Soft-Decision Decoding of Linear Block Codes Based on Ordered Statistics

Marc P. C. Fossorier and Shu Lin, *Fellow, IEEE*

***Abstract*—This paper presents a novel approach to soft decision decoding for binary linear block codes. The basic idea of this approach is to achieve a desired error performance progressively in a number of stages. For each decoding stage, the error performance is tightly bounded and the decoding is terminated at the stage where either near-optimum error performance or a desired level of error performance is achieved. As a result, more flexibility in the tradeoff between performance and decoding complexity is provided. The proposed decoding is based on the reordering of the received symbols according to their reliability measure. In the paper, the statistics of the noise after ordering are evaluated. Based on these statistics, two monotonic properties which dictate the reprocessing strategy are derived. Each codeword is decoded in two steps: 1) hard-decision decoding based on reliability information and 2) reprocessing of the hard-decision-decoded codeword in successive stages until the desired performance is achieved. The reprocessing is based on the monotonic properties of the ordering and is carried out using a cost function. A new resource test tightly related to the reprocessing strategy is introduced to reduce the number of computations at each reprocessing stage. For short codes of lengths $N \leq 32$ or medium codes with $32 < N \leq 64$ with rate $R \geq 0.6$, near-optimum bit error performance is achieved in two stages of reprocessing with at most a computation complexity of $o(K^2)$ constructed codewords, where K is the dimension of the code. For longer codes, three or more reprocessing stages are required to achieve near-optimum decoding. However, most of the coding gain is obtained within the first two reprocessing stages for error performances of practical interest. The proposed decoding algorithm applies to any binary linear code, does not require any data storage, and is well suitable for parallel processing. Furthermore, the maximum number of computations required at each reprocessing stage is fixed, which prevents buffer overflow at low SNR.**

***Index Terms*—Maximum-likelihood decoding, block codes, ordered statistics, reliability information.**

I. Introduction

THE brute-force approach to Maximum-Likelihood Decoding (MLD) of a linear (N, K) block code requires the computation of 2^K conditional probabilities (or equivalently, Euclidean distances from the received sequence for the Additive White Gaussian Noise (AWGN) channel). This method rapidly becomes too complex to be implemented and more effective methods are therefore needed. If a code possesses a trellis structure, the Viterbi decoding algorithm can be applied to reduce the number of computations. Although all linear block codes have a trellis structure [1], the number of states and the branch complexity become prohibitively large for long codes and the Viterbi decoding becomes impractical. Consequently, other efficient algorithms are needed to achieve optimum or near-optimum decoding.

Maximum-likelihood decoding of block codes has been investigated by many coding theorists; a detailed bibliography of contributions in this area can be found in [2]. Most of the early works trade off the optimal performance for reducing the decoding complexity. In Generalized Minimum Distance (GMD) decoding [3], Forney uses an algebraic decoder to produce a list of codeword candidates. This list is determined from the reliability measures of the symbols within each received block. For each candidate, a test is then performed, with respect to a sufficient condition for optimality. The most likely candidate is chosen as decoded codeword. Following the same idea, Chase provided an algorithm where a fixed number of the error patterns corresponding to certain least reliable bit positions are systematically searched [4]. For this algorithm, the maximum number of codewords considered and the error performance depend on the set of tested positions. Chase's algorithm has then been modified to allow the search only on positions corresponding to reliabilities less than a predetermined threshold [5]. For a given set of positions, the error performance depends also on the choice of the threshold, while the maximum number of computations depends on both the choice of the threshold and the signal-to-noise ratio (SNR). These algorithms suffer a slight degradation in performance when used with codes of small dimension, but the gap in error performance with respect to MLD increases with the dimension of the code. Recently, an optimum MLD algorithm based on the same idea was proposed [6]. No limitation on the search space is imposed at the beginning of the algorithm, but at each iteration, a new powerful sufficient condition for optimality is tested. After each test, the search space for the optimum codeword is reduced, up to convergence to a unique solution. Due to the effectiveness of its associated stopping criterion, this optimum algorithm improves the computational complexities of [4], [5] for short codes. However, the complexity of this new algorithm still increases exponentially with the dimension of the code.

Another proposed technique is to perform syndrome decoding on the received sequence, and then use the syndrome information to modify and improve the original hard-decision decoding. This method was introduced in [7], where one of the proposed schemes orders the information bits according

Manuscript received February 7, 1994; revised December 2, 1994. This work was supported by the National Science Foundation under Grants NCR-91-15400 and NCR-94-15374. The material in this paper was presented in part at the IEEE International Symposium on Information Theory, Trondheim, Norway, June 27–July 1, 1994.

The authors are with the Department of Electrical Engineering, University of Hawaii, Honolulu, HI 96822 USA.

IEEE Log Number 9413638.

to their reliability to guide the search of the most likely codeword. Using the same general algorithm, different search schemes, based on binary trees and graphs are presented in [8]. However, the methods presented in [7], [8] require that $N - K$ be relatively small because the search is carried out over most of the column patterns of the parity check matrix of the code. For very-high-rate codes, an efficient method to reduce the search space of [7] was presented in [9]. For a particular code, a predetermined necessary and sufficient list of error patterns is established, based on both the parity check matrix of the code and only a partial ordering of the reliability measures. However, the technique becomes rapidly impractical whenever $N-K$ exceeds 8. Also, not many general conditions on the survivor error patterns which are valid for any codes can be derived [10].

Other methods take advantage of the decomposable structure of certain codes to reduce the overall complexity of trellis decoding [2], [11]–[13]. Nevertheless, the trellis complexity still grows exponentially with the dimension of any sequence of good codes [14]. To maintain the number of computations manageable, many suboptimal multistage decodings based on the trellis structure have been devised [15], [16].

Recently, an optimum algorithm, based on an artificial intelligence technique (Dijkstra's algorithm) has been presented [17]. This algorithm first re-orders the received symbols according to their confidence values and then performs a tree search similar to sequential decoding. The search is guided by a cost function which not only evaluates the present cost of the extended path in the tree, but also estimates its future contributions, allowing a significant reduction of the search space. This algorithm allows optimal decoding of long block codes efficiently for high SNR's. However, for large code lengths, the worst case performance may require both numerous computations and very large memory for low to medium SNR's. Another drawback of this algorithm is the dependency of the cost function on the weight distribution of the code, which may remain unknown. A suboptimum version of this algorithm has also been devised where the maximum number of codeword candidates is limited by a threshold [18].

Preserving the decoding optimality may result in many unnecessary searches or computations, as seen in the above optimum decoding methods. In this paper, a different approach is proposed. For a given range of bit-error rates (BER), we simply guarantee the error performance associated with MLD. Therefore, there exist two events for which the delivered codeword differs from the optimum MLD codeword. First, the MLD codeword is correct and the decoded codeword is in error, but this event is sufficiently rare so that the optimum error performance is not significantly degraded. Second, both are in error, in which case no further degradation occurs. The algorithm is not optimal, but from a practical point of view, no difference in error performance is observed with respect to MLD, while many unnecessary computations are saved. Whenever optimality is negligibly degraded, we refer the obtained error performance as *practically optimum, for the specified BER*.

The proposed algorithm consists of three steps: 1) reordering of the symbols of each received sequence based on their reliability as in [17]; 2) hard-decision decoding of the reordered received sequence which provides a codeword expected to have as few information bits in error as possible; and 3) reprocessing and improving the hard-decision decoding progressively in stages until the practically optimal error performance or a desired error performance is achieved. These reprocessing stages follow directly from the statistics of the noise after ordering, and the corresponding algorithm is very straightforward. We also develop a sufficient condition to recognize the MLD codeword and stop the algorithm. Since the reprocessing of the algorithm follows the monotonic properties of the error probability associated with the ordered symbols, the effectiveness of the sufficient condition increases at each step. A rapid convergence to the MLD codeword is observed in most of the cases, resulting in a very low computational cost on average. Another feature of the algorithm is that after each reprocessing stage, we can tightly bound the error performance achieved based on the ordered statistics. This allows us to determine the number of reprocessing stages needed to decode a given code up to a particular BER and evaluate precisely the maximum number of computations required for the worst case. This last fact is important since many optimum or suboptimum decoding algorithms, despite performing with low computational cost on average, possess a worst case computation cost which is extremely high and very difficult to evaluate exactly, such as the algorithms of [6], [17]. Furthermore, the proposed algorithm does not require storage of survivors or many candidate codewords for the final decision.

The organization of the paper is as follows. The sequence ordering and the hard-decision decoding are described in Section II. The ordered statistics of the noise associated with the ordering are determined and some properties are derived in Section III. Based on the ordered statistics, the algorithm for reprocessing the hard-decision-decoded codeword and its associated stopping criterion are presented in Section IV. The error performance of the algorithm is then analyzed in Section V. Section VI presents simulations results of some well-known codes of lengths up to 128 and Section VII compares our algorithm with other decoding methods. Finally, concluding remarks and possible future research on the proposed decoding algorithm are given in Section VIII.

II. Hard-Decision Decoding Based on Reliability Information

Suppose an (N, K) binary linear code C with generator matrix G and minimum Hamming distance d_H is used for error control over the AWGN channel. Let $\overline{\boldsymbol{c}} = (c_1, c_2, \cdots, c_N)$ be a codeword in C. For BPSK transmission, the codeword $\overline{\boldsymbol{c}}$ is mapped into the bipolar sequence $\overline{\boldsymbol{x}} = (x_1, x_2, \cdots, x_N)$ with $x_i = (-1)^{c_i} \in \{\pm 1\}$. After transmission, the received sequence at the output of the sampler in the demodulator is $\overline{\boldsymbol{r}} = (r_1, r_2, \cdots, r_N)$ with $r_i = x_i + w_i$, where for $1 \leq i \leq N$, w_i's are statistically independent Gaussian random variables with zero mean and variance $N_0/2$. If a hard decision is performed on each r_i independently, the natural choice for measure of reliability is $| r_i |$ since for bipolar signaling, this

value is proportional to the log-likelihood ratio associated with the symbol hard decision. The decoding begins with reordering the components of the received sequence $\overline{r}$ in decreasing order of reliability value. The resultant sequence is denoted

$$\overline{y} = (y_1, y_2, \cdots, y_N) \tag{1}$$

with $| y_1 | > | y_2 | > \cdots > | y_N |$. Since the noise is AWGN, we assume that $y_i = y_j$, with $i \neq j$ has zero probability of occurrence. This reordering defines a permutation function λ_1 for which $\overline{y} = \lambda_1[\overline{r}]$. We permute the columns of the generator matrix G based on λ_1. This results in the following matrix:

$$G' = \lambda_1[G] = [\overline{g_1}', \overline{g_2}' \cdots \overline{g_N}'] \tag{2}$$

where $\overline{g_i}'$ denotes the ith column of G'. Clearly, the binary code C' generated by G' is equivalent to C, and $C' = \lambda_1[C]$. Since the ith column $\overline{g_i}'$ of G' corresponds to the ith component y_i of $\overline{y}$ with reliability value $| y_i |$, we call $| y_i |$ the associated reliability value of $\overline{g_i}'$. Starting from the first column of G', we find the first K independent columns with the largest associated reliability values. Use these K independent columns as the first K columns of a new $K \times N$ matrix G'' maintaining the decreasing order of their reliability values. The remaining $N - K$ columns of G' form the next $N - K$ columns of G'' arranged in the order of decreasing associated reliability values. The above process defines a second permutation function λ_2, such that

$$G'' = \lambda_2[G'] = \lambda_2[\lambda_1[G]].$$

Rearranging the components of $\overline{y}$ according to the permutation λ_2, we obtain the sequence

$$\overline{z} = (z_1, z_2, \cdots, z_K, z_{K+1}, \cdots, z_N) \tag{3}$$

with

$$| z_1 | > | z_2 | > \cdots > | z_K |$$

and

$$| z_{K+1} | > \cdots > | z_N | .$$

It is clear that

$$\overline{z} = \lambda_2[\overline{y}] = \lambda_2[\lambda_1[\overline{r}]].$$

The first K components of $\overline{z}$ are called the *most reliable independent (MRI) symbols* of $\overline{z}$. Now we perform elementary row operations on G'' to obtain a generator matrix G_1 in systematic form

$$G_1 = [I_K P] = \begin{bmatrix} 1 & 0 \cdots & 0 & p_{1,1} & \cdots p_{1,N-K} \\ 0 & 1 \cdots & 0 & p_{2,1} & \cdots p_{2,N-K} \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 \cdots & 1 & p_{K,1} & \cdots p_{K,N-K} \end{bmatrix} \tag{4}$$

where I_K is the $K \times K$ identity matrix and P is the $K \times (N - K)$ parity check matrix. The code C_1 generated by G_1 is equivalent to C' and C.

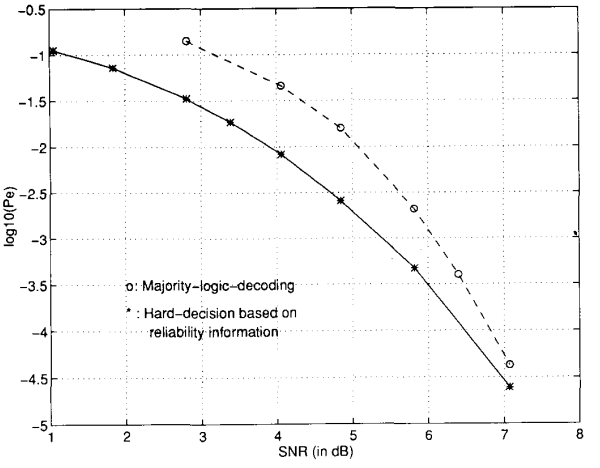

Fig. 1. Error performances for the $(64, 42, 8)$ Reed–Muller code with majority-logic decoding and hard-decision based on reliability information.

Next, we perform hard decisions on the K MRI symbols of $\overline{z}$ such that, for $1 \leq i \leq K$

$$a_i = \begin{cases} 0, & \text{for } z_i > 0 \\ 1, & \text{for } z_i \leq 0. \end{cases} \tag{5}$$

The sequence $(a_1, a_2, \cdots, a_K)$ is used as an information sequence to form a codeword

$$\overline{a} = (a_1, a_2, \cdots, a_K, a_{K+1}, \cdots, a_N)$$

in C_1 based on G_1 where for $1 \leq j \leq N - K$,

$$a_{K+j} = \sum_{1 \leq i \leq K} a_i p_{i,j}.$$

The codeword $\overline{a}$ is taken as the hard-decision-decoded codeword for the received sequence $\overline{z}$. Of course, the estimate $\hat{\overline{c}}_{HD}$ for the transmitted codeword $\overline{c}$ can be obtained by permuting the components of $\overline{a}$ using the inverse permutation $\lambda_1^{-1}\lambda_2^{-1}$, i.e.

$$\hat{\overline{c}}_{HD} = \lambda_1^{-1}\lambda_2^{-1}[\overline{a}]. \tag{6}$$

The above hard-decision decoding based on the reliability information should be close to the optimum decoding. Fig. 1 compares the performance of this decoding process with majority-logic-decoding for the $(64, 42, 8)$ Reed–Muller (RM) code. We see that the hard-decision decoding based on reliability information outperforms majority-logic decoding for low to medium SNR's. For higher SNR, majority-logic decoding provides a slight improvement at error performance of practical interest since its error performance curve possesses a larger slope. Similar results are observed for other RM codes. However, the decoding of $\hat{\overline{c}}_{HD}$ is not aimed to minimize the Euclidean distance between any transmitted codeword of C and the received sequence $\overline{r}$, but instead it minimizes the number of information bits to be in error, which is the fact to be exploited by the algorithm described in this paper.

Using "Mergesort," the ordering of the received sequence is achieved with about $N \log_2 (N)$ comparisons [19, p. 172]

and can be further reduced using a parallel implementation. The entire process to obtain G_1 from G can be realized in $o(NK^2)$ elementary binary additions [20]. However, two levels of parallelism are possible, resulting in K steps of at most K independent summations, each summation consisting of N independent binary additions. Also, for $K > N-K$, if H represents the parity check matrix of G, $H' = \lambda_1[H]$ is clearly the parity check matrix of $G' = \lambda_1[G]$. Then $o(N(N-K)^2)$ elementary binary additions are now required to transform H' into $H_1 = [P^T I_{N-K}]$ and G_1 is easily constructed from H_1. The same permutation λ_2 is determined when dependent columns are present. Note that while the construction of G_1 is carried out from left to right as described in [20], H_1 is formed from right to left. The dominant computational term while constructing G_1 is therefore $o(N \min(K, N-K)^2)$. Finally, computing $\hat{\bar{c}}_{HD}$ requires K sign comparisons and $N-K$ parallel K elementary binary additions. After ordering, the whole process is realized with binary operations only.

III. Ordered Sequence Statistics

Exhaustively testing the 2^K possible changes in the first K positions of $\bar{a}$ and selecting the codeword with smallest Euclidean distance from the received sequence $\bar{z}$ will provide the optimum maximum-likelihood solution. The idea in this paper is to take advantage of the ordering and the fact that $\bar{a}$ contains only a few information bits in error; this reduces the number of possible changes and the remaining discarded changes do not significantly affect the error performance.

A. Definitions

To justify this last fact, we first determine the conditional distribution of the noise W_i in the ordered sequence $\bar{y}$. Note that the ordering is realized with respect to the $|\ y_i\ |$'s and not the w_i's for which we need to evaluate the statistics. From these statistics, we then evaluate the probability that the hard decisions of any group of information bits are jointly in error.

For $i \in \{1, N\}$, define $f_{Wi|Xi}(w_i \mid X_i = s)$ as the density function of the ith noise value in the ordered sequence $\bar{y}$ of length N, conditioned on the fact that $X_i = s$ was transmitted, where $s = \pm 1$. It can be shown that (see Appendix I)

$$f_{Wi|Xi}(w_i \mid X_i = s) = \frac{(\pi N_0)^{-N/2} N!}{(i-1)!(N-i)!} \cdot \left(\int_{-\infty}^{m(w_i)} e^{-x^2/N_0} dx + \int_{M(w_i)}^{\infty} e^{-x^2/N_0} dx \right)^{i-1} \cdot e^{-w_i^2/N_0} \left(\int_{m(w_i)}^{M(w_i)} e^{-x^2/N_0} dx \right)^{N-i} \tag{7}$$

where

$$m(w_i) = \min\{2 + sw_i, -sw_i\}$$

and

$$M(w_i) = \max\{2 + sw_i, -sw_i\}.$$

For equally-likely transmission of bipolar signals normalized to ± 1, the probability that the hard decision of the ith symbol of the sequence $\bar{y}$ of length N is in error is given by

$$\begin{aligned} \mathrm{Pe}(i;N) &= \int_1^{\infty} f_{Wi|Xi}(w_i \mid X_i = -1)\, dw_i \\ &= \frac{(\pi N_0)^{-1/2} N!}{(i-1)!(N-i)!} \int_1^{\infty} \left(\tilde{Q}(n) + 1 - \tilde{Q}(2-n) \right)^{i-1} \\ &\quad \cdot \left(\tilde{Q}(2-n) - \tilde{Q}(n) \right)^{N-i} e^{-n^2/N_0}\, dn \end{aligned} \tag{8}$$

where

$$\tilde{Q}(x) = (\pi N_0)^{-1/2} \int_x^{\infty} e^{-n^2/N_0}\, dn.$$

Similarly, we obtain in Appendix I, for $i < j$, the joint density (see (9) at the bottom of this page), where the indicator function of the set A is $\mathbf{1}_A(x) = 1$ if $x \in A$ and $\mathbf{1}_A(x) = 0$, otherwise. Then, the probability that both hard decisions of the ith and jth symbols of the sequence $\bar{y}$ of length N are in error is

$$\begin{aligned} &\mathrm{Pe}(i,j;N) \\ &= \int_1^{\infty} \int_1^{w_i} f_{Wi,Wj|Xi,Xj}(w_i, w_j \mid X_i = X_j = -1)\, dw_i\, dw_j. \end{aligned} \tag{10}$$

The same approach remains valid for any larger number of considered positions.

B. Monotonic Properties

In the following, we derive monotonicity properties of the probabilities $\mathrm{Pe}(i;N)$ and $\mathrm{Pe}(i,j;N)$, which form the basis of the decoding algorithm.

Theorem 1: For the ordered received sequence $\bar{y}$, the following inequality holds:

$$\mathrm{Pe}(i;N) < \mathrm{Pe}(i+1;N) \tag{11}$$

for $1 \le i < N$.

$$\begin{aligned} f_{Wi,Wj|Xi,Xj}(w_i, w_j \mid X_i = s_i, X_j = s_j) &= \frac{(\pi N_0)^{-N/2} N!}{(i-1)!(j-i-1)!(N-j)!} e^{-(w_i^2 + w_j^2)/N_0} \\ &\quad \cdot \left(\int_{-\infty}^{m(w_i)} e^{-x^2/N_0}\, dx + \int_{M(w_i)}^{\infty} e^{-x^2/N_0}\, dx \right)^{i-1} \left(\int_{m(w_j)}^{M(w_j)} e^{-x^2/N_0}\, dx \right)^{N-j} \\ &\quad \cdot \left(\int_{M(w_j)}^{M(w_i)} e^{-x^2/N_0}\, dx + \int_{m(w_i)}^{m(w_j)} e^{-x^2/N_0}\, dx \right)^{j-i-1} \cdot \mathbf{1}_{[m(w_i), M(w_i)]}(w_j) \end{aligned} \tag{9}$$

Proof: See Appendix II. □

Similarly, we can show that, for $i < j$

$$\text{Pe}(i,j;N) < \text{Pe}(i,j+1;N) < \text{Pe}(i+1,j+1;N) \tag{12}$$

and equivalent results hold for any number of positions considered, as expected.

Corollary 1: For $1 \leq i < N$

$$\text{Pe}(i;N) < \text{Pe}(i;N-1). \tag{13}$$

Proof: Using (8), we easily derive

$$\text{Pe}(i+1;N) - \text{Pe}(i;N) = \frac{N}{i}(\text{Pe}(i;N-1) - \text{Pe}(i;N)) \tag{14}$$

which implies the inequality of (13). □

Rearranging (14), we finally obtain

Corollary 2: For $1 \leq i < N$,

$$\text{Pe}(i;N) < \text{Pe}(i+1;N+1). \tag{15}$$

Theorem 2: For $1 \leq h < i < j < N$

$$\text{Pe}(h,i;N) \leq \text{Pe}(i,j;N) \leq \text{Pe}(i;N). \tag{16}$$

Proof: The proof is immediate using the marginal density definition and the fact that, for $n \geq 1$

$$0 \leq \tilde{Q}(2-n) - \tilde{Q}(n) \leq 1. \quad \square$$

Generalization of Theorem 2 to a larger number of considered positionsin $\overline{\boldsymbol{y}}$ follows the same way. Some important conclusions can be derived from Theorems 1 and 2. Since the first K positions of the re-ordered received sequence $\overline{\boldsymbol{z}}$ satisfy Theorem 1, we are guaranteed that, when considering the positions from K to 1 in $\overline{\boldsymbol{a}}$, the single error probability associated with each position decreases. This result remains valid for any number of positions considered among the K MRI symbols $(z_1, z_2, \cdots, z_K)$. Also, Theorem 2 shows that the more positions we group together, the lower is the probability of these positions to be jointly in error. In the next section, we propose a reprocessing method which exploits these two fundamental facts by testing a stopping criterion which becomes more effective at each decoding step.

C. Other Properties

Next, we present several recurrence relations which can be used to evaluate $\text{Pe}(i;N)$ and other higher order probabilities of error. These recurrence relations reduce the computation complexity of $\text{Pe}(i;N)$ and $\text{Pe}(i,j;N)$ since (8) and (10) are quite complex and hard to evaluate. We omit the proofs and refer to [21], as $\text{Pe}(i;N)$ and $\text{Pe}(i,j;N)$ follow the same recurrence relations as the single moment of ordered statistics.

Theorem 3: The following recurrence relations hold for $\text{Pe}(i;N)$:

$$\text{Pe}(i;N) = \sum_{l=N-i+1}^{N} (-1)^{i+l-N-1} \binom{N}{l} \binom{l-1}{N-i} \text{Pe}(1;l). \tag{17}$$

$$\text{Pe}(i;N) = \sum_{l=i}^{N} (-1)^{l-i} \binom{N}{l} \binom{l-1}{i-1} \text{Pe}(l;l). \tag{18}$$

$$\text{Pe}(i+1;N) = \text{Pe}(i;N) + \binom{N}{i} \sum_{l=0}^{i} (-1)^l \binom{i}{l} \text{Pe}(1;N-i+l). \tag{19}$$

Similar relations hold for higher order probabilities of error, based on the relations for product moments [21]. We finally mention that while these properties are extremely useful for computing $\text{Pe}(i;N)$'s from lower sequence orders, as with any recurrence relation, they also tend to propagate errors.

A legitimate question is whether the statistics of W_i, based on the ordering, could improve the decision device performance, assumed so far to be the usual hard limiter given by (5). We list in this section some properties which show that a straightforward application of the statistics of W_i cannot provide any improvement.

Lemma 1: The likelihood ratio, using the statistics of W_i is the same as the likelihood ratio obtained from the AWGN channel.

Proof: From (7), we observe

$$\frac{f_{Wi|Xi}(y_i - s \mid X_i = s)}{f_{Wi|Xi}(y_i + s \mid X_i = -s)} = \frac{e^{-(y_i-s)^2/N_0}}{e^{-(y_i+s)^2/N_0}} \tag{20}$$

which is simply the likelihood ratio for the AWGN model. □

Lemma 2:

$$\text{Pe}(N;N) \leq \frac{1}{2}. \tag{21}$$

Proof: By integrating (8) by parts, one finds

$$\begin{aligned}\text{Pe}(N;N) &= 1 - (\pi N_0)^{-1/2} N \\ &\quad \cdot \int_1^{\infty} \left(\tilde{Q}(n) + 1 - \tilde{Q}(2-n)\right)^{N-1} e^{-(n-2)^2/N_0} dn \\ &\leq 1 - \text{Pe}(N;N)\end{aligned} \tag{22}$$

which completes the proof. □

Lemma 2 simply shows that even for the last ordered symbol of a sequence of any length, the hard limiter provides the best symbol by symbol decision.

Lemma 3: For any integer $\alpha \in \{1, N\}$

$$\sum_{i_1=1}^{N-(\alpha-1)} \sum_{i_2=i_1+1}^{N-(\alpha-1)+1} \cdots \sum_{i_\alpha=i_{\alpha-1}+1}^{N} \text{Pe}(i_1, i_2, \cdots, i_\alpha; N) = \binom{N}{\alpha} \tilde{Q}(1)^\alpha. \tag{23}$$

Proof: The proof is straightforward using the definition of the joint error probability considered. An equivalent expression with respect to the product moments can also be found [21]. □

Lemma 3 expresses that on average, the probability of α ordered bits to be in error is the same as for independent BPSK signaling.

D. Approximations

For $N \geq 3$, no closed-form expressions for the error probabilities derived in Section III-A have been found. In a separate paper [25], we approximate these quantities based on the central limit theorem, and therefore assume N is sufficiently large. We first show that, for $i < N$

$$\mathrm{Pe}\,(i;N) \cong e^{4(1-m_i)/N_0} \tag{24}$$

where $m_i = \alpha^{-1}(1 - i/N)$, after defining $\alpha(n) = \hat{Q}(2 - n) - \hat{Q}(n)$. Equation (24) is very accurate for $i = N/2$, and becomes less accurate as i approaches the extremities of the ordering. However, the approximation remains very good as long as i does not take values near N.

Following the same approach, we can show that for $n_1 < \cdots < n_j < N$

$$\mathrm{Pe}\,(n_1, n_2, \cdots, n_j; N) \equiv \prod_{l=1}^{j-1} \left(\frac{N}{N-n_l}\right) \mathrm{Pe}\,(n_l;N) \cdot \mathrm{Pe}\,(n_j;N). \tag{25}$$

Equation (25) expresses that although the random variables representing the noise after ordering are not independent, the events associated with any ordered positions in error are almost independent.

IV. Reprocessing

The next decoding step in the proposed decoding algorithm is to reprocess the hard-decision-decoded codeword until either *practically optimum* or a desired error performance is attained. In the following, we first describe the main idea of the reprocessing, then present the reprocessing algorithm, and finally introduce a resource test to reduce the computation complexity.

A. Definitions

Let

$$\overline{\boldsymbol{a}} = (a_1, a_2, \cdots, a_K, a_{K+1}, \cdots, a_N)$$

be the hard-decision-decoded codeword at the first step of the algorithm. For $1 \leq l \leq K$, the *order-l reprocessing* of $\overline{\boldsymbol{a}}$ is defined as follows:

For $1 \leq i \leq l$, make all possible changes of i of the K MRI bits of $\overline{\boldsymbol{a}}$. For each change, reconstruct the corresponding codeword $\overline{\boldsymbol{a}}^{\boldsymbol{\alpha}}$ based on the generator matrix G_1 and determine its corresponding BPSK sequence $\overline{\boldsymbol{x}}^{\boldsymbol{\alpha}}$. Compute the squared Euclidean distance $d^2(\overline{\boldsymbol{z}}, \overline{\boldsymbol{x}}^{\boldsymbol{\alpha}})$ between the ordered received sequence $\overline{\boldsymbol{z}}$ and the signal sequence $\overline{\boldsymbol{x}}^{\boldsymbol{\alpha}}$, and record the codeword $\overline{\boldsymbol{a}}^{\boldsymbol{*}}$ for which $\overline{\boldsymbol{x}}^{\boldsymbol{*}}$ is closest to $\overline{\boldsymbol{z}}$. When all the

$$\sum_{i=0}^{l} \binom{K}{i}$$

possible codewords have been tested, order-l reprocessing of $\overline{\boldsymbol{a}}$ is completed and the recorded codeword $\overline{\boldsymbol{a}}^{\boldsymbol{*}}$ is the final decoded codeword.

For $1 \leq i \leq l$, the process of changing all the possible i of the K MRI bits, reconstructing the corresponding codewords, and computing their squared Euclidean distance from $\overline{\boldsymbol{z}}$ is referred to as *phase-i of the order-l reprocessing*. Clearly, the order-K reprocessing achieves the maximum-likelihood decoding and requires 2^K computations.

Let $I(K) = \{1, 2, \cdots, K\}$ be the index set for the first K positions of $\overline{\boldsymbol{a}}$. For $1 \leq j \leq K$, let $\{n_1, n_2, \cdots, n_j\}$ be a proper subset of j elements of $I(K)$. Then it follows from the ordered statistics developed in Section III that for $n_j \leq K$

$$\mathrm{Pe}\,(n_1, n_2, \cdots, n_j; N) \leq \mathrm{Pe}\,(n_1 + \sigma_1, n_2 + \sigma_2, \cdots, n_j + \sigma_j; N) \tag{26}$$

where $0 \leq \sigma_i < n_{i+1} - n_i + \sigma_{i+1}$ for $1 \leq i \leq j$, with $\sigma_{j+1} = 1$ and $n_{j+1} = K$. For $i < j$, let $\{m_1, m_2, \cdots, m_i\}$ be a proper subset of $\{n_1, n_2, \cdots, n_j\}$. Then

$$\mathrm{Pe}\,(n_1, n_2, \cdots, n_j; N) < \mathrm{Pe}\,(m_1, m_2, \cdots, m_i; N). \tag{27}$$

From (26), it is clear that

$$\mathrm{Pe}\,(n_1, n_2, \cdots, n_j; N) \leq \mathrm{Pe}\,(K - j + 1, \cdots, K - 1, K; N) \tag{28}$$

where equality holds if and only if $n_1 = K - j + 1, \cdots, n_j = K$. Let $\pi(j)$ be the probability that there are j or more errors in the first K positions of $\overline{\boldsymbol{a}}$. Equations (26) and (27) imply that

$$\pi(j) \leq \binom{K}{j} \mathrm{Pe}\,(K - j + 1, \cdots, K - 1, K; N). \tag{29}$$

Clearly, if $\pi(l + 1)$ is sufficiently small, the order-l reprocessing of $\overline{\boldsymbol{a}}$ would give either practically or nearly optimum performance. If P_B represents the block error probability of maximum-likelihood decoding, then

$$\binom{K}{l+1} \mathrm{Pe}\,(K - l, \cdots, K - 1, K; N) << P_B \tag{30}$$

is a sufficient condition for order-l to provide practically optimum error performance at the considered BER, assuming the second permutation λ_2 does not significantly affect the performance. The performance analysis of the complete order-l decoding scheme given in Section V verifies this last fact. It also provides a better evaluation of the performance of order-l reprocessing, since the bound depicted in (30) becomes very loose for long codes. For short codes ($N \leq 32$) and medium codes ($32 < N \leq 64$) with rate $R \geq 0.6$, we find that order-2 reprocessing achieves practically optimum error performance for Pe $\geq 10^{-6}$. For longer codes, at this BER, order-2 reprocessing provides practically optimum or near-optimum performance only for high-rate codes while at least order-3 reprocessing is required to achieve near-optimum performance for medium- to low-rate long codes. In the following, we use the reliability information and the statistics after ordering to

carry out the reprocessing of $\overline{\boldsymbol{a}}$ in a systematic manner that minimizes the number of computations.

For each BPSK signal sequence $\overline{\boldsymbol{x}}^{\boldsymbol{\alpha}} = (\alpha_1, \alpha_2, \cdots, \alpha_N)$ representing a codeword $\overline{\boldsymbol{a}}^{\boldsymbol{\alpha}}$ of C_1, we define, for $i \in [1, N]$

$$\delta_i(\overline{\boldsymbol{a}}^{\boldsymbol{\alpha}}) = \delta_i(\overline{\boldsymbol{x}}^{\boldsymbol{\alpha}}) = 1/4\big((z_i - \alpha_i)^2 - (z_i + \alpha_i)^2\big) = \pm z_i \tag{31}$$

$$D^+(\overline{\boldsymbol{a}}^{\boldsymbol{\alpha}}) = D^+(\overline{\boldsymbol{x}}^{\boldsymbol{\alpha}}) = \{\delta_i(\overline{\boldsymbol{x}}^{\boldsymbol{\alpha}}) : \delta_i(\overline{\boldsymbol{x}}^{\boldsymbol{\alpha}}) > 0, \quad i \in [1, N]\} \tag{32}$$

$$D^-(\overline{\boldsymbol{a}}^{\boldsymbol{\alpha}}) = D^-(\overline{\boldsymbol{x}}^{\boldsymbol{\alpha}}) = \{\delta_i(\overline{\boldsymbol{x}}^{\boldsymbol{\alpha}}) : \delta_i(\overline{\boldsymbol{x}}^{\boldsymbol{\alpha}}) \leq 0, \quad i \in [1, N]\}. \tag{33}$$

Recall that at the first step of decoding, the K MRI symbols of the ordered received sequence $\overline{\boldsymbol{z}}$ are first decoded into $a_1, a_2, \cdots, a_K$ based on (5). Then the remaining $N - K$ bits $a_{K+1}, \cdots, a_N$ are formed from $a_1, a_2, \cdots, a_K$ and the generator matrix G_1 given in (4). Therefore, for $i \in [1, K]$ $\delta_i(\overline{\boldsymbol{a}}) \in D^-(\overline{\boldsymbol{a}})$ since $\delta_i(\overline{\boldsymbol{a}}) = -|z_i|$. Define

$$\begin{aligned}\tilde{\Delta}(\overline{\boldsymbol{a}}^{\boldsymbol{\alpha}}) &= \tilde{\Delta}(\overline{\boldsymbol{x}}^{\boldsymbol{\alpha}}) = 1/4\, d^2(\overline{\boldsymbol{x}}^{\boldsymbol{\alpha}}, \overline{\boldsymbol{z}}) \\ &= 1/4 \sum_{i=1}^{N}(1 + z_i^2) - 1/2 \sum_{i=1}^{N} \alpha_i z_i. \end{aligned} \tag{34}$$

We see that if α_i is changed into $-\alpha_i$ in (34), $\tilde{\Delta}(\overline{\boldsymbol{a}}^{\boldsymbol{\alpha}})$ is changed into $\tilde{\Delta}(\overline{\boldsymbol{a}}^{\boldsymbol{\alpha}}) - \delta_i(\overline{\boldsymbol{a}}^{\boldsymbol{\alpha}})$. We define the cost function Δ (to be minimized over $\overline{\boldsymbol{x}}^{\boldsymbol{\alpha}}$ for maximum-likelihood decoding) associated with each pair $(\overline{\boldsymbol{a}}^{\boldsymbol{\alpha}}, \overline{\boldsymbol{x}}^{\boldsymbol{\alpha}})$ as the inner product

$$\Delta(\overline{\boldsymbol{a}}^{\boldsymbol{\alpha}}) = \Delta(\overline{\boldsymbol{x}}^{\boldsymbol{\alpha}}) = -1/2 \sum_{i=1}^{N} \alpha_i z_i. \tag{35}$$

Equation (35) shows that $\Delta(\overline{\boldsymbol{x}}^{\boldsymbol{\alpha}})$ is now computed only with additions.

B. Reprocessing Algorithm

We start order-1 reprocessing by evaluating the

$$\binom{K}{1}$$

solutions associated by changing the decision of the ith MRI symbol a_i of $\overline{\boldsymbol{a}}$, for i varying from K to 1. Since each of these positions corresponds to a specific row of the parity check matrix P, we obtain a row cost

$$\Delta_i = \delta_i(\overline{\boldsymbol{a}}) + \sum_{j=1}^{N-K} p_{ij}\delta_{K+j}(\overline{\boldsymbol{a}}) \tag{36}$$

where $\delta_i(\overline{\boldsymbol{a}}) < 0$ corresponds to the cost of changing the ith MRI symbol while

$$\sum_{j=1}^{N-K} p_{ij}\delta_{K+j}(\overline{\boldsymbol{a}})$$

represents the cost associated with this change for the parity check bits depending on the ith dimension. We obtain a new codeword $\overline{\boldsymbol{a}}^{\boldsymbol{i}}$ with associated cost $\Delta(\overline{\boldsymbol{a}}^{\boldsymbol{i}}) = \Delta(\overline{\boldsymbol{a}}) - \Delta_i$. If $\Delta_i > 0$, changing the ith MRI symbol reduces the cost associated with the initial decoding $\hat{\boldsymbol{c}}_{\boldsymbol{HD}}$ by the amount Δ_i. At step $K - j$, with $j \in \{1, K\}$, we assume that we have stored the maximum value $\Delta_{\max} = \max_{k>j}\{\Delta_k\}$ (set initially to zero). If $\Delta_j > \Delta_{\max}$, changing the jth MRI symbol improves $\Delta(\overline{\boldsymbol{a}})$ by the biggest amount considered so far in the decoding process. In this case, we set $\Delta_{\max}$ to Δ_j and record the position j of the change. We only record the changed position but do not execute it as it may lead to a local minimum. When the algorithm terminates, the recorded change will be made.

For phase-i of order-l reprocessing, with $i \leq l$, the computations of the

$$\binom{K}{i}$$

changes of the K MRI symbols of $\overline{\boldsymbol{a}}$ are similar, when defining

$$\begin{aligned}\Delta_{j_1,\cdots,j_i} &= \delta_{j_1}(\overline{\boldsymbol{a}}) + \cdots + \delta_{j_i}(\overline{\boldsymbol{a}}) \\ &\quad + \sum_{k=1}^{N-K}(p_{j_1,k} \oplus \cdots \oplus p_{j_i,k})\delta_{K+k}(\overline{\boldsymbol{a}}) \end{aligned} \tag{37}$$

for $j_i > \cdots > j_1$, where $\oplus$ represents the addition in GF(2). When the process terminates, we change the MRI symbols at the position(s) recorded and recompute the parity check symbols. It is then straightforward to permute the symbols back based on the permutation $\lambda_1^{-1}\lambda_2^{-1}$ and declare the resulting codeword as the practically optimum solution. Equation (37) shows that only additions are required by this algorithm.

C. Resource Test for Reducing the Computation Complexity

In general, decoding with order-l reprocessing requires

$$\sum_{j=0}^{l}\binom{K}{j}$$

computations. In the following, we introduce a resource test in the reprocessing to reduce the number of computations.

The maximum-likelihood decoding process can be modeled by different graph representations. In this paper, we consider a complete weighted graph $G(E, V)$ whose vertices are the 2^K BPSK signal sequences representing the code C_1. To each directed edge joining vertex $\overline{\boldsymbol{x}}^{\boldsymbol{\alpha}}$ to vertex $\overline{\boldsymbol{x}}^{\boldsymbol{\beta}}$, we associate the edge cost

$$E(\overline{\boldsymbol{x}}^{\boldsymbol{\alpha}}, \overline{\boldsymbol{x}}^{\boldsymbol{\beta}}) = 1/4\,\big(d^2(\overline{\boldsymbol{x}}^{\boldsymbol{\alpha}}, \overline{\boldsymbol{z}}) - d^2(\overline{\boldsymbol{x}}^{\boldsymbol{\beta}}, \overline{\boldsymbol{z}})\big). \tag{38}$$

Equation (38) implies

$$E(\overline{\boldsymbol{x}}^{\boldsymbol{\alpha}}, \overline{\boldsymbol{x}}^{\boldsymbol{\beta}}) = -E(\overline{\boldsymbol{x}}^{\boldsymbol{\beta}}, \overline{\boldsymbol{x}}^{\boldsymbol{\alpha}}) \tag{39}$$

$$E(\overline{\boldsymbol{x}}^{\boldsymbol{\alpha}}, \overline{\boldsymbol{x}}^{\boldsymbol{\beta}}) + E(\overline{\boldsymbol{x}}^{\boldsymbol{\beta}}, \overline{\boldsymbol{x}}^{\boldsymbol{\gamma}}) = E(\overline{\boldsymbol{x}}^{\boldsymbol{\alpha}}, \overline{\boldsymbol{x}}^{\boldsymbol{\gamma}}). \tag{40}$$

The maximum-likelihood decoding rule can be formulated as follows: Given a starting node $\overline{\boldsymbol{x}}_{\boldsymbol{S}}$ of $G(E, V)$, find the node $\overline{\boldsymbol{x}}_{\boldsymbol{ML}}$ of $G(E, V)$ which maximizes $E(\overline{\boldsymbol{x}}_{\boldsymbol{S}}, \overline{\boldsymbol{x}}^{\boldsymbol{\alpha}})$ over all vertices $\overline{\boldsymbol{x}}^{\boldsymbol{\alpha}}$ of $G(E, V)$.

We choose for starting node $\overline{\boldsymbol{x}}_{\boldsymbol{S}}$ the BPSK signal sequence representing the codeword $\overline{\boldsymbol{a}}$ obtained in Section II and upper bound, for all $\overline{\boldsymbol{x}}^{\boldsymbol{\alpha}}$'s of $G(E, V)$

$$0 \leq E(\overline{\boldsymbol{x}}^{\boldsymbol{\alpha}}, \overline{\boldsymbol{x}}_{\boldsymbol{ML}}) \leq R(\overline{\boldsymbol{x}}^{\boldsymbol{\alpha}}). \tag{41}$$

Using (40), we obtain, for all vertices $\overline{\boldsymbol{x}}^{\boldsymbol{\alpha}}$ of $G(E,V)$

$$E(\overline{\boldsymbol{x}}_{\boldsymbol{S}}, \overline{\boldsymbol{x}}_{\boldsymbol{ML}}) \leq R(\overline{\boldsymbol{x}}_{\boldsymbol{S}}) \tag{42}$$

$$E(\overline{\boldsymbol{x}}_{\boldsymbol{S}}, \overline{\boldsymbol{x}}_{\boldsymbol{ML}}) \leq E(\overline{\boldsymbol{x}}_{\boldsymbol{S}}, \overline{\boldsymbol{x}}^{\boldsymbol{\alpha}}) + R(\overline{\boldsymbol{x}}^{\boldsymbol{\alpha}}). \tag{43}$$

We now use (42) and (43) to reduce the number of computations of the reprocessing algorithm.

For phase-i of order-l reprocessing, $R(\overline{\boldsymbol{x}}^{\boldsymbol{\alpha}})$ introduced above is defined as

$$R_i(\overline{\boldsymbol{x}}^{\boldsymbol{\alpha}}) = \sum_{\delta_j(\overline{\boldsymbol{x}}^{\boldsymbol{\alpha}}) \in D^+(\overline{\boldsymbol{x}}^{\boldsymbol{\alpha}})} \delta_j(\overline{\boldsymbol{x}}^{\boldsymbol{\alpha}}) + \sum_{\delta_j(\overline{\boldsymbol{x}}^{\boldsymbol{\alpha}}) \in D_R^-(\overline{\boldsymbol{x}}^{\boldsymbol{\alpha}})} \delta_j(\overline{\boldsymbol{x}}^{\boldsymbol{\alpha}}) \tag{44}$$

where $D_R^-(\overline{\boldsymbol{x}}^{\boldsymbol{\alpha}})$ represents the $\max\{0, d_H - \mid D^+(\overline{\boldsymbol{x}}^{\boldsymbol{\alpha}}) \mid -i\}$ values of $D^-(\overline{\boldsymbol{x}}^{\boldsymbol{\alpha}})$ with the smallest magnitude (corresponding to parity check positions), and $\mid D^+(\overline{\boldsymbol{x}}^{\boldsymbol{\alpha}}) \mid$ denotes the cardinality of the set $D^+(\overline{\boldsymbol{x}}^{\boldsymbol{\alpha}})$. The first term of (44) represents the best improvement possible when modifying the $\mid D^+(\overline{\boldsymbol{x}}^{\boldsymbol{\alpha}}) \mid$ bits of $\overline{\boldsymbol{x}}^{\boldsymbol{\alpha}}$. Since phase-$i$ of order-l reprocessing modifies at most i bits of $D^-(\overline{\boldsymbol{x}}^{\boldsymbol{\alpha}})$ in the first K positions of $\overline{\boldsymbol{x}}^{\boldsymbol{\alpha}}$ and provides a codeword, at least $\max\{0, d_H - \mid D^+(\overline{\boldsymbol{x}}^{\boldsymbol{\alpha}}) \mid -i\}$ other bits are modified, whose (negative) contribution is bounded by the second summation of (44). The resource function $R_i(\overline{\boldsymbol{x}}_{\boldsymbol{S}})$ evaluated from (44) for $\overline{\boldsymbol{x}}^{\boldsymbol{\alpha}} = \overline{\boldsymbol{x}}_{\boldsymbol{S}}$ represents the best improvement possible (but hardly reachable) on the initial cost $\Delta(\overline{\boldsymbol{a}})$ defined by (35) for the hard-decision-decoded codeword $\overline{\boldsymbol{a}}$. We identify $\Delta_{j_1,\cdots,j_i}$ defined in (37) as

$$\Delta_{j_1,\cdots,j_i} = E(\overline{\boldsymbol{x}}_{\boldsymbol{S}}, \overline{\boldsymbol{x}}_{\boldsymbol{j_1},\cdots,\boldsymbol{j_i}}) \tag{45}$$

where $\overline{\boldsymbol{x}}_{\boldsymbol{j_1},\cdots,\boldsymbol{j_i}}$ is the BPSK signal sequence representing the codeword obtained after inverting the decoded symbols of $\overline{\boldsymbol{x}}_{\boldsymbol{S}}$ in the MRI positions $j_1,\cdots,j_i$. For phase-i of order-l reprocessing, we define for the BPSK signal sequence $\overline{\boldsymbol{x}}_{\boldsymbol{C}}$ such that $\Delta_{\max} = E(\overline{\boldsymbol{x}}_{\boldsymbol{S}}, \overline{\boldsymbol{x}}_{\boldsymbol{C}})$

$$R_{\text{available}}(i) = \min\{R_i(\overline{\boldsymbol{x}}_{\boldsymbol{S}}) - E(\overline{\boldsymbol{x}}_{\boldsymbol{S}}, \overline{\boldsymbol{x}}_{\boldsymbol{C}}), R_i(\overline{\boldsymbol{x}}_{\boldsymbol{C}})\}. \tag{46}$$

We call $R_{\text{available}}(i)$ the *available resource* at phase-i. If $\overline{\boldsymbol{x}}_{\boldsymbol{C}} \neq \overline{\boldsymbol{x}}_{\boldsymbol{ML}}$, (42) and (43) imply that for some $k \geq i$, there exists at least one $\overline{\boldsymbol{x}}_{\boldsymbol{j_1},\cdots,\boldsymbol{j_k}}$ such that

$$E(\overline{\boldsymbol{x}}_{\boldsymbol{S}}, \overline{\boldsymbol{x}}_{\boldsymbol{C}}) < E(\overline{\boldsymbol{x}}_{\boldsymbol{S}}, \overline{\boldsymbol{x}}_{\boldsymbol{j_1},\cdots,\boldsymbol{j_k}}) \leq R_{\text{available}}(i) + E(\overline{\boldsymbol{x}}_{\boldsymbol{S}}, \overline{\boldsymbol{x}}_{\boldsymbol{C}}). \tag{47}$$

Therefore, $-R_{\text{available}}(i) < 0$ represents the smallest value that the contributions from the i MRI bits of $D^-(\overline{\boldsymbol{x}}_{\boldsymbol{S}})$ should provide to further improve $\Delta_{\max} = E(\overline{\boldsymbol{x}}_{\boldsymbol{S}}, \overline{\boldsymbol{x}}_{\boldsymbol{C}})$ for phase-i of order-l reprocessing. The resource test developed in the following exploits this fact to save many unnecessary computations without affecting practically the optimum performance of the algorithm. Note that if both $\mid D^+(\overline{\boldsymbol{x}}_{\boldsymbol{S}}) \mid \geq d_H - i$ and $\mid D^+(\overline{\boldsymbol{x}}_{\boldsymbol{C}}) \mid \geq d_H - i$, which is usually the case when neither $\overline{\boldsymbol{x}}_{\boldsymbol{S}}$ nor $\overline{\boldsymbol{x}}_{\boldsymbol{C}}$ represents the maximum-likelihood codeword, then the two quantities to be minimized in (46) are equal.

We first derive the following simple lemma.

Lemma 4: For $j_1 < j_2 \leq K$

$$\delta_{j_1}(\overline{\boldsymbol{a}}) < \delta_{j_2}(\overline{\boldsymbol{a}}) < 0. \tag{48}$$

Proof: For $j_1 < j_2 \leq K$, (5) and (31) give $\delta_{j_1}(\overline{\boldsymbol{a}}) = - \mid z_{j_1} \mid < 0$ with $\mid z_{j_2} \mid < \mid z_{j_1} \mid$ from the labeling ordering. We also assumed in Section II that $\delta_{j_1}(\overline{\boldsymbol{a}}) = \delta_{j_2}(\overline{\boldsymbol{a}})$ has zero probability of occurrence for AWGN, which completes the proof. □

It is important to notice that the last assumption no longer holds in practical cases where the processed symbols have been previously quantized. The algorithm is then modified in a straightforward fashion by considering the sets of positions corresponding to each quantized value instead of individual positions. In the remaining parts, we ignore this fact and implicitly assume distinct values $\delta_i(\overline{\boldsymbol{a}})$'s.

From Lemma 4, $-\delta_K(\overline{\boldsymbol{x}}_{\boldsymbol{S}}) > 0$ represents the minimum cost necessary to change any MRI symbol(s) of $\overline{\boldsymbol{x}}_{\boldsymbol{S}}$. $R_1(\overline{\boldsymbol{x}}_{\boldsymbol{S}}) \leq -\delta_K(\overline{\boldsymbol{x}}_{\boldsymbol{S}})$ guarantees that $\hat{\overline{\boldsymbol{c}}}_{\boldsymbol{HD}}$ is optimum and no additional computations are required. In the reprocessing algorithm presented above, each time $\Delta_i > \Delta_{\max}$, where Δ_i is computed according to (36), $R_{\text{available}}(1)$ is updated according to (46). As soon as $-\delta_i(\overline{\boldsymbol{x}}_{\boldsymbol{S}}) > R_{\text{available}}(1)$ for some i, we stop computing the remaining Δ_i's. Lemma 4 and (47) guarantee that no further improvement is possible since $-\delta_i(\overline{\boldsymbol{x}}_{\boldsymbol{S}})$ constitutes a decreasing series. Including a test on $R_{\text{available}}(1)$ in the reprocessing algorithm reduces the number of computations per step without requiring a significant amount of additional operations.

In general, we update $R_{\text{available}}(i)$ according to (46) at the beginning of phase-i of order-l decoding since $\Delta_{j_1,\cdots,j_i} < \Delta_{\max}$ does not improve the recorded decoding. In (37), j_i is incremented by "1" and for $k \in [1, i-1]$, j_k is reset to $j_{k+1} - 1$ as soon as $-\delta_{j_1}(\overline{\boldsymbol{a}}) \cdots - \delta_{j_i}(\overline{\boldsymbol{a}}) > R_{\text{available}}(i)$. From the above discussion, we have the following theorem.

Theorem 4: For decoding with order-l reprocessing, if there exists an $i \leq l$ such that

$$-\sum_{j=0}^{i} \delta_{K-j}(\overline{\boldsymbol{a}}) \geq R_{\text{available}}(i+1) \tag{49}$$

then the reprocessing stops after phase-i and modifying $\overline{\boldsymbol{a}}$ at the recorded positions provides the maximum-likelihood decoding solution. □

Theorem 4 improves the extended distance test introduced in [22], due to the reprocessing strategy described in Section IV-B, which exploits the ordering information. In fact, the test of [22] is simply the value $R(\overline{\boldsymbol{x}}^{\boldsymbol{\alpha}})$ depicted in (41) and valid for any node of the graph $G(V,E)$, while the test of Theorem 4 considers the particular state $\overline{\boldsymbol{x}}_{\boldsymbol{S}}$. Consequently, for phase-i reprocessing, i contributions to the resource test from the K MRI positions not only replace contributions from the least reliable positions, but also these contributions monotonically increase. In addition, for phase-i reprocessing, while the available resource defined in (46) always contains the contributions from i MRI bits of $\overline{\boldsymbol{x}}_{\boldsymbol{C}}$, least reliable bits of $D^-(\overline{\boldsymbol{x}}_{\boldsymbol{C}})$ contribute to the test of [22] only when $|D^+(\overline{\boldsymbol{x}}_{\boldsymbol{C}})| < d_H$. Therefore, the efficiency of our new resource test is greatly improved. However, this test is tightly related to the reprocessing algorithm.

D. Computational Analysis

The different computation costs of the algorithm are summarized in Table I. In this table, the binary operations in brackets ([−]) represent sign operations. We ignore the resource updates

TABLE I
COMPUTATIONS REQUIRED FOR DECODING WITH ORDER-l REPROCESSING

Operations	Floating point		Binary
	Computations	Comparisons	
$\lvert y\rvert$			$[N]$
Sorting		$N\log_2(N)$	
G_1			$N\cdot\min(K,N-K)^2$
$\tilde{\mathbf{c}}_{\mathbf{HD}}$			$[K]+K(N-K)$
$\delta_l(\tilde{\mathbf{z}})$			$[N-K]$
$\Delta_{i_1,\cdots,i_l}$	$\leq(N-K-1)\sum_{j=1}^{l}\binom{K}{j}$	$\leq\sum_{j=1}^{l}\binom{K}{j}$	$\leq(N-K)\sum_{j=1}^{l}\binom{K}{j}$
$R_l(\tilde{\mathbf{x}}_{\mathbf{S}})$	$\leq N-K-1$		

depicted in (46) since the computations of the $\Delta_{i_1,\cdots,i_l}$ are largely overestimated, and consider only the dominant order of binary operations when processing the generator matrix G_1.

V. PERFORMANCE ANALYSIS

A. Effects of the Dependent Positions on the Performance

The statistics derived for the ordered sequence $\overline{\boldsymbol{y}}$ in Section III do not directly apply to the reordered sequence $\overline{\boldsymbol{z}}$ since a second permutation λ_2 is required to obtain the generator matrix G_1 with the first K columns to be linearly independent. This permutation changes the initial sorted order of $\overline{\boldsymbol{y}}$. The dependency effect on the performance requires consideration of the generator matrix of each studied code individually. In this paper, however, we look for a general expression which could be applied to any code considered with only a slight dependency on the code parameters. We first notice that the AWGN assumption renders the column ordering independent of each other and of the noise variance value $N_0/2$. The problem can be restated as: What is the probability of choosing $K+i$ columns of the code generator matrix such that the $(K+i)$th pick provides the Kth independent column, for $i\in[0,N-K]$?

We represent the number of dependent columns before the Kth independent one by the random variable X_P and define E_i as the event of picking i dependent columns before the Kth independent one. Then, we have

$$E_0=\bar{E}_1 \tag{50}$$

$$E_{N-K}\subseteq\cdots\subseteq E_1 \tag{51}$$

$$\bar{E}_1\subseteq\cdots\subseteq\bar{E}_{N-K} \tag{52}$$

where $\bar{E}_i$ represents the complement of E_i. It follows from the definitions of X_P and E_i that

$$\begin{aligned}P_i=\mathrm{P}(X_P=i)&=\mathrm{P}\left(E_i\cap\bar{E}_{i+1}\right)\\&=\mathrm{P}\left(E_i\cap\bar{E}_{i+1}\cap\bar{E}_{i+2}\cdots\cap\bar{E}_{N-K}\right).\end{aligned} \tag{53}$$

By using total probability and (52), we rewrite (53) as

$$\begin{aligned}P_i=P(\bar{E}_{N-K})\cdot P(\bar{E}_{N-K-1}\mid\bar{E}_{N-K})\cdots P(\bar{E}_{i+1}\mid\bar{E}_{i+2})\\\cdot(1-P(\bar{E}_i\mid\bar{E}_{i+1})).\end{aligned} \tag{54}$$

Define $N_{\text{ave}}(i)$ as the average number of columns depending on i dimensions for a given code C whose generator matrix is in systematic form. For the set R of all possible row combinations of a given systematic generator matrix G,

$$N_{\text{ave}}(i,G)=\frac{\sum_{R_i\in R}n_1(R_i)}{\binom{K}{i}} \tag{55}$$

where R_i represents any combination of i rows and $n_1(R_i)$ represents the number of columns of R_i containing at least one "1." $N_{\text{ave}}(i)$ is then obtained by averaging $N_{\text{ave}}(i,G)$ over all possible systematic forms of G. Table II compares the values $N_{\text{ave}}(1)$ obtained from simulation with $(N-K)/2+1$ which represents the average number of "1's" per row of a generator matrix in systematic form randomly constructed. For all considered codes, both values closely match and, as expected, the matching is excellent for the extended BCH codes since their weight distribution is almost binomial [20]. Since $N_{\text{ave}}(1)$ represents the average number of ways to add a new dimension for each pick and at most $N-K+1$ column choices are left before the Kth independent column pick, we approximate the average probability to pick one independent column by

$$p=P(\bar{E}_1\mid\bar{E}_2)=\mathrm{P}(A_j^i)=\frac{N_{\text{ave}}(1)}{N-K+1} \tag{56}$$

where A_j^i represents the event that the jth column of at most i dependent columns is independent. With this definition, we obtain

$$P(\bar{E}_i\mid\bar{E}_{i+1})=\mathrm{P}\left(\bigcup_{j=1}^{i}A_j^i\right) \tag{57}$$

and as each event A_j^i is assumed to be equiprobable and independent of j, (57) becomes

$$P(\bar{E}_i\mid\bar{E}_{i+1})=1-\left(1-\frac{N_{\text{ave}}(1)}{N-K+1}\right)^i. \tag{58}$$

Since each dimension is present in at least d_H columns of the generator matrix of any linear block code, we are guaranteed to obtain the last dependent column at the $(N-d_H)$th position in the worst case. Therefore, the maximum number of dependent columns is $N-K-d_H+1$ and $P(\bar{E}_i\mid\bar{E}_{i+1})=P(\bar{E}_i)=1$ for $i>N-K-d_H+1$. Equation (54) can be rewritten as

$$\begin{aligned}P_i=&\left(1-\frac{N_{\text{ave}}(1)}{N-K+1}\right)^i\\&\cdot\left[\prod_{j=i+1}^{N-K-d_H+1}\left(1-\left(1-\frac{N_{\text{ave}}(1)}{N-K+1}\right)^j\right)\right].\end{aligned} \tag{59}$$

For a strict analysis of the random variable X_P, the probability of picking each dependent column depends on both the position it occurs and the number of dependent columns already picked for every systematic generator matrix. This analysis is beyond the scope of this paper. Figs. 2 and 3 depict the distribution approximation obtained from (59) with the real distribution of X_P for, respectively, the $(24,12,8)$ extended Golay code and the $(128,64,22)$ extended BCH code. For both codes, the approximated distribution closely matches the real distribution.

TABLE II
$N_{\text{ave}}(1)$ VERSUS $(N-K)/2+1$ FOR DIFFERENT LINEAR BLOCK CODES
(R = RM, G = Golay, B = BCH, e = extended)

Code	$N_{\text{ave}}(1)$	$(N-K)/2+1$
eG (24,12,8)	8.08	7.00
R (32,16,8)	9.09	9.00
R (32,26,4)	4.46	4.00
R (64,22,16)	20.24	22.00
R (64,42,8)	11.11	12.00
eB (64,36,12)	15.22	15.00
eB (64,45,8)	10.40	10.50
eB (128,64,22)	33.00	33.00
eB (128,99,10)	15.50	15.50
eB (128,120,4)	5.20	5.00

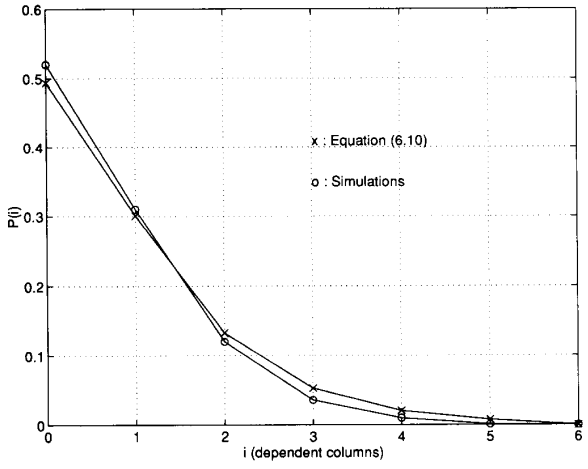

Fig. 2. Distribution of X_P for $(24, 12, 8)$ extended Golay code.

B. Overall Performance Based on Ordering

Define P_{si} as the probability that $\overline{\boldsymbol{a}}$ contains more than i errors in the first K MRI positions.

Phase-0 Error Performance: By referring to the previous notations, we obtain

$$P_{s0}=\sum_{j=0}^{N-K} \mathrm{P}(\text{at least one of the first } K+j \text{ ordered symbols is in error}|X_P=j)P_j \quad (60)$$

and by using the union bound, we find

$$P_{s0} \le \sum_{j=0}^{N-K} P_j \left(\sum_{i=0}^{K+j-1} \mathrm{Pe}\,(K+j-i;N) \right). \quad (61)$$

Equivalently

$$P_{s0} \le \sum_{i=0}^{K-1} \mathrm{Pe}\,(K-i;N) + \sum_{i=1}^{N-K} \left(1-\sum_{j=0}^{i-1} P_j\right) \mathrm{Pe}\,(K+i;N). \quad (62)$$

Assuming that a single error in the MRI positions constitutes the dominant factor for errors in these K positions, $N_{\text{ave}}(1)$ represents the average number of symbol errors in an error block after order-0 reprocessing. Therefore, for an (N, K, d_H) linear block code, we approximate the associated bit-error probability P_{b0} by

$$P_{b0} \approx \frac{N_{\text{ave}}(1)}{N} \cdot \left(\sum_{i=0}^{K-1} \mathrm{Pe}\,(K-i;N) + \sum_{i=1}^{N-K} \left(1-\sum_{j=0}^{i-1} P_j\right) \mathrm{Pe}\,(K+i;N) \right). \quad (63)$$

In (63), all error terms $\mathrm{Pe}\,(i;N)$ corresponding to positions $i \le K$ contribute to P_{b0} while error terms $\mathrm{Pe}\,(i;N)$, with $i > K$ are weighted by their probability of occurrence. Finally, when discarding the dependency factor, the bit-error probability P_{b0} can be approximatively bounded by

$$\frac{N_{\text{ave}}(1)}{N} \mathrm{Pe}\,(K;N) \le P_{b0} \le \frac{N_{\text{ave}}(1)}{N} K\, \mathrm{Pe}\,(K;N). \quad (64)$$

Phase-i Error Performance: More generally, following the same method as previously we get (see (65) at the top of the following page) and

$$\frac{N_{\text{ave}}(1)}{N} \mathrm{Pe}\,(K-i,\cdots,K-1,K;N) \le P_{bi} \le \frac{N_{\text{ave}}(1)}{N} \binom{K}{i} \mathrm{Pe}\,(K-i,\cdots,K-1,K;N). \quad (66)$$

For $i \ge 1$, since phase-i reprocesses the decisions delivered by order-0 decoding, the average number $\tilde{N}_i$ of symbols in error in an error block is no longer $N_{\text{ave}}(1)$ after modifying $\overline{\boldsymbol{a}}$. Despite the fact that minimizing the Euclidean distance between two BPSK signal sequences does not necessarily minimize their corresponding Hamming distance, we expect $\tilde{N}_i \le N_{\text{ave}}(1)$. Due to the difficulty of accurately determining $\tilde{N}_i$, we kept $N_{\text{ave}}(1)$ in (65) and (66).

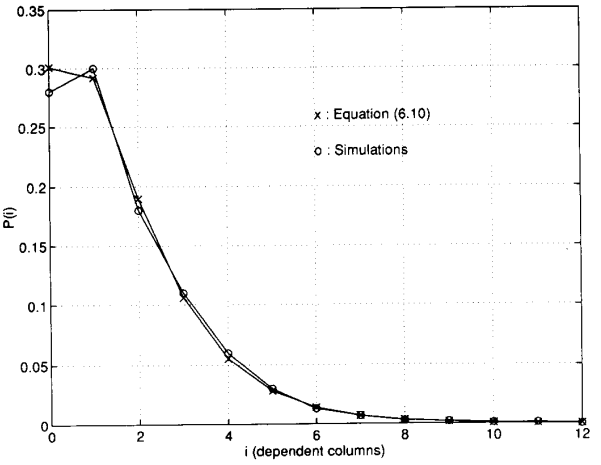

Fig. 3. Distribution of X_P for $(128, 64, 22)$ extended BCH code.

C. Optimum Soft-Decision Performance

The decoding symbol error probability for maximum-likelihood decoding is given by [23, p. 522]

$$\Pr(\epsilon) \approx \left(\frac{d_H}{N}\right) n_d\, \tilde{Q}\left(\sqrt{d_H}\right) \quad (67)$$

$$P_{bi} \approx \frac{N_{\text{ave}}(1)}{N}\left(\sum_{j_1=0}^{K-(i+1)} \cdots \sum_{j_{i+1}=j_i+1}^{K-1} \text{Pe}\,(K-j_{i+1},\cdots,K-j_1;N) + \sum_{j_1=1}^{N-K}\left(1-\sum_{j=0}^{j_1-1} P_j\right)\right.$$
$$\left.\cdot\left(\sum_{j_2=1}^{K+j_1-i} \cdots \sum_{j_{i+1}=j_i+1}^{K+j_1-1} \text{Pe}\,(K+j_1-j_{i+1},\cdots,K+j_1-j_2,K+j_1;N)\right)\right) \quad (65)$$

when the energy per transmitted bit is normalized to unity. In (67), n_d represents the number of minimum-weight codewords of the code. Equation (67) is obtained by considering only the first term of the union bound on the probability of error. It provides a good approximation for small- to medium-dimension codes at medium-to-high SNR. For higher dimensional codes, (67) becomes loose, even at moderate SNR. In this case, the total union bound represents a more accurate performance measure.

D. Algorithm Performance

Define $P_s(i)$ as the probability that a decoded codeword is in error after phase-i of the reprocessing. Then

$$P_s(i) = \text{P}\,(\exists \overline{\boldsymbol{x}}' : d^2(\overline{\boldsymbol{r}},\overline{\boldsymbol{x}}') \le d^2(\overline{\boldsymbol{r}},\overline{\boldsymbol{x}}) \mid \overline{\boldsymbol{x}} \text{ was transmitted, or more than } i \text{ MRI symbols are in error}). \quad (68)$$

Let $P_b(i)$ be the bit-error probability associated with $P_s(i)$. It follows from (67) and the union bound that

$$P_b(i) \le \text{Pr}\,(\epsilon) + P_{bi}. \quad (69)$$

When $P_{bi} << \text{Pr}\,(\epsilon)$, the probability of having at least i of the MRI symbols in error is negligible compared to $\text{Pr}\,(\epsilon)$ and we can stop the decoding process after phase-$(i-1)$ reprocessing. At the corresponding BER, the maximum-likelihood optimum error performance is negligibly altered practically, while at least

$$2^K - \sum_{j=i}^{K}\binom{K}{j}$$

unnecessary computations are saved.

E. Asymptotic Error Performance

In this section, we describe the asymptotic behavior of order-l reprocessing. As N_0 approaches 0, $\text{Pr}\,(\epsilon)$ approaches [23]

$$\text{Pr}\,(\epsilon) \cong \text{e}^{-d_H/N_0}. \quad (70)$$

Based on (24), we obtain, as N_0 approaches 0

$$P_{bl} \cong e^{4\left(1+l-\sum_{j=0}^{l} m_{K-j}\right)/N_0} \quad (71)$$

when assuming that the dependency factor does not influence the asymptotic performance of order-l reprocessing. Also, as N_0 approaches 0

$$m_{K-j} \approx 2 - \tilde{Q}^{-1}(1-(K-j)/N)$$

approaches 2, so that

$$P_{bl} \cong e^{-4(1+l)/N_0}. \quad (72)$$

Combining (70) and (72), the asymptotic performance of the order-l reprocessing is given by

$$P_b(l) \cong \max\left\{e^{-d_H/N_0}, e^{-4(1+l)/N_0}\right\}. \quad (73)$$

Equation (73) implies that for

$$l \ge \min\{\lceil d_H/4 - 1\rceil, K\} \quad (74)$$

order-l reprocessing is asymptotically optimum. Also, whenever (24) dominates (73), an asymptotic coding gain is still achieved by order-l reprocessing. Equation (74) is reached at BER of any practical range, since (72) implies that the error probability for any subset of $l+1$ positions is the same. However, it shows that for many codes of dimension N large enough, our algorithm is not only practically optimum, but also asymptotically optimum.

VI. Simulations Results

A. Error Performance

Figs. 4 to 7 depict the error performances of the $(24, 12, 8)$ extended Golay code, the $(64, 42, 8)$ Reed–Muller (RM) code, the $(128, 99, 10)$ extended BCH code and the $(128, 64, 22)$ extended BCH code, respectively. For each code, the simulated results for various orders of reprocessing are plotted and compared with the theoretical results obtained from (69). Note that for $i \ge 2$, the number of computations involved in (65) becomes extremely large. Therefore, to evaluate (69) for $i \ge 2$, we compute the exact values for the most significant terms which correspond to the least reliable positions considered, and use the approximation of (25) for the remaining secondary ones.

For all codes considered, we observe a close match between the theoretical and simulated results. For the $(24, 12, 8)$ extended Golay code, order-2 reprocessing achieves the practically optimum error performance. In fact, for this code, we also simulated the optimum decoding and found no difference in error performance between the order-2 reprocessing and the optimum decoding. The error performance curves of order-2 reprocessing and optimum ML decoding overlap with each other, as shown on Fig. 4. There is a small performance degradation with order-1 reprocessing. At the BER Pe $= 10^{-6}$,

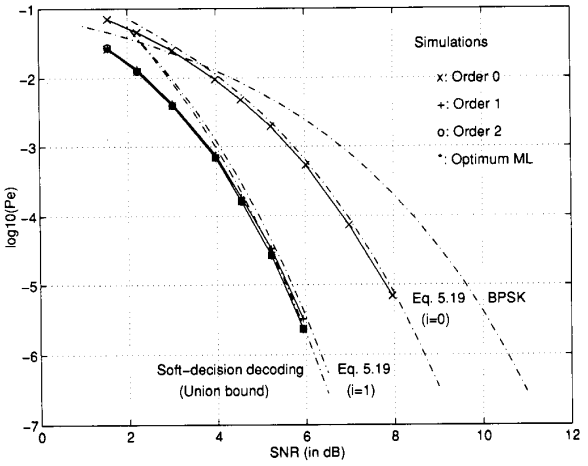

Fig. 4. Error performances for the $(24, 12, 8)$ extended Golay code.

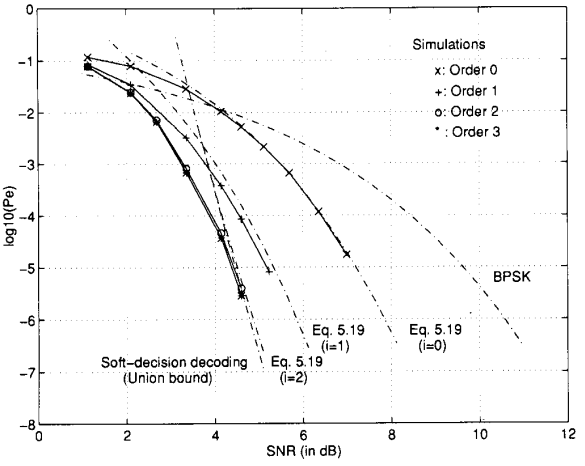

Fig. 6. Error performances for the $(128, 99, 10)$ extended BCH code.

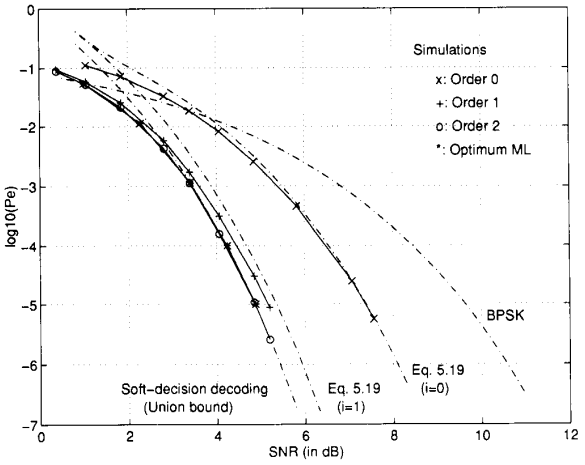

Fig. 5. Error performances for the $(64, 42, 8)$ RM code.

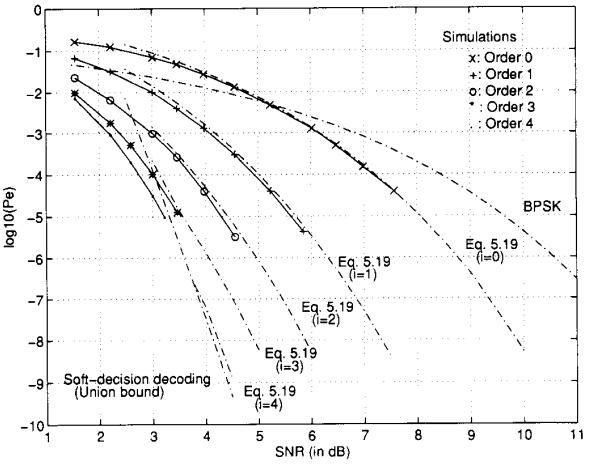

Fig. 7. Error performances for the $(128, 64, 22)$ extended BCH code.

we observe at most a 0.16-dB loss compared to the optimum decoding and at most a loss of 0.3 dB at Pe $= 10^{-11}$, when using the bound of (69) with $i = 1$. For the $(64, 42, 8)$ RM code, order-2 reprocessing achieves practically optimal error performance, and order-1 reprocessing results in at most 0.4-dB degradation in performance compared with the optimum decoding at the BER 10^{-6}. Similar results hold for other short codes of length $N \leq 32$ and rate $R \geq 0.3$, as well as for other medium-length codes of length $32 < N \leq 64$ and rate $R \geq 0.6$ [26]. For all simulated codes in these two classes, at practical BER's, order-2 reprocessing achieves practically optimum error performance, while order-1 reprocessing results in a good tradeoff between error performance and computational complexity. For codes of length $32 < N \leq 64$ and rate $0.3 \leq R < 0.6$, order-3 reprocessing achieves practically optimum error performance, while near-optimum performance is reached by order-2 reprocessing [26].

For longer codes, order-3 or even higher orders of reprocessing might be required to achieve practically optimum or near-optimum error performance. This order depends on both the code length and the rate. For the $(128, 99, 10)$ extended BCH code, decoding with order-3 reprocessing achieves practically optimum error performance, while order-2 reprocessing provides near-optimum performance, within 0.1 dB of the optimum performance for BER Pe $\geq 10^{-6}$. In fact, order-1 reprocessing for this code provides an excellent tradeoff between the error performance and decoding complexity. Despite a degradation in performance of 1.1 dB compared with the optimum decoding, order-1 reprocessing still achieves a 4.8-dB coding gain over the uncoded BPSK with very few computations. For the $(128, 64, 22)$ extended BCH code, order-4 reprocessing is required to achieve practically optimum error performance for BER Pe $\geq 10^{-6}$. At the BER Pe $= 10^{-6}$, the optimum coding gain is 7.0 dB over the uncoded BPSK, while order-2 and order-3 reprocessings achieve 5.6- and 6.5-dB coding gains, respectively. Note finally that for Pe $< 10^{-7}$, even order-4 reprocessing no longer achieves practically optimum error performance for this code.

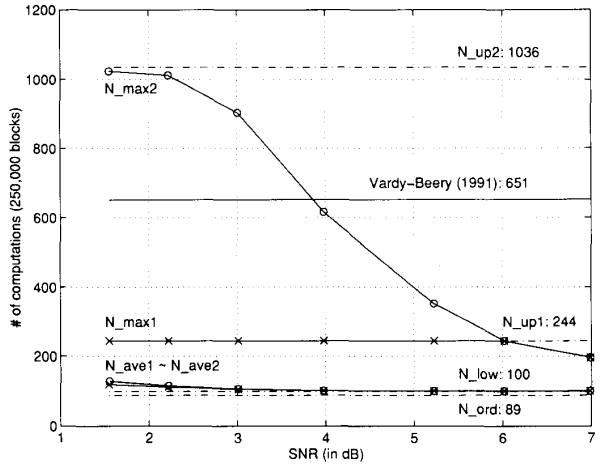

Fig. 8. Number of computations for order-1 (×) and order-2 (o) reprocessings of the (24, 12, 8) extended Golay code.

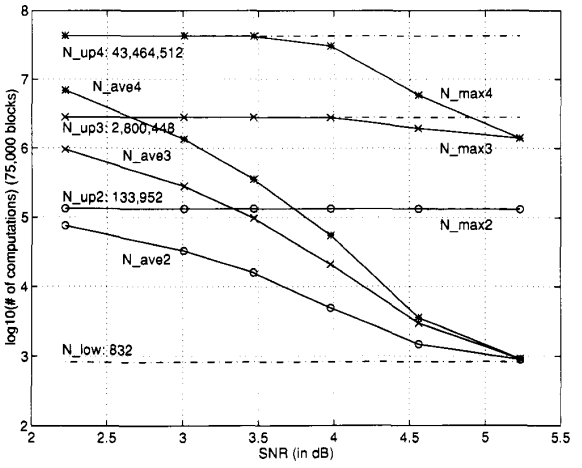

Fig. 9. Number of computations for order-2 (o), order-3 (×), and order-4 (*) reprocessings of the (128, 64, 22) extended BCH code.

B. Number of Computations

$(24, 12, 8)$ *Extended Golay Code:* The best known optimum decoding algorithm for the $(24, 12, 8)$ extended Golay code is provided in [27]. This decoding method requires at most 651 addition-equivalent operations. For this code, order-2 reprocessing achieves practically optimum error performance, as shown in Fig. 4. Evaluating the extreme worst case from Table I while ignoring the binary operations provides 89 comparisons for ordering, at most $N - K - 1 = 11$ operations for order-0, at most $(N - K)\, K = 144$ operations for phase-1, and at most

$$(N - K)\binom{K}{2} = 792$$

operations for phase-2, so a total of 1036 addition-equivalent operations for order-2 reprocessing. We define c_{ave} and $c_{\max}$ as, respectively, the average and the maximum numbers of processed codewords per block. When simulating 250 000 uncoded blocks, Table III depicts the corresponding average number of computations N_{ave} and maximum number of computations $N_{\max}$ for order-2 reprocessing of this code at different BER's. These values are computed from Table I, as described previously for the worst case. They are also compared with order-1 reprocessing in Fig. 8. For Pe $\leq 10^{-3}$, $N_{\max}$ becomes smaller than 651 and decreases further as the SNR increases, while N_{ave} approaches the order-0 decoding complexity 100. It is important to notice that after ordering and order-0 decoding, the additional 2.5-dB asymptotic gain is achieved at the expense of very few computations on average. The average numbers of computations differ slightly between order-1 and order-2 reprocessings, which is not surprising since the corresponding error performances are very close. Also, most computation is due to the ordering, even at low SNR.

$(128, 64, 22)$ *Extended BCH Code:* The simulations for order-2 and order-3 decodings of the $(128, 64, 22)$ extended BCH code are recorded respectively in Tables IV and V. For this code, the ordering requires 769 comparisons and order-0 reprocessing is achieved with at most 63 additions. Then, the number of computations for phase-i reprocessing is evaluated from Table I. These numbers are also compared with order-4 reprocessing in Fig. 9. As expected, the number of computations involved in both order-3 and order-4 reprocessings are enormous. We observe that N_{ave} decreases exponentially as the SNR increases and for order-3 reprocessing, reaches manageable decoding complexities at BER's met in practice. However, only the maximum number of computations of order-2 reprocessing allows a practical implementation of this decoding scheme.

This code was simulated in [17] and for the BER Pe $= 1.57\text{x}10^{-12}$, their optimum decoding algorithm requires the visit of at most 216 052 graph nodes and 42 on average, for 35 000 simulated coded blocks. For 50 000 coded blocks and a similar SNR, order-4 reprocessing suffers about 0.25-dB SNR loss but is required to consider at most 21 812 codewords and 1.17 on average. Note, however, that in both cases, the number of simulated blocks is far too small to obtain reliable

TABLE III
ORDER-2 SIMULATION RESULTS FOR (24, 12, 8) EXTENDED GOLAY CODE
(*: union bound)

SNR (dB)	Pe	c_{ave}	c_{max}	$N_{ave} = 100 + 12c_{ave}$	$N_{max} = 100 + 12c_{max}$
1.55	$10^{-1.56}$	2.39	77	129	1,024
2.22	$10^{-1.90}$	1.33	76	116	1,012
3.01	$10^{-2.40}$	0.55	67	107	904
3.98	$10^{-3.16}$	0.15	43	102	616
5.23	$10^{-4.57}$	0.021	21	101	352
6.02	$10^{-5.72}$	0.005	12	101	244
6.99	$10^{-7.5*}$	0.001	8	101	196

TABLE IV
ORDER-2 SIMULATION RESULTS FOR (128, 64, 22) EXTENDED BCH CODE
(*: union bound)

SNR (dB)	Pe	c_{ave}	c_{max}	$N_{ave} = 832 + 64c_{ave}$	$N_{max} = 832 + 64c_{max}$
2.22	$10^{-2.2}$	1,174	2,080	75,968	133,952
3.01	$10^{-3.0}$	502	2,080	32,960	133,952
3.47	$10^{-3.6}$	236	2,080	15,936	133,952
3.98	$10^{-4.4}$	64.0	2,080	4,928	133,952
4.56	$10^{-5.7}$	9.9	2,060	1,466	132,672
5.23	$10^{-6.6*}$	0.95	2,035	893	131,072

TABLE V
ORDER-3 SIMULATION RESULTS FOR (128, 64, 22) EXTENDED BCH CODE (*: union bound)

SNR (dB)	Pe	c_{ave}	c_{max}	$N_{ave} = 832 + 64c_{ave}$	$N_{max} = 832 + 64c_{max}$
2.22	$10^{-2.8}$	14,819	43,744	949,248	2,800,448
3.01	$10^{-4.0}$	4,415	43,744	283,392	2,800,448
3.47	$10^{-4.9}$	1,505	43,744	97,152	2,800,448
3.98	$10^{-5.8*}$	310	43,237	20,672	2,768,000
4.56	$10^{-7.1*}$	32.9	30,372	2,938	1,944,640
5.23	$10^{-8.9*}$	1.17	21,812	907	1,396,800

information at such BER for which in practice, a concatenated coding scheme would be preferred to a single code. For error performances of practical interest, the order-3 decoding SNR loss is less than 0.4 dB with respect to the optimum one, with $c_{\max} = 43\,744$. We finally point out that our algorithm does not need any storage requirement.

VII. COMPARISON WITH OTHER DECODING ALGORITHMS

A. Comparison with the Chase Algorithm 2

In [4], Chase adds to the HD decoding $\overline{\boldsymbol{d}}$ of the received sequence all possible error patterns $\overline{\boldsymbol{e}}$ obtained by modifying any of the $\lfloor d_H/2 \rfloor$ least reliable bits. These $2^{\lfloor d_H/2 \rfloor}$ vectors are successively decoded by an algebraic decoder, and the best solution is recorded. Therefore, whenever the ML error performance is not achieved, the error performance of this algorithm is dominated by the event that $t+1$ transmission errors are present in the $N - \lfloor d_H/2 \rfloor$ first most reliable positions of the ordered received sequence [4]. In addition to compare Chase Algorithm 2 with our reprocessing method, we propose to use the results of Section III to provide a new analysis of the error performance of this algorithm. Tigher bounds on the error performance than in [4] are derived.

From the results of Section V-D, the bit-error probability $P_b(C2)$ associated with Chase Algorithm 2 is bounded by (see (75) at the bottom of this page) with $j_1 < j_2 < \cdots < j_{t+1}$, W_d representing the weight distribution of the code considered, and n_{w_i} the number of codewords of weight w_i. Equation (75) shows that at high SNR, the error performance of Chase Algorithm 2 is dominated by

$$\max\left\{\tilde{Q}\left(\sqrt{d_H}\right), \text{Pe}\left(N-\lfloor d_H/2\rfloor - t, \cdots, N-\lfloor d_H/2\rfloor; N\right)\right\}. \tag{76}$$

Figs. 10 and 11 compare the simulation results of Chase Algorithm 2 with our reprocessing algorithm for the $(32, 16, 8)$ and $(64, 42, 8)$ RM codes. We also plotted the theoretical bounds of (75). As in Section VI-A, we use (25) to evaluate all terms with secondary contribution to the union bound. We verify, using (69) with $i = 1$ and the upper bound of (75) that Chase Algorithm 2 starts outperforming order-1 reprocessing

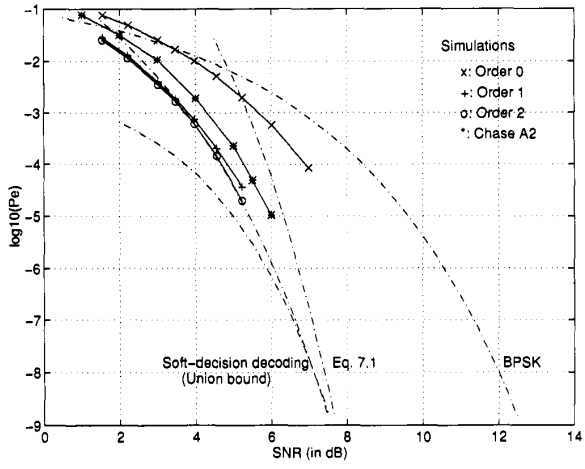

Fig. 10. Error performances for order-i reprocessing and Chase Algorithm 2 decoding of the $(32, 16, 8)$ RM code.

for the BER Pe $= 10^{-7.6}$ for the $(32, 16, 8)$ RM code and Pe $= 10^{-10.1}$ for the $(64, 42, 8)$ RM code. At these crossovers, respectively, 0.35 dB for the $(32, 16, 8)$ RM code and 0.65 dB for the $(64, 42, 8)$ RM code SNR gaps remain between the Chase Algorithm 2 and the order-2 reprocessing error performance curves, obtained from (69) and (75). For both codes no practical difference in error performance is observed between order-2 reprocessing and ML decoding. As expected from Corollary 2 generalized, for a fixed d_H, the performance degradation of Chase Algorithm 2 with respect to the optimum ML error performance at a particular BER increases with N. Therefore, at practical BER's, few orders of reprocessing are sufficient to outperform the Chase Algorithm 2.

B. Comparison with Trellis Decoding

Table VI compares the maximum number of operations needed for order-i reprocessing of some well-known codes with the complexity of Viterbi decoding based on their trellises. The decoding complexity of order-i reprocessing is evaluated from Table I with the assumptions of Section IV-D. The trellis decoding complexities are respectively taken from [11] for the $(24, 12, 8)$ Golay code and the $(32, 16, 8)$ RM code, [29] for the other RM codes and [30] for the extented BCH codes. For order-i reprocessing, we also indicate the loss in coding gain with respect to the practically optimum error performance at Pe $= 10^{-6}$. For the practically optimum error performance, our decoding complexity is less than that of the corresponding trellis decoding. The ratio of the number of computations for trellis decoding to the number

$$\left(\frac{d_H}{N}\right)\max\left\{\tilde{Q}\left(\sqrt{d_H}\right), \text{Pe}\left(N-\lfloor d_H/2\rfloor - t, \cdots, N-\lfloor d_H/2\rfloor: N\right)\right\} \le P_b(C2)$$
$$\le \left(\frac{d_H}{N}\right)\left(\sum_{w_i \in W_d} n_{w_i}\tilde{Q}(\sqrt{w_i}) + \sum_{j_1=1}^{N-\lfloor d_H/2\rfloor - t} \cdots \sum_{j_{t+1}=j_t+1}^{N-\lfloor d_H/2\rfloor} \text{Pe}\left(j_1, j_2, \cdots, j_{t+1}: N\right)\right) \tag{75}$$

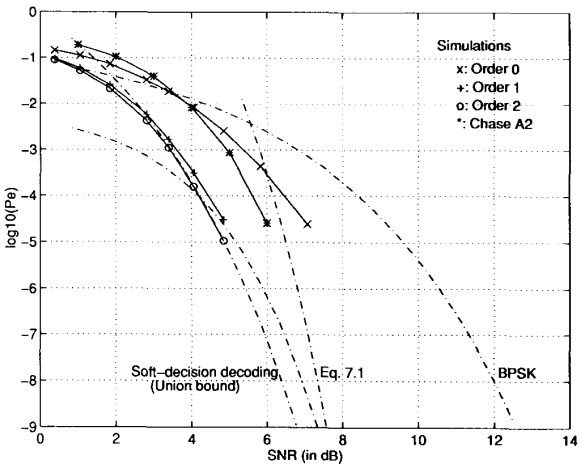

Fig. 11. Error performances for order-i reprocessing and Chase Algorithm 2 decoding of the $(64, 42, 8)$ RM code.

of computations for practically optimum order-i reprocessing at this BER is about 50 for the extended BCH codes, 25 for the (64,42,8) RM code, 4 for the $(32, 26, 8)$ RM code and between 1 and 2 for the other codes. The large advantage of our decoding for the extended BCH codes is due to the fact that there are no structures that are as well decomposable for these codes as there are for RM and Golay codes. Therefore, trellis decoding is not as efficient. For RM codes with $R \geq 0.6$, our decoding approach still largely outperforms the corresponding trellis decoding from a computational point of view since at most order-2 reprocessing is performed. For rate $R < 0.6$ RM codes or the $(24, 12, 8)$ Golay code, only a slight advantage remains in favor of our decoding method. Also, when reprocessing with the resource test described in Section IV-C, the average number of computations becomes much smaller than the maximum one for Pe $= 10^{-6}$. However, for speedup purposes, it is possible to ignore the resource test and implement order-i reprocessing in parallel. Each parallel decoder is assigned a particular subset of positions to process. In such case, the comparisons of Table VI become significant and suggest an advantage of our decoding scheme over trellis decoding at practical BER's. Finally, Table VI also shows that further important reductions of computations are obtained for suboptimum decodings with good tradeoff between error performance and computational complexity.

VIII. Conclusion

In this paper, we have presented a soft-decision decoding scheme for binary linear block codes. The decoding consists of two steps, hard-decision decoding and reprocessing. The hard-decision decoding is based on the ordered reliability values of the received symbols. The purpose of this decoding step is to produce a codeword with relatively few errors in the first K MRI information bit positions. The reprocessing step is designed to improve the hard-decision-decoded codeword progressively until either practically optimum or a desired error performance is achieved. The decoding computation

TABLE VI
Decoding Complexity for Order-i Decoding and Trellis Decoding of Some Well-Known Codes

code	trellis decoding	i	order-i decoding	SNR loss (Pe $= 10^{-6}$)
G(24,12,8)	1,351	1	244	$\leq$ 0.16 dB
		2	1,036	-
RM(32,16,8)	2,399	1	400	$\leq$ 0.3 dB
		2	2,320	-
RM(32,26,4)	1,271	1	290	-
RM(64,22,16)	131,071	2	10,988	$\leq$ 0.25 dB
		3	75,668	-
RM(64,42,8)	544,640	1	1,266	$\leq$ 0.4 dB
		2	20,208	-
eBCH(64,36,12)	13,572,000	2	18,996	$\leq$ 0.25 dB
		3	218,916	-
eBCH(64,45,8)	985,095	1	1,194	$\leq$ 0.47 dB
		2	20,004	-

complexity depends on the order of reprocessing. A cost function together with a resource test have been introduced to reduce the number of computations at each phase of the reprocessing step. The resource test constitutes the most important element of the reprocessing algorithm since its effectiveness increases at each decoding step, ensuring a rapid convergence of the reprocessing algorithm.

For short codes ($N \leq 32$), medium length codes ($32 < N \leq 64$) with rate $R \geq 0.6$ and some long codes of high rates, we have found that order-2 reprocessing achieves practically optimum or near-optimum performance with an $o(K^2)$ constructed codewords, which is much smaller than 2^K. In fact, for medium and high SNR's, the average number of constructed codewords is much smaller than $o(K^2)$. In general, for long codes with large dimensions, higher order reprocessing is required to achieve practically optimum performance. However, decoding with order-2 or order-3 reprocessing provides a good tradeoff between error performance and computation complexity. This is due to the fact that much of the coding gain over the uncoded system is achieved with order-2 or order-3 reprocessing, and higher order reprocessing provides only small improvement.

The main advantage of this decoding is its total abstraction of the algebraic structure of the codes to be decoded. Any binary linear block code can be decoded in the same way. Furthermore, the decoding perfectly fits parallelism since no decision is made before the end of the process. Finally, in [26], we present the equivalent version of the algorithm in the dual space of the code. Therefore, this decoding method provides an efficient processing of the columns of the parity check matrix of the code, and unifies the approaches of processing the G and H matrices.

For long codes with large dimensions, high-order reprocessing may be required to achieve practically optimum performance or a desired level of error performance. High-order reprocessing will result in a large number of computations which may make the proposed decoding scheme impractical for implementation. In this case, certain structural properties of a code may be used to reduce the computation complexity, e.g., decomposable properties [16], [31]. If a code can be decomposed as a multilevel code, e.g., RM codes, the proposed reprocessing scheme may be incorporated with multistage decoding to reduce the computational complexity.

Since each component code of a decomposable code has a smaller dimension than that of the code itself, the number of computations required to reprocess the hard-decision decoding for each component code will be much reduced. This will result in a significant reduction in the overall computation complexity. Of course, the resultant decoding is suboptimum. Investigation in this direction may be worthwhile.

APPENDIX I
CONDITIONAL DENSITIES OF THE NOISE AFTER ORDERING

The analysis in this appendix is restricted to the AWGN model. Generalization to other distributions is straightforward by substituting the desired density function in the following development.

A. Marginal Conditional Density of W_i

In this Appendix, we determine the density function of the ith noise value W_i after ordering, conditioned on the knowledge of the corresponding transmitted symbol $X_i = x_i$. A capital letter represents a random variable while a lower case letter stands for a deterministic value. As described in Section II, we order the received symbols with the new labeling $i < j$ for $| x_i + w_i | > | x_j + w_j |$, where x_i represents the transmitted symbol and w_i the noise value associated with this symbol.

We first assume $X_i = -1$. For $j > 0$

$$| W_{i+j} + x_{i+j} | \leq | w_i - 1 | \tag{77}$$

implies that for $x_{i+j} = -1$

$$\begin{aligned} W_{i+j} &\in [2 - w_i, w_i], \quad \text{if } w_i \geq 1, \\ W_{i+j} &\in [w_i, 2 - w_i], \quad \text{if } w_i \leq 1 \end{aligned} \tag{78}$$

and for $x_{i+j} = 1$

$$\begin{aligned} W_{i+j} &\in [-w_i, w_i - 2], \quad \text{if } w_i \geq 1, \\ W_{i+j} &\in [w_i - 2, -w_i], \quad \text{if } w_i \leq 1. \end{aligned} \tag{79}$$

Since the bipolar transmitted digits are independent, equiprobable, and independent of the zero-mean white Gaussian noise, we observe that for $w_i \geq 1$

$$\begin{aligned} &\mathrm{P}(W_{i+j} \in [2 - w_i, w_i] \mid X_{i+j} = -1) \\ &\quad = \mathrm{P}(W_{i+j} \in [-w_i, w_i - 2] \mid X_{i+j} = 1) \\ &\quad = \mathrm{P}(W \in [2 - w_i, w_i]) \end{aligned} \tag{80}$$

where W represents the white noise before ordering. The same equation holds for $w_i \leq 1$ after permuting the limits of the interval considered.

The case $X_i = 1$ is identical when comparing w_i with respect to -1. Equation (80) becomes, for $w_i \leq -1$

$$\begin{aligned} &\mathrm{P}(W_{i+j} \in [2 + w_i, -w_i] \mid X_{i+j} = -1) \\ &\quad = \mathrm{P}(W_{i+j} \in [w_i, -w_i - 2] \mid X_{i+j} = 1) \\ &\quad = \mathrm{P}(W \in [2 + w_i, -w_i]). \end{aligned} \tag{81}$$

By comparing (80) and (81), we obtain for $j > 0$

$$\begin{aligned} &\mathrm{P}(| W_{i+j} + x_{i+j} | \leq | w_i + s_i | \mid X_i = s_i) \\ &\quad = \mathrm{P}(W \in [m(w_i), M(w_i)] \mid X_i = s_i) \end{aligned} \tag{82}$$

where

$$\begin{aligned} m(w_i) &= \min(2 + s_i w_i, -s_i w_i) \\ M(w_i) &= \max(2 + s_i w_i, -s_i w_i). \end{aligned} \tag{83}$$

For $j > 0$, the same development leads to the complementary conclusion

$$\begin{aligned} &\mathrm{P}(| W_{i-j} + x_{i-j} | \geq | w_i + s_i | \mid X_i = s_i) \\ &\quad = \mathrm{P}(W \in (-\infty, m(w_i)] \cup [M(w_i), \infty) \mid X_i = s_i) \end{aligned} \tag{84}$$

and we note that

$$f_W(w_i) dw_i = \mathrm{P}(w_i \leq W \leq w_i + dw_i). \tag{85}$$

Following the proof of [24, p. 185], the three events

$$\begin{aligned} E_1 &= \{| W + x | > | w_i + s_i |\} \\ E_2 &= \{w_i \leq W \leq w_i + dw_i\} \end{aligned}$$

and

$$E_3 = \{| W + x | < | w_i + dw_i + s_i |\}$$

are disjoint and from (83)–(85), have respective associated probabilities

$$\begin{aligned} \mathrm{P}(E_1) &= (\pi N_0)^{-1/2} \\ &\quad \cdot \left(\int_{-\infty}^{m(w_i)} e^{-x^2/N_0} dx + \int_{M(w_i)}^{\infty} \mathrm{e}^{-x^2/N_0} dx \right) \end{aligned} \tag{86}$$

$$\mathrm{P}(E_2) = (\pi N_0)^{-1/2} e^{-w_i^2/N_0} dw_i \tag{87}$$

$$\mathrm{P}(E_3) = (\pi N_0)^{-1/2} \int_{m(w_i)}^{M(w_i)} e^{-x^2/N_0} dx. \tag{88}$$

When E_1 occurs $i - 1$ times, E_2 once and E_3 $N - i$ times, we obtain

$$\begin{aligned} f_{Wi|Xi}(w_i \mid X_i = s_i) &= \frac{(\pi N_0)^{-N/2} N!}{(i-1)!(N-i)!} \\ &\quad \cdot \left(\int_{-\infty}^{m(w_i)} e^{-x^2/N_0} \, dx + \int_{M(w_i)}^{\infty} e^{-x^2/N_0} \, dx \right)^{i-1} \\ &\quad \cdot e^{-w_i^2/N_0} \left(\int_{m(w_i)}^{M(w_i)} e^{-x^2/N_0} \, dx \right)^{N-i}. \end{aligned} \tag{89}$$

The marginal density of the noise after ordering is easily obtained as, for equiprobable signaling

$$\begin{aligned} f_{Wi}(w_i) &= 1/2 \big(f_{Wi|Xi}(w_i \mid X_i = -1) \\ &\quad + f_{Wi|Xi}(w_i \mid X_i = 1) \big) \\ &= f_{Wi}(-w_i). \end{aligned} \tag{90}$$

B. Joint Conditional Density of W_i and W_j

The joint conditional density is obtained in a similar way. We first assume $X_i = X_j = -1$ and $i < j$. For $k > 0$

$$| W_{j+k} + x_{j+k} | \leq | w_j - 1 | \leq | w_i - 1 | \tag{91}$$

is equivalent to (77) for $| W_{j+k} + x_{j+k} | \leq | w_j - 1 |$. Similarly

$$| w_j - 1 | \leq | w_i - 1 | \leq | W_{i-k} + x_{i-k} | \tag{92}$$

is equivalent to (84) for $| w_i - 1 | \leq | W_{i-k} + x_{i-k} |$. For $k \in (i, j)$, we obtain

$$| w_j - 1 | \leq | W_k + x_k | \leq | w_i - 1 | \tag{93}$$

which implies that for $w_j > 1$ and $w_i > 1$

$$\begin{aligned} W_k &\in [w_j, w_i] \cup [2 - w_i, 2 - w_j], \quad \text{if } x_k = -1 \\ W_k &\in [-w_i, -w_j] \cup [w_i - 2, w_j - 2], \quad \text{if } x_k = 1. \end{aligned} \tag{94}$$

All symmetrical cases for w_i and w_j, as well as the three other possible assumptions for $X_i = \pm 1$ and $X_j = \pm 1$ follow in a straightforward way, as previously. We finally obtain for $i < j$

$$\begin{aligned} &f_{Wi,Wj|Xi,Xj}(w_i, w_j \mid X_i = s_i, X_j = s_j) \\ &= \frac{(\pi N_0)^{-N/2} N!}{(i-1)!(j-i-1)!(N-j)!} e^{-(w_i^2 + w_j^2)/N_0} \\ &\cdot \left(\int_{-\infty}^{m(w_i)} e^{-x^2/N_0}\, dx + \int_{M(w_i)}^{\infty} e^{-x^2/N_0}\, dx \right)^{i-1} \\ &\cdot \left(\int_{m(w_j)}^{M(w_j)} e^{-x^2/N_0}\, dx \right)^{N-j} \cdot \\ &\cdot \left(\int_{M(w_j)}^{M(w_i)} e^{-x^2/N_0}\, dx + \int_{m(w_i)}^{m(w_j)} e^{-x^2/N_0}\, dx \right)^{j-i-1} \\ &\cdot \mathbf{1}_{[m(w_i), M(w_i)]}(w_j) \end{aligned} \tag{95}$$

where $m(w)$ and $M(w)$ are defined in (83) for s_i and s_j, and $\mathbf{1}_A(x) = 1$ if $x \in A$ and $\mathbf{1}_A(x) = 0$, otherwise.

The joint density of W_i and W_j is easily obtained when considering the four possible conditional joint densities. The same method remains available when considering more than two ordered noise values. It becomes straightforward to generalize (95) to any number of noise values [26].

Appendix II
Monotonicity of $\mathrm{Pe}(i; N)$

Define $u = f(n) = \tilde{Q}(2 - n) - \tilde{Q}(n)$ which implies

$$du = (\pi N_0)^{-1/2} e^{-n^2/N_0} (1 + e^{4(n-1)/N_0}) dn.$$

From (8) and the fact that $f(n)$ is one to one, we obtain

$$\begin{aligned} \mathrm{Pe}(i+1; N) - \mathrm{Pe}(i; N) = \binom{N}{i} \int_0^1 (1-u)^{i-1} u^{N-i-1} \\ \cdot \frac{(N-i)(1-u) - i\,u}{1 + \mathrm{e}^{4(f^{-1}(u)-1)/N_0}}\, du. \end{aligned} \tag{96}$$

Integrating by part and defining

$$\alpha(u) = (1 + \mathrm{e}^{4(f^{-1}(u)-1)/N_0})^{-1}$$

$$\begin{aligned} \mathrm{Pe}(i+1; N) - \mathrm{Pe}(i; N) = -\binom{N}{i} \\ \cdot \int_0^1 (1-u)^i u^{N-i} \frac{\partial \alpha(u)}{\partial u}\, du \end{aligned} \tag{97}$$

with

$$\frac{\partial \alpha(u)}{\partial u} = \frac{-4/N_0 \dfrac{\partial f^{-1}(u)}{\partial u} e^{4(f^{-1}(u)-1)/N_0}}{\left(1 + e^{4(f^{-1}(u)-1)/N_0}\right)^2} \tag{98}$$

$$\begin{aligned} \frac{\partial f^{-1}(u)}{\partial u} &= \left. \frac{1}{\dfrac{\partial f(n)}{\partial n}} \right|_{n = f^{-1}(u)} \\ &= \left. \frac{(\pi N_0)^{1/2}}{e^{-n^2/N_0} \left(1 + e^{4(n-1)/N_0}\right)} \right|_{n = f^{-1}(u)}. \end{aligned} \tag{99}$$

Equations (98) and (99) imply

$$\frac{\partial \alpha(u)}{\partial u} \leq 0$$

so that, from (97)

$$\mathrm{Pe}(i+1; N) - \mathrm{Pe}(i; N) > 0 \tag{100}$$

which completes the proof.

Acknowledgment

The authors wish to thank the anonymous reviewers for their useful comments that improved the results presented in the paper. The first author also wishes to thank Dr. N. T. Gaarder for many inspiring discussions.

References

[1] J. K. Wolf, "Efficient maximum likelihood decoding of linear block codes using a trellis," *IEEE Trans. Inform. Theory*, vol. IT-24, pp. 76–80, Jan. 1978.

[2] 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, Jan. 1986.

[3] G. D. Forney Jr, "Generalized minimum distance decoding," *IEEE Trans. Inform. Theory*, vol. IT-12, pp. 125–131, Apr. 1966.

[4] D. Chase, "A class of algorithms for decoding block codes with channel measurement information," *IEEE Trans. Inform. Theory*, vol. IT-18, pp. 170–182, Jan. 1972.

[5] H. Tanaka and K. Kakigahara, "Simplified correlation decoding by selecting codewords using erasure information," *IEEE Trans. Inform. Theory*, vol. IT-29, pp. 743–748, Sept. 1983.

[6] T. Taneko, T. Nishijima, H. Inazumi, and S. Hirasawa, "An efficient maximum likelihood decoding of linear block codes with algebraic decoder," *IEEE Trans. Inform. Theory*, vol. 40, pp. 320–327, Mar. 1994.

[7] 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, pp. 963–975, Sept. 1989.

[8] N. J. C. Lous, P. A. H. Bours, and H. C. A. van Tilborg, "On maximum likelihood soft-decision decoding of binary linear codes," *IEEE Trans. Inform. Theory*, vol. 39, pp. 197–203, Jan. 1993.
[9] J. Snyders, "Reduced lists of error patterns for maximum likelihood soft decoding," *IEEE Trans. Inform. Theory*, vol. 37, pp. 1194–1200, July 1991.
[10] ——, "On survivor error patterns for maximum likelihood soft decoding," in *Proc. IEEE Int. Symp. on Information Theory* (Budapest, Hungary, June 1991), p. 192.
[11] G. D. Forney, Jr, "Coset codes II: Binary lattices and related codes," *IEEE Trans. on Inform. Theory*, vol. 34, pp. 1152–1187, Sept. 1988.
[12] D. Muder, "Minimal trellises for block codes," *IEEE Trans. Inform. Theory*, vol. 34, pp. 1049–1053, Sept. 1988.
[13] T. Kasami, T. Takata, T. Fujiwara, and S. Lin, "On complexity of trellis structure of linear block codes," *IEEE Trans. Inform. Theory*, vol. 39, pp. 1057–1064, May 1993.
[14] A. Lafourcade and A. Vardy, "Asymptotically good codes have infinite trellis complexity," *IEEE Trans. Inform. Theory*, vol. 41, pp. 555–559, Mar. 1995.
[15] F. Hemmati, "Closest coset decoding of $|u|u+v|$ codes," *IEEE J. Selected Areas Commun.*, vol. 7, pp. 982–988, Aug. 1989.
[16] J. Wu, S. Lin, T. Kasami, T. Takata, and T. Fujiwara, "An upper bound on the effective error coefficient of two-stage decoding and good two-level decompositions of some Reed-Muller codes," *IEEE Trans. Commun.*, vol. 42, pp. 813–818, Feb./Mar./Apr. 1994.
[17] Y. S. Han, C. R. P. Hartmann, and C. C. Chen, "Efficient priority-first search maximum-likelihood soft-decision decoding of linear block codes," *IEEE Trans. Inform. Theory*, vol. 39, pp. 1514–1523, Sept. 1993.
[18] Y. S. Han, C. R. P. Hartmann, and K. G. Mehrotra, "Further results on decoding linear block codes using a generalized Dijkstra's algorithm," in *Proc. IEEE Int. Symp. on Information Theory* (Trondheim, Norway, June 1994), p. 342.
[19] R. Sedgewick, *Algorithms in C*. New York: Addison-Wesley, 1990.
[20] W. W. Peterson and E. J. Weldon, Jr., *Error-Correcting Codes*, 2nd ed. Cambridge, MA: M.I.T. Press, 1972.
[21] N. Balakrishnan and A. Clifford Cohen, *Ordered Statistics and Inference: Estimation Methods*. San Diego, CA: Academic Press, 1991.
[22] D. J. Taipale and M. B. Pursley, "An improvement to generalized-minimum-distance decoding," *IEEE Trans. Inform. Theory*, vol. 37, pp. 167–172, Jan. 1991.
[23] G. D. Forney, Jr., *Concatenated Codes*. Cambridge, MA: M.I.T. Press, 1966.
[24] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, 3rd ed. New York: McGraw-Hill, 1991.
[25] M. P. C. Fossorier and S. Lin, "First order approximation of the ordered binary symmetric channel," submitted to *IEEE Trans. Inform. Theory*, Nov. 1994.
[26] M. P. C. Fossorier, "Soft decision decoding of linear block codes based on ordered statistics," Ph.D. dissertation, University of Hawaii at Manoa, Dec. 1994.
[27] A. Vardy and Y. Be'ery, "More efficient soft decoding of the Golay codes," *IEEE Trans. Inform. Theory*, vol. 37, pp. 667–672, May 1991.
[28] S. Lin and D. J. Costello, Jr., *Error Control Coding Fundamentals and Applications*. Englewood Cliffs, NJ: Prentice-Hall, 1983.
[29] H. Thirumoorthy, S. Lin, and T. Kasami, "Soft decision decoding of binary linear block codes based on an iterative search algorithm," submitted to *IEEE Trans. Inform. Theory*, Jan. 1995.
[30] A. Vardy and Y. Be'ery, "Maximum likelihood soft decoding decoding of BCH codes," *IEEE Trans. Inform. Theory*, vol. 40, pp. 546–554, Mar. 1994.
[31] T. Kasami, T. Kakata, T. Fujiwara, and S. Lin, "On complexity of trellis structure of linear block codes," *IEEE Trans. Inform. Theory*, vol. 39, pp. 1057–1064, May 1993.