IEICE TRANS. COMMUN., VOL.E84–B, NO.3 MARCH 2001

PAPER Special Issue on Innovative Mobile Communication Technologies at the Dawn of the 21st Century

Low Complexity Soft Decision Decoding Algorithms for Reed-Solomon Codes

Branka VUCETIC†, Vishakan PONAMPALAM†, *and* Jelena VUČKOVIĆ††, *Nonmembers*

SUMMARY We propose a method to represent non-binary error patterns for Reed-Solomon codes using a trellis. All error patterns are sorted according to their Euclidean distances from the received vector. The decoder searches through the trellis until it finds a codeword. This results in a soft-decision maximum likelihood algorithm with lower complexity compared to other known MLD methods. The proposed MLD algorithm is subsequently modified to further simplify complexity, reflecting in a slight reduction in the error performance.

key words: *Reed-Solomon code, soft decision decoding*

1. Introduction

The demands of higher data rates and higher quality of service in mobile communication systems is growing rapidly. However factors such as limited transmit power, limited bandwidth and multi-path fading continue to restrict the data rates handled by practical systems. Channel coding is a fundermental technique used to combat this problem. Reed-Solomon (RS) codes are a class of non-binary codes defined over finite fields. Binary codes derived from RS codes are effective against burst and random errors which are prevalent on many real channels. For this reason RS codes are widely used for error control with applications including deep space telecommunication systems [2], compact disks [3], digital broadcasting [23] and frequency hopping spread spectrum systems [4].

The most popular decoding method in current applications relies on algebraic hard decisions for each received code symbol. The additional information provided by soft decision can enable between 2 and 3 dB coding gain on Gaussian and in excess of 10 dB on Rayleigh fading channels. One way to reduce the gap between hard decision and maximum likelihood soft decision decoding is to allow the demodulator the option of not making a decision at all when there is no clear indication that one of the symbols from the transmitter alphabet was most probable. This type of demodulator decision is known as erasure. Algebraic decoding algorithms can be modified to handle erasures [17], [18].

Further improvements in the error performance can be obtained by making use of the reliability information of the symbols within each codeword. In the Generalized Minimum Distance (GMD) Decoding algorithm [20] and its variations [7] an algebraic decoder is used to generate a list of codeword candidates. This list is chosen according to the symbol reliability information. The candidates are then compared on the basis of a sufficient optimality criterion and the most likely candidate is chosen. These algorithms are suboptimum. Their performance is slightly worse than the maximum likelihood one for codes of small dimensions but the gap grows with the code dimension.

It is well known that every linear block code can be represented by a minimal trellis [19], [22], [25] which can be used for maximum likelihood soft decision decoding by the Viterbi algorithm or its variations. The complexity of a trellis is expressed by the number of states. The trellises constructed in [19] have no more than q^{n-k} states, where q is the number of symbols in the transmitter alphabet and $n-k$ is the number of redundant symbols in the codeword. In practice, the maximum likelihood algorithm based on the Viterbi method can be applied only to codes with small redundancy or short RS codes.

Vardy and Be'ery proposed an MLD algorithm for RS codes based on their coset properties [14]. The cosets are based on a related binary BCH code. The algorithm first finds the coset that contains the codewords with the least Euclidean distance from the received word. It then finds the codeword with the minimum distance from the received sequence. It was shown that this algorithm was significantly less complex for short codes with a small number of redundant bits.

MLD can also be performed using the algorithm described in [2]. The main idea is in sorting all possible n-tuples from $GF(q)$ in respect to their Euclidean distance from the received sequence. Starting from the closest one, n-tuples are tested, one by one, successively increasing the distance from the received sequence, until the first codeword is found. This algorithm [13] is implemented for a binary code by generating a set of linearly independent columns of the code parity check matrix which add up to the syndrome. They correspond to error patterns from the set of codewords. These error patterns are represented by a binary tree and a binary tree search is performed to find the most likely error pattern. The algorithm suggested in [12] does not

Manuscript received July 3, 2000.
Manuscript revised October 16, 2000.

†The authors are with the School of Electrical and Information Engineering University of Sydney, Sydney, Australia.
††The author is with the Department of Electrical Engineering, California Institute of Technology, Pasadena, USA.

restrict itself to bit patterns that correspond to codewords. It searches through all n-bit patterns until it finds the closets codeword.

In this paper we propose a method to represent error patterns for non-binary codes. All possible error patterns are sorted according to their Euclidean distances from the received word using a trellis. The decoder searches through the trellis till it finds a codeword. This results in a soft maximum likelihood decoding algorithm with computational complexity reduced compared to the methods in [19] and [14]. The proposed MLD algorithm is subsequently modified to further simplify computations. As a result a sub-optimum algorithm is derived with only a slight sacrifice in the error performance.

The aim of this paper is to demonstrate the capability of the proposed algorithms in the general case. Hence we only present simulation results obtained for additive white Gaussian noise (AWGN) channels, which introduces random errors. However burst errors are highly likely under mobile communication environments. In such situations it is common to use interleavers, which distributes errors randomly. The proposed decoding algorithm can be easily applied on fading channels as well.

2. Description of the Algorithm

Consider a Reed-Solomon code of the code length n and information sequence length k, defined over $GF(q)$, where $q = n + 1 = 2^m$. Let

$$\mathbf{v} = (v_0, v_1, ..., v_{n-1}),\ v_i \in GF(q) \tag{1}$$

be the transmitted code word. The binary version of the transmitted codeword can be written as

$$\begin{aligned}\mathbf{v}_b = (&v_0^{(0)}, v_0^{(1)}, ..., v_0^{(m-1)}, v_1^{(0)}, \dots, v_1^{(m-1)}, ..., \\ &v_{n-1}^{(0)}, v_{n-1}^{(1)}, ..., v_{n-1}^{(m-1)}),\quad v_i^{(j)} \in GF(2)\end{aligned} \tag{2}$$

The transmitted codeword is modulated by a BPSK modulator. The transmitted modulated codeword is represented by a vector $\mathbf{v}_t$ given by

$$\begin{aligned}\mathbf{v}_t = (&v_{t,0}^{(0)}, v_{t,0}^{(1)}, ..., v_{t,0}^{(m-1)}, ..., v_{t,n-1}^{(0)}, v_{t,n-1}^{(1)}, \\ &\cdots, v_{t,n-1}^{(m-1)}),\quad v_{t,i}^{(j)} \in \{-1, 1\}\end{aligned} \tag{3}$$

where its components are computed as

$$v_{t,i}^{(j)} = 2v_{b,i}^{(j)} - 1 \tag{4}$$

The sampled channel output, denoted by $\mathbf{y}$ is given by

$$\begin{aligned}\mathbf{y} = (&y_0^{(0)}, y_0^{(1)}, ..., y_0^{(m-1)}, y_1^{(0)}, \dots, y_1^{(m-1)}, ..., \\ &y_{n-1}^{(0)}, y_{n-1}^{(1)}, ..., y_{n-1}^{(m-1)}),\quad y_i^{(j)} \in \mathcal{R}\end{aligned} \tag{5}$$

where $\mathcal{R}$ is the set of real numbers and $y_i^{(j)}$ is given by

$$y_i^{(j)} = v_{t,i}^{(j)} + n_i^{(j)} \tag{6}$$

$n_i^{(j)}$ is a sample of additive white Gaussian noise (AWGN) with variance σ^2.

The demodulator makes hard estimates on each channel output sample $y_i^{(j)}$ which is represented by a vector $\mathbf{r}_b$,

$$\begin{aligned}\mathbf{r}_b = (&r_0^{(0)}, r_0^{(1)}, ..., r_0^{(m-1)}, r_1^{(0)}, \dots, r_1^{(m-1)}, ..., \\ &r_{n-1}^{(0)}, r_{n-1}^{(1)}, ..., r_{n-1}^{(m-1)}),\quad r_i^{(j)} \in GF(2)\end{aligned} \tag{7}$$

where its components are computed as

$$r_i^{(j)} = \begin{cases} 1 & y_i^{(j)} \geq 0 \\ 0 & y_i^{(j)} < 0 \end{cases} \tag{8}$$

The non-binary received sequence can be represented as

$$\mathbf{r} = (r_0, r_1, ..., r_{n-1}),\quad r_i \in GF(q) \tag{9}$$

The RS decoder on the basis of the encoding rule and the channel model makes an estimate of the transmitted codeword, denoted by $\hat{\mathbf{v}}_b$, given by

$$\begin{aligned}\hat{\mathbf{v}}_b = (&\hat{v}_0^{(0)}, \hat{v}_0^{(1)}, ..., \hat{v}_0^{(m-1)}, \hat{v}_1^{(0)}, \hat{v}_1^{(1)} ..., \hat{v}_1^{(m-1)}, ..., \\ &\hat{v}_{n-1}^{(0)}, \hat{v}_{n-1}^{(1)}, ..., \hat{v}_{n-1}^{(m-1)}),\quad \hat{v}_i^{(j)} \in GF(2)\end{aligned} \tag{10}$$

or in the non-binary form

$$\hat{\mathbf{v}} = (\hat{v}_0, \hat{v}_1, ..., \hat{v}_{n-1}),\quad \hat{v}_i \in GF(q) \tag{11}$$

Assume for simplicity in the analysis that the received sequence is ordered: $|y_i^{(j)}| \leq |y_f^{(g)}|$ for $i \leq f$ and $j \leq g$. This is certainly not true in a practical system. However we may order the received sequence in the aim of generating test error patterns. This sequence is reordered when the most likely error pattern is found. The ordering/re-ordering are ignored in the analysis by making the above assumption.

The received vector $\mathbf{r}_b$ can be represented as

$$\mathbf{r}_b = \mathbf{v}_b + \mathbf{e}_b \tag{12}$$

where $\mathbf{e}_b$ is a binary error pattern given by

$$\begin{aligned}\mathbf{e}_b = (&e_0^{(0)}, e_0^{(1)}, ..., e_0^{(m-1)}, ..., e_{n-1}^{(0)}, \\ &e_{n-1}^{(1)}, ..., e_{n-1}^{(m-1)}),\quad e_i^{(j)} \in GF(2)\end{aligned} \tag{13}$$

The components of $\mathbf{e}_b$ are

$$e_i^{(j)} = \begin{cases} 1 & v_i^{(j)} \neq \hat{v}_i^{(j)} \\ 0 & v_i^{(j)} = \hat{v}_i^{(j)} \end{cases} \tag{14}$$

where $i = 0, 1, 2, 3, \dots, n-1$ and $j = 0, 1, 2, 3, \dots, m-1$. The non-binary error pattern is given by

$$\mathbf{e} = (e_0, e_1, ..., e_{n-1}),\quad e_i \in GF(q) \tag{15}$$

The maximum-likelihood soft decoding rule consists in

finding a codeword $\mathbf{w}$ such that the conditional probability $P(\mathbf{w}|\mathbf{y})$ is maximised. For AWGN channels this is equivalent to minimising the Euclidean distance between the received sequence $\mathbf{y}$ and a modulated codeword $\mathbf{w}$, denoted by $d_E^2(\mathbf{w},\mathbf{y})$ and given by

$$d_E^2(\mathbf{w},\mathbf{y}) = \sum_{i=0}^{n-1}\sum_{j=0}^{m-1}\left(y_i^{(j)} - w_i^{(j)}\right)^2 \tag{16}$$

Direct implementation of this algorithm is computationally intensive for long binary codes and even for short RS codes. In several implementations [12], [13] binary error patterns are added to $\mathbf{r}_b$ successively to obtain a codeword. A binary error pattern is defined by a set of non-zero error components $\{i,j\}$ arranged in an error location vector, given by

$$\begin{aligned}\mathbf{e}_l = &\{(i_1,j_1),(i_2,j_2),...,(i_\nu,j_\nu)\},\\ &0 \le i_l \le (n-1), \quad 0 \le j_l \le (m-1),\\ &\nu \le (n-1)\end{aligned} \tag{17}$$

The set of all possible binary error patterns is denoted by I.

Adding an error pattern with one non-zero binary component increases the Euclidean distance, in Eq. (16), by $4|y_i^{(j)}|$. In order to find a codeword $\mathbf{w}$ with the smallest Euclidean distance from $\mathbf{y}$ we need to identify a binary error pattern, defined by a position location vector $\mathbf{e}_l = \{(i,j)\}$ such that

$$\sum_{\{(i,j)\}\in I} |y_i^{(j)}| \tag{18}$$

is minimised. For an error pattern $\mathbf{e}_b$ we define the *Euclidean distance cost*, denoted by c, as

$$c = \sum_{i=0}^{n-1}\sum_{j=0}^{m-1} e_i^{(j)}|y_i^{(j)}| \tag{19}$$

The maximum likelihood decoding rule thus consists in minimising the Euclidean distance cost.

If binary error patterns are organised in the order of increasing Euclidean distance, the decoder tests them, one by one, starting from the closest one, successively increasing the distance, until the first codeword is found. The main challenge in the implementation of this decoding rule is to find a computationally efficient algorithm to search binary error patterns.

Ordering error patterns exactly according to their Euclidean distances requires a large number of calculations. In [12] the binary error patterns are organised into levels, corresponding roughly to Euclidean distances. If the rule proposed in [12] is applied to binary error patterns of a RS code, the level to which an error pattern is assigned, denoted by L, is calculated by the the following equation:

$$L = level(\mathbf{e}) = \sum_{i=0}^{n-1}\sum_{j=0}^{m-1} e_i^{(j)} \cdot (i\cdot m + j + 1) \tag{20}$$

The maximum number of possible levels in which error patterns could be organized is equal to $L_{max} = mn(mn+1)/2$.

The trellis structure depends only on the code length mn and the number of the level for which the trellis is constructed. It does not depend on the number of information bits k or on the received sequence.

The trellis at a level L is constructed by the following rules.

States in the trellis are denoted by S_i, $i = 0,1,...,L_{max}$. The number of states in the trellis is equal to the error pattern level for which the trellis is constructed, L plus 1. The trellis state is defined by the expression

$$S_{d+1} = S_d + \sum_{j=0}^{m-1}(d\cdot m + j + 1)\cdot e_d^{(j)} \tag{21}$$

where $e_d^{(j)}$ is a branch label. Nodes in the trellis are denoted with a pair (d,s) where $d = (0,1,...,mn)$ represents the depth and s represents the state. Every path represented in the trellis starts at node (0,0) and ends at node (n,L).

A node (d,s) is valid if there is at least one branch originating at some node at the depth $d-1$ and pointing out to node (d,s).

A branch can leave a node (d,s) if

$$S_{d+1} \le L \tag{22}$$

where S_{d+1} is defined by (21).

Every path in the trellis represents an error pattern.

The length of the trellis is equal to the code length in bits, mn. Every path which ends in the final state $(S_L\ d_j i)$, $d_j \in \{0,1,...,n-1\}$ must have the level equal to L.

The paths that do not end in the final node (n,S_L) are expurgated from the trellis.

For error patterns in the trellis we can formulate the following theorem.

Theorem 1: If all error patterns at a given level L have the Euclidean distance cost higher than c_{min} so will all error patterns on higher levels.

Proof: All error patterns on a given level $L+1$, determined by (20) can be classified into two groups. Error patterns in Group 1 are of the form

$$\mathbf{e_{L+1}^1} = (1, e_0^{(1)},...,e_0^{(m-1)},...,e_{n-1}^{(0)},e_{n-1}^{(1)},...,e_{n-1}^{(m-1)}) \tag{23}$$

and error patterns belonging to Group 2 may be represented as:

$$\begin{aligned}\mathbf{e_{L+1}^2} = &(e_0^0, e_0^{(1)},...,e_i^{(j-2)},0,1,...,\\ &e_{n-1}^{(0)},e_{n-1}^{(1)},...,e_{n-1}^{(m-1)})\end{aligned} \tag{24}$$

For every error pattern of Group 1 from level $L+1$

there exists an error pattern, e_L^1 on Level L of the form

$$\mathbf{e_L^1} = (0, e_0^{(1)}, ..., e_0^{(m-1)}, ..., e_{n-1}^{(0)}, e_{n-1}^{(1)}, ..., e_{n-1}^{(m-1)}) \tag{25}$$

It is easily shown using Eq. (19), that the cost of e_{L+1}^1 is greater than or equal to the cost of e_L^1.

Similarly for every error pattern from Group 2 of level $L+1$ there exists an error pattern on level L of the form

$$\mathbf{e_L^2} = (e_0^{(0)}, e_0^{(1)}, ..., e_i^{(j-2)}, 1, 0, ..., e_{n-1}^{(0)}, e_{n-1}^{(1)}, ..., e_{n-1}^{(m-1)}) \tag{26}$$

which has a lower cost.

Hence if c_{min} is greater than the cost of all error patterns on level L it must also be greater than the costs of all error patterns on level $L+1$. Using the same argument it is possible to show that c_{min} is also greater than the costs of error patterns belonging to higher levels.

In summary Theorem 1 says that if, while sequentially searching levels, we find a codeword corresponding to a cost c_{min} and it is greater than the cost of all error patterns on level L, we do not have to search any further. In other words this is the terminating condition of the search algorithm.

2.1 An MLD Trellis Based Decoding Algorithm

An MLD algorithm based on Theorem 1 and trellis representation of error patterns is proposed.

The trellises are constructed for all levels and stored in the decoder. Let c_{min} be the Euclidean distance between the received sequence and an intermediate codeword estimate. Its operations can be summarised as follows.

Step 1 Check whether the hard demodulated vector $\boldsymbol{r_b}$ is a codeword. If it is deliver it as an estimate of the transmitted codeword $\hat{\mathbf{v}}$ and stop. If not go to **Step 2**.

Step 2 Set $c_{min} = \infty$ and $L = 1$. Perform hard decision decoding on $\mathbf{r_b}$ and set $\hat{\mathbf{v}}$ to this codeword.

Step 3 Find all error patterns on level L that have a cost less than c_{min}. If there are no such error patterns, return current estimate as the final estimate of the transmitted codeword and **STOP**. Else go to **Step 4**.

Step 4 For each of the error patterns found in **Step 3** check if it produces a codeword when added to the hard demodulated received vector $\mathbf{r}$. If it does and the cost is less than c_{min}, store this codeword as the current estimate and make c_{min} equal to the Euclidean distance between this codeword and the received sequence.

Step 5 Increment L by 1 and go to Step 3.

2.2 A Sub-Optimum Trellis Based Decoding Algorithm

In order to simplify implementation we propose a suboptimum algorithm which is a direct simplification of the MLD trellis based algorithm. It can be summarised as follows.

Step 1 Check whether the hard demodulated vector $\mathbf{r}$ is a codeword. If it is deliver it as an estimate of the transmitted codeword $\hat{\mathbf{v}}$ and stop. If not go to **Step 2**.

Step 2 Set $c_{min} = \infty$ and $L = 1$.

Step 3 For each of the error patterns on level L check if it produces a codeword when added to the hard demodulated received vector $\mathbf{r}$. If it does store this codeword as the current estimate $\hat{v}$ and stop. Return $\hat{\mathbf{v}}$ as the decoder output. Else go to **Step 4**.

Step 4 Increment L by 1 and go to Step 3.

Observe that the terminal condition of the algorithm has been slightly relaxed compared to the MLD algorithm. As seen in the following section this significantly reduces the complexity. Interestingly only a slight reduction in the error performance has been observed for codes with small number of parity symbols.

2.3 A Lexicographic Method for Generating of Error Patterns

The trellis based decoder operates on trellises which are stored in the decoder memory. This algorithm is thus memory intensive. It is possible to compute error patterns for each level based on a Lexicographic method instead of memorising trellises. The error pattern generating rule could be derived by generalising the rule proposed in [12] for binary codes. The error patterns are generated as follows.

Error pattern level L is defined by Eq. (20) for a given (n, k) code with symbols from $GF(2^m)$. The first error pattern on this level is generated by starting with a binary sequence

$$\mathbf{e} = \left(\underbrace{1...1}_{u} \underbrace{0...0}_{nm-u} \right) \tag{27}$$

where u is minimal under the condition that

$$S = \sum_{i=1}^{u} i \geq L \tag{28}$$

If $S = L$, then $\mathbf{e}$ is the first error pattern on level L; if $S > L$ the first error pattern is generated by inverting the $(S-L)$th position in $\mathbf{e}$.

The lexicographically next error pattern on level L is computed as follows.

Reading from left to right find the first 0 following the second 1 in the previously generated error pattern. If such a 0 cannot be found, **e** is the last error pattern on level L. Otherwise, replace that 0 by 1 and put all zeros in the positions preceding it. Add the indices of the 1s in thus generated binary sequence and denote its sum by A. Set the first u positions of the sequence to 1, where u is minimal under the condition that

$$S = \sum_{i=1}^{u} i \geq L - A \tag{29}$$

If $S > (L - A)$ invert the position $S - (L - A)$. The rest of the decoding algorithm is similar to MLD algorithm described previousely.

3. Decoding Complexity

The proposed algorithm, on its way to the estimated codeword $\hat{\mathbf{c}}$, tests all n-tuples $\mathbf{z_i}$ closer to the received vector $\mathbf{y}$ than $\hat{\mathbf{c}}$. Since the density of the codewords in the space of all n-tuples with symbols from $GF(q)$ is larger for a code with smaller number of parity check symbols, the number of the tested error patterns during the decoding process is smaller. Thus the algorithm is faster for such codes [15].

The decoding complexity of the proposed algorithm is a statistical value, since it depends on the level at which the received block is decoded. Let $N_p(L)$ denote the number of all possible paths in the trellis for a given decoding level L. Let us denote the average level on which the decoding operation is completed, for a given E_b/N_0 by $\bar{L}$. Then, the average number of tested paths in trellises until the codeword is decoded denoted by $\bar{N_p}$, is given by the following equation:

$$\bar{N_p} = \sum_{i=1}^{\lceil \bar{L} \rceil} N_p(i) \tag{30}$$

where $\lceil \bar{L} \rceil$ is the smallest integer $\geq \bar{L}$. In order to compare complexities, we computed the number of addition-equivalent operations per decoded codeword, denoted by N_{op}. The number of operations performed while testing each error-pattern, denoted by n_p, is equal to the sum of the number of operations in adding that error pattern to **r** and the number of operations in calculating syndromes. Before the decoding process, the vector **y** must be ordered in non-decreasing order of $|y_i^{(j)}|$. The received sequence is of length $n \cdot m$. The complexity of this ordering process is $\Theta(nm \cdot \log(nm))$ [16]. The complexity of the re-ordering operation is identical to the above. Denote the number of ordering operation by n_s. The total number of operations required per decoded word is equal to:

$$N_{op} = \bar{N_p} \cdot n_p + 2n_s \tag{31}$$

where

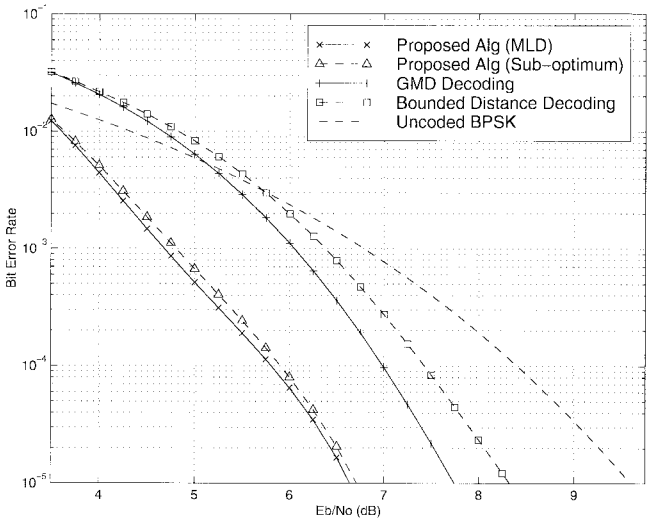

Fig. 1 Bit error rate plots for the (15,13) RS code.

$$n_p = n + 2(n - k) \cdot n \tag{32}$$

While calculating the complexity, we made the assumption that operations of comparison, addition of numbers in $GF(q)$ and the multiplication of numbers from $GF(q)$ have the same complexity.

Additional calculations are required when using the lexicographic method to generate error patterns. The complexity of this process is dependent on the length of the RS code in binary form, $n \cdot m$ and $\bar{N_p}$. However this increase in complexity is small compared to the total number of arithmetic operations required.

4. Simulation Results

The proposed algorithms were tested on a range of Reed-Solomon codes. In this section we present simulation results obtained for the (15,13), (15,11) and (31,29) RS codes, in a transmission system with binary antipodal signals over an AWGN channel.

Figure 1 shows the performance of the decoding algorithms for the (15,13) RS code. Observe that the proposed MLD algorithm has a coding gain of approximately 1.7 dB over the Berlekamp-Massey (BM) algorithm at a BER of 10^{-5}. Interestingly the sub-optimum algorithm has a coding gain of 1.6 dB over the BM algorithm and approximately 1 dB over generalized minimum distance decoding (GMD) at a BER of 10^{-5}. Thus the performance of the sub-optimum algorithm is near-optimum. However as we will see later the sub-optimum algorithm much simpler.

The simulation results obtained for the (15,11) RS code using the proposed algorithms is given in Fig. 2. Figure 2 shows that the proposed MLD and sub-optimum algorithms have a coding gain over the BM algorithm of 2.3 dB and 2.1 dB respectively at a BER of 10^{-5}. Also note that the sub-optimum algorithm has a coding gain of 1.6 dB over GMD at the

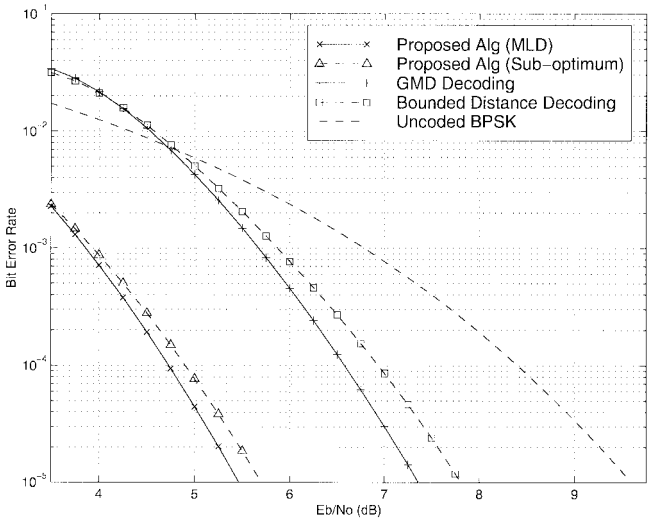

Fig. 2 Bit error rate plots for the (15,11) RS code.

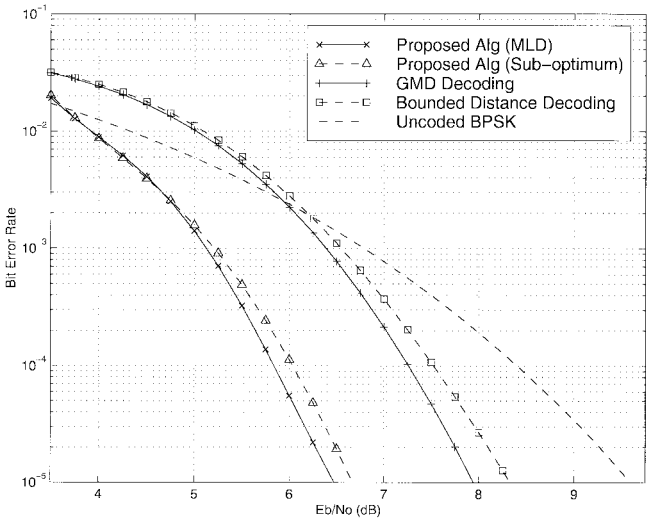

Fig. 3 Bit error rate plots for the (31,29) RS code.

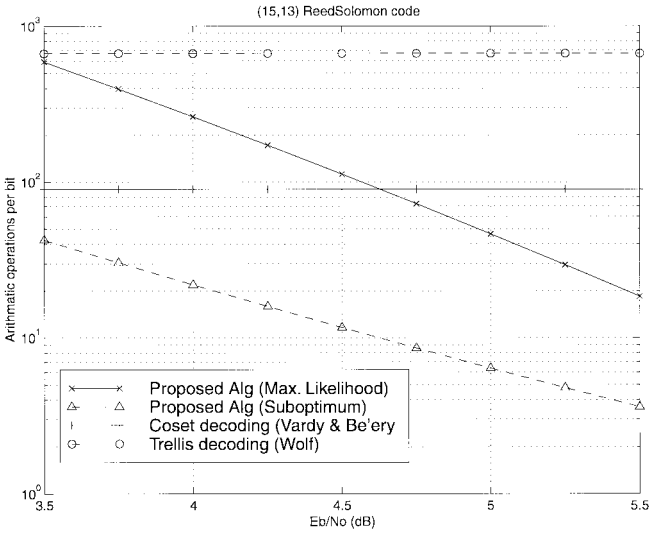

Fig. 4 Complexity of algorithms for the (15,13) RS code.

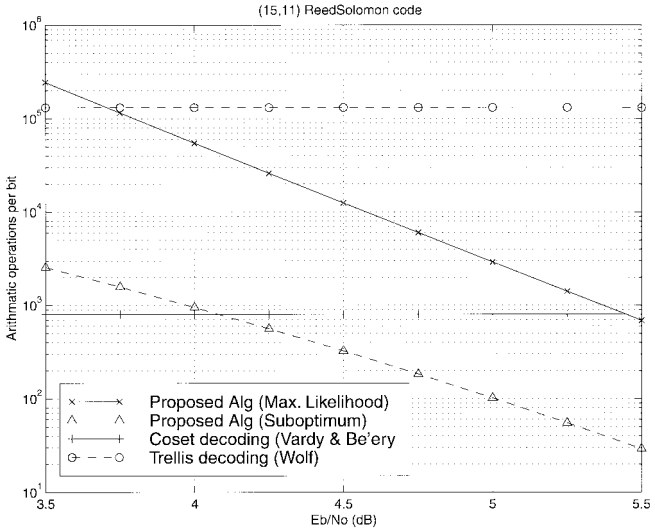

Fig. 5 Average number of arithmetic operations per decoded bit required by various decoding algorithms for the (15,11) RS code.

same BER. Simulation results for the (31,29) RS code are given in Fig. 3. We observe a 1.8 dB coding gain for the MLD and 1.7 dB gain for using the sub-optimal algorithm over the BM algorithm. The coding gain of the maximum likelihood algorithm over the sub-optimum algorithm is 0.1 dB for the (15,13) and (31,29) RS codes and 0.2 dB for the (15,11) RS code.

Figures 4 and 5 show the average number of arithmetic operations required by the proposed algorithms as a function of E_b/N_0 compared against complexities of Wolf's trellis decoding algorithm [19] and the coset decoding algorithm suggested by Vardy & Be'ery [14].

The proposed algorithms require a significantly less number of operations than the other algorithms for E_b/N_0 greater than 4 dB as shown in Figs. 4 and 5. For the (15,13) RS code, at $E_b/N_0 = 5$ dB, the Wolf and the coset algorithms require 670 and 90 operation per decoded bit, respectively. On the other hand the the proposed MLD and sub-optimum algorithm require 50 and 6 operations, respectively at the same E_b/N_0. As seen above the coding gain of a maximum likelihood algorithm is marginal over the proposed sub-optimal algorithm.

At first glance it may seem that the proposed decoding algorithms, and to that matter all SDD algorithms, are too complex to implement on hardware available today. However the proposed decoding algorithms have an important advantage that their complexities depend on the quality of the channel. Since the noise level in mobile channels change continuously, on the average, these algorithms may have reasonably low complexities. Secondly, practical systems have a

target operating bit error rates which may correspond to moderate decoding complexity. High decoding complexities are only encountered at low BER regions, in which the system may not be required to operate. For example, assume we want to obtain a BER of 10^{-4} using the (15,11) RS codes. Now for this BER, we need the channel to have $E_b/N_0 \approx 4.9\,\mathrm{dB}$. For this scenario, we can see from Fig. 5 that the proposed sub-optimal algorithm requires under 200 operations.

5. Conclusion

We propose a new MLD algorithm for RS codes based on representing binary error patterns by trellises. Each error pattern is assigned a level according to its Euclidean distance from the received sequence. Trellises are used to construct error patterns on each level and are stored in the decoder. The decoder searches the trellises starting from the closest level until it finds the closest codeword to the received sequence. The computational complexity is several orders of magnitude less than the classical Wolf trellis and the more recent Vardy-Be'ery coset decoding algorithms for large E_b/N_0.

The MLD algorithm is simplified to produce a sub-optimum algorithm with near MLD error performance. However the complexity of the sub-optimum algorithm is 1–2 orders of magnitude lower than the MLD algorithm for E_b/N_0 in the range of 2–10 dB. The complexities of the two algorithms converge at very high E_b/N_0.

As the proposed trellis based algorithms is memory intensive, a lexicographic method to generate error patterns belonging to a given level is proposed. This increases the computational complexity of the algorithm. However the additional computation required is small compared to the total number of computations.

References

[1] K.Y. Liu and J.-J. Lee, "Recent results on the use of concatenated Reed-Solomon/Viterbi channel coding and data compression for space communications," IEEE Trans. Commun., vol.COM-32, no.5, pp.518–523, May 1984.
[2] S. Wicker and V. Bhargava, eds., Reed-Solomon Codes and Their Applications, IEEE Press, 1994.
[3] K.A.S. Immink, Coding Techniques for Digital Recorders, Prentice-Hall International, 1991.
[4] M.B. Pursley, "Frequency hop transmission for satellite packet switching and terrestrial packet radio networks," IEEE Trans. Inf. Theory, vol.IT-32, no.5, pp.652–667, Sept. 1986.
[5] J. Wolf, "Efficient maximum likelihood decoding of linear block codes using a trellis," IEEE Trans. Inf. Theory, vol.24, no.1, pp.76–80, Jan. 1978.
[6] G.D. Forney, "Generalized minimum distance decoding," IEEE Trans. Inf. Theory, vol.12, no.2, pp.125–131, April 1966.
[7] D. Taipale and M. Pursley, "An improvement to generalized-minimum-distance decoding," IEEE Trans. Inf. Theory, vol.37, no.1, pp.167–172, Jan. 1991.
[8] G.D. Forney, Concatenated Codes, The M.I.T. Press, Cambridge, MA, USA, 1966.
[9] N. Doi, M. Izumita, S. Mita, and H. Imai, "Soft decision decoding for Reed-Solomon codes," Electronics and Communications in Japan, vol.72, pp.72–77, Jan. 1989.
[10] D. Taipale and M. Seo, "An efficient soft-decision Reed-Solomon decoding algorithm," IEEE Trans. Inf. Theory, vol.40, no.4, pp.1130–1139, July 1994.
[11] U. Sorger, "A new Reed-Solomon code decoding algorithm based on Newton's interpolation," IEEE Trans. Inf. Theory, vol.39, no.2, pp.358–365, March 1993.
[12] N. Lous, P. Bours, and H. Tilborg, "On maximum likelihood soft-decision decoding of binary linear codes," IEEE Trans. Inf. Theory, vol.39, no.1, pp.197–203, Jan. 1993.
[13] J. Snyders and Y. Be'ery, "Maximum likelihood soft-decision decoding of binary block codes and decoders for the Golay codes," IEEE Trans. Inf. Theory, vol.35, no.5, pp.963–975, Sept. 1989.
[14] A. Vardy and Y. Be'ery, "Bit-level soft decision decoding of Reed-Solomon codes," IEEE Trans. Commun., vol.39, no.3, pp.440–445, 1991.
[15] J. Conway and N. Sloane, Sphere Packings, Lattices and Groups, Springer-Verlag, 1991.
[16] T.H. Cormen, C. Leiserson, and R. Rivest, Introduction to Algorithms, The M.I.T. Press, 1990.
[17] R.E. Blahut, Theory and Practice of Error Control Codes, Addison-Wesley Publishing Company, 1984.
[18] S. Lin and D. Costello, Error Control Coding: Fundamentals and Applications, Prentice Hall, 1982.
[19] D. Chase, "A class of algorithms for decoding block codes with channel measurement information," IEEE Trans. Inf. Theory, vol.18, no.1, pp.170–182, Jan. 1972.
[20] D.J. Muder, "Minimum trellises for block codes," IEEE Trans. Inf. Theory, vol.34, no.5, pp.1049–1053, 1988.
[21] M. Alard and R. Lassalle, "Principles of modulation and channel coding for digital broadcasting for mobile receivers," EBU Review, no.224, pp.168–190, Aug. 1987.
[22] T. Kaneko, T. Nishijima, H. Inazumi, and S. Hirasawa, "An efficient maximum likelihood decoding of linear block codes with algebraic decoder," IEEE Trans. Inf. Theory, vol.40, no.2, pp.320–327, March 1994.
[23] G.D. Forney, Jr., "Coset codes II: Binary lattices and related codes," IEEE Trans. Inf. Theory, vol.34, no.5, pp.1152–1187, 1988.
[24] A. Lafourcade and A. Vardy, "Optimal sectionalization of a trellis," IEEE Trans. Inf. Theory, vol.42, no.3, pp.689–703, May 1996.
[25] J. Vuckovic and B. Vucetic, "Maximum likelihood decoding of Reed-Solomon codes," The International Symposium on Information Theory, ISIT'97, Ulm, Germany, 29 June–4 July 1997.

Branka Vucetic received the Ph.D. degree in electrical engineering from the University of Belgrade, Belgrade in 1982. In 1986 she joined the University of Sydney where she is currently a Professor and the Director of the Telecommunications Laboratory in the School of Electrical and Information Engineering. Her research interests include mobile communications, error control coding, modulation, multiple access schemes and channel modelling.

Vishakan Ponnampalam received a Bachelor of Engineering degree from the University of Sydney in 1995. Between 1996 and 1997 he was with the Center for Wireless Communications, National University of Singapore working on interference cancellation techniques for multiuser detection applications. Since 1997, Vishakan has been at the University of Sydney pursuing a Doctor of Philosophy degree. His current areas of interest include linear block codes, turbo block codes, and soft decoding algorithms.

Jelena Vučković was born in Niš, Yugoslavia, in 1971. She received her Dipl. Ing. degree from the University of Niš, Faculty of Electronic Engineering in 1993. During 1994 and 1995 she worked as a research and teaching assistant at the University of Niš and during 1996 as a researcher in the group of Prof. Vucetic at the University of Sydney, Australia. In September 1996, she joined California Institute of Technology (CalTech) in Pasadena, U.S.A., where she received her M.S. degree in Electrical Engineering in 1997. She is currently working towards her Ph.D. degree at CalTech.