Abstract—Order-$w$ reprocessing is a suboptimal soft-decision decoding approach for binary linear block codes in which up to $w$ bits are systematically flipped on the so-called most reliable (information) basis (MRB). At each iteration a candidate codeword is recoded and in the end, the most likely codeword is picked from the candidate list. In this paper, we first incorporate two preprocessing rules into order-$w$ reprocessing: i) a test error pattern with Hamming weight up to $w+1$ is discarded without further computing its actual likelihood if it results in more than $\theta$ bit errors within the MRB and the $t$ most reliable bits of the redundancy part; ii) a test error pattern with Hamming weight $w+2$ is discarded if it results in more than 1 bit errors among the $\tau$ most reliable bits of the redundancy part, where $\theta$, $t$, and $\tau$ are parameters to be determined. We show also that, with appropriate choice of parameters, the proposed order-$w$ reprocessing with preprocessing requires comparable complexity to order-$w$ reprocessing but achieves asymptotically the performance of order-$(w+2)$ reprocessing. To complement the MRB, we also employ iterative recoding on a second basis for practical SNRs and systematically extend this approach to a multi-basis order-$w$ reprocessing scheme for high SNRs. We show that the proposed multi-basis scheme significantly enlarges the error-correction radius, a commonly used measure of performance at high SNRs, over the original (single-basis) order-$w$ reprocessing. As a by-product, we also precisely characterize the asymptotic performance of the well-known Chase and GMD decoding algorithms. Our simulation results show that the proposed algorithm successfully decodes the (192, 96) Reed-Solomon concatenated code and the (256, 147) extended BCH code in near optimal manner (within 0.01 dB at a block-error-rate of $10^{-5}$) with affordable computational cost.

Index Terms—Soft-decision decoding, most reliable basis, multi-basis, preprocessing, asymptotic performance, binary linear block codes.

I. INTRODUCTION

Maximum-likelihood (ML) soft-decision decoding provides substantial performance gain in comparison to bounded-distance (hard-decision) decoding. However, ML soft-decision decoding has been shown to be NP-hard whereas polynomial-complexity bounded-distance decoding algorithms exist for many known codes. The problem of finding computationally efficient and practically implementable soft-decision decoding algorithms has been investigated extensively and remains an open and challenging problem, particularly for long block codes. In [8], Forney provided for the first time a suboptimal soft-decision decoding approach that utilizes the concept of generalized minimum distance (GMD) decoding based on channel erasure information. In [5], Chase presented another suboptimal algorithm which uses channel reliability information to search for codewords by successively applying bounded-distance decoding to candidate test error patterns corresponding to certain least reliable bit positions. At practical signal-to-noise ratios (SNRs), the methods in [8], [5] provide mild gains at manageable computational costs; however, these methods rely on an efficient hard-decision bounded-distance decoder which may not be available for many codes.

An active research direction for the soft-decision decoding of general binary linear block codes is iterative recoding based on the so-called most reliable basis (MRB), which can be obtained by applying a greedy search algorithm to the decreasing reliability order of the received word. At each iteration of iterative recoding, a test error pattern (TEP) is added to the information message associated with the MRB and then a candidate codeword is recoded using the systematic generator matrix associated with the MRB. This was first suggested by Dorsch in [7] whose approach was based on flipping TEPs that are iterated in ascending reliability order. In [9], Fossorier and Lin proposed a simple order-$w$ reprocessing scheme that systematically flips up to $w$ bits over the MRB. This method was shown to be asymptotically optimal when $w \approx \min\{\left\lfloor \frac{d_{\text{min}}}{2} \right\rfloor, K\}$, where $K$ denotes the information dimension and $d_{\text{min}}$ denotes the minimum Hamming distance of the code. In [13], Fossorier presented a multi-basis algorithm that does not require a reprocessing order larger than $(w-1)$ in its main phase and that is able to correct any error pattern of at most $w$ errors in the $K+t$ most reliable positions (the parameter $t$ satisfies $0 < t \leq \min\{K, N-K\}$ where $N$ denotes the code length and $K$ the information dimension). In [22], Wu and Hadjicostis proved that Dorsch’s proposed...
order of flipping patterns in [7] is optimal in the sense of minimizing the list error probability under a constraint on the maximum number of recoding operations. The authors also devised a decoding algorithm in which flipping patterns are iterated in natural order and each recoding operation requires at most \( N - K \) Boolean operations. In addition, the average complexity of this algorithm is substantially reduced by employing a suboptimal probabilistic skipping rule which uses the likelihood of the tentatively most likely codeword to dynamically avoid the recoding of unpromising TEPs. Recently, Valamboi and Fossorier applied the so-called “box and match” technique to order-\( w \) reprocessing and were able to correct \( 2w \) bit errors in an extended MRB composed of the MRB information bits and a certain number of the most reliable bits in the redundancy part [20]. This approach was shown to successfully decode the (192, 96) concatenated Reed-Solomon code which had never been done before; it requires, however, a large amount of memory. This breakthrough is achieved by aiming at directly maximizing the error correction capability in an extended MRB (i.e., the MRB plus some of the most reliable bits in the redundancy part) as opposed to the conventional approach of maximizing the error correction capability in the MRB [7], [9], [22].

In this paper we develop two techniques to improve iterative recoding algorithms that are based on the MRB. These techniques are universally applicable but our analysis focuses on order-\( w \) reprocessing due to its simplicity. The first technique, called preprocessing, filters out (unpromising) TEPs whose corresponding codewords have more than a pre-determined number of bit errors within an extended MRB that contains the MRB and a certain number of the most reliable bits in the redundant part. We consider two preprocessing rules. The first rule copes with TEPs whose (Hamming) weights are less than \( w + 2 \) and discards the ones for which the weight of the partial coset composed of the MRB and the \( t \) most reliable bits in the redundancy part is greater than a pre-determined threshold \( \theta \). The second rule copes with TEPs which have weight \( w + 2 \) and discards those that result in more than one bit error among the \( \tau \) most reliable bits in the redundancy part. We show that by appropriately choosing parameters \( t, \theta \) and \( \tau \) the proposed scheme has complexity comparable to that of order-\( w \) reprocessing while achieving asymptotically the performance of order-\( (w + 2) \) reprocessing. The second technique is to complement the MRB with a second basis that is essentially formed by replacing the \( \kappa \) least reliable bits in the MRB with the \( \kappa \) most reliable bits in the redundancy part, and by applying the proposed order reprocessing algorithm with preprocessing to each basis. We also propose a systematic scheme that uses multiple bases in the high SNR regime and show that the use of multiple bases can significantly enlarge error-correction radius, a commonly employed indicator of decoding performance at high SNRs.

The paper is organized as follows. In Section II we provide background on soft-decision decoding and iterative recoding. We present the order-\( w \) reprocessing algorithm with preprocessing in Section III and extend the algorithm to the scenario of multiple bases in Section IV. Simulation results are presented in Section V and pertinent remarks are made in Section VI.

II. PRELIMINARIES

Let \( \mathcal{C}(N, K) \) be a binary linear block code of length \( N \) and dimension \( K \) (and redundancy length \( R = N - K \)) that is used for error control over the additive white Gaussian noise (AWGN) channel, under binary-phase-shift-keying (BPSK) signaling of unit energy. More specifically, a bipolar version of a codeword \( \mathbf{c} = [c_1 c_2 \ldots c_N] \), i.e., \( \left[(-1)^{c_1} (-1)^{c_2} \ldots (-1)^{c_N}\right] \) is used for transmission. At the output of the demodulator, the received unquantized word \( \mathbf{r} = [r_1 \ r_2 \ldots \ r_N] \) takes the form \( r_i = (-1)^{c_i} + n_i \), \( i = 1, 2, \ldots, N \), where \( n_i \) are independent and identically distributed Gaussian random variables with zero mean and variance \( N_0/2 \). We adopt the common assumption that codewords are equally probable, i.e., \( \Pr(\mathbf{c}) = 2^{-K} \) for any \( \mathbf{c} \in \mathcal{C} \). Under this model, the \( i \)-th bit log-likelihood-ratio, which is formally defined as \( \delta_i = \ln \frac{\Pr(\mathbf{c} = \mathbf{0} | r_i)}{\Pr(\mathbf{c} = \mathbf{1} | r_i)} \), can be simplified to \( \delta_i = 4r_i/N_0 \), i.e., the received symbol \( r_i \) is simply a scaled log-likelihood-ratio. Bit-hard-decision sets \( y_i \) to 0 if \( r_i > 0 \) and 1 otherwise; the reliability of the corresponding bit decision, which is defined as a scaled version of the magnitude of the log-likelihood-ratio, is conveniently represented by \( \alpha_i = |r_i| \).

Suppose we are given a binary block code \( \mathcal{C}(N, K) \) with generator matrix \( \mathbf{G} \). Upon receiving \( \mathbf{r} \) (in the form explained above), one can construct a new systematic generator matrix \( \hat{\mathbf{G}} \) associated with the most reliable (information) basis (MRB), via a transformation from the original generator matrix \( \mathbf{G} \). The following greedy search algorithm can be used to obtain the MRB [9].

**Greedy Algorithm for Obtaining the MRB**

1. Sort bit indices in decreasing order of reliability, \( i_1, i_2, \ldots, i_n \) (a permutation of \( 1, 2, \ldots, n \)).
2. Set the index set to be empty.
3. For \( l = 1, 2, \ldots, \), till the index set becomes full, do:
   a. Check whether the \( i_l \)-th column vector of \( \mathbf{G} \) is independent of the column vectors of \( \mathbf{G} \) that are associated to the index set.
   b. If so, add \( i_l \) to the index set, otherwise discard \( i_l \).
4. Permute (column-wise) the original matrix \( \mathbf{G} \) in accordance with the index set and then systematize the resulting matrix.

Note that when \( \frac{1}{n} \geq \frac{1}{2} \), the above procedure (Steps 3 and 4 in particular) can be made computationally more efficient by operating on the parity check matrix instead of the generator matrix [9]. Note that the re-ordered reliabilities in the MRB and the redundancy part satisfy decreasing order respectively, i.e.,

\[
\hat{\alpha}_1 \geq \hat{\alpha}_2 \geq \ldots \geq \hat{\alpha}_K \quad \text{and} \quad \hat{\alpha}_{K+1} \geq \hat{\alpha}_{K+2} \geq \ldots \geq \hat{\alpha}_N. \tag{1}
\]

In the sequel, "\( \tau \)" stands for the ordering associated the MRB; subscript "\( 1 \)" stands for the index set \( \{1, 2, 3, \ldots, K\} \) associated with the basis and "\( 2 \)" stands for the index set \( \{K + 1, K + 2, \ldots, N\} \) associated with the redundancy part. Since matrix \( \hat{\mathbf{G}} \) is in systematic form, we will also write it as \( \hat{\mathbf{G}} = [\mathbf{I}_K \hat{\mathbf{P}}] \), where \( \mathbf{I}_K \) denotes the \( K \times K \) identity matrix and \( \hat{\mathbf{P}} \) reflects the dependency of the (unreliable) redundancy bits on the (reliable) information bits.
Having obtained the reordered $\tilde{G}$, $\tilde{y}$ and $\tilde{\alpha}$ associated with the MRB, one can immediately generate the first candidate codeword by recoding the bit information $\tilde{y}_1$

$$\tilde{c}_0 \triangleq \tilde{y}_1 \tilde{G} = [\tilde{y}_1 \quad \tilde{y}_1 \tilde{P}].$$

Thus, if no errors occur in the MRB, then the above recoding operation successfully retrieves the transmitted codeword. When only a few errors are present in the MRB, one can develop searching strategies that iteratively flip bits of $\tilde{y}_1$ (the information bits corresponding to the MRB), each time recoding the resulting information vector into a codeword (using $\tilde{G}$) and evaluating its likelihood. The action of flipping bits in $\tilde{y}_1$ is equivalent to adding to $\tilde{y}_1$ a binary test error pattern (TEP) $\tilde{e} \triangleq [e_1 \ e_2 \ldots \ e_K]$ and performing a recoding operation with respect to the TEP $\tilde{e}$ as

$$\tilde{c}_e \triangleq (\tilde{y}_1 \oplus \tilde{e}) \tilde{G} = \left[\tilde{y}_1 \oplus \tilde{e} \quad (\tilde{y}_1 \oplus \tilde{e}) \tilde{P}\right],$$

where $\oplus$ denotes the bit-wise XOR operator and $\tilde{y}_1$ denotes the information corresponding to the MRB.

The weighted Hamming distance (WHD) with respect to a TEP $\tilde{e}$, denoted by $D(\tilde{\alpha}, \tilde{e})$, is defined as the sum of the reliabilities corresponding to the entries in which the recoded codeword $\tilde{c}_e$ differs from $\tilde{y}$, i.e.,

$$D(\tilde{\alpha}, \tilde{e}) = \sum_{1 \leq i \leq N} \tilde{\alpha}_i. \quad (4)$$

It is well-known that Maximum-Likelihood (ML) soft-decision decoding is equivalent to searching for the TEP that minimizes the WHD $D(\tilde{\alpha}, \cdot)$. It can be verified that when $\tilde{\alpha}_i = 1$, $i = 1, 2, \ldots, N$, the above formula evaluates to the Hamming distance between $c_0$ and $\tilde{y}$.

Let $e_2 = [e_{2,1} \ e_{2,2} \ldots \ e_{2,R}]$ denote the binary error vector in the redundancy part that corresponds to TEP $e$ in the MRB, i.e.,

$$e_2 \triangleq \tilde{y}_2 \oplus (\tilde{y}_1 \oplus e) \tilde{P}. \quad (5)$$

The WHD $D(\tilde{\alpha}, e)$ can be naturally decomposed into two parts, the information WHD and the redundancy WHD, defined respectively as follows

$$D_1(\tilde{\alpha}, e) \triangleq \sum_{1 \leq i \leq K} \tilde{\alpha}_i, \quad D_2(\tilde{\alpha}, e_2) \triangleq \sum_{R+1 \leq i \leq N} \tilde{\alpha}_{K+i}. \quad (6)$$

In [7], TEPs are ordered in ascending information WHD $D_1(\tilde{\alpha}, \cdot)$, which is shown in [22] to be optimal in the sense of minimizing the list error probability when choosing only a subset among $2^K$ TEPs. This ordering, however, depends on the reliability values and thus requires complex dynamic sorting for each decoding. A simpler TEP ordering is order reprocessing which sorts TEPs in increasing order of primarily Hamming weight and secondary dictionary ordering [9], [14], as shown in Table I. This ordering is independent of reliability values and is thus fixed based on the given reliability order in (1).

### III. ORDER REPROCESSING WITH PREPROCESSING

In this section we present a soft-decision decoding algorithm that is based on iterative recoding in light of the MRB. Recall that order-$w$ reprocessing systematically recodes TEPs with weights up to $w$ in reprocessing ordering, i.e., first the TEP $e = 0$, next all TEPs with weight 1 in the dictionary order, next all TEPs with weight 2 in the dictionary order, ..., finally all TEPs with weight $w$ in dictionary order [9].

An improved implementation of order-$w$ reprocessing is given in [22]. We present two preprocessing rules which can be embedded into any iterative recoding algorithm (possibly with slight modifications). We also study the asymptotic behavior of the proposed algorithm as $N_0$ goes to zero.

#### A. First preprocessing rule

We now consider a thresholding rule on the extended MRB which is composed of the MRB and the $t$ most reliable bits in the redundancy part. Following the order in (1), the extended MRB can be conveniently denoted by $[1, K + t]$.

This preprocessing rule dictates that a codeword $\tilde{c}$ is discarded without further consideration if

$$d_{[1, K+t]}(\tilde{c}, \tilde{y}) > \theta, \quad (7)$$

where $d_{[\alpha, \beta]}(\tilde{c}, \tilde{y})$ denotes the Hamming distance between $\tilde{c}$ and $\tilde{y}$ within bit positions $[\alpha, \beta]$. We next show that this preprocessing rule can be efficiently enforced if TEPs and their resulting codeword information are generated in terms of sub-TEPs (STEPs) as described below.

Consider a decomposition of a TEP $e$ into a unit vector whose only 1 is positioned at the smallest 1’s position in $e$ (i.e., the most reliable nonzero bit in $e$) and a STEP which corresponds to the remaining of $e$, i.e.,

$$e = e^{\text{sub}} + u_{\mu(e)}, \quad (8)$$

where $\mu(e) \triangleq \min\{i : e_i = 1\}$.

Note that since $e^{\text{sub}}\mu(e)$ is equivalent to $e^{\text{sub}} \oplus u_{\mu(e)}$ in the sequel, $e^{\text{sub}} + u_i$ always implies $e^{\text{sub}} = 0$. As we can see, a nonzero STEP $e^{\text{sub}}$ always satisfies $\mu(e^{\text{sub}}) > 1$. A STEP $e^{\text{sub}}$ is used to generate $\mu(e^{\text{sub}}) - 1$ TEPs (where $\mu(e^{\text{sub}})$ is the minimum index among nonzero positions of $e^{\text{sub}}$). More specifically, STEP $e^{\text{sub}}$ generates the TEPs in the set

$$S(e^{\text{sub}}) \triangleq \{e^{\text{sub}} + u_1, e^{\text{sub}} + u_2, \ldots, e^{\text{sub}} + u_{\mu(e^{\text{sub}})-1}\}.$$
We further note that any two STEPs generate disjoint sets of TEPs, i.e., \( S(e_1^{ab}) \cap S(e_2^{ab}) = \emptyset \). This is so because a TEP is uniquely decomposed as \( e = e^{ab} + u_\theta(e) \). This leads us to conclude that all STEPs with weight \( i-1 \) generate all TEPs with weight \( i \); furthermore,

\[
\{ e \in S^{e^{ab}} : w(e) = i \} = \bigcup_{e^{ab} : w(e^{ab}) = i-1} S(e^{ab}). \tag{9}
\]

Eq. (9) indicates that \( (K_i) \) TEPs with weight \( i \) are generated by \( (K_{i-1}) \) STEPs (recall that \( \epsilon_1^{ab} = 0 \), thus \( e^{ab} \) has essentially \( K - 1 \) degrees of freedom). Thus, on average, a STEP with weight \( i-1 \) produces \( K/i \) TEPs.

Note that the information of \( e^{ab} \) is only computed once for all \( \mu(e^{ab}) - 1 \) TEPs in \( S(e^{ab}) \); thus, we can calculate the distance \( d_{1i} = \sum_{k=1}^{w} c_k e^{\sim w}(e) \) through

\[
d_{1i} = \sum_{i=1}^{w} \left( e^{\sim w}(e) \right)_i + d_{K+1}(K_i) \sum_{i=1}^{w} \left( e^{\sim w}(e) \right)_i = \sum_{i=1}^{w} \left( e^{\sim w}(e) \right)_i + d_{1i}(K_i) \sum_{i=1}^{w} \left( e^{\sim w}(e) \right)_i \tag{10}
\]

for \( \mu(e^{ab}) = 1, \ldots, 2, 1 \), where \( e_2^{ab} \) is the redundancy error vector as defined in (5). Therefore, given \( w(e^{ab}) + 1 \) and \( e_2^{ab} \), a preprocessing operation can be efficiently implemented with \( t \) XOR operations and a weight look-up table. Since a STEP produces up to \( K \) TEPs, the complexity of preprocessing associated with a STEP is bounded by \( Kt \) XORs. In fact, if order\( -w \) preprocessing is enforced for STEPs, then the average complexity of preprocessing associated with a STEP is

\[
t \sum_{i=1}^{w+1} (K_i)/(K_{w+1}) = t \frac{K}{w+1} \tag{10}
\]

XOR operations.

We next show how to efficiently compute the WHD \( D(\vec{\alpha}, e) \) in light of the STEP information. If \( D_1(\vec{\alpha}, e^{ab}) \) (defined in (6)) and \( e_2^{ab} \) (defined in (5)) are known, \( D(\vec{\alpha}, e) \) can be computed efficiently as follows:

\[
e_2 = e_2^{ab} \oplus \vec{p}_\mu(e), \tag{11}
\]

\[
D(\vec{\alpha}, e) = D_1(\vec{\alpha}, e) + D_2(\vec{\alpha}, e_2) = D_1(\vec{\alpha}, e^{ab}) + \mu(e) + D_2(\vec{\alpha}, e_2), \tag{12}
\]

where \( \vec{p}_\mu \) denotes the \( i \)-th row of \( \vec{p} \) and \( D_2(\vec{\alpha}, e_2) \) is the redundancy WHD defined in (6). Since we only need to explicitly construct \( e_2^{ab} = [\vec{y} \oplus e \oplus \vec{y}_2 \oplus e_2] \) as the best-offer codeword (to output in the final stage), it suffices to compute \( e_2^{ab} \) and \( D_1(\vec{\alpha}, e^{ab}) \) for a given STEP, so that \( e_2 \) and \( D(\vec{\alpha}, e) \) can be computed through (11) and (12) respectively.

We remark that the task proposed in [13], i.e., to correct any error pattern of at most \( w + 1 \) errors in the extended MRB \([1, K + p]) by requiring a reprocessing order no larger than \( w \), can be straightforwardly implemented by the proposed order\( -w \) reprocessing with the preprocessing rule that we propose in this section (with \( \theta = w + 1 \) and \( t = p \)). Moreover, this (trivial) implementation saves the trouble of generating the multi-basis required in [13].

**B. Second preprocessing rule**

To simplify our presentation we again assume that order\( -w \) reprocessing for STEPs is enforced, i.e., STEPs of order\( -(w + 1) \) are generated and processed. In this subsection, we are interested in processing TEPs with weight \( w + 2 \) using STEPs with weight \( w \). As can be readily seen, if we apply the same strategy as in the preceding subsection, each STEP on average contains \( K(K+1)/2 \) TEPs (note that \( K \) STEPs are coupled with \( K(w+1)/2 \) STEPs). To this end, we propose a slightly easier task so as to avoid such heavy computational burden; in particular, we are interested in finding TEPs \( e \) with weight \( w + 2 \) satisfying

\[
w_{1, \tau}(e_{2}) \leq 1, \tag{13}
\]

where \( \tau \) is a parameter to be determined. The motivation for choosing the threshold to be 1 is to substantially reduce searching complexity (this is addressed subsequently), while at the same time retaining performance in an asymptotic sense (this issue is addressed in the last part of this section).

Suppose we pre-calculate all \( L = (K - w)/2 \) combinations of any two \( \vec{p}_i \), \( i = 1, \ldots, K - w \) (recall \( \vec{G} = [I \vec{P}] \) and \( \vec{P} = [\vec{p}_1 \vec{p}_2 \ldots \vec{p}_K] \)) and (arbitrarily) index them by \( q_1, q_2, \ldots, q_L \). We can uniquely decompose any TEP \( e \) of weight \( w + 2 \) into a vector with two 1’s entries corresponding to the two most reliable 1’s entries of \( e \) and the residual vector which is a STEP of weight \( w \). Note that the vector \( e_2 \) that corresponds to \( e \) with \( w(e) = w + 2 \) can be decomposed into some \( q_i \) and some \( e_{2, \tau} \) that corresponds to \( e^{ab} \) with \( w(e^{ab}) = w \). Thus, for a STEP \( e^{ab} \) of weight \( w \), it suffices to find all \( q_i \)’s satisfying

\[
d_{1, \tau}(q_i, e^{ab}) \leq 1. \tag{14}
\]

We next present two different approaches to implement the above task. The first approach is convenient for software implementation and stores the information regarding \( q_i, 1 \leq i \leq L \), in an array, naturally indexed by \( (q_i)[1, \tau], i = 1, 2, \ldots, L \). Let \( \vec{u}_\tau \) denote the \( \tau \)-dimensional unit vector with the only “1” positioned at the \( \tau \)th entry. Once the redundancy error vector \( e_2^{ab} \) is given, the \( \tau + 1 \) patterns \( (e_2^{ab})[1, \tau], (e_2^{ab})[1, \tau] \oplus \vec{u}_\tau, (e_2^{ab})[1, \tau] \oplus \vec{u}_1, \ldots, (e_2^{ab})[1, \tau] \oplus \vec{u}_{\tau-1} \) are checked one by one: when a pattern is in the array, it is recoded into codeword(s) (note that an index of the array may be empty or contain one or more \( q_i \)’s). This method is called the “box and match” algorithm and a detailed description for it can be found in [20].

The second approach is convenient for hardware implementation and stores \( (q_1)[1, \tau], (q_2)[1, \tau], \ldots, (q_L)[1, \tau] \) in a \( \tau \)-level binary tree with the root set to void and each path from the root to a leaf (with path length \( \tau \)) representing a prototype \( (q_i)[1, \tau] \). The information \( q_i \) is stored in the end node of the path (i.e., a leaf of the tree) and represents \( (q_i)[1, \tau] \). Given this data structure, a matching operation (i.e., checking whether an input vector is a prototype pattern) can be accomplished with at most \( \tau \) XOR operations by searching from the root to a leaf. Thus, the task imposed by (14) can be efficiently implemented by \( \tau + 1 \) match operations for \( (e_2^{ab})[1, \tau], (e_2^{ab})[1, \tau] \oplus \vec{u}_1, (e_2^{ab})[1, \tau] \oplus \vec{u}_2, \ldots, (e_2^{ab})[1, \tau] \oplus \vec{u}_{\tau-1} \), respectively. It can be readily seen that the second preprocessing rule requires at most
\(\tau(\tau + 1)\) binary operations and this complexity is comparable to the complexity of the first preprocessing rule.

C. Asymptotic performance analysis

In this section, we analyze the asymptotic performance of the proposed order-\(w\) reprocessing with preprocessing, as \(N_0\) goes to zero. To simplify the analysis, we make the following assumptions and approximations: (i) The \(K\) most reliable indices are linearly independent and compose the MRB. In fact, for random codes the probability that the \(K + s\) most reliable indices contain a basis is at least \(1 - 2^{-s}\) (see [17], p. 455). Thus, for relatively large \(K\) and \(N\), the MRB and the \(K\) most reliable bits are almost equally reliable. This assumption will be used to simplify the analysis of the list (decoding) error probability which is defined as the probability that a decoder fails to retrieve the transmitted codeword as one of candidate codewords (thus, the decoder outputs an erroneous codeword under any (logical) decision rule). (ii) All redundancy error vectors \(e_2\) satisfy uniform distribution over \([1, \max(t, \tau)]\), i.e., \(\Pr((e_2)[1, \max(t, \tau)]) = 2^{-\max(t, \tau)}\). This assumption will be used to evaluate the number of recodings that are required and comprise the dominant computational burden.

Our analysis will be focusing on the list error probability as \(N_0\) goes to zero. For a given decoding algorithm, the probability of a list decoding error \(P_{\text{list}}\) and the probability of a decoding error \(P_e\) satisfy

\[
\max\{P_{\text{ML}}, P_{\text{list}}\} \leq P_e \leq P_{\text{list}} + P_{\text{ML}},
\]

where \(P_{\text{ML}}\) denotes the error probability of ML decoding.

It has been shown in [18] that as \(N_0 \to 0\) the asymptotic performance of ML decoding takes the form

\[
P_{\text{ML}} = \left(\frac{A_{\text{dmin}}}{2\sqrt{A_{\text{dmin}}}\pi} + o(1)\right)N_0^{1/2}e^{-\frac{2A_{\text{dmin}}}{N_0}},
\]

where \(o(1)\) denotes a negligible term that diminishes as \(N_0\) goes to zero, \(A_{\text{dmin}}\) denotes the minimum Hamming distance of the code \(C(N, K)\), and \(d_{\text{dmin}}\) denotes the number of codewords with Hamming weight \(d_{\text{dmin}}\). An alternative interpretation for (16) is

\[
\lim_{N_0 \to 0} N_0^{1/2}e^{-\frac{2A_{\text{dmin}}}{N_0}} = \frac{A_{\text{dmin}}}{2\sqrt{A_{\text{dmin}}}\pi}.
\]

The following result is proved in the Appendix.

**Theorem 1:** As \(N_0 \to 0\), the probability of more than \(w\) bit errors among the \(N - S\) most reliable bits of a received word is

\[
P_{S, w} = \begin{cases} C_{S, w}^{(1)} + o(1) & \text{if } S \leq w^*, \\ C_{S, w}^{(2)} + o(1) & \text{if } S > w^*, \end{cases}
\]

where \(w^* \triangleq w + 1\) and \(C_{S, w}^{(1)}\) and \(C_{S, w}^{(2)}\) are constants given by

\[
C_{S, w}^{(1)} = \frac{N}{S + w^*} \sum_{i=0}^{S + w^*} \binom{S + w^*}{i} 2^{-(S + w^*)/\pi} (S + w^*)_{(i)}^{-(S + w^*)/2},
\]

\[
C_{S, w}^{(2)} = \frac{N}{w^*} \binom{N - w^*}{S} \pi^{-(S + w^*)/2} S^{-w^*} (S + w^* - S)^{-S}.
\]

**Remarks A:** As shown in the proof in the Appendix, when \(S \leq w^*\) the dominant error events are those that involve between \(w^*\) and \(S + w^*\) bit errors with exactly \(w^*\) errors among the \(N - S\) most reliable bits. On the other hand, when \(S > w^*\) the dominant error events are those with \(w^*\) bit errors occurring among the \(N - S\) most reliable bits and with the \(S\) least reliable bits being error free. The above theorem precisely characterizes the asymptotic list-decoding performance of the Chase-2 algorithm [5] and the generalized minimum distance (GMD) decoding algorithm [8]; a looser bound is given in [18].

We next determine the asymptotic behavior of order-\(w\) reprocessing in light of assumption (i) and Theorem 1. Note that \(S\) in Theorem 1 is replaced by the redundancy length \(R \triangleq N - K\) in this context and that the ML error exponent is \(d_{\text{dmin}}\), which is a fraction of \(R\). Since we are only interested in choosing \(w\) such that the list error exponent is at most as large as the ML error exponent (i.e., \(d_{\text{dmin}}\)), \(w\) must be less than \(R\). Using the above observations, the asymptotic performance of order-\(w\) reprocessing can be closely approximated by ignoring the (possible) linear dependency.

**Corollary 1:** Assuming that the \(K\) most reliable bits are linearly independent, the list decoding error probability of order-\(w\) reprocessing takes, as \(N_0 \to 0\), the form

\[
P_{R, w} = \left(C_{R, w}^{(2)} + o(1)\right)N_0^{(R+w)/2}e^{-\frac{4B_{\text{r}}}{(R+w)}},
\]

where \(R \triangleq N - K\) and \(w^* \triangleq w + 1\).

**Remarks B:** Note that \(\frac{4B_{\text{r}}}{R+w} \approx d_{\text{dmin}}\) due to the fact that \(\frac{4B_{\text{r}}}{R+w} \ll R\). Moreover, we have \(\lim_{N_0 \to 0} P_{R, w}/P_{\text{ML}} = 0\) when \(w\) satisfies \(\frac{4B_{\text{r}}}{R+w} \geq d_{\text{dmin}}\). Thus, roughly speaking, by choosing \(w \approx \frac{d_{\text{dmin}}}{4}\), order-\(w\) reprocessing asymptotically achieves ML decoding as \(N_0 \to 0\) (in the sense that \(\lim_{N_0 \to 0} P_e(\frac{d_{\text{dmin}}}{4})/P_{\text{ML}} = 1\), where \(P_e(w)\) denotes the error probability of order-\(w\) reprocessing).

We next determine the decision region of preprocessing parameters so that preprocessing error events are not the dominant events when compared to error events of the original order-(\(w+2\)) reprocessing for high SNRs. It suffices to show that the list error exponents of the two preprocessing rules are larger than that of order-(\(w+2\)) reprocessing. In order for order-(\(w+2\)) reprocessing to make sense, we assume that at least order-3 reprocessing is required to achieve asymptotically optimal decoding. Hence, we assume that the redundancy length satisfies \(R \geq 1\).

We first introduce two notions. (i) The error-correction radius of a soft-decision decoding algorithm \(A\) is defined as the largest value, denoted by \(D_A^*\), such that \(A\) decodes correctly the transmitted codeword \(c_{\text{tr}}^r\) whenever the received sequence \(r\) is within Euclidean distance \(D_A^*\) to the bipolar version of \(c_{\text{tr}}^r\). (ii) The list error-correction radius (LECR) \(D_A\) of a decoding algorithm \(A\) is defined as the largest value such that \(A\) properly lists \(c_{\text{tr}}^r\) as one of candidate codewords whenever the received vector \(r\) is within Euclidean distance \(D_A\) to the bipolar version of \(c_{\text{tr}}^r\). It is well-known that \(D_{\text{ML}}^* = \sqrt{d_{\text{dmin}}}\), and that the two error-correction radii satisfy
Thus, the LECR is equal to the minimum Euclidean distance between the LECR and the list error exponent. Without loss of generality, we assume the all-zero codeword is transmitted.

We note that the worst-case error event that results in a listing by Theorem 1; herein we have a close look at the relation between the LECR and the list error exponent. Without loss of generality, we assume the all-zero codeword is transmitted.

We proceed to discuss the LECR of the second preprocess-
ing optimization [10]; herein we have a close look at the relation between the LECR and the list error exponent. Without loss of generality, we assume the all-zero codeword is transmitted.

We note that the worst-case error event that results in a listing by Theorem 1; herein we have a close look at the relation between the LECR and the list error exponent. Without loss of generality, we assume the all-zero codeword is transmitted.

We note that the worst-case error event that results in a listing by Theorem 1; herein we have a close look at the relation between the LECR and the list error exponent. Without loss of generality, we assume the all-zero codeword is transmitted.

We note that the worst-case error event that results in a listing by Theorem 1; herein we have a close look at the relation between the LECR and the list error exponent. Without loss of generality, we assume the all-zero codeword is transmitted.

We note that the worst-case error event that results in a listing by Theorem 1; herein we have a close look at the relation between the LECR and the list error exponent. Without loss of generality, we assume the all-zero codeword is transmitted.

We note that the worst-case error event that results in a listing by Theorem 1; herein we have a close look at the relation between the LECR and the list error exponent. Without loss of generality, we assume the all-zero codeword is transmitted.

We note that the worst-case error event that results in a listing by Theorem 1; herein we have a close look at the relation between the LECR and the list error exponent. Without loss of generality, we assume the all-zero codeword is transmitted.

We note that the worst-case error event that results in a listing by Theorem 1; herein we have a close look at the relation between the LECR and the list error exponent. Without loss of generality, we assume the all-zero codeword is transmitted.

We note that the worst-case error event that results in a listing by Theorem 1; herein we have a close look at the relation between the LECR and the list error exponent. Without loss of generality, we assume the all-zero codeword is transmitted.

We note that the worst-case error event that results in a listing by Theorem 1; herein we have a close look at the relation between the LECR and the list error exponent. Without loss of generality, we assume the all-zero codeword is transmitted.

We note that the worst-case error event that results in a listing by Theorem 1; herein we have a close look at the relation between the LECR and the list error exponent. Without loss of generality, we assume the all-zero codeword is transmitted.

We note that the worst-case error event that results in a listing by Theorem 1; herein we have a close look at the relation between the LECR and the list error exponent. Without loss of generality, we assume the all-zero codeword is transmitted.

We note that the worst-case error event that results in a listing by Theorem 1; herein we have a close look at the relation between the LECR and the list error exponent. Without loss of generality, we assume the all-zero codeword is transmitted.

We note that the worst-case error event that results in a listing by Theorem 1; herein we have a close look at the relation between the LECR and the list error exponent. Without loss of generality, we assume the all-zero codeword is transmitted.

We note that the worst-case error event that results in a listing by Theorem 1; herein we have a close look at the relation between the LECR and the list error exponent. Without loss of generality, we assume the all-zero codeword is transmitted.

We note that the worst-case error event that results in a listing by Theorem 1; herein we have a close look at the relation between the LECR and the list error exponent. Without loss of generality, we assume the all-zero codeword is transmitted.

We note that the worst-case error event that results in a listing by Theorem 1; herein we have a close look at the relation between the LECR and the list error exponent. Without loss of generality, we assume the all-zero codeword is transmitted.

We note that the worst-case error event that results in a listing by Theorem 1; herein we have a close look at the relation between the LECR and the list error exponent. Without loss of generality, we assume the all-zero codeword is transmitted.

We note that the worst-case error event that results in a listing by Theorem 1; herein we have a close look at the relation between the LECR and the list error exponent. Without loss of generality, we assume the all-zero codeword is transmitted.

We note that the worst-case error event that results in a listing by Theorem 1; herein we have a close look at the relation between the LECR and the list error exponent. Without loss of generality, we assume the all-zero codeword is transmitted.

We note that the worst-case error event that results in a listing by Theorem 1; herein we have a close look at the relation between the LECR and the list error exponent. Without loss of generality, we assume the all-zero codeword is transmitted.

We note that the worst-case error event that results in a listing by Theorem 1; herein we have a close look at the relation between the LECR and the list error exponent. Without loss of generality, we assume the all-zero codeword is transmitted.

We note that the worst-case error event that results in a listing by Theorem 1; herein we have a close look at the relation between the LECR and the list error exponent. Without loss of generality, we assume the all-zero codeword is transmitted.

We note that the worst-case error event that results in a listing by Theorem 1; herein we have a close look at the relation between the LECR and the list error exponent. Without loss of generality, we assume the all-zero codeword is transmitted.

We note that the worst-case error event that results in a listing by Theorem 1; herein we have a close look at the relation between the LECR and the list error exponent. Without loss of generality, we assume the all-zero codeword is transmitted.

We note that the worst-case error event that results in a listing by Theorem 1; herein we have a close look at the relation between the LECR and the list error exponent. Without loss of generality, we assume the all-zero codeword is transmitted.

We note that the worst-case error event that results in a listing by Theorem 1; herein we have a close look at the relation between the LECR and the list error exponent. Without loss of generality, we assume the all-zero codeword is transmitted.

We note that the worst-case error event that results in a listing by Theorem 1; herein we have a close look at the relation between the LECR and the list error exponent. Without loss of generality, we assume the all-zero codeword is transmitted.

We note that the worst-case error event that results in a listing by Theorem 1; herein we have a close look at the relation between the LECR and the list error exponent. Without loss of generality, we assume the all-zero codeword is transmitted.

We note that the worst-case error event that results in a listing by Theorem 1; herein we have a close look at the relation between the LECR and the list error exponent. Without loss of generality, we assume the all-zero codeword is transmitted.
(herein we do not take into account extremely high rate codes). Therefore, we can safely set $\frac{1}{2} \sum_{i=0}^{w} \binom{K}{i}$ as the bound on the maximum number of recodings.

For the second preprocessing rule, the mean number of additional recodings is $(w+2)^2 \tau (\tau + 1)$ based on assumption (ii) that redundancy error vectors $e_2$ satisfy uniform distribution over $[1, \tau]$. Thus, it suffices to choose

$$\tau = 3 \log_2(K/w)$$

(34)
to accommodate (29) while achieving $(w+2)^2 \tau (\tau + 1) \ll \sum_{i=0}^{w} \binom{K}{i}$ (again we do not take into account extremely high rate codes). Note that we can safely set $\frac{1}{2} \sum_{i=0}^{w} \binom{K}{i}$ as the bound on the maximum number of recodings.

We summarize the above analysis in the following theorem.

**Theorem 2:** By choosing $t, \theta$ and $\tau$ as

$$t = R/2, \quad \theta = 2(w + 3), \quad \tau = 3 \log_2(K/w),$$

(35)
the proposed order-$w$ reprocessing with preprocessing generates less candidate codewords than the original order-$w$ reprocessing, while its decoding error probability is asymptotically (as $N_0 \to 0$) the same as that of order-$(w + 2)$ reprocessing; more specifically,

$$\lim_{N_0 \to 0} \frac{P_e^{(2)}(w)}{P_e^{(0)}(w + 2)} = 1$$

(36)
where $P_e^{(2)}(w)$ denotes the decoding error probability of the proposed order-$w$ reprocessing with preprocessing, and $P_e^{(0)}(w)$ denotes the decoding error probability of the original order-$w$ reprocessing.

**Remarks D:** The choice of parameters in the above theorem is not necessarily optimal, i.e., it does not necessarily minimize the number of candidate codewords subject to the constraints (27) and (29). The proposed scheme is more suitable for high rate codes, say $\frac{K}{N} \geq \frac{1}{2}$.

### IV. Multi-Basis Order Reprocessing

**A. Multi-basis scheme**

A general multi-basis order reprocessing scheme has been presented in [11]. In this section we investigate a special multi-basis order reprocessing scheme for high rate codes, i.e., $\frac{K}{N} \geq \frac{1}{2}$. Consider the scenario in Figure 1 which depicts the distribution of errors after bit hard-decision is applied and the bits are reordered in terms of the MRB and the corresponding redundancy part following the order in (1). In the low SNR regime a certain number, say $\kappa$ (where $\kappa \ll R \triangleq N - K$), of the least reliable bits in the MRB is likely to be almost equally reliable to the $\kappa$ most reliable bits in redundancy part (recall that the reliabilities are sorted in decreasing order in both the MRB and the corresponding redundancy part). In the example of Figure 1, order-5 reprocessing must be employed to correct the errors if the original order-$w$ reprocessing is applied. However, if the interchange of the $\kappa = 7$ bits between the two underlined parts results in another basis, (i.e., the columns $u_1, \ldots, u_{K-k}, \tilde{p}_1, \ldots, \tilde{p}_k$ are linearly independent), then we can correct the errors by only employing order-3 reprocessing on this second basis. This motivates us to construct a second basis by rearranging the bit positions in the order

$$u_1, \ldots, u_{K-k}, \tilde{p}_1, \ldots, \tilde{p}_R, u_{K-k+1}, \ldots, u_K$$

(37)
(recall that $\tilde{G} = [I_{K} \tilde{P}]$) and then applying the greedy algorithm (as described in Section II).

Let $\tilde{G}, \tilde{y}, \alpha$ be the permuted $G, y, \alpha$ corresponding to the second basis $B_2$. Some properties of this second basis are indicated in the following proposition.

**Proposition 1:** (i) If $k < d_{min}$, then the second basis does not involve indices corresponding to $\{\bar{a}_1\}_{i=K-k+1}^K$.

(ii) If $[u_1 \ldots u_{K-k} \tilde{p}_1 \ldots \tilde{p}_R]$ is full rank, then the bit reliabilities $\hat{\alpha}_1 \geq \hat{\alpha}_2 \geq \ldots \geq \hat{\alpha}_K$ associated with the second basis are the most reliable, in the sense that

$$\hat{\alpha}_i \geq \bar{\alpha}_i, \quad i = 1, 2, \ldots, K,$$

where $\hat{\alpha}_1 \geq \hat{\alpha}_2 \geq \ldots \geq \hat{\alpha}_K$ denotes the reliabilities associated with another basis contained by $\{u_1, \ldots, u_{K-k}, \tilde{p}_1, \ldots, \tilde{p}_R\}$.

**Proof:** (i) The first property is due to the fact that at most $R - d_{min} + 1$ linear dependencies can occur during the construction of the MRB (otherwise a nonzero codeword with (Hamming) weight less than $d_{min}$ can be constructed) [14].

(ii) We prove the second property by contradiction. Let $\{\tilde{g}_i\}_{i=1}^K$ and $\{\bar{g}_i\}_{i=1}^K$ be the columns of the generator matrix corresponding to the obtained basis and the other basis respectively. Suppose $l$ is the largest index such that $|\tilde{\alpha}_l| < |\bar{\alpha}_l|$. We observe that $\tilde{g}_1, \tilde{g}_2, \ldots, \tilde{g}_l$ must be linearly dependent of $\tilde{g}_1, \tilde{g}_2, \ldots, \tilde{g}_{l-1}$, otherwise, say $\tilde{g}_i$ ($1 \leq i \leq l$) is not, then $\tilde{g}_i$ must be included in the obtained basis according to the greedy search procedure in Section II. Thus, we obtain consequently

$$\text{Span}\{\tilde{g}_1, \tilde{g}_2, \ldots, \tilde{g}_l\} \subset \text{Span}\{\tilde{g}_1, \tilde{g}_2, \ldots, \tilde{g}_{l-1}\},$$

which is obviously false, since $\tilde{g}_1, \tilde{g}_2, \ldots, \tilde{g}_l$ are linearly independent. We thus conclude the proof of part (ii). □

**Remarks E:** An alternative way to generate a second basis is to apply the greedy search through the following ordering

$$\tilde{p}_1, \ldots, \tilde{p}_k, u_1, \ldots, u_K$$

(38)
Note that $[\tilde{p}_1 \ldots \tilde{p}_k, u_1 \ldots u_K]$ is full rank and thus the greedy search is guaranteed to produce a basis. This ordering has been used in [13]. Our simulations show that, when two-basis order-$w$ reprocessing is exploited, the proposed ordering

\[ \tilde{y} \oplus \tilde{c}^{tr} \]

\[ \begin{array}{cccccccccc}
0 & 0 & 0 & 1 & 0 & 0 & 0 & \ldots & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & | & 0 & 1 & 0
\end{array} \]
We note that in the high SNR regime bit information in the 
\([K + 1, K + \kappa]\) band is almost equally reliable to that of 
the entire MRB \([1, K]\), rather than that of \([K - 1, K]\) 
in the low SNR regime. To this end, we extend the above 
two-basis strategy in the following manner (for simplicity, we 
assume that \(K\) is divisible by \(\kappa\), say \(K = \alpha \kappa\)): we generate 
a additional bases \(B_i, i = 1, 2, \ldots, a\), by applying the greedy 
search to the following sequential orders respectively:

\[
\begin{align*}
(1) & : u_1, \ldots, u_{K-k}, \tilde{p}_1, \\
(2) & : u_1, \ldots, u_{K-2\kappa}, u_{K-\kappa+1}, \ldots, u_{K}, \tilde{p}_1, \\
(3) & : u_1, \ldots, u_{K-3\kappa}, u_{K-2\kappa+1}, \ldots, u_{K}, \tilde{p}_1, \\
& \vdots \\
(a-1) & : u_1, \ldots, u_{\kappa}, u_{2\kappa+1}, \ldots, u_{K}, \tilde{p}_1, \\
(a) & : u_{\kappa+1}, \ldots, u_{K}, \tilde{p}_1,
\end{align*}
\]

Note that in the above ordering, regardless of \(i\), the first \(K - \kappa\) 

vectors are diagonalized. Thus, each greedy searching only 
occurs for the last \(\kappa\) vector choices, and consequently the 
construction of all \(a\) additional bases has equal complexity 
to the construction of the MRB.

B. Asymptotic performance analysis

In this subsection we carry out analysis on the LECR of the 
proposed multi-basis scheme in the high SNR regime. To 
simplify analysis, we assume that \(N, K, w\) are sufficiently large, 
thus we can reasonably ignore \((i)\) the issues of divisibility so that

\[K = \alpha \kappa, \quad w = \alpha f,\]

where \(\kappa\) is the number of the MRB extension bits (i.e., the 
most reliable bits from the redundancy part of the MRB), and 
\((ii)\) the issues of linear dependency (cf. [17], p. 455), so that the 
MRB is exactly composed of the \(K\) most reliable bits and the 
ith basis \(B_i\) is exactly composed by

\[u_1, \ldots, u_{K-\kappa}, u_{K-\kappa+1}, \ldots, u_{K}, \tilde{p}_1, \ldots, \tilde{p}_\kappa\]

(recall that \(\tilde{G} = [I_K, \tilde{P}]\)).

We observe that the proposed multi-basis order-\(w\) reprocessing 
scheme fails (to list the transmitted codeword as one of the 
candidate codewords) if at least \(w+1\) bit errors occur in all 
\(a+1\) bases. In an alternative understanding, a list error occurs 
when there exist at least \(w+1\) errors over all \(a+1\) combinations of 
a out of \(a+1\) regions indexed by \([i\kappa + 1, (i+1)\kappa], i = 0, 1, 2, \ldots, a\). This indicates that at least \(f+1\) 
errors occur at least in two regions out of \(a+1\) regions (otherwise, 
there are \(a\) regions, each of which contains at most \(f\) errors, 
and thus the basis composed of these \(a\) regions contains at 
most \(w\) errors). Apart from the one region that contains at 
least \(f+1\) errors, the remaining \(a\) regions contains at least 
\(w+1\) errors. Consequently, there are at least \(w+f+2\) errors 
over all \(a+1\) regions. Thus, the worst-case error event is 
determined by \(f\) bit errors in the least reliable positions in each 
of the regions \([i\kappa + 1, (i+1)\kappa], i = 0, 1, \ldots, a-2\), 
and \(f+1\) bit errors in the least reliable positions in each of the 
regions \([i\kappa + 1, (i+1)\kappa], i = a - 1, a\). To simplify the 
analysis, we consider a lower bound case where \(f\) errors occur 
in each of the regions \([i\kappa + 1, (i+1)\kappa], i = 0, 1, \ldots, a\). Thus, 
in following the methodology in [10], the LECR is determined 
by the optimization

\[
\text{Minimize} \quad D^2(x) = \sum_{i=1}^{w+f} x_i^2 + \sum_{j=1}^a (k-f)(2-x_{jf})^2 + (R-\kappa)(2-
\text{subject to} \quad x_1 \geq x_2 \geq \ldots \geq x_{w+f} \geq 1,
\]

(39)

where the subscript \(\text{"} jf \text{"}\) stands for the product of \(j \text{ and } f\).

By inspection, we determine that the optimization problem

\[
\text{Minimize} \quad D^2_2(x) = \sum_{i=1}^{w+f} x_i^2 + \sum_{j=1}^a (k-f)(2-x_{jf})^2,
\text{subject to} \quad x_{w+1} \geq x_{w+2} \geq \ldots \geq x_{w+f} \geq 1,
\]

is solved with \(x_1 = x_2 = \ldots = x_{w+f} = 2(k-f) = \frac{2(K-w)}{K}\) 
and results in

\[D^2_2 = \frac{4(K-w)}{K}w\]

Note that by our assumptions

\[\frac{2(K-w)}{K} \geq \frac{2(K-d_{\text{min}}/4)}{K} \geq \frac{2(K-R/4)}{K} \geq \frac{2(K-K/4)}{K} > 1.
\]

Similarly, the optimization problem

\[
\text{Minimize} \quad D^2_2(x) = \sum_{i=1}^{w+f} x_i^2 + (R-\kappa)(2-x_{w+f})^2
\text{subject to} \quad x_{w+1} \geq x_{w+2} \geq \ldots \geq x_{w+f} \geq 1,
\]

is solved with

\[x_{w+1} = x_{w+2} = \ldots = x_{w+f} = \max \left(1, \frac{2(R-\kappa)}{R-K+f} \right)\]

and results in

\[D^2_2 = \left\{ \begin{array}{ll}
\frac{R-K+f}{R-K+f} & , \text{if } \frac{2(R-\kappa)}{R-K+f} < 1, \\
\frac{R-K}{R-K+1} & , \text{otherwise.}
\end{array} \right. \]

(40)

We are interested in maximizing the radius \(D^2\) defined in 
(39) among all choices of \(0 \leq \kappa \leq R\), when given \(N, K, w\) 
(note that parameters \(a, f\) are functions of \(\kappa\)). To this end, we 
first consider how to maximize \(D^2_2\). For notational simplicity, 
let \(\xi = \frac{R}{K-f}\). We therefore simplify the condition for the first 
case of (40) to

\[\frac{2(R-\kappa)}{R-K+f} \leq 1 \quad \Rightarrow \quad \kappa \geq \frac{R}{1+\xi}.
\]

Consequently, we obtain

\[
\max_{\kappa \geq \frac{R}{1+\xi}} \{ D^2_2(\kappa) \} = R-\kappa + f \quad \left|_{\kappa = \frac{R}{1+\xi}} \right. = \frac{2R\xi}{1+\xi},
\]

(41)

Note that the expression for the second case in (40) is a convex 
function of \(\kappa\) and thus has a unique maximal point. To this 
end, we take its derivative with respect to \(\kappa\) and set it to zero:

\[
0 = (1-\xi)(\kappa^2 - R\kappa) - (2\kappa - R)((1-\xi)\kappa - R)
= -\kappa^2(1-\xi) + 2\kappa R - R^2.
\]

Solving, we obtain

\[\kappa = \frac{R}{1+\sqrt{\xi}},\]
which consequently results in the maximum radius 
\[
\frac{\max_{\kappa = \frac{R}{2 - \xi}} D_2^2(\kappa)}{\kappa = \frac{R}{1 + \sqrt{\xi}}} = \frac{4\xi R}{(1 + \sqrt{\xi})^2}.
\]
(42)

Combining (41) and (42), \(D_2^2\) achieves the maximum value
\[
\max_{\kappa > 0} \{D_2^2(\kappa)\} = \max \left(\frac{2R\xi}{1 + \sqrt{\xi}}, \frac{4\xi R}{(1 + \sqrt{\xi})^2}\right) = \frac{4\xi R}{(1 + \sqrt{\xi})^2}
\]
with the maximum point at \(\kappa = \frac{R}{1 + \sqrt{\xi}}\). We next consider the combination of \(D_1^2\) and \(D_2^2\). The case when
\[
\frac{2(R - \kappa)}{R - \kappa + \bar{f}} \leq \frac{2(\kappa - f)}{\kappa}
\]
requires, in conjunction with \(f = \frac{\bar{w}}{K}\kappa\), that
\[
\kappa \geq \frac{KR}{2K - \bar{w}} = \frac{R}{2 - \xi}.
\]
(43)

Note that \(D^2 = D_1^2 + D_2^2\). Since \(D_1^2\) is independent of the choice of \(\kappa\) and the maximal point \(\kappa = \frac{R}{1 + \sqrt{\xi}}\) is within the legitimate range defined by (43), \(D^2(\frac{R}{1 + \sqrt{\xi}})\) reaches the maximum value
\[
\max_{\kappa \geq \frac{R}{1 + \sqrt{\xi}}} \{D^2(\kappa)\} = D^2\left(\frac{R}{1 + \sqrt{\xi}}\right) = \frac{4\xi R}{(1 + \sqrt{\xi})^2} + 4K\xi(1 - \xi).
\]
(44)

On the other hand, when \(\kappa \leq \frac{R}{2 - \xi}\), \(D_1^2\) and \(D_2^2\) are combined into a single degree of freedom, that is \(D^2\) defined in (39) is solved with
\[
x_1 = x_2 = \ldots = x_{w + f} = \frac{N - w - \kappa}{N - \kappa + f},
\]
and has minimum value
\[
D^2(\kappa) = \frac{4(N - w - \kappa)(w + f)}{N - \kappa + f} = \frac{4\xi(N - \xi K - \kappa)(K + \kappa)}{N - (1 - \xi)\kappa}.
\]
(45)

Note that (45) is a convex function and thus has derivative equal to zero at the unique maximal point. To obtain the maximal point, we solve
\[
0 = (N - \xi K - \kappa)(K + \kappa)(1 - \xi) + (N - (1 - \xi)\kappa)(N - \xi K - \kappa)^2 = \kappa^2(1 - \xi) - 2\kappa N + (N^2 - 2\xi K N - (1 - \xi)K^2)
\]
and conclude that the maximal point is
\[
\kappa = \frac{N}{1 + \sqrt{\xi}} - \sqrt{\xi}K.
\]
(46)

In order to show that the above expression (46) is beyond the range \([0, \frac{R}{2 - \xi}]\), we proceed to establish a tight upper bound on \(\xi\) in light of the following established bound on the minimum distance (cf. [3], p. 394).

**Proposition 2:** For binary codes of fixed rate \(\frac{K}{N}\), the function \(d_{\text{min}}\) asymptotically satisfies
\[
\frac{d_{\text{min}}}{N} \leq 2\rho(1 - \rho)
\]
(47)

for any \(\rho\) satisfying \(\rho\log_2\frac{1}{\rho} + (1 - \rho)\log_2\frac{1}{1 - \rho} = 1 - \frac{K}{N}\).

Since \(\frac{K}{N} \geq 0.5\), we obtain \(\rho(1 - \rho) < 0.1\) and \(d_{\text{min}} < 0.2N\). Thus
\[
\xi = \frac{w}{K} \leq \frac{d_{\text{min}}}{4K} < \frac{N}{20K} \leq 0.1.
\]

Apparently, under the above range of \(\xi\), we have
\[
\frac{R}{2 - \xi} \leq \frac{N}{1 + \sqrt{\xi}} - \sqrt{\xi}K.
\]

Thus, (45) is a monotonic function within the defined range \([0, \frac{R}{2 - \xi}]\) and the maximum function value is achieved at the boundary point \(\kappa = \frac{R}{2 - \xi}\), i.e.,
\[
\max_{\kappa \leq \frac{R}{2 - \xi}} \{D(\kappa)\} \leq D\left(\frac{R}{2 - \xi}\right) = 4K\xi(1 - \xi) + 4R\xi(1 - \xi)\frac{1}{2 - \xi}.
\]
(48)

Finally, the overall maximum radius is obtained by choosing the larger value between (44) and (48). Note that \(D^2(\kappa)\) is a continuous function (though it cannot be expressed in a single formula); thus, the value \(D^2(\frac{R}{2 - \xi})\) achieved based on the decision region \([0, \frac{R}{2 - \xi}]\) is equal to the value achieved based on the decision region \([\frac{R}{2 - \xi}, \frac{R}{1 + \sqrt{\xi}}]\). Hence (44) has larger value than (48). The following theorem summarizes our analysis.

**Theorem 3:** The LECR \(D^2\) of the proposed multi-basis scheme is determined by the optimization problem defined in (39) and has the following solution:
\[
D^2(\kappa) = \begin{cases} 
4K\xi(1 - \xi) + R - (1 - \xi)\kappa, & \text{if } \kappa > \frac{R}{1 + \sqrt{\xi}}, \\
4K\xi(1 - \xi) + \frac{4\xi(R - \kappa)}{N - (1 - \xi)\kappa}, & \text{if } \frac{R}{1 + \sqrt{\xi}} \geq \kappa > \frac{R}{2 - \xi}, \\
\frac{4\xi(N - \xi K - \kappa)(K + \kappa)}{N - (1 - \xi)\kappa}, & \text{otherwise},
\end{cases}
\]
(49)

where \(R \triangleq N - K\) and \(\xi \triangleq \frac{w}{K}\). Moreover, \(D^2(\kappa)\) has a unique maximal point and achieves the maximum value of
\[
\max_{\kappa} \{D^2(\kappa)\} = \frac{4\xi R}{(1 + \sqrt{\xi})^2} + 4K\xi(1 - \xi)
\]
(50)

when
\[
\kappa = \frac{R}{1 + \sqrt{\xi}}.
\]
(51)

Consider soft-decision decoding of the \((1024, 512, 112)\) concatenated code. This code is obtained by concatenating a \((128, 73, 56)\) Reed-Solomon code with the \((8, 7, 2)\) binary parity code and inserting one additional information bit [4]. In order to achieve bounded-distance soft-decision decoding, i.e., \(D^2 \leq \sqrt{d_{\text{min}}}\), we need
\[
\frac{4(1024 - 512)(w + 1)}{1024 - 512 + w + 1} \geq 112
\]
for order-\(w\) reprocessing. Solving, we obtain \(w \geq 29\), which implies that order-29 reprocessing is necessary (and also sufficient) to achieve asymptotically optimal decoding. On the other hand, for the multi-basis scheme, we have
\[
\frac{4 \times 512\xi}{(1 + \sqrt{\xi})^2} + 4 \times 512\xi(1 - \xi) \geq 112.
\]

Solving, we obtain \(w \geq 18\). Therefore, multi-basis order-18 reprocessing suffices to achieve the maximum decoding radius. As can be easily seen, the resulting computational savings over single basis order reprocessing can be tremendous. Figure 2 depicts the squared LECR \(D^2\) as a function of \(\kappa\) by fixing \(w = 18\).
Fig. 2. The squared LECR of the (1024, 512, 112) function of $\kappa$ by fixing $w = 18$.

V. SIMULATIONS

We first briefly comment on the appropriateness of the second assumption that we used for the analysis of preprocessing rules in Section III.C, i.e., the assumption that a redundancy error vector $\mathbf{e}_2$ follows a uniform distribution over $[1, t]$ (or equivalently, $\Pr(\mathbf{e}_2[1, t]) = 2^{-t}$). Essentially, our interest lies in the mean number of recodings rather than the exact distribution, where the mean number of recodings for order-$w$ reprocessing using the first preprocessing rule was determined in (32). We simulate order-5 reprocessing on the (192, 96) concatenated RS code with $\theta = 9$, $t = 20$, and order-4 reprocessing on the (256, 147) extended BCH code with $\theta = 11$, $t = 28$, at various signal-to-noise-ratios. Table II shows results that are averaged over 5000 samples. As we can see, the estimated number of TEPs obtained from (32) is close to the actual estimates. Thus, (32) is a good approximation of the mean number of recodings and the uniform distribution is a plausible assumption.

<table>
<thead>
<tr>
<th>Code/SNR</th>
<th>Estimate</th>
<th>0.5dB</th>
<th>1.0dB</th>
<th>1.5dB</th>
<th>2.0dB</th>
</tr>
</thead>
<tbody>
<tr>
<td>(192, 96)</td>
<td>$4.39 \times 10^9$</td>
<td>$7.01 \times 10^9$</td>
<td>$7.15 \times 10^9$</td>
<td>$7.25 \times 10^9$</td>
<td>$7.39 \times 10^9$</td>
</tr>
<tr>
<td>(256, 147)</td>
<td>$1.27 \times 10^9$</td>
<td>$1.32 \times 10^9$</td>
<td>$1.32 \times 10^9$</td>
<td>$1.32 \times 10^9$</td>
<td>$1.32 \times 10^9$</td>
</tr>
</tbody>
</table>

TABLE II
COMPARISONS BETWEEN ESTIMATED AND SIMULATED AVERAGE NUMBER OF RECodings AT VARIOUS SNRs.

We proceed to consider the soft-decision decoding of the (256, 147, 30) extended BCH code over the AWGN channel. Note that for this code there exists an efficient algebraic bounded-distance decoder (cf. [3]) that successfully corrects up to 14 bit errors, thus the Chase-2 algorithm is applicable. This algorithm systematically flips the 15 least reliable bits, each time applying algebraic decoding to correct up to 14 bit errors among the remaining bits. Clearly, the Chase-2 algorithm requires $2^{15} = 32768$ times algebraic decodings. Its performance in terms of block-error-rate is plotted in Figure 3 and can be seen to be far worse than ML performance, although it is well-known that this algorithm is asymptotically optimal in the high SNR regime. If, on the other hand, we want to see how many least reliable bits are needed to be systematically flipped in order to achieve near-optimal performance at a block-error-rate of $10^{-5}$, we can compute the list decoding error probability through a two-fold integration (as addressed in the Appendix) and require the value of this probability to be roughly a quarter of the ML lower bound value (the ML lower bound is obtained by tracking the fraction of cases where the codeword picked by the proposed two-basis (2-B) order-4 reprocessing turns out to be more likely than the transmitted word). Through numerical computation, this performance qualification is achieved by choosing the number of bits to be 60, i.e., in order to achieve near optimal performance at a block-error-rate of $10^{-5}$, the computational cost of the Chase-2 algorithm is of the order of $2^{60} = 10^{18}$ algebraic decodings.

For comparison purposes, we simulated the proposed 2-B order-$w$ ($w = 3, 4$) reprocessing with the number of STEPs set to $\sum_{i=0}^{w} \binom{K}{i}$ (i.e., $5.3 \times 10^6$ and $1.9 \times 10^7$ for $w = 3, 4$ respectively). When $w = 3$, the other parameters are chosen to be $\theta = 12$, $t = 32$, $\tau = 19$, $\kappa = 27$. Figure 3 shows that the proposed 2-B order-3 reprocessing outperforms the original order-5 reprocessing. The performance data for the latter algorithm is computed through the two-fold integration.
determined by the probability of the event that their gap is invisible. The exact values are shown in Table VI. As can be seen, the maximum number of candidate codewords for the proposed scheme is less than that of order-4 reprocessing (\(\sum_{i=0}^{4} \binom{96}{i} = 3.5 \times 10^6\)), while the proposed scheme clearly outperforms order-5 reprocessing as shown in Figure 4. Our simulation results also indicate that the proposed scheme evidently outperforms the “Box and Match” method, which searches over a maximum of \(4.5 \times 10^6\) candidate codewords (in addition to requiring a large memory to store \(\sum_{i=0}^{4} \binom{96}{i}\) codewords in \(2^{24} = 2 \times 10^6\) boxes) to achieve near-optimal performance [20].

### VI. Conclusions and Discussions

In this paper, we have investigated the soft-decision decoding of binary linear block codes. We incorporated two preprocessing rules into the new version of order-\(w\) reprocessing. The first rule discards a test error pattern (TEP) if the weight of its partial coset, composed of the most reliable (information) basis (MRB) and the \(t\) most reliable bits in the redundancy part, is above a pre-determined threshold \(\theta\). The second preprocessing rule discards TEPs with weight \(w + 2\) which exhibit more than one bit error over the \(\tau\) most reliable bits in the redundancy part. We showed that by appropriately choosing the parameters...
but achieves asymptotically the performance of the original
θ
practical SNRs the proposed algorithm is more ef-
measure at high SNRs. Our extensive simulations show that at
the list error-correction radius, a commonly used performance
showed that the use of multiple bases signi-

not amenable to VLSI design due to the tremendous hardware
near-ML performance for practical SNRs) compared to the
shown in this paper to be superior (in terms of achieving
the original order

will make it even more attractive for off-line applications, such
Further study of the complexity of software implementations
approaches and/or the possible existence of special classes of
functions:

Lemma 3: Let \( f(\cdot) \) be smooth and bounded in \([x, y]\) \( x > 0 \)
and independent of \( N_0 \). Furthermore, assume \( f(x) \neq 0 \). Then,

\[
\int_x^y f(u) e^{-u^2/N_0} du = \left( \frac{f(x)}{2x} + O(1) \right) N_0 e^{-x^2/N_0}
\]
as \( N_0 \to 0 \).

Lemma 4: If \( n > w^* \triangleq w + 1 \), then

\[
q_{n, w, 0} = \left( \pi^{1/2} 4^{-n-w} n^{-w^*} n^{n+w-1/2} + O(1) \right) N_0^{n+1/2} e^{-w^*}
\]
as \( N_0 \to 0 \).

VII. APPENDIX: PROOF OF THEOREM 1

We will use the notation \( o(1) \) to represent a term which
diminishes as \( N_0 \to 0 \), i.e., \( \lim_{N_0 \to 0} o(1) = 0 \) and \( O(1) \)
to mean \( \lim_{N_0 \to 0} O(1) = a \), where \( a \) denotes some nonzero
constant. We first prove a series of lemmas. Lemmas 1–3 are
basic calculus results that can also be found in [2], but we

<table>
<thead>
<tr>
<th>SNR (dB)</th>
<th>0.0</th>
<th>0.5</th>
<th>1.0</th>
<th>1.5</th>
<th>2.0</th>
<th>2.5</th>
<th>3.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-B order-4</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
</tr>
<tr>
<td>Lower bound</td>
<td>994</td>
<td>988</td>
<td>985</td>
<td>975</td>
<td>970</td>
<td>948</td>
<td>915</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SNR (dB)</th>
<th>0.0</th>
<th>0.5</th>
<th>1.0</th>
<th>1.5</th>
<th>2.0</th>
<th>2.5</th>
<th>3.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>STEPs (max)</td>
<td>10^6</td>
<td>10^6</td>
<td>10^6</td>
<td>10^6</td>
<td>10^6</td>
<td>10^6</td>
<td>10^6</td>
</tr>
<tr>
<td>STEPs (mean)</td>
<td>684,995</td>
<td>460,568</td>
<td>234,071</td>
<td>81,280</td>
<td>18,861</td>
<td>2,757</td>
<td>293</td>
</tr>
<tr>
<td>Codewords (max)</td>
<td>567,191</td>
<td>575,345</td>
<td>654,822</td>
<td>577,198</td>
<td>741,274</td>
<td>1,986,132</td>
<td></td>
</tr>
<tr>
<td>Codewords (mean)</td>
<td>122,925</td>
<td>86,129</td>
<td>46,880</td>
<td>18,808</td>
<td>5,087</td>
<td>1,046</td>
<td>211</td>
</tr>
</tbody>
</table>

TABLE VI

Recorded numbers of decoding errors and ML lower-bound
decoding errors by simulating the proposed 2-B order-4
reprocessing for the (192, 96, 28) concatenated Reed-Solomon
code.

<table>
<thead>
<tr>
<th>SNR (dB)</th>
<th>0.0</th>
<th>0.5</th>
<th>1.0</th>
<th>1.5</th>
<th>2.0</th>
<th>2.5</th>
<th>3.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Codewords (max)</td>
<td>567,191</td>
<td>575,345</td>
<td>654,822</td>
<td>577,198</td>
<td>741,274</td>
<td>1,986,132</td>
<td></td>
</tr>
<tr>
<td>Codewords (mean)</td>
<td>122,925</td>
<td>86,129</td>
<td>46,880</td>
<td>18,808</td>
<td>5,087</td>
<td>1,046</td>
<td>211</td>
</tr>
</tbody>
</table>

TABLE VII

Statistics about the computational cost of the proposed 2-B
order-4 reprocessing on decoding the (192, 96, 28)
concatenated Reed-Solomon code.

t, \( \theta \) and \( \tau \) the proposed order−\( w \) reprocessing with preprocessing
requires comparable complexity to order−\( w \) reprocessing
but achieves asymptotically the performance of the original
order−\( (w+2) \) reprocessing.

We employed iterative redecoding on a second basis to com-
plement the inefficiency of the MRB for practical SNRs,
and extended this approach systematically to a multi-basis
order−\( w \) reprocessing scheme in the high SNR regime. We
showed that the use of multiple bases significantly enlarges
the list error-correction radius, a commonly used performance
measure at high SNRs. Our extensive simulations show that at
practical SNRs the proposed algorithm is more efficient than
the original order−\( w \) reprocessing algorithm.

Before closing, we comment on the applicability of the
methodology of iterative redecoding. Iterative redecoding has
been shown in this paper to be superior (in terms of achieving
near-ML performance for practical SNRs) compared to the
methodology of iterative bounded-distance decoding (for alge-
braic codes of interest). The major concern regarding iterative
redecoding approaches, however, lies in the fact that they are
not amenable to VLSI design due to the tremendous hardware
complexity associated with the construction of the MRB,
including the sorting of the generator matrix (alternatively
the parity check matrix) and the on-the-fly binary Gaussian
elimination process (for some discussion on the complexity of
VLSI implementations for binary Gaussian elimination refer
to [6], [15]). Thus, it is imperative that future work investigates
the complexity of VLSI implementations for iterative redecoding
approaches and/or the possible existence of special classes of
codes for which iterative redecoding may be more attractive.
Further study of the complexity of software implementations
of iterative redecoding approaches is also important because it
will make it even more attractive for off-line applications, such
as deep-space communications.
Proof: Note that there exists a small positive number \( \xi \) such that
\[
\min\{w^* + n(1 - \xi)^2, \ w^*(2 - \xi)^2\} > \frac{4nw^*}{n + w^*},
\]
(57)
where \( \sigma \) is a number satisfying
\[
0 < \sigma < \min\{\frac{2n}{n + w^*} - (1 + \xi), 2 - \xi - \frac{2n}{n + w^*}\}.
\]
By treating each integration separately
\[
0 \leq \int_{1+\xi}^{2-\xi} x^{-w}e^{-(n+w^*)(x-2n/(n+w^*))^2/N_0} dx
\]
(62)
\[
0 \leq \int_{2n/(n+w^*)+\sigma}^{2n/(n+w^*)-\sigma} x^{-w}e^{-(n+w^*)(x-2n/(n+w^*))^2/N_0} dx
\]
(63)
\[
\leq \int_{1+\xi}^{2-\xi} x^{-w}e^{-(n+w^*)x^2/N_0} dx
\]
(64)
Note that, as \( N_0 \to 0 \),
\[
\int_{-\sigma}^{\infty} e^{-(n+w^*)x^2/N_0} dx = \frac{1 + o(1)}{2\sigma} \frac{N_0}{n + w^*} e^{-(n+w^*)^2/N_0} = o(1)N_0^{1/2}.
\]
Thus, we have
\[
\int_{-\sigma}^{\infty} e^{-(n+w^*)x^2/N_0} dx = 2 \left( \sqrt{\pi} \left( \frac{N_0}{n + w^*} \right)^{1/2} - o(1) \right) N_0^{1/2}
\]
(65)
as \( N_0 \to 0 \). It follows from (62)–(65) and the arbitrariness of \( \sigma \) that
\[
\int_{1+\xi}^{2-\xi} x^{-w}e^{-(n+w^*)(x-2n/(n+w^*))^2/N_0} dx
\]
(66)
Combining (61) and (66) results in
\[
\int_{1+\xi}^{2-\xi} x^{-w}e^{-(n+w^*)x^2/N_0} dx
\]
(67)
as \( N_0 \to 0 \).

We next show that the dominant term of the integration (55) is contained in the second part, i.e., the integration over \([1 + \xi, 2 - \xi]\). We proceed to determine this dominant term.

Plugging in (53), we obtain
\[
\int_{1+\xi}^{2-\xi} \left( \int_{2-x}^{2-\xi} e^{-u^2/N_0 du} \right)^n (2x)^{-w}e^{-w^*x^2/N_0} dx
\]
(68)
\[
= (1 + o(1)) \int_{1+\xi}^{2-\xi} \left( \frac{1}{2(2-x)} N_0 e^{-(2-x)^2/N_0} \right)^n (2x)^{-w}e^{-w^*x^2/N_0} dx
\]
(69)
\[
= (2^{-n+w} + o(1)) N_0^n e^{-(2n+w^*)/2n(n+w^*)} \int_{1+\xi}^{2-\xi} x^{-w}e^{-(n+w^*)(x-2n/(n+w^*))^2/N_0} dx
\]
(70)
We further decompose the interval \([1 + \xi, 2 - \xi]\) into three disjoint intervals:
\[
[1+\xi, 2-\xi] = \left[ 1 + \xi, \frac{2n}{n + w^*} - \sigma \right] \cup \left[ \frac{2n}{n + w^*} - \sigma, \frac{2n}{n + w^*} + \sigma \right] \cup \left[ \frac{2n}{n + w^*} + \sigma, 2 - \xi \right]
\]
(71)
Condition (57) indicates that (59), (60) are negligible terms compared to (64), which concludes the lemma. ☐
The following lemma is an extension of Lemma 4. The proof is analogous to the proof of Lemma 4 and is omitted.

**Lemma 5:** If \( n > w^* \triangleq w + 1 \) and \( k > 0 \), then
\[
q_{n,w,k} = (\pi^{1/2} (n + 2w + k) - w)^{n - 1} (n + w)^{n + w - 1/2} + o(1)
\]
as \( N_0 \to 0 \).

The following lemma accounts for the opposite case of Lemma 4, i.e., the case where \( n \leq w^* \).

**Lemma 6:** If \( n \leq w^* \triangleq w + 1 \), then
\[
q_{n,w,0} = \left( \frac{2 - (n + w)}{n + w^*} + o(1) \right) N_0^{n+1} e^{-(n+w^*)/N_0}
\]
as \( N_0 \to 0 \).

**Proof:** We first show that the integration over \([1.5, \infty)\) is a negligible term. Note that as \( N_0 \to 0 \)
\[
\int_{1.5}^{\infty} \left( \int_{2-x}^{1} e^{-u^2/N_0} du \right)^n (2x)^{-w} e^{-w^* x^2/N_0} dx
\]
\[
\leq \int_{1.5}^{\infty} \left( \int_{-\infty}^{2-x} e^{-u^2/N_0} du \right)^n (2x)^{-w} e^{-w^* x^2/N_0} dx
\]
\[
= (\pi N_0)^{n/2} (2x) - w e^{-w^* x^2/N_0} dx
\]
\[
= (\pi N_0)^{n/2} \left( 3 - w^* + o(1) \right) N_0 e^{-2.25 w^*/N_0}
\]
\[
= o(1) \cdot e^{-(n+w^*)/N_0},
\]
where the second to last equality is due to Lemma 3, and the last one is due to the condition \( n \leq w^* \).

We now turn to the remaining term of \( q_{n,w,0} \) which integrates within \([1, \ 1.5]\). Note that Lemma 2 is not applicable to the inner integration \( \int_{2-x}^{1} e^{-u^2/N_0} du \) since \( 2-x \) can be arbitrarily close to 1. To this end, we first derive the asymptotic formula for the sub-term which integrates within \([1 + \sigma, \ 1.5]\), where \( \sigma < 0.5 \) is an arbitrarily positive number, and then take the limit as \( \sigma \to 0 \). We have
\[
\int_{1+\sigma}^{1.5} \left( \int_{2-x}^{1} e^{-u^2/N_0} du \right)^n (2x)^{-w} e^{-w^* x^2/N_0} dx
\]
\[
= \int_{1+\sigma}^{1.5} (2(2-x))^{-n} + o(1) N_0^{-n} e^{-n(2-x)^2/N_0} (2x)^{-w} e^{-w^* x^2/N_0} dx
\]
\[
= N_0^n e^{-n(2-x)^2/N_0} \left( 2 - (n + w) \right) \left( 1 - \sigma \right)^{-n} (1 + \sigma)^{-w} + o(1) N_0 e^{-(n+w^*)/N_0}
\]
\[
= \frac{2 - (n+w)}{n + w^*} \left( 1 - \sigma \right)^{-n} (1 + \sigma)^{-w} + o(1) N_0^{n+1} e^{- \frac{(n+w^*)}{N_0}}
\]
as \( N_0 \to 0 \), where the first equality is due to Lemma 2 and the third equality is due to Lemma 3. If we define
\[
\Delta(\sigma, N_0) \triangleq \int_{1+\sigma}^{1.5} \left( \int_{2-x}^{1} e^{-u^2/N_0} du \right)^n (2x)^{-w} e^{-w^* x^2/N_0} dx
\]
then (70) and (71) imply
\[
\lim_{N_0 \to 0} \Delta(\sigma, N_0) = 1, \quad 0 < \sigma < 0.5,
\]
which further implies
\[
\lim_{\sigma \to 0} \lim_{N_0 \to 0} \Delta(\sigma, N_0) = 1.
\]
Note that \( \Delta(\sigma, N_0) \) by definition (72) is continuous within \([0, a] \times (0, b]\), where \( a, b < 0.5 \). If we define
\[
\Delta(\sigma, 0) = 1, \quad 0 \leq \sigma < 0.5,
\]
then, as indicated by (73), \( \Delta(\sigma, N_0) \) is continuous within the closed region \([0, a] \times [0, b]\), where \( a, b < 0.5 \), and thus is uniformly continuous within the neighborhood \([0, a] \times [0, b]\).

This property enables us to exchange two limits on \( \Delta \), i.e.,
\[
\lim_{N_0 \to 0} \lim_{\sigma \to 0} \Delta(\sigma, N_0) = \lim_{\sigma \to 0} \lim_{N_0 \to 0} \Delta(\sigma, N_0) = 1.
\]
We thus conclude the lemma.

The following lemma is an extension of Lemma 6. The proof is analogous to that of Lemma 6 and is omitted.

**Lemma 7:** If \( n \leq w^* \triangleq w + 1 \), and \( k > 0 \), then
\[
q_{n,w,k} = \left( \frac{2 - (n+k+w)}{n + w^*} + o(1) \right) N_0^{n+k+1} e^{-(n+k+w^*)/N_0}
\]
as \( N_0 \to 0 \).
We first establish an explicit formula for the probability that more than $w$ bit errors occurred among the $N-S$ most reliable bits, following the developments in [1], [12]. Let $f^c_\alpha(x)$ and $f^e_\alpha(x)$ be the density functions associated with a reliability value $\alpha$, for a correct hard decision and an erroneous hard decision at position $i$. It follows that

$$f^c_\alpha(x) = \frac{\hat{q}(x+1)}{Q(1)} u(x),$$

$$f^e_\alpha(x) = \frac{\hat{q}(x-1)}{1-Q(1)} u(x),$$

where $u(x)$ is the standard step function and

$$\hat{q}(x) = (\pi N_0)^{-1/2} \exp(-x^2/N_0), \quad \hat{Q}(x) = \int^\infty_x \hat{q}(z)dz.$$

The corresponding cumulative distribution functions are given respectively by

$$F^c_\alpha(x) = \frac{Q(1)-\hat{Q}(x+1)}{Q(1)} u(x),$$

$$F^e_\alpha(x) = \frac{1 - \hat{Q}(x) - \hat{Q}(x-1)}{1-Q(1)} u(x).$$

For $1 \leq j \leq i$, let $\beta_j(i)$ represent the $j$-th ordered reliability value among $i$ hard-decision errors in a received sequence $\mathbf{r}$, so that

$$\beta_1(i) \geq \beta_2(i) \geq \ldots \geq \beta_i(i).$$

Similarly, the remaining $N-i$ reliability values corresponding to correct hard decisions are also reored in decreasing order, denoted by

$$\gamma_1(N-i) \geq \gamma_2(N-i) \geq \ldots \geq \gamma_{N-i}(N-i).$$

The density functions of $\beta_j(i)$ and $\gamma_j(N-i)$ are then given by

$$f_{\beta_j(i)}(x) = \frac{i!}{(j-1)! (i-j)!} [1 - F^c_\alpha(x)]^{j-1} f^c_\alpha(x) f^e_\alpha(x)^{i-j},$$

$$f_{\gamma_j(N-i)}(x) = \frac{(N-i)!}{(i-1)! (N-i-i)!} [1 - F^c_\alpha(x)]^{i-1} f^c_\alpha(x) f^e_\alpha(x)^{N-i-i},$$

respectively.

Consequently, the probability of the event $\beta_j(i) \geq \gamma_j(N-i)$ is given by

$$\Pr(\beta_j(i) \geq \gamma_j(N-i)) = \int_0^\infty f_{\beta_j(i)}(x) \int_0^x f_{\gamma_j(N-i)}(z)dz dx,$$

and the error probability that more than $w$ bit errors occur among the $N-S$ most reliable bits is determined by

$$P_{S,w} = \sum_{i=w^*}^{S+w} \Gamma_i \Pr(\beta_{w^*}(i) \geq \gamma_{N-S-w}(N-i)) + \sum_{i=S+w^*}^{N} \Gamma_i,$$

where

$$w^* \triangleq w + 1, \quad \Gamma_i \triangleq \binom{N}{i} (Q(1))^{i} (1 - Q(1))^{N-i}.$$

The above formula (76) was derived in [1], [12]. We proceed to explore its asymptotic behavior as $N_0 \to 0$. In light of Lemma 1, we have,

$$\hat{Q}(1) = \left( \frac{1}{2\sqrt{\pi}} + o(1) \right) N_0^{1/2} e^{-1/N_0}$$

as $N_0 \to 0$. Thus,

$$\Gamma_i = \binom{N}{i} \left( \left( \frac{1}{2\sqrt{\pi}} + o(1) \right) N_0^{1/2} e^{-1/N_0} \right)^i (1 - o(1))^{N-i}$$

$$= \binom{N}{i} 2^{-i} \pi^{-i/2} + o(1) N_0^{i/2} e^{-i/N_0}.$$ (78)

Note that the inner integration in (75) can be explicitly expressed as

$$\int_0^x f_{\gamma_{N-S-w}(N-i)}(u) du$$

$$= \int_0^x \frac{(N-i)!}{(N-S-w^!)(S+w-i)!} \left[ 1 - F^c_\alpha(u) \right]^{N-S-w^*} f^c_\alpha(u) f^e_\alpha(u)^{S+w-i} du$$

$$= \int_0^x \frac{(N-i)!}{(N-S-w^!)(S+w-i)!} \left[ 1 - F^c_\alpha(u) \right]^{N-S-w^*} F^c_\alpha(u) f^e_\alpha(u)^{S+w-i} du$$

$$= \frac{(N-i)!}{(N-S-w^!)(S+w-i)!} \sum_{k=0}^{N-S-w^*} (-1)^k \binom{N-S-w^*}{k} (N-S-w^*)^k (S+w-i)^k$$

$$= \frac{(N-i)!}{(N-S-w^!)(S+w-i)!} \sum_{k=0}^{N-S-w^*} (-1)^k \binom{N-S-w^*}{k} (N-S-w^*)^k (S+w-i)^k (\int_0^x f_{\gamma_{N-S-w}(N-i)}(u) du)$$

$$= \left( \frac{1}{2\sqrt{\pi}} + o(1) \right) N_0^{1/2} e^{-1/N_0}.$$
Note also that
\[ \Gamma_i f_{\beta_{w^*}(i)}(x) = \binom{N}{i} \tilde{Q}(1)^i \left( 1 - \tilde{Q}(1) \right)^{N-i} \left( \frac{\tilde{Q}(x+1)}{\tilde{Q}(x+1)} \right)^w \frac{\tilde{q}(x)}{\tilde{q}(x+1)} \frac{p_{\tilde{w^*},k} \tilde{Q}(x+1)}{\tilde{Q}(1)} \right)^i \]  
Thus, the terms \( p_{w^*,k} \) with \( k > w^* - S \) are again negligible.

We next tackle the case \( i > w^* \). It suffices to consider only \( \Gamma_i \frac{\tilde{q}(x)}{\tilde{q}(x+1)} \frac{p_{\tilde{w^*},k} \tilde{Q}(x+1)}{\tilde{Q}(1)} \). This is because, when \( i > S + w^* \),

Substituting (79) and (80) into (75) yields

\[ \Gamma_i \frac{\tilde{q}(x)}{\tilde{q}(x+1)} \frac{p_{\tilde{w^*},k} \tilde{Q}(x+1)}{\tilde{Q}(1)} = \frac{(N-w^*)^{i-1}}{w^*} \frac{p_{\tilde{w^*},k} \tilde{Q}(x+1)}{\tilde{Q}(1)} \]  

We next simplify \( p_{i,k} \) as \( N_0 \to 0 \), utilizing Lemma 2:

\[ p_{i,k} = (\pi N_0)^{-(n+i)/2} \int_1^\infty \int_2 e^{-x^2/2N_0} \left( \int_1^\infty e^{-y^2/2N_0} dy \right)^n \right)^i \frac{e^{-x^2/2N_0}}{2x} N_0 e^{-x^2/2N_0} \]  

where \( n = S + w^* - i + k \).

**Case 1:** \( S \leq w^* \).

When \( i = w^* \) and \( k = 0 \), Lemma 6 immediately indicates

\[ p_{w^*,0} = \left( \pi(S+w^*)^2 \frac{2^{-2(S+w^*)}}{S+w^*} + o(1) \right) N_0^{-(S+w^*)/2} e^{-(S+w^*)/N_0} \]  

We next show that for the case \( i = w^* \), \( k > 0 \)

\[ p_{w^*,k} = o(1) \cdot p_{w^*,0} \] (84)

When \( i = w^* \) and \( 0 < k \leq w^* - S \), Lemma 6 indicates that the exponent in \( e \) of \( p_{w^*,k} \) is \( -\frac{S+k+w^*}{N_0} \). Thus all corresponding terms \( p_{w^*,k} \) \( 0 < k \leq w^* - S \) are negligible compared to \( p_{w^*,0} \). When \( i = w^* \) and \( k > w^* - S \), Lemma 4 indicates that the exponent of \( p_{w^*,k} \) is \( -\frac{4(S+k)w^*}{(S+k+w^*)N_0} \). Noting that \( \frac{4(S+k)w^*}{S+k+w^*} > \frac{4(S+k)w^*}{2w^*} = 2(S+k) > S+w^* \), we arrive at the conclusion

\[ P_{S,w} = \sum_{i=w^*}^{S+w} \left( \binom{N}{i} \frac{(S+w^*)}{S+w^*} \right) \frac{2^{-2(S+w)}}{S+w^*} + o(1) \right) N_0^{-(S+w^*)/2} e^{-(S+w^*)/N_0} \]  

as \( N_0 \to 0 \).

**Case 2:** \( S > w^* \).

Lemmas 5 and 7 indicate that

\[ p_{i,k} = o(1) \cdot p_{i,0}, \quad k > 0 \] (90)
Hence, we focus on the case $k = 0$. Lemma 4 reveals that, when $i = w^*$ and $k = 0$,

$$ p_{w^*, 0} = \left( \pi^{-(S+w)/2 - S-i - S} S - w^* - S (S + w^*)^{S+w-1/2} + o(1) \right) $$

\[(91)\]

When $S > i > w^*$, Lemma 6 is invoked regarding $p_{i,0}$. Thus its (dominant) exponent coefficient is expressed by

$$ g(i) = i - w^* + \frac{4(S - i + w^*)w^*}{S - i + 2w^*}, \quad w^* \leq i \leq S. $$

We proceed to characterize it. Taking the derivative, we obtain

$$ g'(i) = 1 - \frac{4w^*}{(S - i + 2w^*)}. $$

Clearly, $g(i)$ is a concave function and achieves its minimum value at boundary points $i = w^*$ or $i = S$. Furthermore, the condition $S > w^*$ indicates $g(w^*) < g(S)$. Thus, $g(i)$ achieves the minimum value $\frac{4Sw^*}{S+w^*}$ at $i = w^*$, i.e.,

$$ p_{i,0} = o(1) \cdot p_{w^*, 0}, \quad w^* < i < S. \quad (92) $$

When $S \leq i < S + w^*$, Lemma 7 is applicable to $p_{i,0}$. Thus, its dominant exponent coefficient is expressed by

$$ [(S - (i - w^*)) + w^*] + (i - w^*) = S + w^* > \frac{4Sw^*}{S+w^*}, $$

which indicates that

$$ p_{i,0} = o(1) \cdot p_{w^*, 0}, \quad S \leq i < S + w^*. \quad (93) $$

When $i \geq S + w^*$, we have similarly

$$ \Gamma_i = \left( \binom{N}{i} 2^{-i} \pi^{-i/2} + o(1) \right) N_0^{i/2} e^{-i/N_0} = o(1) p_{w^*, 0}. $$

\[(94)\]

Summarizing over (76), (81), (90)–(94), we conclude the second part of the theorem

$$ P_{S, w} = (1 + o(1)) \left( \binom{N}{w^*} \binom{N - w^*}{S} p_{w^*, 0} \right) $$

\[= \left( \binom{N}{w^*} \binom{N - w^*}{S} \pi^{-S+w} 4^{-(S+w)} S - w^* - S (S + w^*)^{S+w-1/2} + o(1) \right) N_0^{-S+w} e^{-4S+w^{*}/N_0}. \]