International Symposium on Information Theory and its Applications, ISITA2008
Auckland, New Zealand, 7-10, December, 2008

A Self-Orthogonal Code and Its Maximum-Likelihood Decoder for Combined Channel Estimation and Error Protection

Chia-Lung Wu†, Po-Ning Chen†, Yunghsiang S. Han‡ and Shih-Wei Wang†

† Dept. of Communications Engineering
National Chiao-Tung University
Hsin Chu, Taiwan 30056, ROC
E-mail (C.-L. Wu): clwu@banyan.cm.nctu.edu.tw
E-mail (P.-N. Chen): qponing@mail.nctu.edu.tw
E-mail (S.-W. Wang): swwang.cm94g@nctu.edu.tw

‡ Graduate Institute of Communications Engineering
National Taipei University
Taipei, Taiwan, ROC
E-mail: yshan@mail.ntpu.edu.tw

Abstract

Recent researches have confirmed that better system performance can be obtained by jointly considering channel equalization and channel estimation in the code design, when compared with the system with individually optimized devices. However, the existing codes are mostly searched by computers, and hence, exhibit no good structure for efficient decoding. In this paper, a systematic construction for the codes for combined channel estimation and error protection is proposed. Simulations show that it can yield codes of comparable performance to the computer-searched best codes. In addition, the structural codes can now be maximum-likelihoodly decodable in terms of a newly derived recursive metric for use of the priority-first search decoding algorithm. Thus, the decoding complexity reduces considerably when compared with that of the exhaustive decoder. In light of the rule-based construction, the feasible codeword length that was previously limited by the capability of code-search computers[1] can now be further extended, and therefore facilitates their applications.

1. Background

The channel model considered in this work is the quasi-static block fading channel [7], which can be described by:

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

where $\boldsymbol{y} = [y_1, y_2, \cdots, y_L]^T$ is the received vector, $\boldsymbol{h} = [h_1, h_2, \cdots h_P]^T$ is the channel coefficient vector, $\boldsymbol{n} = [n_1, n_2, \cdots, n_L]^T$ is i.i.d. zero-mean complex Gaussian distributed with $E[|n_j|^2] = \sigma_n^2$, and

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

is formed by the P shift counterparts of system input $\boldsymbol{b} = [b_1, \cdots, b_N]^T$ with each $b_j \in \{\pm 1\}$. It can then be understood from (1) that in absence of the additive noise $\boldsymbol{n}$, the system output $\boldsymbol{y}$ is equal to the convolution of the system input $\boldsymbol{b}$ and the channel coefficients $\boldsymbol{h}$, and the channel coefficients remain constant within the block of length $L = N + P - 1$. Throughout the paper, we assume that $N \geq P$.

Based on the system model, the least square estimate of channel coefficients $\boldsymbol{h}$ for a given $\boldsymbol{b}$ (interchangeably, $\mathbb{B}$) is given by

$$\hat{\boldsymbol{h}} = (\mathbb{B}^T\mathbb{B})^{-1}\mathbb{B}^T\boldsymbol{y},$$

and the joint maximum-likelihood (ML) decision on the transmitted codeword becomes:

$$\boldsymbol{b}_{\text{ML}} = \arg\min_{\boldsymbol{b}\in\mathcal{C}} \|\boldsymbol{y} - \mathbb{B}\hat{\boldsymbol{h}}\|^2 = \arg\min_{\boldsymbol{b}\in\mathcal{C}} \|\boldsymbol{y}\boldsymbol{y}^H - \mathbb{P}_B\|^2, \tag{2}$$

where $\mathbb{P}_B = \mathbb{B}(\mathbb{B}^T\mathbb{B})^{-1}\mathbb{B}^T$, and $\mathcal{C}$ is the channel code. Notably, codeword $\boldsymbol{b}$ and transformed codeword $\mathbb{P}_B$ is not one-to-one corresponding unless b_1 is fixed. For convenience, we will set $b_1 = -1$ for the channel codes we construct in the sequel.

This work was supported by the National Science Council of Taiwan, R.O.C., under the projects of NSC 95-2221-E-009-054-MY3 and Aiming for the Top University and Elite Research Center Development Plan..

[1]The best half-rate code search took around 10 days for codeword length $N = 18$, and may lengthen up to one month for $N = 24$.

In literature, no systematic code constructions have been proposed for combined channel estimation and error protection. The existing code constructions were mostly found by computer searches [1, 4, 7]. Efficient-decodability for the perhaps structureless computer-searched codes therefore becomes an engineering challenge.

In 2002, Skoglund et al [7] searched by computers for the binary code suitable for combined estimation and error protection by minimizing the sum of the pairwise error probabilities (PEP) under equal prior, i.e.

$$\frac{1}{2^K}\sum_{\boldsymbol{b}\in\mathcal{C}}\sum_{\hat{\boldsymbol{b}}\in\mathcal{C},\hat{\boldsymbol{b}}\neq\boldsymbol{b}}\Pr\left(\text{decision}=\boldsymbol{b}\,\middle|\,\hat{\boldsymbol{b}}\text{ transmitted}\right), \quad (3)$$

where $\mathcal{C}$ denotes the (N, K) nonlinear block code. Their simulations showed that in comparison with the Hamming error correcting code under perfect channel estimate, evident performance improvement can be obtained by adopting their computer-searched nonlinear code. Although their code was searched at the signal-to-noise ratio (SNR) of 10 dB, they observed that the found code can perform well in a wide range of different SNRs. Later in [1], the PEP in (3) was replaced by a pairwise distance measure between two codewords, and a suboptimal greedy algorithm was proposed in order to speed up the code search process. In 2007, Giese and Skoglund [4] re-applied their original idea to the single- and multiple-antenna systems, and used the asymptotic PEP and the generic gradient-search algorithm in place of the PEP and the simulated annealing algorithm to reduce the system complexity.

In [7], the authors wrote at the end "an important topic for further research is to study how the decoding complexity of the proposed scheme can be decreased." They proceeded to state that along this research line, "one main issue is to investigate what kind of structure should be enforced on the code to allow for simplified decoding." Motivated by these statements, a different approach for code design was taken in this work. Specifically, we first confirmed that the codeword that maximizes the system SNR is orthogonal to its own delayed counterparts. Then, a systematic code design can be conducted based on the self-orthogonality rule. It was subsequently shown by simulations that our systematic constructed code yields comparable performance to the computer-optimized code. As so happened that the computer-searched code in [7] satisfies such rule, its insensitivity to operating SNRs found the theoretical footing. Enforced by the systematic structure of our rule-based constructed codes, we can then derive a recursive ML decoding metric for use of priority-first search decoding algorithm. The decoding complexity is accordingly significantly decreased at moderate-to-high SNRs, when compared with the exhaustive decoder for the structureless computer-searched codes.

It should be mentioned that although the codes searched by computers in [4, 7] target the unknown channels, for which the channel coefficients are assumed unknown constant within a decoding block, the evaluation of the PEP criterion does require to presume certain knowledge of channel statistics. The code constructed based on the rule we proposed however is guaranteed to maximize the system SNR regardless of the statistics of the channels. This hints the potential of self-orthogonal codes to the situation where channel blindness becomes a strict system restriction.

2. Self-Orthogonal Codes

We now begin to present the code design rule that guarantees the maximization of the system SNR regardless of the channel statistics.

A known inequality [6] for the multiplication of two positive semidefinite Hermitian matrices, $\mathbb{A}$ and $\mathbb{B}$, is that

$$\text{tr}(\mathbb{A}\mathbb{B}) \leq \text{tr}(\mathbb{A})\cdot\lambda_{\max}(\mathbb{B}), \quad (4)$$

where $\text{tr}(\cdot)$ represents the matrix trace operation, and $\lambda_{\max}(\mathbb{B})$ is the maximal eigenvalue of $\mathbb{B}$ [5]. The above inequality holds with equality when $\mathbb{B}$ is an identity matrix.

From the system model in (1), it can be derived that the average SNR satisfies:

$$\begin{aligned}\text{Average SNR} &= \frac{E[\|\mathbb{B}\boldsymbol{h}\|^2]}{E[\|\boldsymbol{n}\|^2]}\\ &= \frac{\text{tr}(E[\boldsymbol{h}\boldsymbol{h}^H]\mathbb{B}^T\mathbb{B})}{L\sigma_n^2}\\ &= \frac{N}{L}\frac{1}{\sigma_n^2}\text{tr}\left(E[\boldsymbol{h}\boldsymbol{h}^H]\frac{1}{N}\mathbb{B}^T\mathbb{B}\right)\\ &\leq \frac{N}{L}\frac{1}{\sigma_n^2}\text{tr}(E[\boldsymbol{h}\boldsymbol{h}^H])\lambda_{\max}\left(\frac{1}{N}\mathbb{B}^T\mathbb{B}\right).\end{aligned}$$

Then, the theories on Ineq. (4) lead to that taking

$$\frac{1}{N}\mathbb{B}^T\mathbb{B} = \mathbb{I}_P \triangleq \begin{bmatrix}1 & 0 & \cdots & 0\\ 0 & 1 & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \cdots & 1\end{bmatrix}_{P\times P} \quad (5)$$

will optimize the average SNR regardless of the statistics of $\boldsymbol{h}$ [3].

Existence of codeword sequences satisfying (5) is promised only for $P = 2$ with N odd (and trivially,

$P = 1$). In some other cases such as $P = 3$, one can only design codes to approximately satisfy (5) as:

$$\mathbb{B}^T\mathbb{B} = \begin{bmatrix} N & \pm1 & 0 \\ \pm1 & N & \pm1 \\ 0 & \pm1 & N \end{bmatrix} \text{ for } N \text{ even,}$$

and

$$\mathbb{B}^T\mathbb{B} = \begin{bmatrix} N & 0 & \pm1 \\ 0 & N & 0 \\ \pm1 & 0 & N \end{bmatrix} \text{ for } N \text{ odd.}$$

Owing to this observation, we will relax (5) to allow some off-diagonal entries in $\mathbb{B}^T\mathbb{B}$ to be either 1 or -1 whenever a strict maintenance of (5) is impossible.

Empirical examination by simulated-annealing code-search algorithm shows that for $N = 22$, the best half-rate codes that minimize the sum of PEPs in (3) under[2] complex zero-mean Gaussian distributed $\boldsymbol{h}$ with $E[\boldsymbol{h}\boldsymbol{h}^H] = (1/2)\mathbb{I}_P$ and $P = 2$ all satisfy that

$$\mathbb{B}^T\mathbb{B} = \begin{bmatrix} N & \pm1 \\ \pm1 & N \end{bmatrix}.$$

The operational meaning of the condition $\mathbb{B}^T\mathbb{B} = N \cdot \mathbb{I}_P$ is that the codeword is orthogonal to all of its shifted counterparts, and hence, a space-diversity nature is implicitly enforced. This coincides with the conclusion made in [2] that the training sequence satisfying that $\mathbb{B}^T\mathbb{B}$ is proportional to $\mathbb{I}_P$ can provide optimal channel estimation performance. It should be mentioned that codeword condition (5) has been identified in [4], and the authors in [4, pp. 1591] remarked that a code sequence with certain aperiodic autocorrelation property can possibly be exploited in future code design approaches, which is one of the main research goals of this paper.

Next, we will construct a code tree, consisting of 2^K codewords of length N, which are self-orthogonal in the sense that $\mathbb{B}^T\mathbb{B} = \mathbb{H}$ for some $\mathbb{H}$ that well-approximates the identity matrix.

A code tree of a (N, K) binary code represents every codeword as a path on a binary tree. The code tree consists of $(N + 1)$ levels. The single leftmost node at level zero is usually called the *origin node*. There are at most two branches leaving each node at each level. The 2^K rightmost nodes at level N are called the *terminal nodes*.

Each branch on the code tree is labeled with the appropriate code bit b_i. As a convention, the path from the single origin node to one of the 2^K terminal nodes is termed the *code path* corresponding to the codeword. Since there is a one-to-one correspondence between the codeword and the code path of $\mathcal{C}$, a codeword can be interchangeably referred to by its respective code path or the branch labels that the code path traverses. Similarly, for any node in the code tree, there exists a unique path traversing from the single original node to it; hence, a node can also be interchangeably indicated by the path (or the path labels) ending at it. We can then denote the path ending at a node at level ℓ by the branch labels $[b_1, b_2, \cdots, b_\ell]$ it traverses. For convenience, we abbreviates $[b_1, b_2, \cdots, b_\ell]^T$ as $\boldsymbol{b}_{(\ell)}$, and will drop the subscript when $\ell = N$. The successor pathes of a path $\boldsymbol{b}_{(\ell)}$ are those whose first ℓ labels are exactly the same as $\boldsymbol{b}_{(\ell)}$.

By denoting by $\mathcal{A}(\boldsymbol{b}_{(\ell)})$ the set of all possible ±1-sequences of length N, whose first ℓ bits equal $b_1, b_2, \cdots, b_\ell$ and whose $\mathbb{B}$-representation satisfies $\mathbb{B}^T\mathbb{B} = \mathbb{H}$, we propose the encoding algorithm below.

Step 1. Fix $b_1 = -1$, *and let* $\Delta = \left\lfloor \frac{|\mathcal{A}(b_{(1)})|-1}{2^K-1} \right\rfloor$. *Find* 2^K *codewords of the* (N, K) *code by repeating Steps 2–4 for* $0 \le i \le 2^K - 1$.

Step 2. Let $\underline{\rho} = 0$, $\rho = i \times \Delta$ *and* $\bar{\rho} = |\mathcal{A}(b_{(1)})| - 1$.

Step 3. For $\ell = 2$ *to* N, *assign* b_ℓ *and adjust* $\underline{\rho}$ *and* $\bar{\rho}$ *according to that if* $\rho < \underline{\rho} + \gamma$, *then* $\bar{\rho} = \underline{\rho} + \gamma - 1$ *and* $b_\ell = -1$*; else,* $\underline{\rho} = \underline{\rho} + \gamma$ *and* $b_\ell = +1$, *where* $\gamma = |\mathcal{A}(\boldsymbol{b}_{(\ell-1)}, b_\ell = -1)|$.

Step 4. Retain the i*th codeword* $\boldsymbol{b}$, *and goto Step 2 for the next codeword until all* 2^K *codewords are retained.*

The basic idea behind the code construction algorithm above is that it is reasonable from (2) that a good code should exist at the time $\min_{\boldsymbol{b}\neq\tilde{\boldsymbol{b}}} \|\mathbb{P}_B - \tilde{\mathbb{P}}_B\|^2$ is large. Hence, a uniform pick among all ±1-sequences that are self-orthogonal should result in a good code as confirmed later by simulations.

3. Maximum-Likelihood Decoding of the Self-Orthogonal Codes

By denoting $\mathbb{D} = (\mathbb{B}^T\mathbb{B})^{-1} = \mathbb{H}^{-1}$, and letting the matrix entry of $\mathbb{D}$ be $\delta_{i,j}$, the joint ML decision in (2) can be reformulated as:

$$\begin{aligned} \boldsymbol{b}_{\text{ML}} &= \arg\min_{\boldsymbol{b}\in\mathcal{C}} -\text{tr}(\mathbb{P}_B \boldsymbol{y}\boldsymbol{y}^H) \\ &= \arg\min_{\boldsymbol{b}\in\mathcal{C}} -\text{tr}(\mathbb{B}\mathbb{D}\mathbb{B}^T \boldsymbol{y}\boldsymbol{y}^H) \\ &= \arg\min_{\boldsymbol{b}\in\mathcal{C}} \frac{1}{2}\left[\sum_{m=1}^{N}\sum_{n=1}^{N}(-w_{m,n} b_m b_n)\right], \end{aligned}$$

[2]The adopted statistical parameters of $\boldsymbol{h}$ follow those in [7].

where

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

By adding a constant, independent of the codeword $\boldsymbol{b}$, the ML decision remains unchanged. Hence,

$$\boldsymbol{b}_{\mathrm{ML}} = \arg\min_{\boldsymbol{b}\in\mathcal{C}} g(\boldsymbol{b}), \tag{6}$$

where

$$g(\boldsymbol{b}) \triangleq \sum_{m=1}^{N} \left(\sum_{n=1}^{m-1} |w_{m,n}| + \frac{1}{2}|w_{m,m}| \right) - \frac{1}{2} \sum_{m=1}^{N} \sum_{n=1}^{N} w_{m,n} b_m b_n.$$

We then found that $g(\boldsymbol{b})$ can be obtained in a recursive fashion through

$$g(\boldsymbol{b}_{(\ell+1)}) = g(\boldsymbol{b}_{(\ell)}) + \alpha_{\ell+1} - b_{\ell+1} \sum_{i=0}^{P-1} \sum_{j=0}^{P-1} \delta_{i,j} \mathrm{Re}\left\{ y_{\ell+i+1} \cdot u_j(\boldsymbol{b}_{(\ell+1)}) \right\},$$

for $1 \leq \ell \leq N-1$, where

$$\alpha_{\ell+1} \triangleq \sum_{n=1}^{\ell} |w_{\ell+1,n}| + \frac{1}{2}|w_{\ell+1,\ell+1}| \tag{7}$$

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

$$u_j(\boldsymbol{b}_{(\ell+1)}) = u_j(\boldsymbol{b}_{(\ell)}) + \frac{1}{2}\left(b_\ell y^*_{\ell+j} + b_{\ell+1} y^*_{\ell+1+j} \right).$$

As a result, $g(\boldsymbol{b}_{(\ell+1)})$ and $\{u_j(\boldsymbol{b}_{(\ell+1)})\}_{0\leq j\leq P-1}$ can be computed from the previous $g(\boldsymbol{b}_{(\ell)})$ and $\{u_j(\boldsymbol{b}_{(\ell)})\}_{0\leq j\leq P-1}$ with the knowledge of $y_{\ell+1}$, $y_{\ell+2}$, $\cdots$, $y_{\ell+P}$ and $b_{\ell+1}$, and the initial condition satisfies that $g(\boldsymbol{b}_{(0)}) = u_j(\boldsymbol{b}_{(0)}) = b_0 = 0$ for $0 \leq j \leq P-1$. Notably, although the computation burden of α_ℓ in (7) increases linearly with ℓ, such a linearly growing load can be moderately compensated by the fact that α_ℓ is only necessary to compute it once for each ℓ, because it can be shared for all paths ending at level ℓ over the code tree. It accordingly remains to prove that when guided by the recursive $g(\boldsymbol{b}_{(\ell)})$, the priority-first search over the code tree as put in the following algorithm guarantees to locate the ML code path $\boldsymbol{b}_{\mathrm{ML}}$.

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

Step 2. *Evaluate the g-function values of the successor paths of the current top path in the Stack, and 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 ascending g-function values of them.*

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.*

Suppose that $\hat{\boldsymbol{b}}$ is the first top path that reaches a terminal node, and hence, is the output code path of the above algorithm. Then, *Step 3* of the algorithm ensures that $g(\hat{\boldsymbol{b}})$ is no larger than the g-function values of any other paths currently in the Stack. Since g is non-decreasing along every path in the code tree, i.e. $g(\boldsymbol{b}_{(\ell+1)}) \geq g(\boldsymbol{b}_{(\ell)})$, and since any other code paths should be the offspring of some path $\boldsymbol{b}_{(\ell)}$ existing in the Stack, we conclude that $g(\hat{\boldsymbol{b}})$ should have the smallest g-function value among all code paths, and thus, from (6), $\hat{\boldsymbol{b}} = \boldsymbol{b}_{\mathrm{ML}}$.

4. Simulation Results

In this section, the performance of the self-orthogonal codes proposed in Section 2 is examined. For ease of comparison, the channel parameters used in our simulations follow those in [7], where $\boldsymbol{h}$ is complex zero-mean Gaussian distributed with $E[\boldsymbol{h}\boldsymbol{h}^H] = (1/P)\mathbb{I}_P$ and $P = 2$.

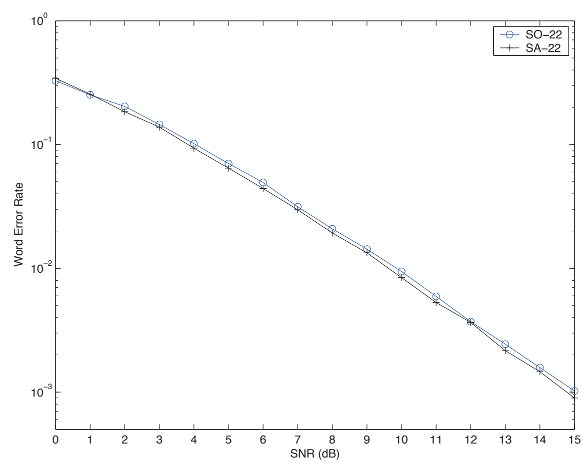

Figure 1: The ML word error rates of the SA-22 code and the SO-22 code.

There are two half-rate codes of length 22 simulated in Fig. 1: the computer-searched code obtained in [7] (SA-22), and the self-orthogonal code proposed in this work (SO-22). We observe from Fig. 1 that the SO-22 code and the SA-22 code have comparable performance. Actually, extensive simulations in Fig. 2 show that the performance of the rule-based half-rate codes is as good as the computer-searched half-rate codes for all $N > 12$ simulated.

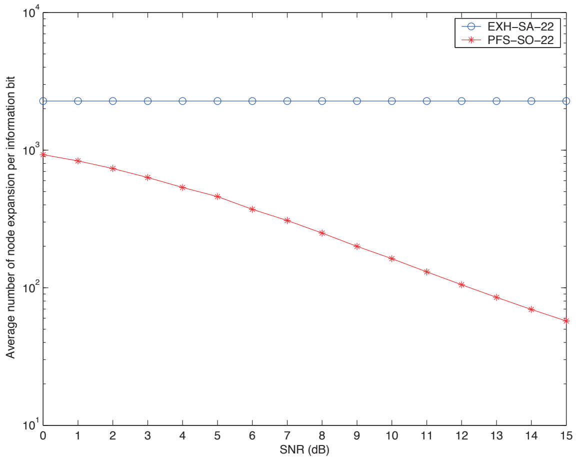

Figure 3: The decoding complexity of the SA-22 code by exhaustive decoder (EXH-SA-22), and the SO-22 code by priority-first search decoding algorithm (PFS-SO-22).

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 nodes expanded is exactly the number of tree branch metrics (i.e. one recursion of g-function values) computed, the equivalent complexity of exhaustive decoder is correspondingly plotted. It can then be observed that in comparison with the exhaustive decoder, a significant reduction in computational burdens can be obtained at moderate-to-high SNRs by adopting the SO-22 code and the priority-first search decoder.

5. Conclusion

In this paper, we established the systematic rule to construct codes based on the optimal signal-to-noise ratio framework that requires every codeword to be self-orthogonal. In terms of the self-orthogonal structure, we further derive a recursive ML metric for the tree-based priority-first search decoding algorithm, and hence, avoid the use of the time-consuming exhaustive decoder that was previously used in [1, 4, 7] to decode the structureless computer-optimized codes. Empirical examinations indicate that the rule-based codes we constructed has almost identical performance to the computer-optimized codes, but its decoding complexity, as anticipated, is much lower than the exhaustive decoder.

Acknowledgement

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 computer-searched codes for further study in this paper.

References

[1] O. Coskun and K. M. Chugg, "Combined coding and training for unknown ISI channels," *IEEE Trans. Commun.*, vol. 53, no. 8, pp. 1310-1322, August 2005.

[2] S. N. Crozier and 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, August 1991.

[3] G. Ganesan and P. Stoica, "Space-time block codes: A maximum SNR approach," *IEEE Trans. Inform. Theory*, vol. 47, no. 4, pp. 1650-1659, May 2001.

[4] J. Giese and M. Skoglund, "Single- and multiple-antenna constellations for communication over unknown frequency-selective fading channels," *IEEE Trans. Inform. Theory*, vol. 53, no. 4, pp. 1584-1594, April 2007.

[5] D. Harville, *Matrix Algebra From a Statistician's Perspective*, 1st edition, Springer, 2000.

[6] R. Patel and M. Toda, "Trace inequalities involving Hermitian metrices," *Linear Alg. Its Applic.*, vol. 23, pp. 13-20, 1979.

[7] M. Skoglund, J. Giese and S. Parkvall, "Code design for combined channel estimation and error protection," *IEEE Trans. Inform. Theory*, vol. 48, no. 5, pp. 1162-1171, May 2002.

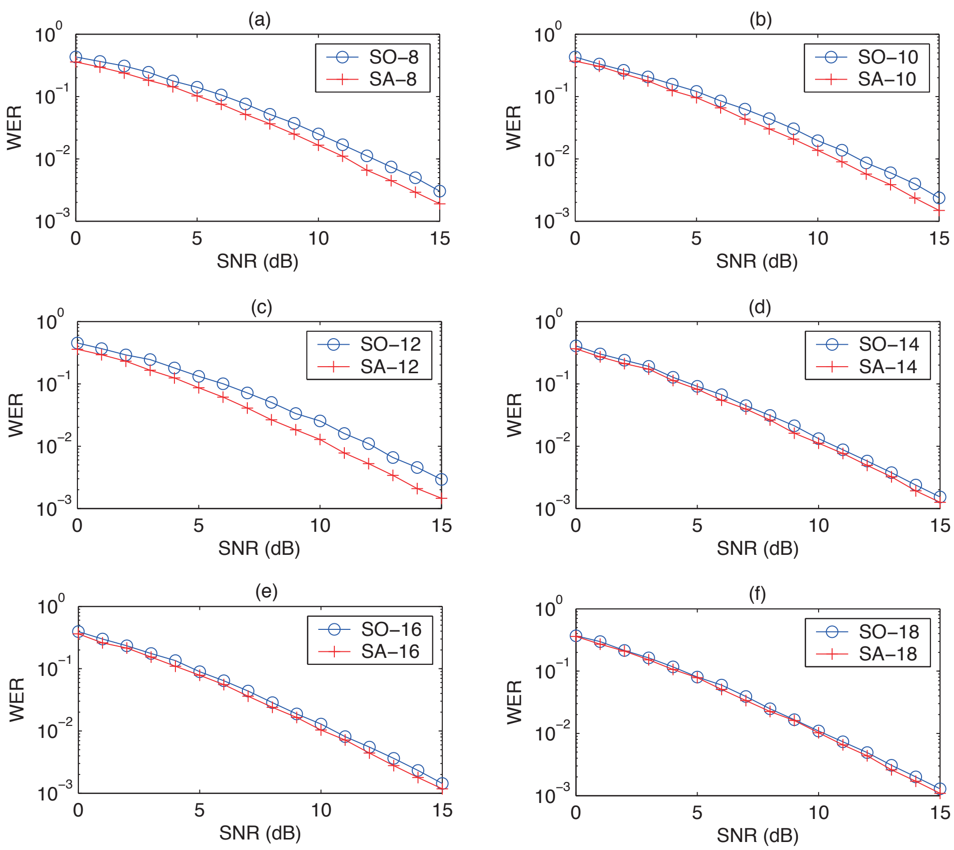

Figure 2: The ML word error rates (WERs) of the computer-searched half-rate codes by simulated annealing (SA-N) and the rule-based half-rate codes (SO-N).