Soft-Decision Decoding of Reed–Muller Codes: A Simplified Algorithm

Ilya Dumer, *Senior Member, IEEE*

***Abstract*—Soft-decision decoding is considered for general Reed–Muller (RM) codes of length n and distance d used over a memoryless channel. A recursive decoding algorithm is designed and its decoding threshold is derived for long RM codes. The algorithm has complexity of order $n \ln n$ and corrects most error patterns of the Euclidean weight of order $\sqrt{n/\ln n}$, instead of the decoding threshold $\sqrt{d}/2$ of the bounded distance decoding. Also, for long RM codes of fixed rate R, the new algorithm increases $4/\pi$ times the decoding threshold of its hard-decision counterpart.**

***Index Terms*—Decoding threshold, memoryless channel, Plotkin construction, recursive decoding, Reed–Muller (RM) codes.**

I. Introduction

BELOW we consider decoding algorithms for Reed–Muller (RM) codes and their subcodes. Given two positive integers r, m, where $r \le m$, RM codes—denoted below as $\left\{ {m \atop r} \right\}$—have length n, dimension k, and Hamming distance d, where

$$n = 2^m, \quad k = \sum_{i=0}^{r} \binom{m}{i}, \quad d = 2^{m-r}.$$

To describe the decoding performance of these codes in the Hamming metric, we use the notion of an asymptotic decoding threshold.

Definition 1: Given a sequence of codes A_i of growing length $n_i \to \infty$ in the Hamming spaces, we say that some decoding algorithm $\Psi(A_i)$ has decoding thresholds δ_i if for $i \to \infty$ there exists a residual sequence $\tau_i \to 0$ such that

- $\Psi(A_i)$ fails on a vanishing fraction of all error patterns that have (Hamming) weight up to $\delta_i(1-\tau_i)$;
- $\Psi(A_i)$ fails on a nonvanishing fraction of the error patterns that have (Hamming) weight up to δ_i.

Similarly, when codes A_i are used in the Euclidean or any other metric space, the thresholds δ_i and residuals τ_i are defined by considering the error weights in the corresponding metric. Note that in all cases, we estimate the thresholds δ_i up to some marginal error of order $\delta_i \tau_i$.

A number of decoding algorithms have been developed for RM codes. In the following, we briefly discuss those for which both the asymptotic performance and the algorithmic complexity are already known. Majority decoding considered in the seminal paper [1] was the first algorithm developed for RM codes. This decoding has complexity order bounded from above by nk. Subsequently, it was proven in [2] that for long RM codes of fixed rate R, majority decoding achieves the Hamming threshold of

$$\delta = (d \ln d)/4. \tag{1}$$

Another—recursive—technique is based on the Plotkin construction$(\boldsymbol{u}, \boldsymbol{u}+\boldsymbol{v})$, which decomposes RM codes $\left\{ {m \atop r} \right\}$ into the two codes

$$\left\{ {m-1 \atop r} \right\} \quad \text{and} \quad \left\{ {m-1 \atop r-1} \right\}.$$

Various recursive algorithms are introduced in [3]–[6]. For a general metric space, these algorithms guarantee bounded distance decoding [5] with a low complexity order $O(n \min(r, m-r))$. One particular design [6] addresses bounded distance decoding in the Euclidean metric, by correcting all error patterns of the Euclidean weight up to $\sqrt{d}/2$.

An efficient technique developed in [7] employs the symmetry group of RM codes. This algorithm has been analyzed for RM codes of the second order, where it substantially outperforms majority decoding. Finally, feasible maximum *a posteriori* probability (MAP) algorithms have been derived for biorthogonal and Hamming codes in [8].

Recently, new recursive algorithms were developed in [13] and [14] for decoding in the Hamming spaces. The results are summarized in the following statement, and will later be compared with a more general algorithm proposed in this paper.

Theorem 2: Long RM codes $\left\{ {m \atop r} \right\}$ can be decoded with complexity of order $(3n \log_2 n)/2$ and achieve the following thresholds in the Hamming metric:

$$\begin{aligned} &\begin{cases} \delta_* = n/2 \\ \tau_* = (4m/d)^{1/2^r}, \end{cases} && \text{if } \frac{r}{\ln m} \to 0 \\ &\begin{cases} \delta_* = (d \ln d)/2 \\ \tau_* = (\ln 4m)/\ln d, \end{cases} && \text{if } \frac{\min(r, m-r)}{\ln m} \to \infty. \end{aligned} \tag{2}$$

Note that the first case corresponds to the low-rate codes. Here the threshold (2) increases the former decoding thresholds of [3] and [5] from the order of $d/2$ to $n/2$. For codes of fixed rate R, the threshold (2) increases $\ln d$ times the threshold $d/2$ of the bounded distance decoding and also doubles that of the

Manuscript received June 4, 2004; revised November 16, 2005. This work was supported by the National Science Foundation under Grant CCR-0097125. The material of this paper was presented in part at the 37th Annual Allerton Conference on Communication, Control, and Computing, Monticello, IL, September 1999.

The author is with the College of Engineering, University of California, Riverside, CA 92521 USA (e-mail: dumer@ee.ucr.edu).

Communicated by A. E. Ashikhmin, Associate Editor for Coding Theory.
Digital Object Identifier 10.1109/TIT.2005.864425

majority decoding. Our goal is to generalize these results for an arbitrary memoryless channel.

II. Summary of the Results

In the sequel, we study *soft-decision* recursive algorithms. In doing so, we shall use the following setting. Suppose that the two equiprobable symbols ± 1 are transmitted over a symmetric memoryless channel $\mathcal{Z}_g$ with an additive noise $v \in \mathbb{R}$ that has probability density function (pdf) $g(v)$. For any received symbol x, one can readily calculate the two posterior probabilities

$$p(1\,|\,x) = \frac{g(x-1)}{g(x-1)+g(x+1)}$$
$$p(-1\,|\,x) = \frac{g(x+1)}{g(x-1)+g(x+1)}$$

of sending a 1 and a -1, respectively. Then we find their difference

$$y(x) = p(1\,|\,x) - p(-1\,|\,x) = \frac{g(x-1)-g(x+1)}{g(x-1)+g(x+1)}. \quad (3)$$

Given that the symbol 1 is being transmitted over the channel $\mathcal{Z}_g$, the first two moments of the random variable (RV) $y(x)$ equal

$$\mathsf{E}y = \int_{-\infty}^{\infty} y(x)\cdot g(x-1)\,dx$$
$$\mathsf{E}y^2 = \int_{-\infty}^{\infty} y^2(x)\cdot g(x-1)\,dx. \quad (4)$$

We shall see that our decoding thresholds can be defined in terms of parameter

$$\theta = \mathsf{E}y \cdot (\mathsf{E}y^2)^{-1/2} \quad (5)$$

which is defined by the pdf $g(v)$. The main result is given in the following theorem.

Theorem 3: Consider long RM codes $\left\{ {m \atop r} \right\}$ of length $n = 2^m$ and distance $d = 2^{m-r}$ that satisfy the restriction

$$\frac{m-r}{\ln m} \to \infty, \qquad \text{as } m \to \infty. \quad (6)$$

Let these codes be used on a memoryless channel $\mathcal{Z}_g$ such that

$$\mathsf{E}y \geq \left(\frac{1}{md}\right)^{1/2^{r-1}}. \quad (7)$$

Then these codes can be decoded with complexity of order $(3n\log_2 n)/2$ and give

- a vanishing block error probability if
$$\theta \geq \left(\frac{4m}{d}\right)^{1/2^r}; \quad (8)$$
- a nonvanishing block error probability if
$$\theta \leq \left(\frac{1}{d}\right)^{1/2^r}. \quad (9)$$

Note that Theorem 3 holds for all long RM codes with the exception of those whose distance is bounded from above by m^c for some constant $c > 0$. In Section VI, we shall consider the applications for the binary-symmetric channels (BSC) and the additive white Gaussian noise (AWGN) channels, and see that parameter θ serves as a measure of channel quality. We shall also see that for both channels, condition (7) is superseded by condition (8) and can be removed. As one particular example, let us replace any symbol x received on a general channel $\mathcal{Z}_g$ by its sign ± 1. The corresponding (hard-decision) BSC has transition error probability

$$p = \int_{-\infty}^{-1} g(v)dv. \quad (10)$$

In this case, it is readily verified that $\mathsf{E}y = 1-2p$, $\mathsf{E}y^2 = 1$, and $\theta = \mathsf{E}y$. Then Theorem 3 reads as follows.

Corollary 4: Consider long RM codes $\left\{ {m \atop r} \right\}$ that satisfy restriction (6). On a BSC with transition error probability p, these codes can be decoded with complexity of order $(3n\log_2 n)/2$ and give

- a vanishing block error probability if
$$p \leq (1 - (4m/d)^{1/2^r})/2; \quad (11)$$
- a nonvanishing block error probability if
$$p \geq (1 - (1/d)^{1/2^r})/2. \quad (12)$$

In essence, Corollary 4 shows that if equality holds in (11), then the decoding corrects most error patterns of the weight pn, but fails to do so on the weight $(p+\varepsilon)n$ for an arbitrarily small $\varepsilon > 0$. It can also be verified that the earlier Theorem 2 can be obtained by estimating the right-hand side in (11) as a function of m and r. Another important corollary is obtained for an AWGN channel with a noise power σ^2 and the pdf

$$g(v) \overset{\text{def}}{=} (2\pi\sigma^2)^{-1/2} e^{-v^2/2\sigma^2}. \quad (13)$$

In this case, recursive decoding yields the following threshold in the Euclidean metric.

Theorem 5: Long RM codes $\left\{ {m \atop r} \right\}$ can be decoded with complexity of order $(3n\log_2 n)/2$ and achieve the Euclidean threshold $\delta_\diamond$, where

$$\left(\frac{d}{4m}\right)^{2^{-r}} \lesssim \frac{\delta_\diamond}{\sqrt{n}} \lesssim d^{2^{-r}}, \quad \text{if } \frac{r}{\ln m} \to 0 \quad (14)$$
$$\frac{\delta_\diamond}{\sqrt{n}} \sim (r\ln 4)^{-1/2}, \quad \text{if } \frac{\min(r, m-r)}{\ln m} \to \infty \quad (15)$$

Note that the bounds in (14) and (15) increase about $2(n/d)^{1/2}$ times the Euclidean threshold $\sqrt{d}/2$ of the algorithms [5] and [6] that perform bounded distance decoding.

Finally, we compare two different settings: first, when decoding employs the original soft-decision outputs and, second, when it directly proceeds to the corresponding hard-decision BSC. For the latter case, the decoding thresholds δ_* are already given in (2). For comparison, we perform a similar calculations on the AWGN channels, and count the number of symbols δ_- randomly inverted in the transmission process. Namely, we shall derive the tight upper bound on the number δ_- that yields a vanishing decoding error probability. Then we have the following corollary to Theorem 3.

Corollary 6: For long RM codes $\left\{{m \atop r}\right\}$ of fixed code rate $R \in (0,1)$, the recursive soft-decision decoding of Theorem 3 increases $4/\pi$ times the corresponding hard-decision threshold δ_*

$$\delta_- = 4\delta_*/\pi.$$

The paper is organized as follows. We briefly summarize recursive properties of RM codes in Section III, where we mostly follow a more detailed discussion of [13]. Then, in Section IV, we introduce the new soft-decision algorithm Φ_r^m. In Section V, we shall begin our study of decoding performance. The analysis will be based on the central moments of the RVs recalculated in the decoding process. Theorems 3 and 5 conclude this study in Section VI.

III. Recursive Structure of RM Codes

To define RM code $\left\{{m \atop r}\right\}$, represent all 2^m positions i as points $(i_1,\ldots,i_m)$ in E_2^m. Let

$$f = f_r^m = a_0 + \sum_{s=1}^{r} \sum_{1\le j_1<\cdots<j_s\le m} a_{j_1,\ldots,j_s} i_{j_1}\cdots i_{j_s}$$

be any Boolean polynomial of degree r or less in variables $i_1,\ldots,i_m$. The corresponding codeword $\boldsymbol{c}(f)$ is obtained by assigning the value $f(i)$ to any position $(i_1,\ldots,i_m)$. We then decompose any polynomial f in the two parts

$$f_r^m(i_1,\ldots,i_m) = f_r^{m-1}(i_2,\ldots,i_m) + i_1 f_{r-1}^{m-1}(i_2,\ldots,i_m)$$

using polynomials f_r^{m-1} and f_{r-1}^{m-1} in $m-1$ variables. The corresponding codewords $\boldsymbol{u} = \boldsymbol{c}(f_r^{m-1})$ and $\boldsymbol{v} = \boldsymbol{c}(f_{r-1}^{m-1})$ belong to the codes $\{{m-1 \atop r}\}$ and $\{{m-1 \atop r-1}\}$, respectively. Also, now $\boldsymbol{c}$ is represented in the form $\boldsymbol{c} = \boldsymbol{u}, \boldsymbol{u}+\boldsymbol{v}$, which is the well-known Plotkin construction [9]. By continuing this process, we obtain four RM codes of length 2^{m-2} and proceed further. Finally, we end this splitting at the biorthogonal codes $\left\{{g \atop 1}\right\}$ or full spaces $\left\{{h \atop h}\right\}$, where

$$g = 1,\ldots,m-r+1, \qquad h = 1,\ldots,r.$$

Remark: Later we will see that ending this process at the codes $\left\{{g \atop 1}\right\}$ increases the decoding threshold when compared to the full splitting that ends at the repetition codes $\left\{{g-1 \atop 0}\right\}$.

We also enumerate the above procedures as follows. We say that $\boldsymbol{c}$ is split onto two "paths" $\boldsymbol{u}$ and $\boldsymbol{v}$ and assign the path value $\xi_1 = 0$ to a $\boldsymbol{v}$-component and $\xi_1 = 1$ to a $\boldsymbol{u}$-component. In each step i of our splitting, we repeat the same procedure for the component ξ_i. If our splitting ends at some left-end code $\left\{{g \atop 1}\right\}$, this gives us the corresponding binary path

$$\xi \stackrel{\text{def}}{=} (\xi_1,\ldots,\xi_{m-g})$$

of length $m-g$. Otherwise, if the splitting ends at the right-end node $\left\{{h \atop h}\right\}$, we obtain a path of length $m-h$, which is denoted as

$$\xi \stackrel{\text{def}}{=} (\xi_1,\ldots,\xi_{m-h}).$$

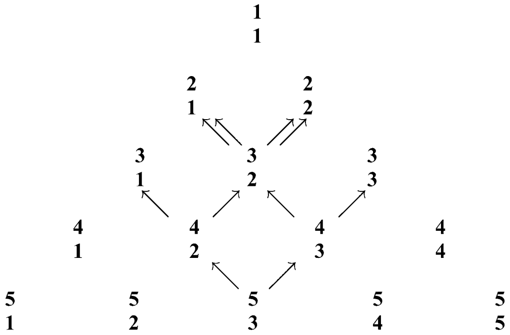

Fig. 1. Decomposition of RM code $\left\{{5 \atop 3}\right\}$.

Example: In Fig. 1, this decomposition process is shown for the RM code $\left\{{5 \atop 3}\right\}$ of length 32.

This code is first split into $\left\{{4 \atop 2}\right\}$ code $\{\boldsymbol{v}\}$ and $\left\{{4 \atop 3}\right\}$ code $\{\boldsymbol{u}\}$. Code $\{\boldsymbol{v}\}$ is then split into codes $\left\{{3 \atop 1}\right\}$ and $\left\{{3 \atop 2}\right\}$, while $\{\boldsymbol{u}\}$ is split into $\left\{{3 \atop 2}\right\}$ and $\left\{{3 \atop 3}\right\}$. In the final step, two codes $\left\{{3 \atop 2}\right\}$ obtained in the previous step are again split into codes $\left\{{2 \atop 1}\right\}$ and $\left\{{2 \atop 2}\right\}$. This gives six different paths:

the leftmost path $(0,0)$, which ends at the code $\left\{{3 \atop 1}\right\}$;
the paths $(0,1,0)$ and $(1,0,0)$, which end at $\left\{{2 \atop 1}\right\}$;
the paths $(0,1,1)$ and $(1,0,1)$, which end at $\left\{{2 \atop 2}\right\}$;
the rightmost path $(1,1)$, which ends at $\left\{{3 \atop 3}\right\}$.

Note that our polynomial f_r^m is defined by the set of binary coefficients

$$\boldsymbol{a}_r^m = \{a_{j_1,\ldots,j_s} \,|\, s = 0,\ldots,r\}$$

which in essence form k information bits that encode a vector $(\boldsymbol{u}, \boldsymbol{u}+\boldsymbol{v})$. Then the preceding procedure also decomposes $\boldsymbol{a}_r^m$ into two information subblocks which encode vectors $\boldsymbol{u}$ and $\boldsymbol{v}$. Proceeding in the same way, we see that any codeword can be encoded from the information strings assigned to the different end nodes $\left\{{g \atop 1}\right\}$ and $\left\{{h \atop h}\right\}$. Also, the encoding follows the same splitting process and can be enumerated by the above paths ξ. Thus, any path ξ that ends at the left-end code $\left\{{g \atop 1}\right\}$ gives $g+1$ information bits, while any path terminated at the right-end code $\left\{{h \atop h}\right\}$ gives 2^h bits. In the sequel, we use notation $a(\xi)$ for the specific information string associated with any path ξ.

IV. Recursive Decoding Algorithm

Let any binary symbol c be mapped onto $(-1)^a$. Then any codeword of RM code belongs to $\{1,-1\}^n$ and has the form

$$\boldsymbol{c} = (\boldsymbol{u}, \boldsymbol{uv}).$$

This codeword is transmitted over a memoryless channel $\mathcal{Z}_g$. The received block $x \in \mathbb{R}^n$ consists of the two halves $\boldsymbol{x}'$ and $\boldsymbol{x}''$, which are the corrupted images of vectors $\boldsymbol{u}$ and $\boldsymbol{u}v$. By taking the symbols x_i' and x_i'' for each $i = 1, \dots, n/2$, the decoder finds the posterior probabilities

$$q_i' \stackrel{\text{def}}{=} \Pr\{u_i = 1 \,|\, x_i'\}, \quad q_i'' \stackrel{\text{def}}{=} \Pr\{u_i v_i = 1 \,|\, x_i''\}.$$

We first try to find the codeword $\boldsymbol{v}$ from $\left\{ \begin{smallmatrix} m-1 \\ r-1 \end{smallmatrix} \right\}$ and then proceed with the block $\boldsymbol{u} \in \left\{ \begin{smallmatrix} m-1 \\ r \end{smallmatrix} \right\}$.

Step 1. Here we use both probabilities q_i' and q_i'' to derive the posterior probability

$$q_i^v \stackrel{\text{def}}{=} \Pr\{v_i = 1 \,|\, x_i', x_i''\}$$

of the symbol $v_i = u_i \cdot (u_i v_i)$ of codeword $\boldsymbol{v}$. Obviously, $v_i = 1$ iff the symbols u_i and $u_i v_i$ are equal. Thus, using the formula of total probability, we find

$$q_i^v = q_i' q_i'' + (1 - q_i')(1 - q_i''). \tag{16}$$

Symbols q_i^v are combined into the vector $\boldsymbol{q}^v$ of length $n/2$. Then we use some soft-decision decoder $\Psi_v(\boldsymbol{q}^v)$ specified later. The decoding result $\hat{\boldsymbol{v}} \in \left\{ \begin{smallmatrix} m-1 \\ r-1 \end{smallmatrix} \right\}$ is then passed to Step 2. □

Step 2. We first take the channel outputs x_i'' and the decoded symbols $\hat{v}_i$ to find the posterior probability

$$\hat{q}_i \stackrel{\text{def}}{=} \Pr\{u_i = 1 \,|\, x_i'', \hat{v}_i\}$$

of the symbol u_i on the *right half* $\boldsymbol{uv}$. Here we assume that *Step 1 gives the correct vector* $\hat{\boldsymbol{v}} = \boldsymbol{v}$, and take

$$\hat{q}_i = \begin{cases} q_i'', & \text{if } \hat{v}_i = +1 \\ 1 - q_i'', & \text{if } \hat{v}_i = -1. \end{cases}$$

Now we have the two posterior probabilities q_i' and $\hat{q}_i$ of the same symbol u_i. By using the Bayes' rule, we find the combined estimate

$$\hat{q}_i^u \stackrel{\text{def}}{=} \Pr\{u_i = 1 \,|\, q_i', \hat{q}_i\} = \frac{q_i' \hat{q}_i}{q_i' \hat{q}_i + (1 - q_i')(1 - \hat{q}_i)}. \tag{17}$$

Symbols $\hat{q}_i^u$ form the vector $\hat{\boldsymbol{q}}^u$ of length $n/2$. Then we use some (soft-decision) decoding algorithm $\Psi_u(\hat{\boldsymbol{q}}^u)$ and find a subblock $\hat{\boldsymbol{u}} \in \left\{ \begin{smallmatrix} m-1 \\ r \end{smallmatrix} \right\}$. □

To simplify our notation, in the following we use an equivalent description. Given position i, we use (3) to find *the difference* between the two posterior probabilities q_i' and $1 - q_i'$ of sending 1 and -1

$$y_i' \stackrel{\text{def}}{=} 2q_i' - 1.$$

Similarly, define the differences $y_i'' = 2q_i'' - 1$ on the right half and combine all n symbols y_i' and y_i'' into the vector $\boldsymbol{y} = (\boldsymbol{y}', \boldsymbol{y}'')$. It is readily verified that formulas (16) and (17) are rewritten as follows:

$$y_i^v = y_i' y_i'' \tag{18}$$
$$\hat{y}_i^u = (y_i' + y_i'' \hat{v}_i)/(1 + y_i' y_i'' \hat{v}_i). \tag{19}$$

However, it is yet an open problem to estimate the asymptotic thresholds when function (19) is used in our recalculations. Therefore, we keep the first rule (18) but replace (19) with approximation rule

$$y_i^u = (y_i' + y_i'' \hat{v}_i)/2. \tag{20}$$

We will see in Section VII that such a replacement does not degrade the decoding threshold.

In a more general scheme Φ_r^m, we keep decomposing subblocks $\hat{\boldsymbol{v}}$ and $\hat{\boldsymbol{u}}$ using vectors $\boldsymbol{y}^v$ and $\boldsymbol{y}^u$ as our new inputs, instead of the former vectors $\boldsymbol{q}^v$ and $\boldsymbol{q}^u$. In all intermediate steps, we only recalculate the quantities y_i^v and y_i^u using (18) and (20). Finally, we perform (soft-decision) minimum-distance (MD) decoding once we reach the biorthogonal codes $\left\{ \begin{smallmatrix} g \\ 1 \end{smallmatrix} \right\}$ or full spaces $\left\{ \begin{smallmatrix} h \\ h \end{smallmatrix} \right\}$. In the latter case, the decoder performs trivial bit-by-bit decision, while in the biorthogonal codes this is done by choosing the code vector $\hat{\boldsymbol{c}}$ with the maximum inner product

$$\forall \boldsymbol{c} \in \left\{ \begin{smallmatrix} g \\ 1 \end{smallmatrix} \right\} : (\hat{\boldsymbol{c}}, \boldsymbol{y}) \geq (\boldsymbol{c}, \boldsymbol{y}). \tag{21}$$

The decoded codeword $\hat{\boldsymbol{c}}$ and the corresponding information block $\hat{\boldsymbol{a}}$ are now obtained as follows.

```
Algorithm $\Phi_r^m\ (\boldsymbol{y})$ for an input vector $\boldsymbol{y}$.
1. If $1 < r < m$, execute the following.
    1.1. Calculate quantities $y_i^v = y_i' y_i''$.
    Decode $\boldsymbol{y}^v$ into the vector $\hat{\boldsymbol{v}} = \Phi_{r-1}^{m-1}(\boldsymbol{y}^v)$.
    Pass $\hat{\boldsymbol{v}}$ and $\hat{\boldsymbol{a}}^v$ to Step 1.2
    1.2. Calculate quantities $y_i^u = (y_i' + y_i'' \hat{v}_i)/2$.
    Decode $\boldsymbol{y}^u$ into the vector $\hat{\boldsymbol{u}} = \Phi_r^{m-1}(\boldsymbol{y}^u)$.
    Output decoded components
    $\hat{\boldsymbol{a}} := (\hat{\boldsymbol{a}}^v \,|\, \hat{\boldsymbol{a}}^u)$; $\hat{\boldsymbol{c}} := (\hat{\boldsymbol{u}} \,|\, \hat{\boldsymbol{u}}\hat{\boldsymbol{v}})$.
2. If $r = 1$, use MD decoding (21) at $\left\{ \begin{smallmatrix} r \\ 1 \end{smallmatrix} \right\}$.
3. If $r = m$, use MD decoding at $\left\{ \begin{smallmatrix} r \\ r \end{smallmatrix} \right\}$.
```

Note that recalculations (18) require $n/2$ operations. In addition, note that MD decoding (21) of any node gives the same results if all symbols of the input vector $\boldsymbol{y}$ are scaled proportionally. Therefore, we can replace (20) by a simpler rule

$$y_i^u = y_i' + y_i'' \hat{v}_i$$

which requires n (floating-point) operations. Therefore, our decoding complexity satisfies the following recursion:

$$|\Phi_r^m| \leq |\Phi_{r-1}^{m-1}| + |\Phi_r^{m-1}| + 3n/2.$$

To find the complexity order $|\Phi_r^m|$, we also use the following "boundary" conditions. MD decoding Φ_h^h of trivial codes of any length n can be executed in n operations. MD decoding of biorthogonal codes of any length n requires at most $n \log_2 2n + 3$ operations (see [9, Sec. 14.4]). Now we obtain the complexity estimate, which is similar to that of [13].

Lemma 7: Algorithm Φ_r^m decodes RM codes $\left\{ \begin{smallmatrix} m \\ r \end{smallmatrix} \right\}$ with complexity

$$|\Phi_r^m| \leq 2n \min(r, m - r) + n(m - r + 1).$$

Note that $|\Phi_r^m|$ has the maximum order of $3nm/2$ operations claimed in Theorems 3 and 5. Our goal now is to estimate the correcting capability of the algorithm, which will be done in the following two sections.

V. Preliminary Probabilistic Analysis of the Algorithm

A. Recalculation of the Outputs for Φ_r^m

Our next goal is to analyze the decoding error probabilities obtained on the different paths ξ. First, consider two paths ξ and ς, and let l be the first (senior) position where these two disagree. Then we say that ξ succeeds ς and write $\xi > \varsigma$, if $\xi_l = 1$ and $\varsigma_l = 0$. Note that after $l-1$ identical decoding steps, path ς moves left while ξ moves right. Therefore, ξ is processed after ς.

Now let us consider any left-end subpath ξ, that ends on the biorthogonal code $\{ {g \atop 1} \}$. For any $s = 1, \dots, 2^{g+1}$, let $c^{(s)}$ denote the sth codeword of this code, and let $I^{(s)}$ be its support, that is the subset of code positions with symbols-1. Let also a^l denote the vector that consists of l identical symbols $a \in \{0,1\}$. Here we take $c^{(1)} = 1^{2^g}$ and $c^{(2)} = -c^{(1)}$. Then

$$|I^{(1)}| = 0, \quad |I^{(2)}| = 2^g, \quad |I^{(s)}| = 2^{g-1}, \quad s = 3, \dots, 2^{g+1}.$$

Without loss of generality, we assume that the codeword 1^n is transmitted. Then the correct decoding outputs an all-one codeword $c^{(1)}$ on any end node $\left\{ {g \atop 1} \right\}$. Given the original vector $\boldsymbol{y}$, let $\boldsymbol{y}(\xi)$ be the vector obtained by recalculations (18) and (20) on a subpath ξ. For any support $I^{(s)}$, it will be convenient to use the weighted sum

$$Y^{(s)}(\xi) = \sum_{i \in I^{(s)}} y_i(\xi)/2^{g-1}, \qquad s = 2, \dots, 2^{g+1}. \tag{22}$$

It follows from definition (21) of MD decoding, that the codeword $c^{(1)}$ is not chosen if[1] $Y^{(s)}(\xi) < 0$ for any $s \geq 3$. To simplify our recalculations (18) and (20) on any path ξ, we wish to consider only the case when all preceding MD decodings (21) return *correct vectors* $\boldsymbol{v}(\varsigma) = \mathbf{1}$ from all the previous paths $\varsigma < \xi$. In this case, recalculations (18) and (20) are reduced on any path ξ to a simpler form

$$\boldsymbol{y}^v = \boldsymbol{y}'\boldsymbol{y}'', \quad \boldsymbol{y}^u = (\boldsymbol{y}' + \boldsymbol{y}'')/2. \tag{23}$$

Below we also consider any incomplete (sub)path ξ of some length $l \in [1, m-1]$ and its immediate prefix $\underline{\xi}$

$$\xi = (\xi_1, \dots, \xi_l), \quad \underline{\xi} = (\xi_1, \dots, \xi_{l-1}).$$

For any subpath ξ and any step l, we can now recalculate the outputs $\boldsymbol{y}(\xi)$ using a simplified recursion

$$\boldsymbol{y}(\xi) = \begin{cases} \boldsymbol{y}'(\underline{\xi}) \cdot \boldsymbol{y}''(\underline{\xi}), & \text{if } \xi_l = 0 \\ \boldsymbol{y}'(\underline{\xi})/2 + \boldsymbol{y}''(\underline{\xi})/2, & \text{if} \xi_l = 1 \end{cases} \tag{24}$$

which is *independent of the previous results*. Then the event

$$\boldsymbol{Y}_-(\xi) = \bigcup_{s=3}^{2^{g+1}} \{Y^{(s)}(\xi) < 0\} \tag{25}$$

represents the decoding failure on any path ξ. This has probability

$$p(\xi) = \Pr \boldsymbol{Y}_-(\xi). \tag{26}$$

Now consider any right-end path ξ, that ends on the code $\left\{ {h \atop h} \right\}$ of length 2^h. Here, the vector $\boldsymbol{y}(\xi)$ consists of 2^h symbols $y_i(\xi)$. We shall see that the index i can be dropped; therefore, in the sequel $y(\xi)$ denotes any symbol obtained on this path ξ. Correspondingly, MD decoder (21) makes a bit-wise decision

$$a(\xi) = \operatorname{sign} y(\xi)$$

which is incorrect with probability

$$p(\xi) = \Pr\{y(\xi) < 0\}. \tag{27}$$

These probabilities $p(\xi)$ are used in the following lemma.

Lemma 8: Block error probability P taken over all k information paths ξ satisfies inequalities

$$p(0^{r-1}) \leq P \leq \sum_{\xi} p(\xi) \leq k \max_{\xi} p(\xi). \tag{28}$$

Proof: Here the lower bound is the probability of the immediate failure on the first (leftmost) path 0^{r-1}. To derive the upper bound, consider the probability

$$P(\xi) = \Pr \left\{ \boldsymbol{Y}_-(\xi) \bigcap_{\varsigma < \xi} \boldsymbol{Y}_+(\varsigma) \right\} \tag{29}$$

that ξ is the first erroneous path. Here we take the complementary events $\boldsymbol{Y}_+(\varsigma) = \overline{\boldsymbol{Y}}_-(\varsigma)$ on all previous paths $\varsigma < \xi$. Obviously, $p(\xi) \geq P(\xi)$, since the right-hand side of (29) includes intersecting events, of which definition (26) keeps only the event $\boldsymbol{Y}_-(\xi)$. Also, note that $P = \sum_{\xi} P(\xi)$. Thus, we obtain the upper bound (28). □

B. Asymptotic Setting

Our goal now is to estimate the maximum error probability $p(\xi)$ obtained over all paths ξ. To proceed further, we use the following setup.

1. First, note that the original blocks $\boldsymbol{y}'$ and $\boldsymbol{y}''$ are derived from the different channel bits x_i. Consequently, their descendants $\boldsymbol{y}'$ and $\boldsymbol{y}''$ are also obtained from the different channel bits. Then this process repeats itself in (24). Thus, all symbols $y_i(\xi)$ of any vector $\boldsymbol{y}(\xi)$ are independent and identically distributed (i.i.d.) RVs. It is for this reason that we use the common notation $y(\xi)$ for any random symbol $y_i(\xi)$ in (27).

2. Next, note that any RV $Y^{(s)}(\xi)$ of (22) is the sum of 2^{g-1} i.i.d. RV $y_i(\xi)$ for any $s \geq 3$. Therefore, it has the same pdf for any subset $I^{(s)}$, and we can consider the single RV

$$Y(\xi) = \sum_{i=1}^{2^{g-1}} y_i(\xi)/2^{g-1}. \tag{30}$$

In the sequel, we also add the suffix 1^{g-1} to any left-end path ξ ending at the node $\left\{ {g \atop 1} \right\}$ and obtain the *extended paths*, all of which have the same length $m-1$ and end at the same "sink"

[1] Here we assume that $Y^{(s)} = 0$ is accounted as a negative quantity with probability $1/2$.

$\left\{{1 \atop 1}\right\}$. In this case, RV $Y(\xi)$ can be considered as the result of recalculations (24) performed on the suffix 1^{g-1}.

3. Let $\mathsf{E}(\xi) = \mathsf{E}, Y(\xi)$ denote the expectation of an RV $Y(\xi)$. Originally, all RV y_i have positive expectations $\mathsf{E}(y_i)$. Thus, the expectations $\mathsf{E}(\xi)$ remain positive on any path ξ due to recalculations (24). Below, we consider the normalized RV

$$\varkappa(\xi) = Y(\xi)/\mathsf{E}(\xi). \tag{31}$$

By replacing the sum $Y^{(s)}(\xi)$ in definition (25) with $\varkappa(\xi)$, we obtain the following bounds:

$$\Pr\{\varkappa(\xi) \le 0\} \le p(\xi) \le 2^{g+1} \Pr\{\varkappa(\xi) < 0\}. \tag{32}$$

Here, the probability of incorrect decoding into any single codeword $c^{(s)}$ serves as a lower bound, and the union bound is used as its upper counterpart.

4. For any path ξ, we consider the RV $\varkappa(\xi)$ and define its jth central moment

$$E_j(\xi) \overset{\text{def}}{=} \mathsf{E}(\varkappa(\xi) - 1)^j \tag{33}$$

where j is a positive integer. To upper-bound the error probabilities $p(\xi)$, we shall combine (32) with the Chebyshev inequality for the moments of any even order j

$$\begin{aligned} \Pr\{\varkappa(\xi) < 0\} &\le E_j(\xi) \\ p(\xi) &\le 2^{g+1} E_j(\xi). \end{aligned} \tag{34}$$

5. We shall take $j = 4m$ and use a relatively high central moment $E_{4m}(\xi)$, which will give the best estimates in the upper bound (32). By contrast, the low moments—say $E_2(\xi)$—yield much weaker bounds. In the sequel, we shall estimate the largest moment $\max E_{4m}(\xi)$ taken over all paths ξ.

C. *Ordering of the Paths*

Our goal now is to find the maximum moment $E_{4m}(\xi)$. Direct calculations —using recursive formulas (24)—yield very bulky expressions for the general moments $E_j(\xi)$ and even the variances $E_2(\xi)$ on most paths ξ. Therefore, we shall replace exact calculations with a partial ordering of the moments $E_j(\xi)$. This ordering will be performed in Theorems 9 and 10, and shows that $\max E_j(\xi)$ is achieved on the leftmost (first) path

$$\pounds = 0^{r-1}, 1^{m-r}. \tag{35}$$

Thus, our problem will be reduced to finding $E_{4m}(\pounds)$.

Discussion: Recall that for any path $\xi = (\xi_1, \ldots, \xi_{m-1})$, we take two i.i.d. RVs y_1 and y_2 on any intermediate step i. Then for $\xi_i = 0$, we perform a "non-Gaussian" recalculation $y_1 y_2$, while for $\xi_i = 0$, the transformation $(y_1 + y_2)/2$ makes the result more Gaussian-like. Therefore, we can assume that the Gaussian approximation is valid for any path ξ with a long all-one tail 1^g. However, it is easy to verify that most paths ξ end at the nodes $\left\{{g \atop 1}\right\}$ with a small g, and have distributions different from the normal pdf (similarly to the message-passing algorithms). It is for this reason that we seek the worst path $\pounds$. Now consider two subpaths ξ_{01} and ξ_{10} that disagree in their last two positions

$$\begin{cases} \xi_{01} = (\xi_1, \ldots, \xi_{l-2}, 0, 1) \\ \xi_{10} = (\xi_1, \ldots, \xi_{l-2}, 1, 0). \end{cases} \tag{36}$$

We also add the same suffix $\bar{\xi}$ to both paths and say that the paths $\xi_{\text{left}} = (\xi_{01}, \bar{\xi})$ and $\xi_{\text{right}} = (\xi_{10}, \bar{\xi})$ are neighbors. The following theorem is central to our analysis. Its proof—given in Appendix I—does not depend on the original pdf of random variables x and can be applied to a general channel $\mathcal{Z}_g$.

Theorem 9: For any even positive j, any two neighbors ξ_{left} and ξ_{right} satisfy inequality

$$E_j(\xi_{\text{left}}) \ge E_j(\xi_{\text{right}}). \tag{37}$$

This theorem leads to the following important statement proven in Appendix II.

Theorem 10: For any path ξ and any even positive j, the error probability $p(\xi)$ on any path ξ can be bounded using the moment $E_j(\pounds)$ of the leftmost path $\pounds$

$$p(\xi) \le 2^{m-r+2} E_j(\pounds). \tag{38}$$

The overall block error probability can be bounded as follows:

$$\Pr\{\varkappa(\pounds) \le 0\} \le P \le (4kd) E_j(\pounds). \tag{39}$$

VI. Decoding Thresholds

A. *Recalculation of the Largest Variance E_2*

It is readily verified that for any position i, the normalized channel outputs $y_i/\mathsf{E}y_i$ have the range $[-1/\mathsf{E}y, 1/\mathsf{E}y]$ and the variance

$$\mathsf{E}y^2/(\mathsf{E}y)^2 - 1 = \theta^{-2} - 1 \tag{40}$$

that only depends on the parameter θ of (5). Our next goal is to estimate the variance E_2 on the leftmost path $\pounds$.

Lemma 11: On a memoryless channel $\mathcal{Z}_g$, the leftmost path $\pounds$ has the variance

$$E_2 = 2^{-(m-r)}(\theta^{-2^r} - 1). \tag{41}$$

Proof: We use equalities (24). Consider any path $\xi = (\underline{\xi}, \xi_l)$ with the last bit ξ_l. Then it is easy to verify that the variance $E_2(\xi)$ satisfies recursions

$$\begin{aligned} E_2(\xi) + 1 &= (E_2(\underline{\xi}) + 1)^2, &&\text{if } \xi_l = 0 \\ E_2(\xi) &= E_2(\underline{\xi})/2, &&\text{if } \xi_l = 1. \end{aligned} \tag{42}$$

The first recursion in (42) shows that on the prefix 0^{r-1}, the original variance $\theta^{-2} - 1$ of (40) is replaced with

$$E_2(0^{r-1}) = \theta^{-2^r} - 1.$$

Then the second recursion of (42) is applied to the suffix 1^{m-r}. This gives equality (41). □

B. *Asymptotic Threshold*

Proof of Theorem 3: To prove estimates (8) and (9), we take the corresponding boundary values

$$\hat{\theta} = (4m/d)^{1/2^r} \tag{43}$$

$$\tilde{\theta} = (1/d)^{1/2^r} \tag{44}$$

and find the two values of the variance $E_2(\pounds)$ in (41)

$$\hat{E}_2 = (4m)^{-1} - 2^{r-m}, \quad \tilde{E}_2 = 1 - 2^{r-m}. \tag{45}$$

Next, recall that $\varkappa(\pounds)$ is the sum of $d = 2^{m-r}$ i.i.d. RVs $\varkappa_i(0^{r-1})$. All these variables have the mean $\bar{\varkappa}_i = 1$ and the variance $\sigma^2(\varkappa_i) = \theta^{-2^r} - 1$. Note that $\sigma^2(\varkappa_i) \le d$ for both values of θ in (43) and (44). Also, condition (7) shows that all RVs $\varkappa_i$ are limited as

$$\varkappa_i \in [-(\mathsf{E}y)^{-2^{r-1}}, (\mathsf{E}y)^{-2^{r-1}}] \subseteq [-md, md]. \qquad (46)$$

Now it follows from [10, Theorem VIII.3] that the sum $\varkappa(\mathcal{L})$ has the pdf that tends to the Gaussian distribution $\mathcal{N}(1, E_2)$ where the variance E_2 is defined in (41). Also, restriction (6) shows that the number d of these RVs $\varkappa_i$ grows faster than any polynomial order m^c. For this reason, we can approximate the residual behavior of $\varkappa(\mathcal{L})$ by the Gaussian distribution as $m \to \infty$. (See also Remark 1 following the proof.) In particular, the lower bound (39) is approximated as

$$\Pr\{\varkappa(\mathcal{L}) \le 0\} \to Q(\tilde{E}_2^{-1/2}) \to Q(1), \qquad m \to \infty \qquad (47)$$

where

$$Q(t) \overset{\text{def}}{=} (2\pi)^{-1/2} \int_t^{\infty} \exp\{-t^2/2\} dt.$$

Thus, our choice of $\tilde{\theta}$ gives a nonvanishing block error probability, and (9) is proven.

To prove (8), we approximate $\varkappa(\mathcal{L})$ by the Gaussian distribution $\mathcal{N}(1, \hat{E}_2)$. To apply the upper bound (39), we need to estimate the moment $E_{4m}(\mathcal{L})$. Here we first calculate the central moment $\hat{E}_{4m}$ of the normal RV $\mathcal{N}(1, \hat{E}_2)$

$$\begin{aligned}\hat{E}_{4m} &= (4m-1)!! \cdot (\hat{E}_2)^{2m} \\ &< (4m/e)^{2m} \cdot (4m)^{-2m} = e^{-2m}. \qquad (48)\end{aligned}$$

Next, we use $\hat{E}_{4m}$ as an asymptotic approximation of $E_{4m}(\mathcal{L})$. (See also Remark 2 following the proof.) Then the upper bound in (39) reads

$$P \lesssim (4kd)\hat{E}_{4m} < 2^{2m} e^{-2m}. \qquad (49)$$

Thus, we obtain the vanishing block error probability for $m \to \infty$, and the proof is completed. □

Remarks:

1. Consider the sum $\varkappa(\mathcal{L})$ of $L = 2^{m-r}$ i.i.d. RV with the mean value of 1. It follows from [10, Theorem XVI.7.1] that the residual probability $\Pr\{\varkappa(\mathcal{L}) \le 0\}$ tends to its Gaussian approximation (47) if the corresponding number of standard deviations $E_2^{-1/2}$ is small relative to L

$$E_2^{-1/2} = o(L^{1/6}).$$

Both quantities $\hat{E}_2$ and $\tilde{E}_2$ satisfy this asymptotic due to the restriction (6).

2. Asymptotic approximation $E_{4m}(\mathcal{L}) \sim \hat{E}_{4m}$ for large m is also related to Remark 1. Namely, to estimate the moment

$$E_{4m}(\mathcal{L}) = \int_{-\infty}^{\infty} (x-1)^{4m} p(x)\, dx$$

we first take $|x| \le m^2$ and approximate the pdf $p(x)$ by $\mathcal{N}(1, \hat{E}_2)$. For $x > m^2$, we use Hoeffding's or Bennett's inequalities [11, eqs. 2.8 or 2.9], which show that for the RV $\varkappa_i$ limited by (46), the pdf $p(x)$ of their sum declines faster than $x^{-(4x \ln m)/m}$, which is smaller than $x^{-4m \ln m}$.

3. Finally, note that the normal RV $\mathcal{N}(1, \hat{E}_2)$ considered above attains its smallest moment $\hat{E}_j$ at $j = 4m$, and that $\hat{E}_{4m}$ has the same exponential order as the Gaussian approximation $Q(\hat{E}_2^{-1/2})$. It is for this reason that we use $\hat{E}_{4m}$ in (48).

Now consider a slightly different algorithm Ψ_r^m, which does not stop at the biorthogonal codes but proceeds further to the repetition codes $\left\{ {g-1 \atop 0} \right\}$. The following—almost identical—theorem shows that Ψ_r^m requires a bigger parameter θ that is the square root of (43). Thus, this algorithm Ψ_r^m is inferior to Φ_r^m.

Theorem 12: Algorithm Ψ_r^m has complexity order of $(3n \log_2 n)/2$ for decoding RM codes $\left\{ {m \atop r} \right\}$ on a memoryless channel $\mathcal{Z}$. Provided that $(m-r)/\ln m \to \infty$ and $\mathsf{E}y \ge (1/md)^{1/2^r}$, algorithm Ψ_r^m gives

- a vanishing decoding error probability if
$$\theta \ge (2m/d)^{1/2^{r+1}};$$
- a nonvanishing decoding error probability if
$$\theta \le (1/d)^{1/2^{r+1}}.$$

C. Applications for BSC and AWGN Channels

First, we apply Theorem 3 for a BSC with transition error probability p. Then the RV $y(x)$ is defined according to (3) as

$$y(x) = \begin{cases} 1-2p, & \text{if } x = +1 \\ 2p-1, & \text{if } x = -1. \end{cases}$$

Then it is easy to verify that $\mathsf{E}y$ and $\theta = \mathsf{E}y \cdot (\mathsf{E}y^2)^{-1/2}$ are equal

$$\theta = \mathsf{E}y = 1 - 2p.$$

In this case, we remove restriction (7) and obtain Corollary 4.

Next, consider an AWGN channel with normal pdf $g(v) = \mathcal{N}(0, \sigma^2)$. If the symbol 1 is transmitted, the channel output x has pdf $\mathcal{N}(1, \sigma^2)$. Then definition (3) gives RV

$$y(x) = \frac{g(x-1) - g(x+1)}{g(x-1) + g(x+1)} = \tanh\left(\frac{2x}{\sigma^2}\right).$$

We will need the following results of [12].

Lemma 13: Consider the normal RV $x = \mathcal{N}(1, \sigma^2)$ and its function

$$y(x) = \tanh(2x/\sigma^2).$$

Then both moments $\mathsf{E}y$ and $\mathsf{E}y^2$ tend to 0 only if $\sigma \to \infty$, and they tend to 1 only if $\sigma \to 0$. Also,

$$\begin{aligned} &\mathsf{E}y \sim \mathsf{E}y^2 \sim \sigma^{-2}, && \text{if } \sigma \to \infty \\ &1 - \mathsf{E}y \sim 1 - \mathsf{E}y^2 \sim \pi Q(\sigma^{-1}), && \text{if } \sigma \to 0. \qquad (50)\end{aligned}$$

Proof of Theorem 5: Recall that we obtain a vanishing error probability if we take $\theta = \hat{\theta}$ in (43). It is also easy to verify that in this case

$$\theta \to 0, \qquad \text{if } \frac{r}{\ln m} \to 0 \qquad (51)$$

$$1 - \theta \to 2^{-r} \ln\left(\frac{d}{4m}\right), \qquad \text{if } \frac{\min(r, m-r)}{\ln m} \to \infty. \qquad (52)$$

First, we study the case $\theta \to 0$ of (51). By definition of $\theta = \mathsf{E}y \cdot (\mathsf{E}y^2)^{-1/2}$, (43) and (50) give

$$\theta^{-1} \sim \sigma \sim (d/4m)^{1/2^r}. \qquad (53)$$

Now we see that restriction (7) can be removed due to the stronger restriction (53).

Let a noise vector $\boldsymbol{z}$ with the normal components $\mathcal{N}(0,\sigma^2)$ be added to the transmitted codeword $\boldsymbol{c}$. Then the vectors $\boldsymbol{c}$ and $\boldsymbol{c}+\boldsymbol{z} \in \mathbb{R}^n$ are separated by the squared Euclidean distance

$$\rho^2 = \sum_i z_i^2.$$

The squared sum ρ^2/σ^2 has χ^2-distribution with n degrees of freedom. It is well known [10] that ρ^2/σ^2 tends to the normal distribution $\mathcal{N}(n,2n)$ as $n \to \infty$. Thus, we have asymptotic equality

$$\Pr\{\rho^2 \le n\sigma^2\} \to Q(0) = 1/2.$$

Now we see that the error patterns of the squared Euclidean weight up to $n\sigma^2$ have probability about $1/2$. However, all but a vanishing fraction of these errors get corrected given σ of (53). Therefore, the Euclidean threshold satisfies the lower bound (14)

$$\delta_\diamond \ge n^{1/2}\sigma \gtrsim n^{1/2}(d/4m)^{1/2^r}.$$

On the other hand, (44) shows that a slightly different $\theta = \tilde{\theta}$ gives a nonvanishing error probability. In this case, condition (53) is now replaced by the asymptotic equality

$$\sigma \sim d^{1/2^r}.$$

This immediately gives the upper bound in (14).

Similarly, consider the second case (52). Since $\theta \to 1$, we have $\sigma \to 0$. Then $\mathsf{E}y$ also satisfies restriction (7). Also, (50) gives the approximation $1-\theta \sim \pi Q(\sigma^{-1})/2$. Then condition (52) guarantees a vanishing block error probability provided that σ satisfies condition

$$Q(1/\sigma) \sim \pi^{-1}2^{-r+1}\ln(d/4m). \qquad (54)$$

Taking the logarithms of both sides, we rewrite the latter as

$$\sigma^2 \sim 1/(r\ln 4).$$

Thus, the squared threshold $\delta_\diamond^2$ satisfies the lower bound

$$\delta_\diamond^2 \gtrsim n/(r\ln 4).$$

To obtain the upper bound, we apply the same arguments to (44). Then equality (54) is only slightly modified to

$$Q(1/\sigma) \sim \pi^{-1}2^{-r+1}\ln d \qquad (55)$$

which gives the same asymptotic threshold $\delta_\diamond^2 \sim n/(r\ln 4)$ in (15) after we take the logarithms of both sides. □

Proof of Corollary 6: Consider any RM code $\left\{ {m \atop r} \right\}$ of code rate

$$R = 2^{-m}\sum_{i=0}^{r}\binom{m}{i}.$$

It is easy to verify that R is bounded away from both 0 and 1 only if $r/m \to 1/2$ as $m \to \infty$. Thus, we use restrictions (52) to consider long codes of nonvanishing rate $R \in (0,1)$.

Next, note that on the AWGN channels, each symbol is inverted with the probability $Q(1/\sigma)$. Then all the error patterns with $nQ(1/\sigma)$ or fewer inversions have combined probability that tends to $1/2$. Most of these error patterns are corrected if $Q(1/\sigma)$ satisfies condition (54). On the other hand, a nonvanishing fraction of them is not corrected given condition (55). Now we see that both conditions give a similar threshold δ_-, which satisfies asymptotic equality

$$\delta_- \sim nQ(1/\sigma) \sim 2d\ln(d/4m)/\pi \sim (2d\ln d)/\pi$$

and increases $4/\pi$ times the hard-decision threshold $\delta_* = (d\ln d)/2$ of (2). □

VII. Possible Improvements and Concluding Remarks

In this paper, we designed a recursive decoding algorithm for RM codes and developed probabilistic tools that give its asymptotic decoding threshold on an arbitrary memoryless channel. Our study yet leaves many open problems. First, note that the thresholds of different paths ξ can only show when the output error rates begin their asymptotic decline. Therefore, we need to tightly estimate the error probabilities $p(\xi)$ instead of these thresholds. This problem is especially important if decoding is performed on good channels since computer simulation becomes prohibitively time consuming. Estimating the *exponential moments* $\mathsf{E}e^{\lambda y(\xi)}$ instead of the power moments $E_{4m}(\xi)$ can give a solution to this problem.

The second direction is to refine the algorithm Φ_r^m and further increase its threshold. In particular, we can consider the algorithm $\hat{\Phi}_r^m$ that performs the original recalculations

$$(y'+y'')/(1+y'y'') \qquad (56)$$

instead of the simplified rule $(y'+y'')/2$. To date, no analytical techniques are known for this modification, due to the nonlinear transformation (56) of the two random variables. However, the following simple statement shows that this algorithm $\hat{\Phi}_r^m$ has limited advantages.

Lemma 14: Algorithm $\hat{\Phi}_r^m$ cannot increase the threshold of the simplified algorithm Φ_r^m.

Proof: Both algorithms use the same transformation $y'y''$ in (18) while proceeding on the weakest (leftmost) subpath 0^{r-1}. Thus, we obtain the same RV $y(0^{r-1})$ and the same error probability in MD decoding (21). On the other hand, this path 0^{r-1} completely defines the decoding threshold of the simplified algorithm Φ_r^m, according to Theorem 10. Therefore, the threshold of Φ_r^m is at least equal to that of $\hat{\Phi}_r^m$. □

It is interesting to note that algorithm $\hat{\Phi}_r^m$ can be further advanced by using the likelihoods $(1+y)/(1-y)$ of the received symbols in the decoding rule (21) instead of their differences y used in Φ_r^m. Simulation results show that this enhanced algorithm $\hat{\Phi}_r^m$ slightly outperforms the simplified version Φ_r^m at the expense of a higher complexity. For example, in Fig. 2, we present simulation results for RM code $\left\{ {7 \atop 3} \right\}$ and also give thecorresponding complexity estimates using the number of the floating-point operations. We note, however, that the algorithm $\hat{\Phi}_r^m$ increases its gain over Φ_r^m for the longer codes that we tested.

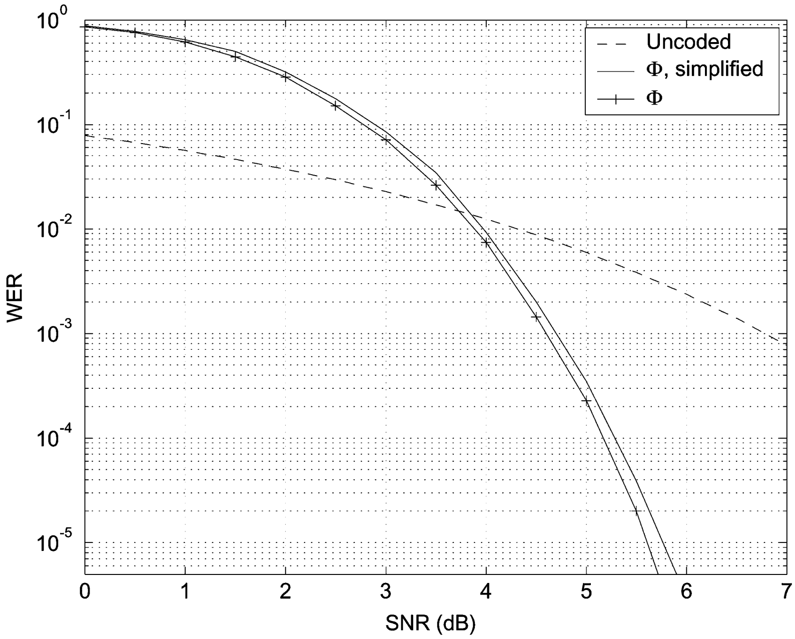

Fig. 2. $(128, 64)$ RM code $\left\{\begin{smallmatrix}7\\3\end{smallmatrix}\right\}$. Word error rate (WER) for the algorithms Φ_r^m and $\hat{\Phi}_r^m$. Number of operations: $|\Phi_r^m| = 1072, |\hat{\Phi}_r^m| = 3888$.

Finally, it is important to extend the preceding analysis to the subcodes of RM codes. In particular, it turns out that the probabilities $p(\xi)$ rapidly decline as the decoding proceeds to the new paths ξ. Thus, code performance can be substantially improved by pruning a few leftmost paths. For example, this can be done by setting the corresponding information bits as zeros. To date, however, the path ordering established in this paper is rather incomplete; therefore, optimal pruning is yet another open problem.

Appendix I
Proof of Theorem 9

The proof consists of two parts and generalizes the proof of Theorem 13 of [14].

1. We first prove inequality (37) for the paths ξ_{01} and ξ_{10} defined in (36). Let us consider four different i.i.d. RVs $\varkappa_1, \varkappa_2, \varkappa_3$, and $\varkappa_4$, which all have expected values 1 and represent four different outputs obtained on the common part $\xi_1, \dots \xi_{l-2}$ of ξ_{01} and ξ_{10}. Then it is readily verified that ξ_{01} and ξ_{10} have the outputs

$$\begin{aligned}\varkappa_{01} &\overset{\text{def}}{=} \varkappa(\xi_{01}) = \varkappa_1\varkappa_2/2 + \varkappa_3\varkappa_4/2\\ \varkappa_{10} &\overset{\text{def}}{=} \varkappa(\xi_{10}) = (\varkappa_1 + \varkappa_2)(\varkappa_3 + \varkappa_4)/4.\end{aligned}$$

We then study their moments $E_j(\xi_{01})$ and $E_j(\xi_{10})$ defined in (36). First, we replace $\varkappa_{01}$ with the new RV

$$\tilde{\varkappa}_{01} = \varkappa_1\varkappa_3/2 + \varkappa_2\varkappa_4/2$$

that has the same pdf and the moments $E_j(\xi_{01})$. Then $\tilde{\varkappa}_{01}$ and $\varkappa_{10}$ are related by the equality

$$\tilde{\varkappa}_{01} = \varkappa_{10} + z \tag{57}$$

where

$$z = (\varkappa_1 - \varkappa_2)(\varkappa_3 - \varkappa_4)/4.$$

Note that the function $\varkappa_1 - \varkappa_2$ has a symmetric distribution for any given sum $\varkappa_1 + \varkappa_2$ (though these two variables are obviously dependent). The same fact holds for $\varkappa_3 - \varkappa_4$ given any $\varkappa_3 + \varkappa_4$. Therefore, RV z satisfies the equality

$$\Pr\{z \mid \varkappa_{10}\} = \Pr\{-z \mid \varkappa_{10}\} \tag{58}$$

for any value of $\varkappa_{10}$. Below we say that z is *symmetric given* $\varkappa_{10}$. Equality (57) also shows that RV $\tilde{\varkappa}_{01}$ and $\varkappa_{10}$ have the same expectation, which is equal to 1.

Since z is symmetric given $\varkappa_{10}$, the conditional moments $\mathsf{E}(z^i \mid \varkappa_{10})$ satisfy conditions

$$\begin{cases}\mathsf{E}(z^i \mid \varkappa_{10}) \equiv 0, & \text{if } i \text{ is odd}\\ \mathsf{E}(z^i \mid \varkappa_{10}) \geq 0, & \text{if } i \text{ is even}\end{cases}$$

where i is a positive integer. Now we study the power expansion

$$(\tilde{\varkappa}_{01} - 1)^j = \{(\varkappa_{10} - 1) + z\}^j \tag{59}$$

and take the expectations of both sides. Since RVs $\tilde{\varkappa}_{01}$ and $\varkappa_{01}$ have the same pdf, the left-hand side gives

$$\mathsf{E}(\tilde{\varkappa}_{01} - 1)^j = \mathsf{E}(\varkappa_{01} - 1)^j = E_j(\xi_{01}).$$

Now consider the expansion of the right-hand side in (59)

$$\{(\varkappa_{10} - 1) + z\}^j = \sum_{i=0}^{j} \binom{i}{j} (\varkappa_{10} - 1)^i z^{j-i}.$$

For an even j, its expectation can be represented as

$$\mathsf{E}\sum_{i=0}^{j}\binom{i}{j}(\varkappa_{10}-1)^i\mathsf{E}(z^{j-i}\,|\,\varkappa_{10}) = E_j(\xi_{10})$$
$$+\mathsf{E}\sum_{i=0,2,\ldots}^{j-2}\binom{i}{j}(\varkappa_{10}-1)^i\mathsf{E}(z^{j-i}\,|\,\varkappa_{10}) \geq E_j(\xi_{10}).$$

Indeed, we obtain the first equality, by removing all the summands, where $j-i$ is odd, since $\mathsf{E}(z^{j-i}\,|\,\varkappa_{10}) \equiv 0$. For any even j, the remaining summands include the terms with even powers i and $j-i$. These terms are nonnegative. Thus, (37) holds for ξ_{01} and ξ_{10}.

2. Next, we prove general property (37) for any two neighbors $\xi_{\text{left}} = (\xi_{01}, \bar{\xi})$ and $\xi_{\text{right}} = (\xi_{10}, \bar{\xi})$. Note that Part 1 of the proof only used the fact that z is a symmetric RV in the representation (57). Obviously, it suffices to prove that this property holds for both immediate suffixes $\bar{\xi} = 0$ and $\bar{\xi} = 1$ provided that it holds on ξ_{01} and ξ_{10}. This directly follows from recursion (24) and equality (57). Indeed, for $\bar{\xi} = 1$, we have the equalities

$$\begin{aligned}\varkappa_{\text{left}} &= \varkappa'_{01} + \varkappa''_{01} = \varkappa'_{10} + \varkappa''_{10} + z' + z''\\ &= \varkappa_{\text{right}} + z' + z''.\end{aligned}$$

Note that $z' + z''$ is a symmetric RV given $\varkappa_{\text{right}} = \varkappa'_{10} + \varkappa''_{10}$. Similarly, for $\bar{\xi} = 0$, we have equalities

$$\varkappa_{\text{left}} = \varkappa'_{01}\varkappa''_{01} = \varkappa_{\text{right}} + z'\varkappa''_{10} + z''\varkappa'_{10} + z'z''.$$

Given two symmetric RVs z' and z'', we now see that the last three summands are again symmetric RVs given the product $\varkappa_{\text{right}} = \varkappa'_{10}\varkappa''_{10}$. Thus, both descendant subpaths also satisfy conditions (57) and (58). □

APPENDIX II
PROOF OF THEOREM 10

Consider any left-end path $\xi = \xi_1, \ldots, \xi_{m-1}$. From Theorem 9 we see that for any even j, the central moment $E_j(\xi)$ does not decrease if any combination of the two consecutive bits 10 in ξ is replaced with 01. These changes can be performed until all zeros precede all ones, which gives the leftmost path $\pounds$. Thus, any left-end path satisfies inequalities

$$\Pr\{\varkappa(\xi) < 0\} \leq E_j(\xi) \leq E_j(\pounds). \tag{60}$$

Next, we need to prove (60) for the right-end paths ξ, which end at the different nodes $\left\{ {h \atop h} \right\}$. Note that all right-end paths are decoded bit-wise. Therefore, any decoding error probability $\Pr\{\varkappa(\xi) < 0\}$ is accounted separately as $p(\xi)$.

To proceed, we first use the all-zero suffix 0^{h-1} and extend any right-end path ξ into $\bar{\xi} = \xi, 0^{h-1}$. This extension makes all paths $\bar{\xi}$ end at the same node $\left\{ {1 \atop 1} \right\}$ as their left-end counterparts. Thus, we can order all extended right-end paths and see that they also satisfy inequalities similar to (60)

$$\Pr\{\varkappa(\bar{\xi}) < 0\} \leq E_j(\bar{\xi}) \leq E_j.$$

Secondly, we prove that $E_j(\xi) \leq E_j(\bar{\xi})$ for any original path ξ and its extension $\bar{\xi}$. Obviously, it suffices to consider a one-bit extension $\bar{\xi} = \xi, 0$. Consider two i.i.d. outputs $\varkappa_1, \varkappa_2$ of the path ξ and the output $\varkappa_1\varkappa_2$ of the path $\xi, 0$. Then

$$\begin{aligned}E_j(\bar{\xi}) &= \mathsf{E}(\varkappa_1\varkappa_2 - 1)^j = \mathsf{E}(\mathsf{E}(\varkappa_1\varkappa_2 - 1)^j\,|\,\varkappa_1)\\ &\geq \mathsf{E}(\varkappa_1\bar{\varkappa}_2 - 1)^j = E_j(\xi)\end{aligned}$$

In the latter inequality, we use the fact that $(\varkappa_1\varkappa_2 - 1)^j$ is a convex function of $\varkappa_2$ for any given $\varkappa_1$ and any even j. Thus, we can use the Jensen inequality and replace $\varkappa_2$ with its mean $\bar{\varkappa}_2 = 1$. Now we see that inequalities (60) hold for all paths ξ. Finally, note that any path ξ ends at the node with $g \leq m-r+1$. Now we combine (60) with our original bound (34) to obtain inequality (38). In turn, by combining (38) and (28), we obtain our main statement (39). □

ACKNOWLEDGMENT

The author wishes to thank R. Blahut and M. Burnashev for helpful suggestions.

REFERENCES

[1] I. S. Reed, "A class of multiple error correcting codes and the decoding scheme," *IEEE Trans. Inf. Theory*, vol. IT-4, no. 4, pp. 38–49, Sep. 1954.

[2] R. E. Krichevskiy, "On the number of Reed-Muller code correctable errors," *Dokl. Sov. Acad. Sci.*, vol. 191, pp. 541–547, 1970.

[3] S. N. Litsyn, "On decoding complexity of low-rate Reed-Muller codes," in *Proc. 9th All-Union Conf. Coding Theory and Information Transmission*, Odessa, Ukraine, U.S.S.R., 1988, pp. 202–204. in Russian.

[4] F. Hemmati, "Closest coset decoding of $u\,|\,u+v\,|$ codes," *IEEE Sel. Areas Commun.*, vol. 7, no. 6, pp. 982–988, Aug. 1989.

[5] G. A. Kabatyanskii, "On decoding Reed-Muller codes in semicontinuous channel," in *Proc. 2nd Int. Workshop "Algebraic and Combinatorial Coding Theory"*, Leningrad, U.S.S.R., 1990, pp. 87–91.

[6] G. Schnabl and M. Bossert, "Soft-decision decoding of Reed-Muller codes as generalized multiple concatenated codes," *IEEE Trans. Inf. Theory*, vol. 41, no. 1, pp. 304–308, Jan. 1995.

[7] V. Sidel'nikov and A. Pershakov, "Decoding of Reed-Muller codes with a large number of errors," *Probl. Inf. Transm.*, vol. 28, no. 3, pp. 80–94, 1992.

[8] A. Ashikhmin and S. Litsyn, "Simple MAP decoding of first-order Reed-Muller and Hamming codes," *IEEE Trans. Inf. Theory*, vol. 50, no. 8, pp. 1812–1818, Aug. 2004.

[9] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*. Amsterdam, The Netherlands: North-Holland, 1981.

[10] W. Feller, *An Introduction to Probability Theory and Its Applications*. New York: Wiley, 1971, vol. 2.

[11] W. Hoeffding, "Probability inequalities for sums of bounded random variables," *J. Amer. Statist. Assoc.*, vol. 58, no. 301, pp. 13–30, Mar. 1963.

[12] I. Dumer and R. Krichevskiy, "Soft decision majority decoding of Reed-Muller Codes," *IEEE Trans. Inf. Theory*, vol. 46, no. 1, pp. 258–264, Jan. 2000.

[13] I. Dumer, "Recursive decoding and its performance for low-rate Reed-Muller codes," *IEEE Trans. Inf. Theory*, vol. 50, no. 5, pp. 811–823, May 2004.

[14] I. Dumer and K. Shabunov, "Recursive error correction for general Reed-Muller codes," *Discr. Appl. Math.*, vol. 154, pp. 253–269, Jan. 2006.