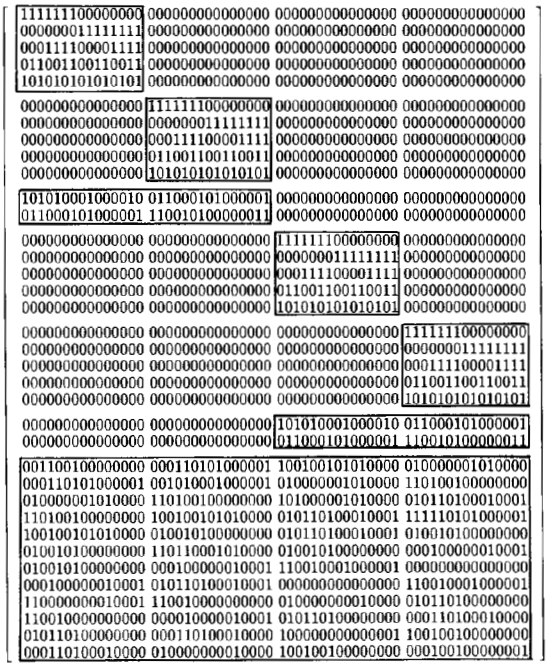

Fig. 3. A binary generator matrix for the (15, 9, 7) RS code over $\mathrm{GF}(2^4)$ generated by $g(x) = \prod_{l=1}^{6}(x - \alpha^l)$ where α is a zero of $\pi_1(x) = x^4 + x + 1$. The factorization of m is: $4 = 2 \cdot 2$, i.e., $q = 2$ and $p_1 = p_2 = 2$; $\mathscr{B}^{(q)} = \mathscr{B}^{(2)} = \mathscr{R}(15, 9)$; $\mathscr{B}^{(1)}$ is the (15, 6, 7) BCH code over $\mathrm{GF}(2^2) \subset \mathrm{GF}(2^4)$; $\mathscr{B}$ is the binary (15, 5, 7) BCH code. We arbitrarily chose $\{1, \alpha\}$ as a basis for $\mathrm{GF}(2^4)$ over $\mathrm{GF}(2^2)$ and $\{1, \beta\}$ as a basis for $\mathrm{GF}(2^2)$ over GF(2) where β is a zero of $\pi_2(x) = x^2 + x + 1$. Thus the basis for $\mathrm{GF}(2^4)$ over GF(2) becomes $\{1, \beta, \alpha, \alpha\beta\} = \{1, \alpha^5, \alpha, \alpha^6\}$.

III. Soft-Decision Decoding

In this section two decoding algorithms are derived. In the development of Algorithm 1, we assume a binary memoryless channel, which greatly simplifies the derivation. This, however, makes the value of Algorithm 1 more conceptual than practical as this algorithm does not provide for the well-known burst correction capability of Reed-Solomon codes. The practical value of Algorithm 2, which is obtained as a modification of Algorithm 1, is apparently much higher since this algorithm does account for channel memory and thus accommodates a bursty channel.

Suppose that using the linear mapping ϕ a codeword of $\mathscr{R}$ is transmitted through a binary channel. We assume through-

out a continuous-output, say additive white Gaussian noise (AWGN) channel, characterized by transition probability densities $f_j(v) = f(v/j)$, $j \in \mathrm{GF}(2)$, $v \in \mathbb{R}$ where $\mathbb{R}$ is the real line. In case of a discrete channel with output alphabet $\mathbb{F}$, $f_j(v)$ should be replaced by transition probabilities $p_j(v) = p(v/j)$, $j \in \mathrm{GF}(2)$, $v \in \mathbb{F}$. Now let the word $\boldsymbol{v} = (v_0, v_1, \cdots v_{N-1}) = (v_0^1, v_0^2, \cdots v_0^m, v_1^1, v_1^2, \cdots v_1^m m \cdots v_{N-1}^1, v_{N-1}^2, \cdots v_{N-1}^m)$ be observed at the output. Maximum likelihood decoding consists of finding a codeword $\boldsymbol{c} \in \mathscr{B}$ that maximizes $P(\boldsymbol{v}/\boldsymbol{c})$, that is maximizes the *a posteriori* probability $P(\boldsymbol{c}/\boldsymbol{v})$, provided that $P(\boldsymbol{c})$ is the same for all $\boldsymbol{c} \in \mathscr{B}$. On a memoryless channel one may as well search the maximum of

$$M(\boldsymbol{c}) = \sum_{i=0}^{N-1} \sum_{j=1}^{m} \log f(v_i^j / c_i^j).$$

Using the partition into cosets (4) and interchanging the order of summation we may perform the maximization as follows:

$$\max_{\boldsymbol{c} \in \mathscr{B}} M(\boldsymbol{c}) = \max_{\mathbb{G}_l} \max_{\boldsymbol{c} \in \mathbb{G}_l} \sum_{j=1}^{m} \sum_{i=0}^{N-1} \log f(v_i^j / c_i^j).$$

It follows from the structure of the generator matrix obtained in Theorem 1 that in a given coset the choice of $(c_0^j, c_1^j, \cdots c_{N-1}^j)$ may be made independently for each $j = 1, 2, \cdots m$. Hence, we may interchange summation over j with maximization within a coset, i.e.,

$$\max_{\boldsymbol{c} \in \mathscr{B}} M(\boldsymbol{c}) = \max_{\mathbb{G}_l} \sum_{j=1}^{m} \left[\max_{\boldsymbol{c} \in \mathbb{G}_l} \sum_{i=0}^{N-1} \log f(v_i^j / c_i^j) \right].$$

Yet, the maximization within the square brackets is just the soft decoding of the inner BCH code $\mathscr{B}$. This implies the following decoding algorithm.

Algorithm 1: For each of the 2^Δ cosets $\mathbb{G}_l$, $l = 0, 1, \cdots, 2^\Delta - 1$, and for each coset representative $\boldsymbol{r} = (r_0^1, r_0^2, \cdots r_0^m, r_1^1, r_1^2, \cdots r_1^m, \cdots r_{N-1}^1, r_{N-1}^2, \cdots r_{N-1}^m)$.

1) Find the m codewords $\hat{\boldsymbol{b}}_1, \hat{\boldsymbol{b}}_2, \cdots \hat{\boldsymbol{b}}_m$ where $\hat{\boldsymbol{b}}_j = (\hat{b}_0^j, \hat{b}_1^j, \cdots \hat{b}_{N-1}^j) \in \mathscr{B}$ by maximizing

$$M_j(\boldsymbol{b}) = \sum_{i=0}^{N-1} \log f(v_i^j / b_i + r_i^j) \tag{8}$$

with respect to all $\boldsymbol{b} \in \mathscr{B}$ for $j = 1, 2, \cdots, m$.

2) Evaluate

$$M(\boldsymbol{c}) = \sum_{j=0}^{m} M_j(\hat{\boldsymbol{b}}_j) = \sum_{j=1}^{m} \sum_{i=0}^{N-1} \log f(v_i^j / c_i^j) \tag{9}$$

where $c_i^j = \hat{b}_i^j + r_i^j$.

Decode to the codeword $\hat{\boldsymbol{c}} \in \mathscr{B}$ that maximizes (9). ■

For decoding the inner BCH code transforms [17], [18], trellises [14], [19] or other efficient methods [16], [20] are available. Let Ω denote the computational complexity of either of these methods. Then the number of real addition equivalent operations required by the above decoding algorithm is given by

$$\mathscr{N}_1 = \left[m\Omega + m \right] \cdot 2^\Delta = m(\Omega + 1) \cdot 2^{m(K-k)} \tag{10}$$

Evidently [17], Ω is upper bounded by $k \cdot 2^k$. However, in most cases Ω is much less than that due mainly to a precomputation stage which is common to all the cosets. For instance for the $(7, 4, 3)$ binary Hamming code $\Omega = 14$ without precomputation, and $\Omega = 3$ if a precomputation stage of 66 real operations is employed (for details see [20]).

A well-known soft-decision decoding technique is the Viterbi algorithm based on the conventional Wolf's trellis [14]. The complexity of this technique when applied to RS code is given [17] by

$$W = \left[3m(2K - 2^m + 1) + 5 \right] \cdot 2^{m(2^m - K - 1)}. \tag{11}$$

Let us compare the exponents in (10) and (11) for the case of high-rate RS codes. Define $\rho = (2^m - 1) - K$, the redundancy of $\mathscr{R}$. If $\rho < \sqrt{2^m}$ and ρ is even then the redundancy of $\mathscr{B}$ is $m\rho/2$. Hence, $k = (2^m - 1) - m\rho/2$, and therefore

$$m(K - k) = m \cdot \left((2^m - 1) - \rho - \left[(2^m - 1) - \frac{m\rho}{2} \right] \right)$$
$$= \left(\frac{m - 2}{2} \right) \cdot m(2^m - K - 1).$$

Thus for high-rate RS codes the exponent of (10) is less than or equal to the exponent of (11) for $m = 3, 4$; whereas for $m \geq 5$ the exponent of (11) is lower than that of (10).

For half-rate RS codes and also for some RS codes of slightly higher rate (e.g., $\mathscr{R}(15, 9)$, $\mathscr{R}(31, 17)$, $\mathscr{R}(31, 21)$, etc.) the situation is different. It may be easily verified that for these codes $m(K - k)$ is lower than $m(2^m - K - 1)$ for any $m = 3, 4, 5 \cdots$. Obviously, the computational gain would be even greater for extended and doubly extended RS codes.

Finally, the exponent in (10) may be further reduced by means of a recursive algorithm based on the "recursive" structure of the generator matrix derived in Fig. 2. The recursive algorithm, which is derived below, is in some cases considerably more efficient than the existing decoding techniques even for quite high-rate RS codes. Let k_j, $j = 1, 2, \cdots, q$ denote the dimension of $\mathscr{B}^{(j)}$ (to keep the notation rigorous we also define $\mathscr{B}^{(0)} = \mathscr{B}$ and $k_0 = k$). The main idea of recursive algorithm is representing each $\mathscr{B}^{(j)} \subset \mathscr{B}$ as a union of cosets in a way analogous to (4)

$$\mathscr{B}^{(j)} = \bigcup_{l=0}^{2^{\Delta_j} - 1} \mathbb{G}_l^j$$

such that $\Delta_j = \Pi_{i=1}^{j} p_i (k_j - k_{j-1})$ and $\mathbb{G}_l^j = \{ \boldsymbol{r}^l + \boldsymbol{c} \mid \boldsymbol{c} \in \mathbb{G}^j \}$ where

$$\mathbb{G}^j = \mathscr{B}_1^{(j-1)} \oplus \mathscr{B}_2^{(j-1)} \oplus \cdots \oplus \mathscr{B}_{p_j}^{(j-1)}$$

and the vectors $\boldsymbol{r}^l$, $l = 0, 1, \cdots 2^{\Delta_j} - 1$ are coset representatives for $\mathbb{G}^j$ in $\mathscr{B}^{(j)}$. Given this recursive partition into cosets we may apply Algorithm 1 recursively with $\mathscr{B}$ and $\mathscr{R}$ replaced by, respectively, $\mathscr{B}^{(j-1)}$ and $\mathscr{B}^{(j)}$ at each stage of recursion. The complexity of such recursive decoding is upper bounded by

$$\mathscr{N}_2 = m(\Omega + q) \cdot 2^{\sum_{j=1}^{q} \Delta_j}. \tag{12}$$

The primary advantage of the recursive algorithm is that $\sum_{j=1}^{q} \Delta_j$ is obviously strictly less than Δ. Thus, for instance, for the $(15, 11)$ RS code over $\mathrm{GF}(2^4)$ we have $\sum_{j=1}^{2} \Delta_j = p_1(k_1 - k) + p_1 p_2 (K - k_1) = 2(9 - 7) + 2 \cdot 2(11 - 9) = 12$ and $\Delta = m(K - k) = 16$; and for the $(15, 9)$ RS code $\sum_{j=1}^{2} \Delta_j = 14$ and $\Delta = 16$.

As the proposed decoders maximize the sum of bit and not symbol likelihoods they do not necessarily provide for the inherent burst error correction capability of Reed–Solomon codes. This, as we have already mentioned, is a direct consequence of our initial assumption of a binary *memoryless* channel. However, with only a slight modification Algorithm 1

becomes suitable for a "bursty" channel as well. We shall refer to the modified version as Algorithm 2. Assuming as in [10], [11], [21], independent noise on each transmitted symbol we may characterize a general binary channel with memory by the 2^m transition probability densities

$$f(v/\xi) = f(v^1, v^2, \cdots, v^m/\xi^1, \xi^2, \cdots, \xi^m) \quad (13)$$

where $\xi \in \mathrm{GF}(2^m)$ and $(\xi^1, \xi^2, \cdots, \xi^m)$ is a radix-2 expansion of ξ. A binary AWGN channel with bursts may be viewed as a special case of (13). Now recall that each coset of $\mathscr{R}$ is an interleaver. It is well known [22], [23] that interleaving converts a channel with memory (especially a channel with bursts) to one that can be treated as memoryless. Hence, *within a given coset* of $\mathscr{R}$ the maximization may be still performed separately for each of the m interleaved codes and the first step of Algorithm 2 is identical to that of Algorithm 1. However, at step 2 of the modified algorithm, we take into account channel memory by evaluating

$$\begin{aligned} M(c) &= \sum_{i=0}^{N-1} \log f(v_i/c_i) \\ &= \sum_{i=0}^{N-1} \log f(v_i^1, v_i^2 \cdots, v_i^m/\hat{b}_i^1 + r_i^1, \\ &\qquad \hat{b}_i^2 + r_i^2, \cdots, \hat{b}_i^m + r_i^m) \end{aligned} \quad (14)$$

and decoding to the codeword $\hat{c} \in \mathscr{R}$ that maximizes (14). Evidently the above modification almost does not affect the decoding complexity, viz. the number of real operations required by Algorithm 2 is given by

$$\mathscr{N}_3 = \left[m\Omega + 2^m - 1 \right] \cdot 2^{m(K-k)}. \quad (15)$$

Examples: In the following examples, we compare the complexities of Algorithm 2 and the conventional trellis decoding. For the (7, 5) RS code over $\mathrm{GF}(2^3)$ we have $K = 5$, $k = 4$ and $\Omega = 3$ with a precomputation stage of 66 real operations. Hence $\mathscr{N}_3 = 128$ and the total complexity of soft-decision decoding is 194 real addition-equivalent operations per codeword or about 13 operations per information bit. Decoding the same code with Viterbi algorithm based on Wolf's trellis we need $W = 2048$ real operations per codeword, whereas straightforward maximization requires about 230 000 operations. For the extended (8, 5) RS code we have $\mathscr{N}_3 = 136$ and the total complexity of soft decoding is 208 real operations per codeword or 14 operations per information bit as compared to $W = 11\,776$ operations per codeword or 785 operations per information bit. For the (15, 13), (15, 11) and (15, 9) RS codes over $\mathrm{GF}(2^4)$ we need approximately 90, 800, and 2000 real operations per information bit, respectively. The same numbers using trellis decoding are, respectively, 670, 132 000 and 19 000 000 operations per information bit. The soft-decoding complexity per information bit is plotted versus the asymptotic soft-decision coding gain $10 \log [(K/N) \cdot (N - K + 1)]$ in Fig. 4.

As illustrated by the foregoing examples the proposed algorithms are in many cases several orders of magnitude more efficient than the existing optimal techniques. In addition the structure regularity of our decoders makes them much more suitable for VLSI implementation than the conventional trellis decoding. Nevertheless, the algorithms presented herein are practically applicable only to small RS codes. Hence this paper should be viewed as just the first step towards the implementation of maximum-likelihood Reed-Solomon decoders that make full use of bit soft-decision information. However, the structure of the RS generator matrix that is derived in Section II may serve a basis for further research into, possibly suboptimal, bit level soft decoding algorithms which would be practically applicable to long RS codes.

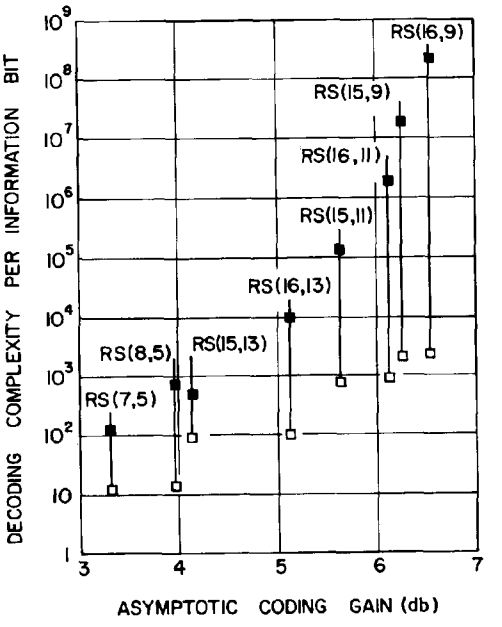

Fig. 4. Bit-level soft-decision decoding complexity for Reed-Solomon codes over $\mathrm{GF}(2^3)$ and $\mathrm{GF}(2^4)$. ■—conventional trellis decoding. □—proposed soft decoding algorithms.

ACKNOWLEDGMENT

The authors are indebted to J. Snyders for many constructive discussions and to the referees whose remarks have improved the presentation of the paper. A. Vardy wishes to thank H. Itzkowitz for her invaluable help.

REFERENCES

[1] E. R. Berlekamp, R. E. Peile, and S. P. Pope, "The application of error control to communications," *IEEE Commun. Mag.*, vol. 25, pp. 44–57, 1987.

[2] W. W. Wu, D. Haccoun, R. E. Peile, and Y. Hirata, "Coding for satellite communication," *IEEE J. Select. Areas Commun.*, vol. SAC-5, pp. 724–785, 1987.

[3] Consultative Committee for Space Data Systems, "Recommendations for Space Data System Standards: Telemetry Channel Coding," *Blue Book*, 1984.

[4] E. R. Berlekamp, J. Shifman, and W. Toms, "An application of Reed-Solomon codes to a satellite TDMA system," *MILCOM'86*, Monterey, CA.

[5] B. C. Mortimer, M. J. Moore, and M. Sablatash, "The design of a high-performance error-correcting coding scheme for the Canadian broadcast telidon system based on Reed-Solomon codes," *IEEE Trans. Commun.*, vol. COM-35, pp. 1113–1138, 1987.

[6] M. B. Pursley and W. E. Stark, "Performance of Reed-Solomon coded frequency-hop spread-spectrum communication in partial-band interference," *IEEE Trans. Commun.*, vol. COM-33, pp. 767–774, 1985.

[7] D. Divsalar, R. M. Gagliardi, and J. H. Yuen, "PPM performance for Reed-Solomon decoding over an optical-RF relay

link," *IEEE Trans. Commun.*, vol. COM-32, pp. 302–305, 1984.
[8] J. G. Proakis, *Digital Communications*. New York: McGraw-Hill, 1983.
[9] G. C. Clark and J. B. Cain, *Error Correction Coding for Digital Communications*. New York: Plenum, 1981.
[10] U. Cheng and G. K. Huth, "Bounds on the bit error probability of a linear cyclic code over GF(2^l) and its extended code," *IEEE Trans. Inform. Theory*, vol. 34, pp. 776–785, 1988.
[11] G. D. Forney, Jr., "Generalized minimum distance decoding," *IEEE Trans. Inform. Theory*, vol. IT-12, pp. 125–131, 1966.
[12] N. Doi, H. Imai, M. Izumita, and S. Mita, "Soft decision decoding for Reed-Solomon codes," in *Proc. GLOBECOM'87*, pp. 2090–2094.
[13] L. R. Welch and E. R. Berlekamp, "Error-correction for algebraic block codes," U.S. Pat. 536591, 1983.
[14] J. K. Wolf, "Efficient maximum likelihood decoding of linear block codes," *IEEE Trans. Inform. Theory*, vol. IT-24, pp. 76–80, 1978.
[15] F. J. McWilliams and N. J. A. Sloane, *The Theory of Error Correcting Codes*. Amsterdam, the Netherlands: North Holland, 1977.
[16] J. H. Conway and N. J. A. Sloane, "Soft decoding techniques for codes and lattices, including the Golay code and the Leech lattice," *IEEE Trans. Inform. Theory*, vol. IT-32, pp. 41–50, 1986.
[17] Y. Be'ery and J. Snyders, "Optimal soft decision block decoders based on fast Hadamard transform," *IEEE Trans. Inform. Theory*, vol. IT-32, pp. 355–364, 1986.
[18] ——, "A recursive Hadamard transform optimal soft decision decoding algorithm," *SIAM J. Algebraic Discrete Meth.*, vol. 8, no. 4, pp. 778–789, 1987.
[19] G. D. Forney, Jr., "Coset codes II: Binary lattices and related codes," *IEEE Trans. Inform. Theory*, vol. 34, pp. 1152–1187, 1988.
[20] J. Snyders and Y. Be'ery, "Maximum likelihood soft decoding of binary block codes and decoders for the Golay codes," *IEEE Trans. Inform. Theory*, vol. 35, 1989.
[21] M. Rice, D. J. Tait, and P. G. Farell, "A soft decision Reed-Solomon decoder," presented at the IEEE Int. Symp. Inform. Theory, Kobe, Japan, 1988.
[22] A. J. Viterbi and J. K. Omura, *Principles of Digital Communication and Coding*. New York: McGraw-Hill, 1977.
[23] J. Wolfowitz, *Coding Theorems of Information Theory*. New York: Springer-Verlag, 1978.

Alexander Vardy was born in Moscow, U.S.S.R., on November 12, 1963. He received the B.Sc. degree (Summa Cum Laude) in electrical engineering from the Technion—Israel Institute of Technology, Haifa, in 1985. He is presently working toward the Ph.D. degree in electrical engineering.

In the summer of 1985 he was with the P.T.T., Berne, Switzerland and during 1985–1990 he was a Research and Development Engineer in the Israeli Air Force. Since 1986 he has also been a Teaching Assistant at the Tel-Aviv University. He was a recipient of the Wolf Fellowship for graduate studies in 1986. His research interests include coding theory, combinatorics, and systolic arrays.

Yair Be'ery (M'87) was born in Tel-Aviv, Israel. He received the B.Sc., M.Sc., and Ph.D. degrees in electrical engineering from Tel-Aviv University, Tel-Aviv, Israel.

He is currently with the Department of Electronic Communications, Control and Computer Systems, Tel-Aviv University, Tel-Aviv, Israel. From 1979 to 1985 he was involved in research and development of digital signal processing systems and design of VLSI architectures for special purpose processors. He received the Eliyahu Golomb Award from the Ministry of Defense, Israel, in 1984, and the Rothschild Fellowship for postdoctoral studies at the Rensselaer Polytechnic Institute, Troy, NY, in 1986. His research interests include: error control coding, VLSI architectures and systolic arrays, and adaptive algorithms for speech and data transmission.