Maximum-Likelihood Priority-First Search Decodable Codes for Combined Channel Estimation and Error Correction

Chia-Lung Wu, *Student Member, IEEE*, Po-Ning Chen, *Senior Member, IEEE*, Yunghsiang S. Han, *Senior Member, IEEE*, and Ming-Hsin Kuo

***Abstract*—The coding technique that combines channel estimation and error correction has received attention recently, and has been regarded as a promising approach to counter the effects of multipath fading. It has been shown by simulation that a proper code design that jointly considers channel estimation can improve the system performance subject to a fixed code rate as compared to a conventional system which performs channel estimation and error correction separately. Nevertheless, the major obstacle that prevents the practice of such coding technique is that the existing codes are mostly searched by computers, and subsequently exhibit no apparent structure for efficient decoding. Hence, the operation-intensive exhaustive search becomes the only decoding option, and the decoding complexity increases dramatically with codeword length. In this paper, a systematic construction is derived for a class of structured codes that support joint channel estimation and error correction. It is confirmed by simulation that these codes have comparable performance to the best simulated-annealing-based computer-searched codes. Moreover, the systematically constructed codes can now be maximum-likelihoodly decoded with respect to the unknown-channel criterion in terms of a newly derived recursive metric for use by the priority-first search decoding algorithm. Thus, the decoding complexity is significantly reduced as compared with that of an exhaustive decoder.**

***Index Terms*—Channel coding, fading channels, multipath channels, frequency-selective fading, maximum likelihood decoding, sequential decoding.**

I. Introduction

CURRENTLY, a typical receiver in a wireless communication system performs channel estimation and data estimation separately. The former task estimates channel characteristics based on a known training sequence or pilot, while the latter uses these characteristics to estimate the transmitted coded data.

Manuscript received December 12, 2007; revised December 13, 2008. Current version published August 19, 2009. This work was supported by the NSC of Taiwan, R.O.C., by Grant NSC 95-2221-E-009-054-MY3. The material in this paper was presented in part at the International Symposium on Information Theory and Its Applications, Auckland, New Zealand, December 2008.

C.-L. Wu and P.-N. Chen are with the Department of Communication Engineering, National Chiao-Tung Universtiy, Taiwan, R.O.C. (e-mail: clwu@banyan.cm.nctu.edu.tw; qponing@mail.nctu.edu.tw).

Y. S. Han is with the Graduate Institute of Communication Engineering, National Taipei University, 237 Taiwan, R.O.C. He is also with the Department of Computer Science and Information Engineering, National Chi Nan University, Taiwan, R.O.C. (e-mail: yshan@mail.ntpu.edu.tw).

M.-H. Kuo is with Pagetron Corp., Taipei, Taiwan, R.O.C. (e-mail: shin30@gmail.com).

Communicated by H.-A. Loeliger, Associate Editor for Coding Techniques.

Color versions of Figures 1–6 in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIT.2009.2025548

Recent research results [3], [6], [13], [14] have confirmed that better system performance can be obtained by jointly performing channel and data estimation, as compared to a typical system that performs these tasks separately. In 1994, Seshadri [13] proposed a blind maximum-likelihood sequence estimator (MLSE) that performs the two tasks simultaneously. Skoglund *et al.* [14] later provided a milestone evidence that a code design that jointly considers channel estimation and error correction is able to counter multipath block fading more efficiently than the approach with a separate error-correcting code and channel estimation scheme. They also applied the same idea to a multiple-input multiple-output (MIMO) system as described in a subsequent publication [6]. In short, Skoglund *et al.*, by computer search, identified nonlinear codes that support joint channel estimation and error correction in a multipath block fading channel. Through simulations, they found that a communication system using these nonlinear codes can outperform a typical communication system with perfect channel estimation by 2 dB. Their results hint that a single, perhaps nonlinear, code may improve the transmission rate in a highly mobile environment in which traditional channel estimation becomes technically infeasible. A similar idea was also proposed by [3], and the authors actually named such codes *training codes*.

One of the drawbacks of these joint estimation codes found by computer search is that they lack a systematic structure, and can therefore be decoded only by an operation-intensive exhaustive search. This naturally leads to the research query of how to construct an efficiently decodable code that supports joint channel estimation and error correction.

In this paper, this query was resolved first by discovering that regardless of the fading statistics, the codeword that maximizes the system signal-to-noise ratio (SNR) must be orthogonal to the delayed version of itself. We termed this property *self-orthogonality*. Second, we found that the code that consists of properly chosen self-orthogonal codewords has a performance comparable to that of the simulated-annealing-based computer-searched code. Because the maximum-likelihood metrics for self-orthogonal codewords can be equivalently transformed into a recursively formulated metric, it is finally shown that these structured codes can be maximum-likelihoodly decoded by the priority-first search algorithm [2], [7], [9], [12], resulting in a decoding complexity significantly smaller than that required by exhaustive decoding.

The paper is organized as follows. Section II describes the system model, followed by the technical background required for this work. Section III establishes the self-orthogonal codeword-selection condition that optimizes the system SNR regardless of the fading statistics, and then uses it to construct codes for joint channel and data estimation. The recursive maximum-likelihood decoding metrics for the constructed codes are derived in Section IV. Simulations are summarized and discussed in Section V. Section VI concludes the paper.

In this work, superscripts "H" and "T" are specifically reserved for the matrix operations of Hermitian transpose and transpose, respectively [8].

II. Background

A. System Model and Maximum-Likelihood Decoding Criterion

Suppose a codeword $\boldsymbol{b} = [b_1, \ldots, b_N]^T$ of an (N, K) code $\mathcal{C}$ is transmitted over a block fading (specifically, quasi-static fading) channel of memory order $P-1$, where each $b_j \in \{-1, +1\}$. Denote the channel coefficients by $\boldsymbol{h} = [h_1, \ldots, h_P]^T$, and assume that they are *constant* within a coding block of length $L = N + P - 1$. By letting the codeword matrix be

$$\mathbb{B} \triangleq \begin{bmatrix} b_1 & 0 & \cdots & 0 \\ \vdots & b_1 & \ddots & \vdots \\ b_N & \vdots & \ddots & 0 \\ 0 & b_N & \ddots & b_1 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & b_N \end{bmatrix}_{L \times P}$$

the complex-valued received vector $\boldsymbol{y}$ is given by

$$\boldsymbol{y} = \mathbb{B}\boldsymbol{h} + \boldsymbol{n} \tag{1}$$

where $\boldsymbol{n}$ is zero-mean complex-Gaussian distributed with $E[\boldsymbol{n}\boldsymbol{n}^H] = \sigma_n^2 \mathbb{I}_L$, and $\mathbb{I}_L$ is the $L \times L$ identity matrix. We then make the following assumptions: both transmitter and receiver know nothing about the channel coefficients $\boldsymbol{h}$, but have knowledge of the multipath parameter P. Also, there are adequate guard periods between consecutive encoding blocks such that zero interblock interference is guaranteed. Based on the system model in (1) and the above two assumptions, the least square estimate of the channel coefficients $\boldsymbol{h}$ for a given $\boldsymbol{b}$ (alternatively, $\mathbb{B}$) equals $\hat{\boldsymbol{h}} = (\mathbb{B}^T\mathbb{B})^{-1}\mathbb{B}^T\boldsymbol{y}$, and the joint maximum-likelihood (ML) decision for the transmitted codeword becomes [1]

$$\begin{aligned} \hat{\boldsymbol{b}} &= \arg\min_{\boldsymbol{b}\in\mathcal{C}} \min_{\boldsymbol{h}} \|\boldsymbol{y} - \mathbb{B}\boldsymbol{h}\|^2 \\ &= \arg\min_{\boldsymbol{b}\in\mathcal{C}} \|\boldsymbol{y} - \mathbb{B}\hat{\boldsymbol{h}}\|^2 \\ &= \arg\min_{\boldsymbol{b}\in\mathcal{C}} \|\boldsymbol{y} - \mathbb{P}_B\boldsymbol{y}\|^2 \end{aligned} \tag{2}$$

where $\mathbb{P}_B \triangleq \mathbb{B}(\mathbb{B}^T\mathbb{B})^{-1}\mathbb{B}^T$. Note that the mapping from a codeword $\boldsymbol{b}$ to the corresponding transformed codeword $\mathbb{P}_B$ is not one-to-one unless b_1 is fixed. For convenience, we will always set $b_1 = -1$ for the codes we construct.[1]

[1]Under the setting, it is obvious that the largest code rate attainable by our code design is $(N-1)/N$.

B. Code Designs for Joint Channel and Data Estimation

In the literature, no systematic code constructions have been proposed for joint channel and data estimation in quasi-static fading channels. Efforts have mostly been invested in computer searches for codes that counter channel fading [3], [6], [10], [11], [14], [15], [17]. The decoding of such structureless computer-searched codes thus becomes an engineering challenge.

In 2002, Skoglund *et al.* [14] relied on simulated annealing to search for nonlinear binary block codes suitable for joint channel and data estimation in quasi-static fading channels. As optimization criterion, they used the sum of all pairwise error probabilities (PEP) under equal prior probabilities. Although the operating signal-to-noise ratio (SNR) for the code search was set at 10 dB, their simulation results demonstrated that their codes perform well under a wide range of SNRs. In addition, the mismatch in the relative powers of different channel coefficients, as well as in the channel Rice factors [16], has no big effect on the resulting performance either. Their results indicate that the nonlinear estimation codes can outperform a typical linear error correcting code operated with a perfect channel estimator.

Later in 2005, Coskun and Chugg [3] replaced the PEP sum by a properly defined pairwise distance measure between two codewords, and proposed a suboptimal greedy algorithm to speed up the code search process. In 2007, Giese and Skoglund [6] re-applied their original idea to single and multiple-antenna systems, and used the asymptotic PEP and the generic gradient-search algorithm, respectively, in place of the PEP and the simulated-annealing algorithm in [14] to reduce system complexity.

In [14], the authors point out that "an important topic for further research is to study how the decoding complexity of the proposed scheme can be decreased." Moreover, they state that "one main issue is to investigate what kind of structure should be enforced on the code to allow for simplified decoding." Motivated by these remarks, we take here a different approach for code design. Specifically, we establish a systematic code design constraint for joint channel and data estimation in quasi-static fading channels, and show that the codes constructed based on this constraint can maximize the system SNR regardless of the fading statistics. As it so happens that the computer-searched codes in [14] also satisfy this constraint, their insensitivity to SNR and channel mismatch now find a theoretical support.

Although a recursive metric had been derived in [1] from joint maximum-likelihood decoding metric, however, there is no efficient decoding algorithm that can exploit it due to structureless code design. Taking advantage of the systematic structure of our codes, we can then derive a recursive maximum-likelihood decoding metric that can be used in the priority-first search decoding algorithm. The decoding complexity is therefore significantly decreased in contrast to that of the exhaustive decoder required by the structureless computer-searched codes.

It is worth mentioning that although the codes selected by computer search in [6] and [14] target unknown channels, for which the channel coefficients are assumed constant within a given coding block, the evaluation of the PEP criterion does presume knowledge of the channel statistics. Even if the dependence of the code design on channel statistics is relaxed in [3], the pairwise distance criterion proposed therein is still for computer search, and no systematic code design is resulted. The code constructed based on the algorithm we propose, however, is guaranteed to achieve an acceptable system SNR regardless of the statistics of the channel. This suggests that our systematically constructed codes are also suitable in cases where channel blindness becomes a stringent system restriction.

C. The Maximum-Likelihood Priority-First Search Decoding Algorithm

A code tree of an (N,K) binary code represents every codeword as a path on a binary tree. Each branch on the code tree is labeled with the appropriate code bit b_i. We can then denote the path ending at a node at level ℓ by the sequence of branch labels $[b_1, b_2, \ldots, b_\ell]$ it traverses. For convenience, we abbreviate $[b_1, b_2, \ldots, b_\ell]^T$ as $\boldsymbol{b}_{(\ell)}$, and will drop the subscript when $\ell = N$. The successor paths of a path $\boldsymbol{b}_{(\ell)}$ are those whose first ℓ labels are exactly the same as $\boldsymbol{b}_{(\ell)}$.

The *priority-first search algorithm* (also known as the *best-first search algorithm*) is a common graph search algorithm that explores a graph by expanding the most promising path selected according to some criterion. Examples are *Algorithm* A^* [12], *Dijkstra's Algorithm* [2], or *Stack Algorithm* [9]. In implementation, the most promising path is usually drawn from a list of candidates in a stack or a priority queue. One of the main distinctions among the family of priority-first search algorithms is the *metric* associated with paths on the search graph.[2] By adopting different metrics, some algorithms guarantee optimal search results, while some can only yield suboptimal ones. A typical priority-first search algorithm is exemplified by the following sequence of operations.

Step 1. Load the stack with the path that ends at the original node.

Step 2. Evaluate the metric values of the successor paths of the current top path in the stack. Then delete this top path from the stack.

Step 3. Insert the successor paths obtained in Step 2 into the stack such that the paths in the stack are ordered according to their ascending metric values.

Step 4. If the top path in the stack ends at a terminal node in the code tree, output the labels corresponding to the top path, and the algorithm stops; otherwise, go to Step 2.

Next, we give a sufficient condition under which the above priority-first search algorithm is guaranteed to locate the path with the smallest metric among all paths.

Lemma 1: If the metric f is nondecreasing along every path $\boldsymbol{b}_{(\ell)}$ in the code tree, i.e.

$$f\left(\boldsymbol{b}_{(\ell)}\right) \le \min_{\left\{\tilde{\boldsymbol{b}}\in\mathcal{C}:\tilde{\boldsymbol{b}}_{(\ell)}=\boldsymbol{b}_{(\ell)}\right\}} f(\tilde{\boldsymbol{b}}) \tag{3}$$

then the priority-first search algorithm always yields the code path with the smallest metric value among all code paths of $\mathcal{C}$.

Proof: Let $\boldsymbol{b}^*$ be the first top path that reaches a terminal node (and hence, is the output code path of the priority-first search algorithm.) Then, *Step 3* of the algorithm ensures that $f(\boldsymbol{b}^*)$ is no larger than the metric value of any path currently in the stack. Since condition (3) guarantees that the metric value of any other code path, which should be the offspring of some path $\boldsymbol{b}_{(\ell)}$ currently existing in the stack, is no less than $f(\boldsymbol{b}_{(\ell)})$, we have

$$f(\boldsymbol{b}^*) \le f\left(\boldsymbol{b}_{(\ell)}\right) \le \min_{\{\tilde{\boldsymbol{b}}\in\mathcal{C}:\tilde{\boldsymbol{b}}_{(\ell)}=\boldsymbol{b}_{(\ell)}\}} f(\tilde{\boldsymbol{b}}).$$

Consequently, the lemma follows. □

[2]In the optimization literature, this metric is sometimes called *evaluation function*. Since we apply the algorithm in decoding, we adopt the term *metric* in this work.

When defining a metric f, it is convenient to represent it as the sum of two components

$$f\left(\boldsymbol{b}_{(\ell)}\right) \triangleq g\left(\boldsymbol{b}_{(\ell)}\right) + \varphi\left(\boldsymbol{b}_{(\ell)}\right).$$

The first component g is directly defined based on the maximum-likelihood metric such that

$$\arg\min_{\boldsymbol{b}\in\mathcal{C}} g(\boldsymbol{b}) = \arg\min_{\boldsymbol{b}\in\mathcal{C}} \|\boldsymbol{y} - \mathbb{P}_B \boldsymbol{y}\|^2.$$

After g is defined, the second component φ is designed to validate (3) with $\varphi(\boldsymbol{b}) = 0$ for any $\boldsymbol{b} \in \mathcal{C}$. Then from $f(\boldsymbol{b}) = g(\boldsymbol{b}) + \varphi(\boldsymbol{b}) = g(\boldsymbol{b})$ for all $\boldsymbol{b} \in \mathcal{C}$, the desired maximum-likelihood priority-first search decoding algorithm is established. A typical interpretation of the so-called *heuristic function* φ is that it helps predict a future route from the end node of the current path to a terminal node [7]. Notably, the design of the heuristic function φ that validates (3) is not unique. Different designs may result in variations in computational complexity.

III. Code Construction

A. Code Constraint That Maximizes the Average SNR Regardless of Channel Statistics

From the system model in (1), it can be derived that the average SNR conditional on the input $\mathbb{B}$ satisfies

$$\frac{E[\|\mathbb{B}\boldsymbol{h}\|^2 \mid \mathbb{B}]}{E[\|\boldsymbol{n}\|^2]} = \frac{N}{L\sigma_n^2}\operatorname{tr}\left[E[\boldsymbol{h}\boldsymbol{h}^H]\left(\frac{1}{N}\mathbb{B}^T\mathbb{B}\right)\right]. \tag{4}$$

Since both transmitter and receiver know nothing about the channel coefficients $\boldsymbol{h}$, the average SNR can be as worse as

$$\min_{\{\boldsymbol{h}:\operatorname{tr}(E[\boldsymbol{h}\boldsymbol{h}^H])=\tau\}} \frac{E[\|\mathbb{B}\boldsymbol{h}\|^2 \mid \mathbb{B}]}{E[\|\boldsymbol{n}\|^2]}$$

where τ is a certain (possibly unknown) power level on the channel coefficients $\boldsymbol{h}$. We then found that such a worst-case SNR can be upper-bounded by a constant, i.e.

$$\min_{\{\boldsymbol{h}:\operatorname{tr}(E[\boldsymbol{h}\boldsymbol{h}^H])=\tau\}} \frac{E[\|\mathbb{B}\boldsymbol{h}\|^2 \mid \mathbb{B}]}{E[\|\boldsymbol{n}\|^2]} \le \frac{E[\|\mathbb{B}\tilde{\boldsymbol{h}}\|^2 \mid \mathbb{B}]}{E[\|\boldsymbol{n}\|^2]} = \left(\frac{N}{L\sigma_n^2}\right)\tau$$

where the above inequality holds since an upper bound can be resulted by taking any $\boldsymbol{h}$ that satisfies $\operatorname{tr}(E[\boldsymbol{h}\boldsymbol{h}^H]) = \tau$ into $E[\|\mathbb{B}\boldsymbol{h}\|^2 \mid \mathbb{B}]/E[\|\boldsymbol{n}\|^2]$, and here we take $\tilde{\boldsymbol{h}}$ to be zero-mean i.i.d. with $\operatorname{tr}(E[\tilde{\boldsymbol{h}}\tilde{\boldsymbol{h}}^H]) = \tau$. It is thus straightforward from (4) that this constant SNR bound can be achieved even if the system is totally blind on channel coefficients $\boldsymbol{h}$ (as well as the power level τ), when the codeword is designed to be *self-orthogonal* in the sense that

$$\frac{1}{N}\mathbb{B}^T\mathbb{B} = \mathbb{I}_P. \tag{5}$$

Condition (5) actually has an operational meaning. It ensures that every codeword is orthogonal to the shifted version of itself, and hence temporal diversity can be implicitly realized even under completely no knowledge on channel statistics. We henceforth say that codewords constrained on (5) maximize the average SNR attainable regardless of the statistics of $\boldsymbol{h}$ [5].

Unfortunately, a codeword sequence satisfying (5) is only guaranteed to exist for $P = 2$ with N odd (and trivially, for $P = 1$). In some other cases, such as $P = 3$, one can only design codes to approximately satisfy (5). For example

$$\frac{1}{N}\mathbb{B}^T\mathbb{B} = \frac{1}{N}\begin{bmatrix} N & \pm 1 & 0 \\ \pm 1 & N & \pm 1 \\ 0 & \pm 1 & N \end{bmatrix} \text{ for } N \text{ even}$$

and

$$\frac{1}{N}\mathbb{B}^T\mathbb{B} = \frac{1}{N}\begin{bmatrix} N & 0 & \pm 1 \\ 0 & N & 0 \\ \pm 1 & 0 & N \end{bmatrix} \text{ for } N \text{ odd.}$$

We therefore relax (5) and allow some off-diagonal entries in $\mathbb{B}^T\mathbb{B}$ to be either -1 or 1 whenever it is impossible to strictly satisfy (5). We will denote such a matrix as $\mathbb{G}$.

After the establishment of (5), we find that this particular structure of $\mathbb{G}$ can really be observed in the simulated-annealing-based computer-searched codes. Specifically, for $4 \le N \le 18$ and N even, the best computer-searched half-rate codes that minimize the sum of PEPs under complex zero-mean Gaussian distributed $\boldsymbol{h}$ with $E[\boldsymbol{h}\boldsymbol{h}^H] = (1/2)\mathbb{I}_P$ and $P = 2$ all satisfy the relation

$$\mathbb{B}^T\mathbb{B} = \begin{bmatrix} N & \pm 1 \\ \pm 1 & N \end{bmatrix}. \tag{6}$$

We have also obtained and examined the computer-searched code used in [14] for $N = 22$, and found as anticipated that every codeword satisfies (6).

We close this subsection by stating some existing results in the literature that correspond to condition (5). The authors in [4] suggest that for an optimal channel estimation, the training sequences $\boldsymbol{b}$ can be chosen such that $\mathbb{B}^T\mathbb{B}$ is proportional to $\mathbb{I}_P$. Their observation agrees with what we obtained in (5). Moreover, condition (5) also has been identified in [6] where the authors remark ([6, p. 1591]) that a code sequence with a certain aperiodic autocorrelation property possibly could be exploited in future code design approaches. This is indeed one of the main research goals of this paper.

B. Equivalent System Model for Joint Channel and Data Estimation

By noting that $\mathbb{P}_B$ is idempotent and symmetric, and both $\mathrm{tr}(\mathbb{P}_B)$ and $\|\mathrm{vec}(\mathbb{P}_B)\|^2$ equal P, where $\mathrm{vec}(\cdot)$ denotes the operation to transform a matrix into a vector,[3] the joint ML decision in (2) can be reformulated as

$$\begin{aligned}\hat{\boldsymbol{b}} &= \arg\min_{\boldsymbol{b}\in\mathcal{C}}(\boldsymbol{y} - \mathbb{P}_B\boldsymbol{y})^H(\boldsymbol{y} - \mathbb{P}_B\boldsymbol{y}) \\ &= \arg\min_{\boldsymbol{b}\in\mathcal{C}} -\mathrm{tr}(\mathbb{P}_B\boldsymbol{y}\boldsymbol{y}^H) \\ &= \arg\min_{\boldsymbol{b}\in\mathcal{C}}(\|\mathrm{vec}(\boldsymbol{y}\boldsymbol{y}^H)\|^2 - \mathrm{vec}(\mathbb{P}_B)^T\mathrm{vec}(\boldsymbol{y}\boldsymbol{y}^H) \\ &\quad - \mathrm{vec}(\boldsymbol{y}\boldsymbol{y}^H)^H\mathrm{vec}(\mathbb{P}_B) + \|\mathrm{vec}(\mathbb{P}_B)\|^2) \\ &= \arg\min_{\boldsymbol{b}\in\mathcal{C}}\|\mathrm{vec}(\boldsymbol{y}\boldsymbol{y}^H) - \mathrm{vec}(\mathbb{P}_B)\|^2. \end{aligned} \tag{7}$$

This implies that the ML decision can be obtained by finding the codeword $\mathbb{P}_B$ whose Euclidean distance to $\boldsymbol{y}\boldsymbol{y}^H$ is the smallest. We can then bound the ML error probability by

$$\begin{aligned} P_{\mathrm{e}} \le \frac{1}{2^K}\sum_{i=1}^{2^K}\sum_{j=1, j\ne i}^{2^K} &\Pr(\|\mathrm{vec}(\boldsymbol{y}\boldsymbol{y}^H) - \mathrm{vec}\left(\mathbb{P}_{B_j}\right)\|^2 \\ &< \|\mathrm{vec}(\boldsymbol{y}\boldsymbol{y}^H) - \mathrm{vec}(\mathbb{P}_{B_i})\|^2 \,|\, \boldsymbol{b}_i \text{ transmitted}) \end{aligned} \tag{8}$$

where $\boldsymbol{b}_i$ is the ith codeword of an (N, K) block code, and $\mathbb{P}_{B_i}$ denotes the equivalent ith codeword in the $\mathbb{P}_B$-domain. By the self-orthogonal property, $\mathbb{P}_B = \mathbb{B}(\mathbb{B}^T\mathbb{B})^{-1}\mathbb{B}^T = \frac{1}{N}\mathbb{B}\mathbb{B}^T$. The PEP-based upper bound in (8) then suggests that a good self-orthogonal code design should have an adequately large pairwise Euclidean distance

$$\left\|\mathrm{vec}\left(\mathbb{B}_i\mathbb{B}_i^T\right) - \mathrm{vec}\left(\mathbb{B}_j\mathbb{B}_j^T\right)\right\|^2 \tag{9}$$

between all codeword pairs $\mathbb{B}_i$ and $\mathbb{B}_j$, where $\mathbb{B}_i$ is the equivalent ith codeword in the $\mathbb{B}$-domain. Based on this observation, we may infer under equal prior probabilities that a uniform draw of codewords satisfying $\mathbb{B}^T\mathbb{B} = N \cdot \mathbb{I}_P$ may asymptotically result in a good code. This is conceptually equivalent to a uniform pick of codewords in a set of self-orthogonal binary sequences.

We recall that our initial research query is how to construct an efficiently decodable code that supports joint channel estimation and error correction. In order to achieve this goal for the priority-first search decoding algorithm, we need an efficient and systematic way to generate the successor paths of the top path. In particular, we would like to have a code tree that can be spanned in an on-the-fly or bit-by-bit fashion. The uniform pick principle then suggests that considering only the self-orthogonal sequences with the same prefix $\boldsymbol{b}_{(\ell-1)}$, the ratio of the number of self-orthogonal codewords satisfying $b_\ell = -1$ to the number of all self-orthogonal sequences having the same b_ℓ must be made equal to the similar ratio for self-orthogonal codewords satisfying $b_\ell = 1$, whenever possible. Mathematically, this can be expressed as

$$\begin{aligned} &\frac{|\mathcal{C}(b_1, b_2, \ldots, b_{\ell-1}, b_\ell = 1)|}{|\mathcal{A}(b_1, b_2, \ldots, b_{\ell-1}, b_\ell = 1 \,|\, \mathbb{G})|} \\ &\qquad\approx \frac{|\mathcal{C}(b_1, b_2, \ldots, b_{\ell-1}, b_\ell = -1)|}{|\mathcal{A}(b_1, b_2, \ldots, b_{\ell-1}, b_\ell = -1 \,|\, \mathbb{G})|} \end{aligned} \tag{10}$$

[3]For an $M \times N$ matrix $\mathbb{A}$, $\mathrm{vec}(\mathbb{A})$ is defined as

$$\mathrm{vec}(\mathbb{A}) = [a_{1,1} \,\ldots\, a_{M,1}\, a_{1,2} \,\ldots\, a_{M,2}\, a_{1,N} \,\ldots\, a_{M,N}]^T.$$

where $\mathcal{C}(\boldsymbol{b}_{(\ell)})$ is the set of all codewords whose first ℓ bits equal $b_1, b_2, \ldots, b_\ell$, and $\mathcal{A}(\boldsymbol{b}_{(\ell)} \mid \mathbb{G})$ is the set of all binary sequences of length N, whose first ℓ bits equal $b_1, b_2, \ldots, b_\ell$, and whose $\mathbb{B}$-representation satisfies $\mathbb{B}^T\mathbb{B} = \mathbb{G}$. Accordingly, given the index i of the codeword, where $0 \le i \le 2^K - 1$, and given the previous $\ell - 1$ bits $b_1, b_2, \ldots, b_{\ell-1}$, whether the next code bit b_ℓ is -1 or $+1$ can be determined conceptually by checking whether i is less than or larger than $\sum_{\tilde{b}_1+\tilde{b}_2\cdot 2+\ldots+\tilde{b}_{\ell-1}\cdot 2^{\ell-2} < b_1+b_2\cdot 2+\cdots+b_{\ell-1}\cdot 2^{\ell-2}} |\mathcal{C}(\tilde{\boldsymbol{b}}_{(\ell-1)} \mid \mathbb{G})| + |\mathcal{C}(\boldsymbol{b}_{(\ell-1)}, b_\ell = -1 \mid \mathbb{G})|$. A specific code design algorithm will be given in the next subsection.

C. *Exemplified Code Design Algorithm for Channels of Memory Order One*

In this subsection, we provide an exemplified code design algorithm based on the uniform pick principle for channels of memory order 1, namely, $P = 2$. The code design algorithm for channels with higher memory order can be similarly built.

For $\theta \in \{-1, 0, +1\}$, we define

$$\mathbb{G}_\theta \triangleq \begin{bmatrix} N & \theta \\ \theta & N \end{bmatrix}.$$

Note that when $\mathbb{B}^T\mathbb{B} = \mathbb{G}_0 (= N \cdot \mathbb{I}_2)$ cannot be satisfied as aforementioned for N even, $\mathbb{G}_{-1}$ and $\mathbb{G}_1$ will be used instead to define the relaxed self-orthogonal codewords. In such case, the uniform pick principle again suggests that half of the codewords should be uniformly drawn from binary sequences satisfying $\mathbb{B}^T\mathbb{B} = \mathbb{G}_{-1}$, and the other half of codewords are selected according to $\mathbb{B}^T\mathbb{B} = \mathbb{G}_1$. The proposed codeword selection process is simply to list all the sequences satisfying the desired self-orthogonal property in binary-alphabetical order, starting from zero, and uniformly pick the codewords from the ordered list in every Δ_θ interval with

$$\Delta_\theta = \frac{|\mathcal{A}(b_1 = -1 \mid \mathbb{G}_\theta)| - 1}{2^K/|\Theta| - 1} \quad \text{for } \theta \in \Theta \tag{11}$$

where $\Theta = \{0\}$ for N odd, and $\Theta = \{-1, 1\}$ for N even. As a result, the selected codewords are those sequences with indices closest to $\lfloor (i \bmod (2^K/|\Theta|)) \cdot \Delta_\theta \rfloor$ for $0 \le i \le 2^K - 1$. The codeword mapping algorithm is summarized by the following list.

Step 1. Input the index i of the requested codeword in the (N, K) block code, where $0 \le i \le 2^K - 1$.

Step 2. Set $\Theta = \{0\}$ for N odd, and $\Theta = \{-1, 1\}$ for N even. Also, set $\theta = ((N + 1) \bmod 2) \cdot (-1)^{\lceil (i+1)/(2^K/|\Theta|) \rceil}$. Compute Δ_θ according to (11). Initialize $b_1 = -1, \ell = 1$ and $\rho = \lfloor (i \bmod (2^K/|\Theta|)) \cdot \Delta_\theta \rfloor$. Let the minimum sequence index $\rho_{\min} = 0$.

Step 3. Execute $\ell = \ell + 1$, and compute $\gamma_\ell = |\mathcal{A}(\boldsymbol{b}_{(\ell-1)}, b_\ell = -1 \mid \mathbb{G}_\theta)|$.
If $\rho < \rho_{\min} + \gamma_\ell$, then choose the next code bit $b_\ell = -1$;
otherwise choose the next code bit $b_\ell = 1$, and readjust $\rho_{\min} = \rho_{\min} + \gamma_\ell$.

Step 4. If $\ell = N$, output the corresponding codeword $\boldsymbol{b}$, and the algorithm stops; otherwise, go to Step 3.

In implementing the above algorithm, it is perhaps more convenient to calculate γ_ℓ recursively[4] such that the codeword mapping can be performed in an on-the-fly or bit-by-bit systematic fashion with respect to the given codeword index i. This recursive nature also facilitates the priority-first decoding search at the receiver, since branches of the code tree will only be spanned when necessary.

IV. Maximum-Likelihood Metrics for Priority-First Search Decoding

In this section, we will establish two different metric functions to be used by the priority-first search algorithm. The first metric is

$$f_1\left(\boldsymbol{b}_{(\ell)}\right) = g\left(\boldsymbol{b}_{(\ell)}\right) + \varphi_1\left(\boldsymbol{b}_{(\ell)}\right) \tag{12}$$

where $g(\boldsymbol{b}_{(\ell)})$ is derived in Section IV-A, and $\varphi_1(\boldsymbol{b}_{(\ell)}) = 0$ is the all-zero function (cf. Section IV-B). The second metric is

$$f_2\left(\boldsymbol{b}_{(\ell)}\right) = g\left(\boldsymbol{b}_{(\ell)}\right) + \varphi_2\left(\boldsymbol{b}_{(\ell)}\right) \tag{13}$$

with $g(\boldsymbol{y})$ the same as in f_1, and with $\varphi_2(\boldsymbol{b}_{(\ell)})$ defined in Section IV-C. Both metrics will lead to an ML decoding. The difference is that f_1 can be computed on-the-fly, and will therefore cause much less delay in the decoding. For the evaluation of f_2, however, one needs to know *all* received symbols, but the computational complexity of f_2 is one order of magnitude less than that of f_1.

A. *Recursive Maximum-Likelihood Metric g*

Let subcode $\mathcal{C}_\theta$ be the set of codewords that satisfy $\mathbb{B}^T\mathbb{B} = \mathbb{G}_\theta$, where θ takes value in Θ. Hence, $\mathcal{C} = \cup_{\theta\in\Theta}\mathcal{C}_\theta$, and $\mathcal{C}_\theta \cap \mathcal{C}_\eta = \emptyset$ whenever $\theta \ne \eta$. Since a transmitted codeword belongs to only one of the subcodes, to maintain *individual stacks* for priority-first codeword searching over *each* subcode will introduce considerable unnecessary decoding burden, especially for the subcodes that the transmitted codeword does not belong to. Hence, only one stack is maintained during the entire priority-first search, and the metric function values for different subcodes are compared and sorted in the same stack. The path

[4]Initializing $b_0 = 0$, $m_0 = \theta$ and $\gamma_1 = |\mathcal{A}(b_1 \mid \mathbb{G}_\theta)|$, and setting $m_{\ell+1} = m_\ell - b_\ell b_{\ell+1}$ for $0 \le \ell < N$, we obtain for $P = 2$ that if $|m_{\ell-1} + b_{\ell-1}| \le N - \ell$, and $(b_{\ell-1}, b_\ell) = (-1, 1)$,

$$\gamma_{\ell+1} = \gamma_\ell \cdot \frac{1}{2(N-\ell)} \cdot \left(\frac{(N - \ell - m_{\ell-1})^2 - 1}{N - \ell + m_{\ell-1} + 1} \right) \cdot \mathbf{1}\{|m_{\ell-1} + 2| \le N - \ell - 1\}$$

where $\mathbf{1}\{\cdot\}$ is the set indicator function. If $|m_{\ell-1} + b_{\ell-1}| \le N - \ell$, but $(b_{\ell-1}, b_\ell) \ne (-1, 1)$, then

$$\gamma_{\ell+1} = \gamma_\ell \cdot \frac{1}{2(N-\ell)} \cdot (N - \ell + m_{\ell-1} + 1 - b_{\ell-1}b_\ell + b_\ell) \cdot \mathbf{1}\{|m_{\ell-1} - b_{\ell-1}b_\ell + b_\ell| \le N - \ell - 1\}.$$

If however $|m_{\ell-1} + b_{\ell-1}| > N - \ell$, then

$$\gamma_{\ell+1} = \begin{cases} 0, & \text{for } (b_{\ell-1}, b_\ell) \ne (-1, 1) \\ & \text{or } ((b_{\ell-1}, b_\ell) = (-1, 1) \text{ and } m_{\ell-1} \ne -N + \ell - 1) \\ 1, & \text{otherwise.} \end{cases}$$

to be expanded next is therefore the one whose metric function value is the smallest globally.

By denoting $\mathbb{D}_\theta = \mathbb{G}_\theta^{-1} = (\mathbb{B}^T\mathbb{B})^{-1}$, and letting the matrix entry of $\mathbb{D}_\theta$ be $\delta_{i,j}^{(\theta)}$, we can continue the derivation from (7) as follows:

$$\begin{aligned}\hat{\boldsymbol{b}} &= \arg\min_{\boldsymbol{b}\in\mathcal{C}} \sum_{\theta\in\Theta} \left[-\mathrm{tr}(\mathbb{B}\mathbb{D}_\theta\mathbb{B}^T\boldsymbol{y}\boldsymbol{y}^H)\right]\cdot \mathbf{1}\{\boldsymbol{b}\in\mathcal{C}_\theta\}\\ &= \arg\min_{\boldsymbol{b}\in\mathcal{C}} \sum_{\theta\in\Theta} \left[-\mathrm{vec}(\mathbb{D}_\theta)^T\mathrm{vec}(\mathbb{B}^T\boldsymbol{y}\boldsymbol{y}^H\mathbb{B})\right]\cdot \mathbf{1}\{\boldsymbol{b}\in\mathcal{C}_\theta\}\\ &= \arg\min_{\boldsymbol{b}\in\mathcal{C}} \sum_{\theta\in\Theta} \left[-\sum_{i=0}^{P-1}\sum_{j=0}^{P-1}\delta_{i,j}^{(\theta)}\sum_{m=1}^{L}\sum_{n=1}^{L} b_{m+i}b_{n+j}y_m y_n^*\right]\\ &\quad \cdot \mathbf{1}\{\boldsymbol{b}\in\mathcal{C}_\theta\}\end{aligned}$$

where for convenience, we put $b_j = 0$ for $j > N$. After adjusting indices, the derivation can be resumed as

$$\hat{\boldsymbol{b}} = \arg\min_{\boldsymbol{b}\in\mathcal{C}} \frac{1}{2}\sum_{\theta\in\Theta}\left[\sum_{m=1}^{N}\sum_{n=1}^{N}\left(-w_{m,n}^{(\theta)}b_m b_n\right)\right]\cdot\mathbf{1}\{\boldsymbol{b}\in\mathcal{C}_\theta\} \tag{14}$$

where

$$w_{m,n}^{(\theta)} = \sum_{i=0}^{P-1}\sum_{j=0}^{P-1}\delta_{i,j}^{(\theta)}\mathrm{Re}\{y_{m+i}y_{n+j}^*\}.$$

As the maximum-likelihood decision remains unchanged by adding a constant that is independent of the codeword $\boldsymbol{b}$, we add a constant to make the decision criterion nonnegative[5]:

$$\begin{aligned}\hat{\boldsymbol{b}} &= \arg\min_{\boldsymbol{b}\in\mathcal{C}}\left\{\sum_{m=1}^{N}\max_{\eta\in\Theta}\left(\sum_{n=1}^{m-1}\left|w_{m,n}^{(\eta)}\right| + \frac{1}{2}\left|w_{m,m}^{(\eta)}\right|\right)\right.\\ &\quad\left. -\frac{1}{2}\sum_{\theta\in\Theta}\left[\sum_{m=1}^{N}\sum_{n=1}^{N}w_{m,n}^{(\theta)}b_m b_n\right]\mathbf{1}\{\boldsymbol{b}\in\mathcal{C}_\theta\}\right\}\\ &= \arg\min_{\boldsymbol{b}\in\mathcal{C}}\sum_{\theta\in\Theta}\left[\sum_{m=1}^{N}\max_{\eta\in\Theta}\left(\sum_{n=1}^{m-1}\left|w_{m,n}^{(\eta)}\right| + \frac{1}{2}\left|w_{m,m}^{(\eta)}\right|\right)\right.\\ &\quad\left.-\frac{1}{2}\sum_{m=1}^{N}\sum_{n=1}^{N}w_{m,n}^{(\theta)}b_m b_n\right]\mathbf{1}\{\boldsymbol{b}\in\mathcal{C}_\theta\}.\end{aligned}$$

It remains to prove that the metric of

$$\sum_{m=1}^{N}\max_{\eta\in\Theta}\left(\sum_{n=1}^{m-1}\left|w_{m,n}^{(\eta)}\right| + \frac{1}{2}\left|w_{m,m}^{(\eta)}\right|\right) - \frac{1}{2}\sum_{m=1}^{N}\sum_{n=1}^{N}w_{m,n}^{(\theta)}b_m b_n$$

can be computed recursively. To that aim, we define for every path $\boldsymbol{b}_{(\ell)}$ over code tree θ f

$$\begin{aligned}g\left(\boldsymbol{b}_{(\ell)}\right) &\triangleq \sum_{m=1}^{\ell}\max_{\eta\in\Theta}\left(\sum_{n=1}^{m-1}\left|w_{m,n}^{(\eta)}\right| + \frac{1}{2}\left|w_{m,m}^{(\eta)}\right|\right)\\ &\quad -\frac{1}{2}\sum_{m=1}^{\ell}\sum_{n=1}^{\ell}w_{m,n}^{(\theta)}b_m b_n.\end{aligned}$$

Then, by $w_{m,n}^{(\theta)} = w_{n,m}^{(\theta)}$ for every $1 \le m, n \le N$ and $\theta \in \Theta$, we have for $1 \le \ell \le N-1$

$$\begin{aligned}g(\boldsymbol{b}_{(\ell+1)}) &= g\left(\boldsymbol{b}_{(\ell)}\right)\\ &\quad + \max_{\eta\in\Theta}\left(\sum_{n=1}^{\ell}\left|w_{\ell+1,n}^{(\eta)}\right| + \frac{1}{2}\left|w_{\ell+1,\ell+1}^{(\eta)}\right|\right)\\ &\quad - \sum_{n=1}^{\ell}w_{\ell+1,n}^{(\theta)}b_{\ell+1}b_n - \frac{1}{2}w_{\ell+1,\ell+1}^{(\theta)}\\ &= g\left(\boldsymbol{b}_{(\ell)}\right) + \max_{\eta\in\Theta}\alpha_{\ell+1}^{(\eta)}\\ &\quad - b_{\ell+1}\sum_{i=0}^{P-1}\sum_{j=0}^{P-1}\delta_{i,j}^{(\theta)}\mathrm{Re}\left\{y_{\ell+i+1}\cdot u_j\left(\boldsymbol{b}_{(\ell+1)}\right)\right\}\end{aligned}$$

where

$$\alpha_{\ell+1}^{(\eta)} \triangleq \sum_{n=1}^{\ell}\left|w_{\ell+1,n}^{(\eta)}\right| + \frac{1}{2}\left|w_{\ell+1,\ell+1}^{(\eta)}\right| \tag{15}$$

and for $0 \le j \le P-1$,

$$\begin{aligned}u_j(\boldsymbol{b}_{(\ell+1)}) &\triangleq \sum_{n=1}^{\ell}b_n y_{n+j}^* + \frac{1}{2}b_{\ell+1}y_{\ell+j+1}^*\\ &= u_j\left(\boldsymbol{b}_{(\ell)}\right) + \frac{1}{2}\left(b_\ell y_{\ell+j}^* + b_{\ell+1}y_{\ell+1+j}^*\right).\end{aligned}$$

This shows that we can recursively compute $g(\boldsymbol{b}_{(\ell+1)})$ and $\{u_j(\boldsymbol{b}_{(\ell+1)})\}_{0\le j\le P-1}$ from the previous $g(\boldsymbol{b}_{(\ell)})$ and $\{u_j(\boldsymbol{b}_{(\ell)})\}_{j=0}^{P-1}$ using $y_{\ell+1}, y_{\ell+2}, \ldots, y_{\ell+P}$ and $b_{\ell+1}$, and setting as initial condition $g(\boldsymbol{b}_{(0)}) = u_j(\boldsymbol{b}_{(0)}) = b_0 = 0$ for $0 \le j \le P-1$.

A final remark in this discussion is that although the computational burden of $\alpha_\ell^{(\eta)}$ in (15) increases linearly with ℓ, such a linearly increasing burden can be moderately compensated for by the fact that it is only necessary to compute $\alpha_\ell^{(\eta)}$ once for each ℓ and η, because it can be shared for all paths ending at level ℓ over the code tree η.

B. Heuristic Function φ_1

We next derive the first heuristic function that validates (3). Taking the maximum-likelihood metric g into the sufficient condition in (3) yields

$$\begin{aligned}&\sum_{m=1}^{\ell}\max_{\eta\in\Theta}\alpha_m^{(\eta)} - \frac{1}{2}\sum_{m=1}^{\ell}\sum_{n=1}^{\ell}w_{m,n}^{(\theta)}b_m b_n + \varphi\left(\boldsymbol{b}_{(\ell)}\right)\\ &\le \min_{\{\tilde{\boldsymbol{b}}\in\mathcal{C}:\tilde{\boldsymbol{b}}_{(\ell)}=\boldsymbol{b}_{(\ell)}\}}\left[\sum_{m=1}^{N}\max_{\eta\in\Theta}\alpha_m^{(\eta)}\right.\\ &\quad\left.-\frac{1}{2}\sum_{m=1}^{N}\sum_{n=1}^{N}w_{m,n}^{(\theta)}b_m b_n + \varphi(\tilde{\boldsymbol{b}})\right].\end{aligned}$$

[5]Here, a *nonnegative* maximum-likelihood criterion makes possible the later definition of path metric $g(\boldsymbol{b}_{(\ell)})$ to be *nondecreasing* along any path in the code tree. It can then be anticipated (cf. Section IV-B) that letting the heuristic function be zero for all paths in the code tree suffices to result in a metric function satisfying the condition (3) in Lemma 1.

Note that the additive constant that makes the metric function nondecreasing along any path in the code tree can also be obtained by first defining g based on (14), and then determining its respective φ according to (3). Such an approach however complicates the determination of the heuristic function φ when we additionally require the metric function to be recursive-computable. The alternative approach that directly defines a recursive-computable g based on a nonnegative maximum-likelihood criterion is accordingly adopted in this work.

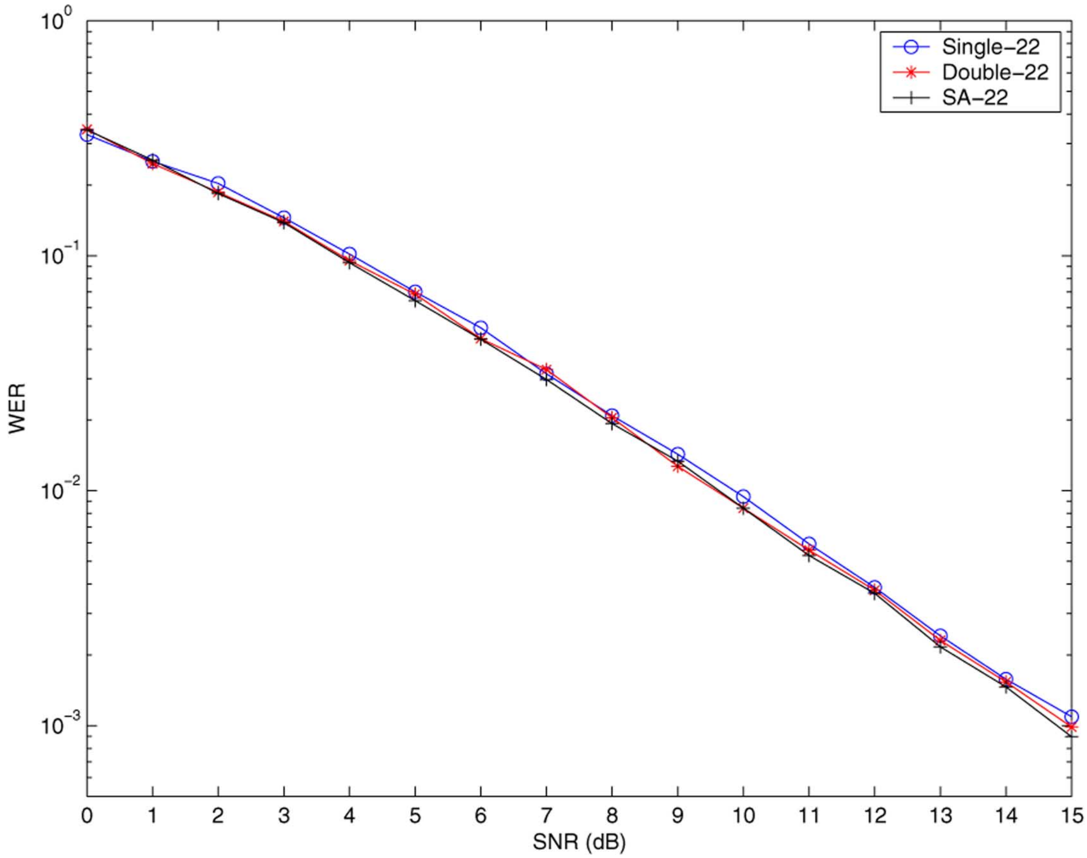

Fig. 1. The maximum-likelihood word error rates (WERs) of the computer-searched half-rate code by simulated annealing in [14] (SA-22), the constructed half-rate code with double code trees (Double-22), and the constructed half-rate code with single code tree (Single-22). The codeword length is $N = 22$.

Hence, in addition to $\varphi(\tilde{\boldsymbol{b}}) = 0$, the heuristic function should satisfy

$$\varphi\left(\boldsymbol{b}_{(\ell)}\right) \leq \sum_{m=\ell+1}^{N} \max_{\eta\in\Theta} \alpha_m^{(\eta)} - \max_{\left\{\tilde{\boldsymbol{b}}\in\mathcal{C}:\tilde{\boldsymbol{b}}_{(\ell)}=\boldsymbol{b}_{(\ell)}\right\}} \left(\sum_{m=\ell+1}^{N} \tilde{b}_m \sum_{n=1}^{\ell} w_{m,n}^{(\theta)} b_n + \frac{1}{2}\sum_{m=\ell+1}^{N}\sum_{n=\ell+1}^{N} w_{m,n}^{(\theta)} \tilde{b}_m \tilde{b}_n\right). \tag{16}$$

It is apparent that the all-zero function is the largest one that satisfies this inequality subject to no dependence on the future route and future receptions, i.e., $\{\tilde{b}_m\}_{m\geq\ell+1}$ and $\{w_{m,n}^{(\theta)}\}_{m\geq\ell+1,n\geq\ell+1}$. Hence, we choose $\varphi_1(\boldsymbol{b}_{(\ell)}) = 0$.

Note that φ_1 is trivially on-the-fly computable, and hence so is f_1. In comparison with the exhaustive-search decoding, decoding based on recursive priority-first search shows a significant decrease in computational complexity especially at medium-to-high SNRs.

C. Heuristic Function φ_2

If we drop the requirement that the metric f must be independence of future receptions, we can further reduce the computational complexity. Upon reception of all $y_1,\ldots,y_L$, the heuristic function that satisfies (16) regardless of $\tilde{b}_{\ell+1},\ldots,\tilde{b}_N$ can be increased to

$$\begin{aligned}\varphi_2\left(\boldsymbol{b}_{(\ell)}\right) &\triangleq \sum_{m=\ell+1}^{N} \max_{\eta\in\Theta} \alpha_m^{(\eta)} - \sum_{m=\ell+1}^{N} \left|\sum_{n=1}^{\ell} w_{m,n}^{(\theta)} b_n\right| - \frac{1}{2}\sum_{m=\ell+1}^{N}\sum_{n=\ell+1}^{N}\left|w_{m,n}^{(\theta)}\right| \\ &= \sum_{m=\ell+1}^{N} \max_{\eta\in\Theta} \alpha_m^{(\eta)} - \sum_{m=\ell+1}^{N}\left|v_m^{(\theta)}\left(\boldsymbol{b}_{(\ell)}\right)\right| - \beta_\ell^{(\theta)}\end{aligned} \tag{17}$$

where for $1 \leq \ell, m \leq N$ and $\theta \in \Theta$

$$v_m^{(\theta)}\left(\boldsymbol{b}_{(\ell)}\right) \triangleq \sum_{n=1}^{\ell} w_{m,n}^{(\theta)} b_n = v_m^{(\theta)}(\boldsymbol{b}_{(\ell-1)}) + b_\ell w_{\ell,m}^{(\theta)}$$

and

$$\begin{aligned}\beta_\ell^{(\theta)} &\triangleq \sum_{m=\ell+1}^{N}\left(\sum_{n=\ell+1}^{m-1}\left|w_{m,n}^{(\theta)}\right| + \frac{1}{2}\left|w_{m,m}^{(\theta)}\right|\right) \\ &= \beta_{\ell-1}^{(\theta)} - \sum_{n=\ell+1}^{N}\left|w_{\ell,n}^{(\theta)}\right| - \frac{1}{2}\left|w_{\ell,\ell}^{(\theta)}\right|\end{aligned}$$

with initial conditions $v_m^{(\theta)}(\boldsymbol{b}_{(0)}) = b_0 = 0$, and $\beta_0^{(\theta)} = \sum_{m=1}^{N} \alpha_m^{(\theta)}$. Simulations show that compared to the zero-heuristic function φ_1, the heuristic function in (17)

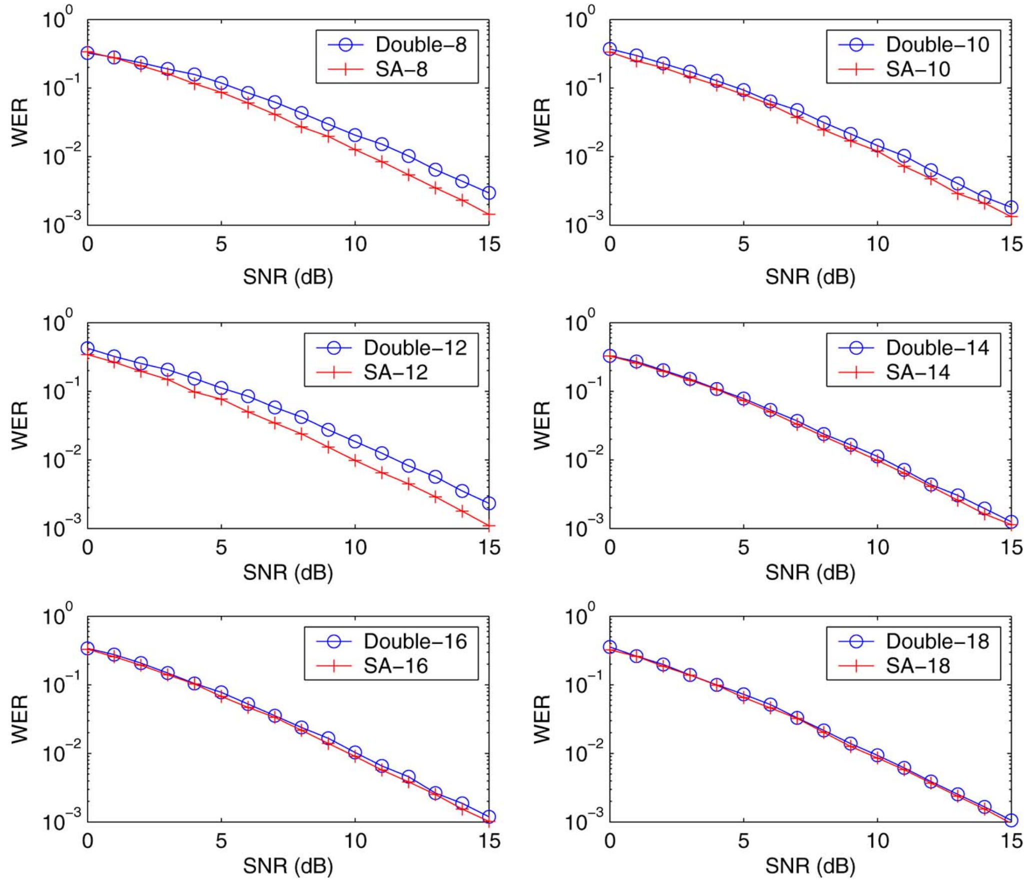

Fig. 2. The maximum-likelihood word error rates (WERs) of the computer-searched code by simulated annealing (SA-N) and the constructed half-rate code with double code trees (Double-N).

further reduces the number of path expansions during the decoding process up to one order of magnitude (cf. Table I).

V. Simulation Results

In this section, we examine the performance of the codes proposed in Section III. We also illustrate the decoding complexity of the maximum-likelihood priority-first search decoding algorithm presented in the previous section. For ease of comparison, the channel parameters used in our simulations follow those in [14], where $\boldsymbol{h}$ is zero-mean complex-Gaussian distributed with $E[\boldsymbol{h}\boldsymbol{h}^H] = (1/P)\mathbb{I}_P$ and $P = 2$. The average system SNR is, thus, given by

$$\frac{N}{L\sigma_n^2}\mathrm{tr}\left(E[\boldsymbol{h}\boldsymbol{h}^H]\frac{1}{N}\mathbb{B}^T\mathbb{B}\right) = \frac{N}{L\sigma_n^2}\mathrm{tr}\left(\frac{1}{NP}\mathbb{B}^T\mathbb{B}\right) = \frac{N}{L\sigma_n^2} \quad (18)$$

since $\mathrm{tr}(\mathbb{B}^T\mathbb{B}) = NP$ for all simulated codewords.[6]

[6]The authors in [14] directly define the channel SNR as $1/\sigma_n^2$. It is apparent that their definition is exactly the limit of (18) as N approaches infinity.

Since it is assumed that an adequate guard period between two encoding blocks exists (so that there is no interference between two consecutive decoding blocks), the computation of the system SNR for finite N should be adjusted to account for this muting (but still part-of-the-decoding-block) guard period. For example, in comparison of the $(6,3)$ and $(20,10)$ codes over channels with memory order 1 (i.e., $P = 2$), one can easily observe that the former can only transmit 18 code bits in the time interval of 21 code bits, while the latter pushes out up to 20 code bits in the period of the same duration. Thus, under fixed code bit transmission power and fixed component noise power σ_n^2, it is reasonable for the $(20,10)$ code to result in a higher SNR than the $(6,3)$ code.

Fig. 1 illustrates the simulation results of three codes: the computer-searched half-rate code obtained by the simulated annealing algorithm in [14] (SA-22), the constructed double-tree code with half of the codewords satisfying $\mathbb{B}^T\mathbb{B} = \mathbb{G}_{-1}$ and the remaining half satisfying $\mathbb{B}^T\mathbb{B} = \mathbb{G}_1$ (Double-22), and the constructed single-tree code whose codewords are all selected from the candidate sequences satisfying $\mathbb{B}^T\mathbb{B} = \mathbb{G}_{-1}$ (Single-22). We observe from Fig. 1 that the Double-22 code performs almost the same as the SA-22 code. Actually, the simulations illustrated in Fig. 2 provide evidence that the performance of the constructed double-tree half-rate codes is as good as the computer-searched half-rate codes for all $N > 12$. However, when $N \le 12$, the Double-N code performs slightly worse than the SA-N code. This is because for $N \le 12$ the approximation in (10) can no longer be well maintained due to the restriction that $|\mathcal{A}(\boldsymbol{b}_{(\ell)}\,|\,\mathbb{G})|$ must be an integer.

In addition to the Double-22 code, Fig. 1 also depicts simulation results of the Single-22 code. Since the pairwise codeword distance in the sense of (9) for the Single-22 code is in general smaller than that of the Double-22 code, its performance has a 0.2 dB degradation compared with that of the Double-22 code. However, we will see in Fig. 3 that the Single-22 code actually has the smallest decoding complexity among the three codes. This suggests that to select codewords uniformly from a single code tree should not be ruled out as a candidate design, especially when the decoding complexity becomes the main system concern.

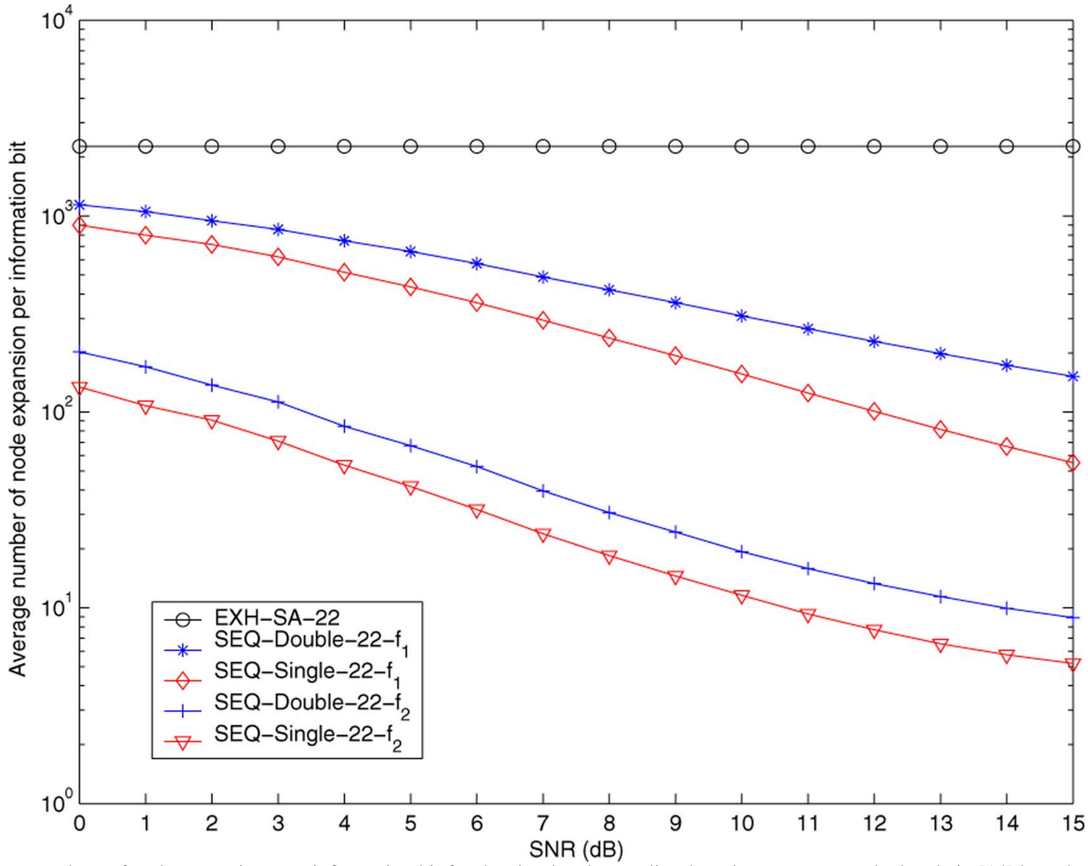

Fig. 3. The average numbers of node expansions per information bit for the simulated-annealing-based computer-searched code in [14] by exhaustive decoding (EXH-SA-22), and the constructed single-tree (SEQ-Single-22) and double-tree (SEQ-Double-22) codes using the priority-first search decoding guided by either metric function f_1 or metric function f_2.

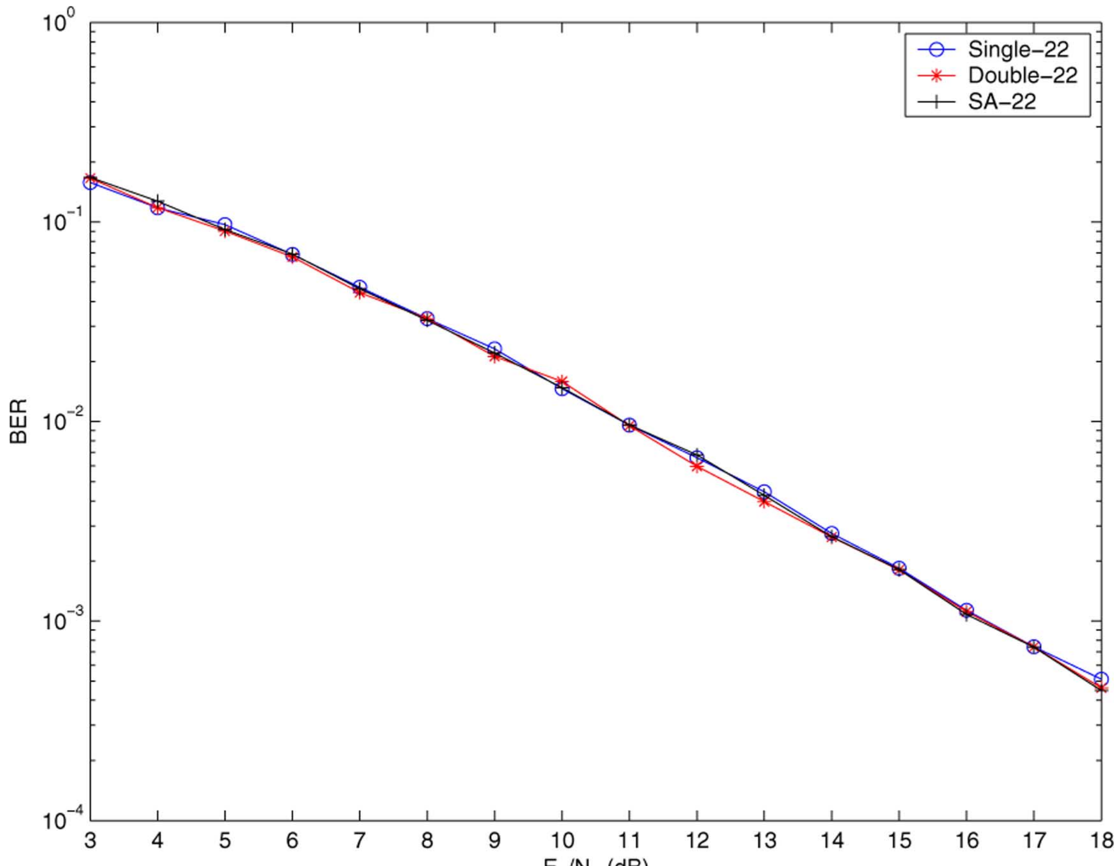

Fig. 4. BERs for the simulation of codes illustrated in Fig. 1.

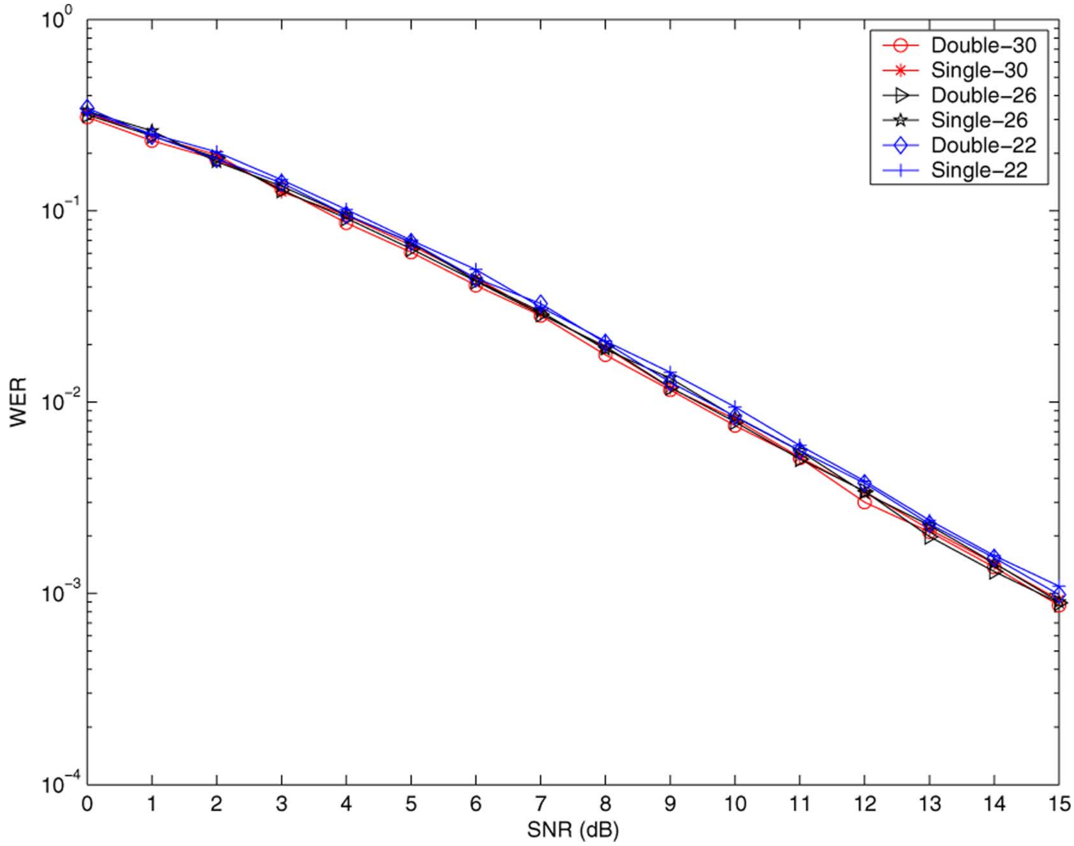

Fig. 5. WERs for the codes of Single-22, Double-22, Single-26, Double-26, Single-30, and Double-30.

In Fig. 3, the average numbers of node expansions per information bit are illustrated for the codes examined in Fig. 1. Since the number of node expansions is exactly equal to the number of tree branch metrics (i.e., one recursion of f-function values) computed, the equivalent complexity of exhaustive decoding is correspondingly plotted. It can then be observed that in comparison with the exhaustive decoder, a significant reduction in computational burden is achieved at moderate-to-high SNRs by adopting the Double-22 code and the priority-first search decoder with on-the-fly computable metric f_1 [see (12)]. Further reduction can be achieved if the Double-22 code is replaced with the Single-22 code. This is because performing the sequential search over multiple code trees introduces extra node expansions for those code trees that the transmitted codeword does not belong to. An additional order-of-magnitude reduction in node expansions can be achieved when the metric $f_2 = g + \varphi_2$ [see (13)] is used instead.

The authors in [3] and [14] only focus on the word error rate (WER). No bit error rate (BER) performances that involve the mapping design between the information bit patterns and the codewords are presented. Yet, in certain applications, such as voice transmission and digital radio broadcasting, the BER is generally considered a more critical performance index. In addition, the adoption of the BER performance index, as well as the signal-to-noise ratio per information bit, facilitates the comparison of codes of different code rates.

Fig. 4 depicts the BER performance of the same codes whose WER performances were depicted in Fig. 1. The corresponding $E_{\rm b}/N_0$ is computed according to $E_{\rm b}/N_0 = {\rm SNR}/R$, where $R = K/N$ is the code rate. The mapping between the bit patterns and the codewords of the given computer-searched code is obtained through simulated annealing by minimizing the upper bound of

$$\text{BER} \le \frac{1}{2^K} \sum_{i=1}^{2^K} \sum_{\substack{j=1\\ j\neq i}}^{2^K} \frac{d(\boldsymbol{m}_i, \boldsymbol{m}_j)}{K} \Pr(\hat{\boldsymbol{b}} = \boldsymbol{b}_j \,|\, \boldsymbol{b}_i \text{ transmitted})$$

where, other than the notations defined in (8), $\boldsymbol{m}_i$ is the information sequence corresponding to the ith codeword, and $d(\cdot,\cdot)$ is the Hamming distance. For the constructed codes of Section III-C, the binary representation of the index of the requested codeword in Step 1 is directly taken as the information bit pattern corresponding to the requested codeword. The result illustrated in Fig. 4 then indicates that the BER performance of the three curves are almost the same. Hence we conclude that taking the binary representation of the requested codeword index as the information bit pattern for the constructed code not only makes its implementation easy, but also yields a BER performance similar to that of the best simulated-annealing-based computer-searched codes.

Last, we demonstrate the WER and BER performances, respectively, of Single-26, Double-26, Single-30, and Double-30 codes, together with those of Single-22 and Double-22 codes, over the quasi-static fading channels in Figs. 5 and 6. Both figures show that the Double-30 code has the best maximum-likelihood performance not only in WER but also in BER. This result concurs with the intuition that a longer code will perform better provided that the channel coefficients remain unchanged

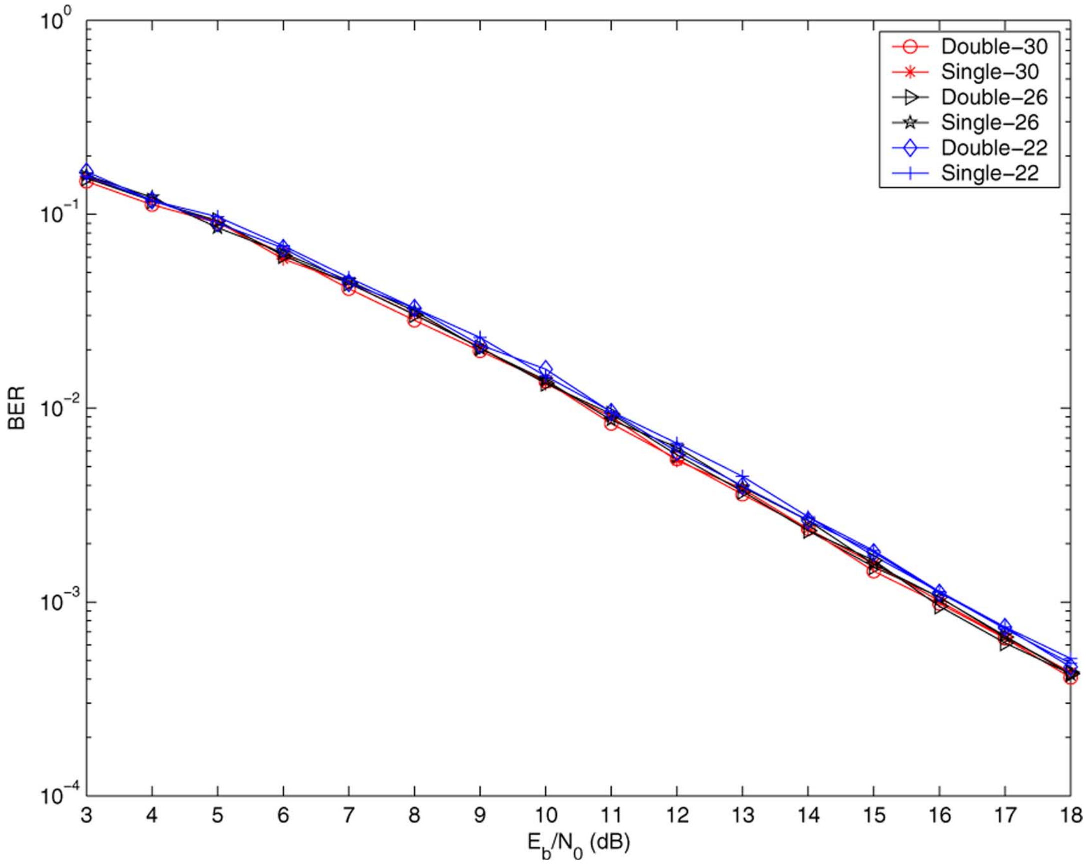

Fig. 6. BERs for the codes of Single-22, Double-22, Single-26, Double-26, Single-30, and Double-30.

TABLE I
AVERAGE NUMBER OF NODE EXPANSIONS PER INFORMATION BIT FOR THE PRIORITY-FIRST SEARCH DECODING OF THE CONSTRUCTED HALF-RATE CODES OF LENGTH 22, 26, AND 30

SNR	5dB	6dB	7dB	8dB	9dB	10dB	11dB	12dB	13dB	14dB	15dB	∞ dB
Double-22-f_1	671	590	506	436	375	320	274	236	204	178	156	54
Double-22-f_2	68	55	42	32	26	20	17	14	12	10	9	6
ratio of f_1/f_2	9.8	10.7	12.0	13.6	14.4	16.0	16.1	16.8	17.0	17.8	17.3	9.0
Double-26-f_1	2361	2006	1695	1416	1189	981	813	677	523	499	392	105
Double-26-f_2	175	130	94	69	53	39	29	23	18	15	13	6
ratio of f_1/f_2	13.5	15.4	18.0	20.5	22.4	25.2	28.0	29.4	29.1	33.3	30.2	17.5
Double-30-f_1	8455	7073	5760	5133	3759	3430	2644	1996	1765	1368	1081	192
Double-30-f_2	459	332	232	166	119	86	60	44	33	25	20	7
ratio of f_1/f_2	18.4	21.3	24.8	30.9	31.6	39.9	44.1	45.4	53.4	54.7	54.1	27.4
Single-22-f_1	460	371	308	250	200	163	130	105	85	69	57	12
Single-22-f_2	45	33	26	20	15	12	10	8	7	6	5	4
ratio of f_1/f_2	10.2	11.2	11.8	12.5	13.3	13.5	13.0	13.1	12.1	11.5	11.4	3.0
Single-26-f_1	1635	1328	1061	839	666	522	403	312	244	191	152	21
Single-26-f_2	112	79	57	42	31	23	17	13	11	9	7	4
ratio of f_1/f_2	14.6	16.8	18.6	20.0	21.5	22.7	23.7	23.9	22.2	21.2	21.7	5.3
Single-30-f_1	5871	4695	3857	2924	2335	1813	1328	884	805	572	416	39
Single-30-f_2	284	199	144	101	72	51	35	26	18	14	11	4
ratio of f_1/f_2	20.6	23.6	26.8	29.0	32.4	35.5	38.0	34.0	44.7	40.9	37.8	9.8

in a coding block. The decoding complexities of the codes are listed in Table I, from which we observe that the saving of decoding complexity of metric f_2 with respect to metric f_1 increases as the codeword length increases. It is worth mentioning that at very high SNR, the priority-first search decoding over the AWGN channels will directly go all the way down to the terminal nodes, and result in a decoding complexity of approximately two node expansions per information bit. However, for fading channels, the decoding complexity cannot reach the ideal two node expansions per information bit even with zero additive noise, as shown in the last column of Table I. In this regard, metric f_2 still reaches a better ultimate decoding complexity than metric f_1.

We close this section by commenting on the attained diversity level d of the simulated codes. The diversity level d serves as approximation of the word error probability at high SNR, i.e.,

TABLE II
THE ATTAINED DIVERSITY LEVELS OF CODES, WHICH ARE LEAST-SQUARE-APPROXIMATED BASED ON WER PERFORMANCE CURVES WITHIN 8–15 dB

Diversity	$N=8$	$N=10$	$N=12$	$N=14$	$N=16$	$N=18$	$N=22$	$N=26$	$N=30$
SA-N	1.84	1.84	1.88	1.87	1.89	1.87	1.91	—	—
Single-N	—	—	—	—	—	—	1.89	1.90	1.87
Double-N	1.67	1.80	1.79	1.85	1.88	1.87	1.87	1.89	1.87

$P_{\mathrm{e}} \approx \mathrm{SNR}^{-d}$. From Table II, we observe that the attained diversities of codes of length 22 are around 1.9, which is close to the anticipated value of $P=2$. The tables also suggest that the diversities degrade at small N, and the computer-searched codes have somewhat higher diversities within the considered SNR range. We conclude that under the constraint of the self-orthogonal structure, the simulated codes can turn the second delayed channel path into another diversity. This results in a blind detection performance of diversity level close to P.

VI. CONCLUSION

In this paper, we introduce an algorithm to construct codes that allow joint channel estimation and error correction at the receiver side of a block fading channel. In contrast with previously published codes, our codes are designed systematically and allow for an ML decoding with a much smaller computational complexity than the operation-intensive exhaustive decoding that was previously used in [3], [6], [14] to decode the structureless computer-searched codes. The given algorithm is based on the optimal signal-to-noise ratio framework and requires every codeword to satisfy a self-orthogonal property that helps to counter the effects of multipath fading.

The improved decoding algorithm is a tree-based priority-first search decoding algorithm that uses a recursive maximum-likelihood metric. Simulations demonstrate that the constructed codes have almost identical performance as the best computer-searched codes, but with much smaller decoding complexity.

Moreover, we propose two different maximum-likelihood decoding metrics. The first one can be used in an on-the-fly fashion, while the second one that results in a much lower decoding complexity requires the knowledge of all channel outputs. We hence have a tradeoff between decoding complexity versus decoding delay.

Note that so far we have ignored an implicit problem of codes that absorb the training sequence into the error-correcting codewords: in traditional packet-switched systems, frame synchronization is often achieved by the same training sequence. Without synchronizing the codeword margins, decoding may become technically infeasible. Nevertheless, there are recent standards starting to consider to partly separate the tasks of frame synchronization and channel estimation. For example, in IEEE 802.16e, initial frame synchronization is performed by means of a preamble, and is later shared by all users. Pilots are then added amid user data for individual channel estimation during data transmission [18]. It is then fair to say that at this stage, the joint channel estimation and error correction codes may only fit well in an initial-sync, or circuit-switched, or TDD-based system environment. It will be an interesting, but quite challenging, future task to further enhance the proposed codes to possess self-synchronization capability.

ACKNOWLEDGMENT

The authors would like to thank Prof. M. Skoglund, Dr. J. Giese and Prof. S. Parkvall of the Royal Institute of Technology (KTH), Stockholm, Sweden, for kindly providing us their codes for further study in the preparation of this paper.

REFERENCES

[1] K. M. Chugg and A. Polydoros, "MLSE for an unknown channel—Part I: Optimality considerations," *IEEE Trans. Commun.*, vol. 44, no. 7, pp. 836–846, Jul. 1996.
[2] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd ed. Cambridge, MA: MIT Press, 2001.
[3] O. Coskun and K. M. Chugg, "Combined coding and training for unknown ISI channels," *IEEE Trans. Commun.*, vol. 53, no. 8, pp. 1310–1322, Aug. 2005.
[4] S. N. Crozier, D. D. Falconer, and S. A. Mahmound, "Least sum of squared errors (LSSE) channel estimation," *Proc. Inst. Elect. Eng.*, vol. 138, pt. F, pp. 371–378, Aug. 1991.
[5] G. Ganesan and P. Stoica, "Space-time block codes: A maximum SNR approach," *IEEE Trans. Inf. Theory*, vol. 47, no. 4, pp. 1650–1659, May 2001.
[6] J. Giese and M. Skoglund, "Single- and multiple-antenna constellations for communication over unknown frequency-selective fading channels," *IEEE Trans. Inf. Theory*, vol. 53, no. 4, pp. 1584–1594, Apr. 2007.
[7] Y. S. Han and P.-N. Chen, "Sequential decoding of convolutional codes," in *The Wiley Encyclopedia of Telecommunications*, J. Proakis, Ed. New York: Wiley, 2002.
[8] D. Harville, *Matrix Algebra From a Statistician's Perspective*, 1st ed. New York: Springer, 2000.
[9] S. Lin and D. J. Costello, Jr., *Error Control Coding*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 2004.
[10] J. C. L. Ng, K. B. Letaief, and R. D. Murch, "Complex optimal sequences with constant magnitude for fast channel estimation initialization," *IEEE Trans. Commun.*, vol. 46, no. 3, pp. 305–308, Mar. 1998.
[11] S. I. Park, S. R. Park, I. Song, and N. Suehiro, "Multiple-access interference reduction for QS-CDMA systems with a novel class of polyphase sequence," *IEEE Trans. Inf. Theory*, vol. 46, no. 4, pp. 1448–1458, Jul. 2000.
[12] J. Pearl, *Heuristics: Intelligent Search Strategies for Computer Problem Solving*. Reading, MA: Addison-Wesley, 1984.
[13] N. Seshadri, "Joint data and channel estimation using blind trellis search techniques," *IEEE Trans. Commun.*, vol. 42, no. 2/3/4, pp. 1000–1011, Feb./Mar./Apr. 1994.
[14] M. Skoglund, J. Giese, and S. Parkvall, "Code design for combined channel estimation and error protection," *IEEE Trans. Inf. Theory*, vol. 48, no. 5, pp. 1162–1171, May 2002.
[15] C. Tellambura, Y. J. Guo, and S. K. Barton, "Channel estimation using aperiodic binary sequences," *IEEE Commun. Lett.*, vol. 2, no. 5, pp. 140–142, May 1998.
[16] C. Tepedelenlioğlu, A. Abdi, and G. B. Giannakis, "The Ricean K factor: Estimation and performance analysis," *IEEE Trans. Wireless Commun.*, vol. 2, no. 4, pp. 799–810, Jul. 2003.
[17] S.-A. Yang and J. Wu, "Optimal binary training sequence design for multiple-antenna systems over dispersive fading channels," *IEEE Trans. Veh. Technol.*, vol. 51, no. 5, pp. 1271–1276, Sep. 2002.
[18] *Draft Standard for Local and Metropolitan Area Networks, Part 16: Air Interface for Broadcast Wireless Access Systems*, P802.16Rev2/D6, Jul. 2008.

Chia-Lung Wu (S'08) received the B.Eng. degree in information and computer engineering from the Chung-Yuan Christian University, Chung-Li, Taiwan, in 1994, the M.S. degree in communications engineering from the National Chiao-Tung University, Hsin Chu, Taiwan, in 2000.

He is currently working toward the Ph.D. degree under the advisory of Prof. P.-N. Chen. He held visiting positions with Queen's University, Kingston, Canada, in 2005, and the Royal Institute of Technology (KTH), Stockholm, Sweden, in 2008. His research interests lie in noncoherent communications and channel coding.

Po-Ning Chen (S'93–M'95–SM'01) was born in Taipei, R.O.C., in 1963. He received the B.S. and M.S. degrees in electrical engineering from the National Tsing-Hua University, Taiwan, in 1985 and 1987, respectively, and the Ph.D. degree in electrical engineering from University of Maryland, College Park, in 1994.

From 1985 to 1987, he was with Image Processing Laboratory, National Tsing-Hua University, where he worked on the recognition of Chinese characters. During 1989, he was with Star Tech. Inc., where he focused on the development of fingerprint recognition systems. After he received the Ph.D. degree in 1994, he jointed Wan Ta Technology Inc. as a Vice General Manager, conducting several projects on Point-of-Sale systems. In 1995, he became a member of Research Staff with the Advanced Technology Center, Computer and Communication Laboratory, Industrial Technology Research Institute, Taiwan, where he led a project on Java-based Network Managements. Since 1996, he has been an Associate Professor with the Department of Communications Engineering, National Chiao-Tung University, Taiwan, and was promoted to a full Professor in 2001. His research interests generally lie in information and coding theory, large deviation theory, distributed detection, and sensor networks.

Dr. Chen received the annual Research Award from the National Science Council, Taiwan, R.O.C., five years in a row since 1996. He then received the 2000 Young Scholar Paper Award from Academia Sinica, Taiwan. His Experimental Handouts for the course of Communication Networks Laboratory have been awarded as the Annual Best Teaching Materials for Communications Education by the Ministry of Education, Taiwan, R.O.C., in 1998. He was elected Chair of the IEEE Communications Society Taipei Chapter, in 2006 and 2007, during which the IEEE ComSoc Taipei Chapter won the 2007 IEEE ComSoc Chapter Achievement Awards (CAA) and 2007 IEEE ComSoc Chapter of the Year (CoY). He has served as Chairman of the Department of Communications Engineering, National Chiao-Tung University, during 2007–2009. He has been selected as the Outstanding Tutor Teacher of the National Chiao-Tung University in 2002. He was also the recipient of the Distinguished Teaching Award from the College of Electrical and Computer Engineering, National Chiao-Tung University, in 2003.

Yunghsiang S. Han (S'90–M'93–SM'08) was born in Taipei, Taiwan, on April 24, 1962. He received the B.Sc. and M.Sc. degrees in electrical engineering from the National Tsing Hua University, Hsinchu, Taiwan, in 1984 and 1986, respectively, and the Ph.D. degree from the School of Computer and Information Science, Syracuse University, Syracuse, NY, in 1993.

He was a Lecturer with Ming-Hsin Engineering College, Hsinchu, from 1986 to 1988 . He was a Teaching Assistant from 1989 to 1992, and a Research Associate with the School of Computer and Information Science, Syracuse University, from 1992 to 1993. He was an Associate Professor with the Department of Electronic Engineering, Hua Fan College of Humanities and Technology, Taipei Hsien, Taiwan, from 1993 to 1997. He was with the Department of Computer Science and Information Engineering, National Chi Nan University, Nantou, Taiwan, from 1997 to 2004, where he was promoted to Professor in 1998. He was a visiting scholar with the Department of Electrical Engineering, University of Hawaii at Manoa, from June to October 2001, and the SUPRIA Visiting Research Scholar with the Department of Electrical Engineering and Computer Science and CASE center at Syracuse University, from September 2004 to January 2004. He is now with the Graduate Institute of Communication Engineering, National Taipei University, Taipei. His research interests are in error-control coding, wireless networks, and security.

Dr. Han was a winner of the 1994 Syracuse University Doctoral Prize.

Ming-Hsin Kuo was born in Tainan City, Taiwan, R.O.C., in 1984. He received the B.S. and M.S. degrees in communications engineering from the National Chiao-Tung University, Taiwan, in 2006 and 2008, respectively.

He is currently with Pegatron Corporation, Taiwan. His research interests are in channel coding, wireless communication, and networking.