Transactions Papers

Efficiently Decoded Full-Rate Space-Time Block Codes

Don Torrieri, *Senior Member, IEEE*, and Matthew C. Valenti, *Senior Member, IEEE*

***Abstract*—Space-time block codes with orthogonal structures typically provide full-diversity reception and simple receiver processing. However, rate-1 orthogonal codes for complex constellations have not been found for more than two transmit antennas. By using a genetic algorithm, rate-1 space-time block codes that accommodate very simple receiver processing at the cost of reduced diversity are designed in this paper for more than two transmit antennas. Simulation results show that evolved codes combined with efficient outer codes provide better performance over fading channels than minimum-decoding-complexity quasi-orthogonal codes at typical operating signal-to-noise ratios. When the fading is more severe than Rayleigh fading, the spectral efficiency is specified, and an efficient outer code is used, evolved codes outperform orthogonal space-time block codes.**

***Index Terms*—Space-time block codes, genetic algorithm, diversity, quasi-orthogonal codes.**

I. Introduction

SPACE-TIME codes transmitted by multiple antennas improve the performance of a communication system in a fading environment without the need for multiple receive antennas or channel-state information at the transmitter [1], [2]. An orthogonal space-time block code (STBC), such as the Alamouti code, provides full diversity at full transmission rate and maximum-likelihood decoding that entails only linear processing. However, rate-1 orthogonal STBCs for complex constellations exist only for two transmit antennas. Orthogonal STBCs for more than two transmit antennas require a code rate that is less than unity [2], which implies a reduced spectral efficiency.

When used with rotated constellations, quasi-orthogonal (QO) [3], [4], minimum-decoding-complexity quasi-orthogonal (MDC-QO) [5], [6] and coordinate-interleaved [7] STBCs can provide full diversity at full rate but require more complex decoding than the decoupled decoding of each real-valued symbol that is possible with orthogonal STBCs. In this paper, rate-1 linear dispersion codes [8] that are nearly orthogonal STBCs are generated by a genetic algorithm [9]. Since the evolved codes are nearly orthogonal, simple suboptimal decoupled decoding that would be optimal for orthogonal codes causes only a minor increase in intersymbol interference and reduction of diversity gain and, hence, a performance loss only at high signal-to-noise ratios. The MDC-QO STBCs [6] offer full diversity when used with a maximum-likelihood decoder that can decouple complex symbols. Compared with these codes, the evolved codes presented in this paper have a simplified implementation in both the decoder (decoupling of real symbols) and the modulator (no constellation rotation). As shown subsequently, when the spectral efficiency is specified, an efficient outer code is used, and the fading is severe, the evolved codes provide better performance than both rate-1 MDC-QO and orthogonal STBCs. Other applications of genetic algorithms to the design of different types of space-time block codes may be found in the literature [10]–[13].

Section II defines linear dispersion STBCs and the requirements of decoupled decoding. Section III describes the details and the options of the genetic algorithm. In Section IV, the algorithm optimization and the performance comparisons of various STBCs are presented. The Appendix lists the dispersion matrices of several discrete-alphabet evolved codes.

II. STBC Requirements

Let N_T denote the number of transmit antennas, N denote the number of distinct transmitted constellation symbols, and L denote the length of a space-time codeword. The space-time code rate is N/L, the number of information symbols conveyed per signaling interval. A transmitted symbol that belongs to a complex signal constellation is represented as

$$x_n = x_n^r + jx_n^i, \ 1 \le n \le N \tag{1}$$

where x_n^r and x_n^i are the real and imaginary components of the symbol, respectively, and $j = \sqrt{-1}$. The $L \times N_T$ transmission matrix representing a transmitted codeword of a linear dispersion STBC may be expanded as

$$\mathbf{G} = \sum_{n=1}^{N} a_n \mathbf{A}_n + j \sum_{n=1}^{N} b_n \mathbf{B}_n \tag{2}$$

where each a_n and b_n is a real or imaginary component of the symbols and the *dispersion matrices* $\mathbf{A}_n$ and $\mathbf{B}_n$

Paper approved by N. Benvenuto, the Editor for Modulation and Detection of the IEEE Communications Society. Manuscript received January 3, 2009; revised July 22, 2009.

D. Torrieri is with the US Army Research Laboratory, Adelphi, MD (e-mail: dtorr@arl.army.mil).

M. C. Valenti is with West Virginia University, Morgantown, WV (e-mail: mvalenti@wvu.edu).

Portions of this paper were presented at the IEEE Military Communications Conference (MILCOM), San Diego, CA, Nov. 2008. M.C. Valenti's contribution was sponsored by the West Virginia University Research Corporation and the National Science Foundation under Award No. CNS-0750821.

Digital Object Identifier 10.1109/TCOMM.2010.02.090003

0090-6778/10$25.00 © 2010 IEEE

have real-valued elements. Each space-time encoder linearly combines weighted real and imaginary components of N symbols and produces a sequence of L complex numbers. Each real and imaginary component of each of these complex numbers becomes a separate in-phase and quadrature component, respectively, of a modulated carrier. In contrast to most orthogonal STBCs, linear dispersion codes have transmission matrices that are not restricted to complex symbols and their complex conjugates with the same magnitudes.

Example 1: Consider the MDC-QO rate-1 STBC that maps $N = 4$ information symbols into $N_T = 4$ antenna outputs over $L = 4$ signaling intervals [6]. The coefficients in (2) are $a_1 = x_1^r$, $a_2 = x_2^r$, $a_3 = x_1^i$, $a_4 = x_2^i$, $b_1 = x_3^r$, $b_2 = x_4^r$, $b_3 = x_3^i$, and $b_4 = x_4^i$. The transmission matrix is

$$\mathbf{G} = \begin{bmatrix} x_1^r + jx_3^r & x_2^r + jx_4^r & -x_1^i + jx_3^i & -x_2^i + jx_4^i \\ -x_2^r + jx_4^r & x_1^r - jx_3^r & x_2^i + jx_4^i & -x_1^i - jx_3^i \\ -x_1^i + jx_3^i & -x_2^i + jx_4^i & x_1^r + jx_3^r & x_2^r + jx_4^r \\ x_2^i + jx_4^i & -x_1^i - jx_3^i & -x_2^r + jx_4^r & x_1^r - jx_3^r \end{bmatrix}.$$

The dispersion matrices are

$$\mathbf{A}_1 = \begin{bmatrix} +1 & 0 & 0 & 0 \\ 0 & +1 & 0 & 0 \\ 0 & 0 & +1 & 0 \\ 0 & 0 & 0 & +1 \end{bmatrix}, \mathbf{B}_1 = \begin{bmatrix} +1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & +1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

$$\mathbf{A}_2 = \begin{bmatrix} 0 & +1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & +1 \\ 0 & 0 & -1 & 0 \end{bmatrix}, \mathbf{B}_2 = \begin{bmatrix} 0 & +1 & 0 & 0 \\ +1 & 0 & 0 & 0 \\ 0 & 0 & 0 & +1 \\ 0 & 0 & +1 & 0 \end{bmatrix}$$

$$\mathbf{A}_3 = \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix}, \mathbf{B}_3 = \begin{bmatrix} 0 & 0 & +1 & 0 \\ 0 & 0 & 0 & -1 \\ +1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix}$$

$$\mathbf{A}_4 = \begin{bmatrix} 0 & 0 & 0 & -1 \\ 0 & 0 & +1 & 0 \\ 0 & -1 & 0 & 0 \\ +1 & 0 & 0 & 0 \end{bmatrix}, \mathbf{B}_4 = \begin{bmatrix} 0 & 0 & 0 & +1 \\ 0 & 0 & +1 & 0 \\ 0 & +1 & 0 & 0 \\ +1 & 0 & 0 & 0 \end{bmatrix}.$$

■

Only one receive antenna is assumed for simplicity, and the extension of the analysis to multiple receive antennas is straightforward. The complex channel response to transmit antenna ℓ at the sampled demodulator output is

$$h_\ell = h_\ell^r + jh_\ell^i, \ 1 \le \ell \le N_T \tag{3}$$

where h_ℓ^r and h_ℓ^i are the real and imaginary components of h_ℓ, respectively. The channel responses are assumed to be known at the receiver and constant for L symbol periods. Let $\mathbf{z}_n$ denote the $2L \times 1$ vector with its first L elements equal to the real–parts of the responses to a_n, $1 \le n \le N$, or b_{n-N}, $N+1 \le n \le 2N$, and its second L elements equal to the imaginary-parts of the responses to $a_n, 1 \le n \le N$, or b_{n-N}, $N+1 \le n \le 2N$. Thus,

$$\mathbf{z}_n = a_n \mathcal{A}_n \mathbf{h}, \ 1 \le n \le N \tag{4}$$

$$\mathbf{z}_n = b_{n-N} \mathcal{B}_{n-N} \mathbf{h}, \ N+1 \le n \le 2N \tag{5}$$

where $\mathbf{h} = [h_1^r \ h_2^r \cdots h_{N_T}^r \ h_1^i \ h_2^i \cdots h_{N_T}^i]^T$ and $\mathcal{A}_n$ and $\mathcal{B}_n$ are $2L \times 2N_T$ matrices. They have the forms

$$\mathcal{A}_n = \begin{bmatrix} \mathbf{A}_n & \mathbf{0}_{L \times N_T} \\ \mathbf{0}_{L \times N_T} & \mathbf{A}_n \end{bmatrix}, \ \mathcal{B}_n = \begin{bmatrix} \mathbf{0}_{L \times N_T} & -\mathbf{B}_n \\ \mathbf{B}_n & \mathbf{0}_{L \times N_T} \end{bmatrix}. \tag{6}$$

where $\mathbf{0}_{L \times N_T}$ is the $L \times N_T$ matrix of zeros. Let $\mathbf{d} = [a_1 \ a_2 \cdots a_N \ b_1 \ b_2 \ldots b_N]^T$. The $2L \times 1$ vector of received symbols in the absence of noise is

$$\mathbf{y}_s = \sum_{n=1}^{2N} \mathbf{z}_n = \mathbf{Hd} \tag{7}$$

where

$$\mathbf{H} = [\mathcal{A}_1 \mathbf{h} \ \mathcal{A}_2 \mathbf{h} \cdots \mathcal{A}_N \mathbf{h} \ \mathcal{B}_1 \mathbf{h} \ \mathcal{B}_2 \mathbf{h} \cdots \mathcal{B}_N \mathbf{h}] \tag{8}$$

is the $2L \times 2N$ channel matrix.

For full-diversity reception and simple (decoupled) maximum-likelihood processing in the receiver, an *orthogonality condition* must be satisfied:

$$\mathbf{H}^T \mathbf{H} = ||\mathbf{h}||^2 \mathbf{I}_{2N \times 2N} \tag{9}$$

for any vector $\mathbf{h}$ with real-valued components, where $||\mathbf{h}||$ denotes the Euclidean norm of $\mathbf{h}$ and $\mathbf{I}_{2N \times 2N}$ is the $2N \times 2N$ identity matrix. The orthogonality condition is satisfied if and only if

$$\mathbf{A}_n^T \mathbf{A}_n = \mathbf{B}_n^T \mathbf{B}_n = \mathbf{I}_{N_T \times N_T}, 1 \le n \le N \tag{10}$$

$$\mathbf{A}_n^T \mathbf{A}_\ell + \mathbf{A}_\ell^T \mathbf{A}_n = \mathbf{B}_n^T \mathbf{B}_\ell + \mathbf{B}_\ell^T \mathbf{B}_n = \mathbf{0}_{N_T \times N_T}, \quad n \ne \ell, \ 1 \le n, \ell \le N \tag{11}$$

$$\mathbf{A}_n^T \mathbf{B}_\ell = \mathbf{B}_\ell^T \mathbf{A}_n, \ n, \ell \le N. \tag{12}$$

A compact proof using linear algebra that $(10) - (12)$ are both necessary and sufficient for the orthogonality condition can be found in [14]. An alternative, much more complicated proof, can be derived from the theory of amicable orthogonal designs [2], [15]. Equation (10) also ensures that the total power transmitted for symbol-component a_n or b_n by each transmit antenna is equal to a_n^2 or b_n^2, respectively. Thus, the total power transmitted by each antenna is equal. The dispersion matrices of Example 1 do not satisfy (11) and are therefore not orthogonal.

Let $\mathcal{E}_s$ denote the average energy per symbol transmitted by all the transmit antennas and $\mathcal{E}_b$ the average energy per data bit transmitted by all the antennas. In the presence of additive white Gaussian noise with spectral-density N_0, the vector of received symbols is

$$\mathbf{y}_r = \mathbf{Hd} + \mathbf{n} \tag{13}$$

where $\mathbf{n}$ is the zero-mean noise with covariance matrix $E\left[\mathbf{nn}^T\right] = N_0 \mathbf{I}$, $E\left[|d_n|^2\right] = \mathcal{E}_s / N_T$ for each component d_n of $\mathbf{d}$, and E[$\cdot$] denotes the expected value. If the orthogonality condition is satisfied, then the receiver computes the $2N \times 1$ vector

$$\mathbf{y} = ||\mathbf{h}||^{-2} \mathbf{H}^T \mathbf{y}_r = \mathbf{d} + \mathbf{n}_1 \tag{14}$$

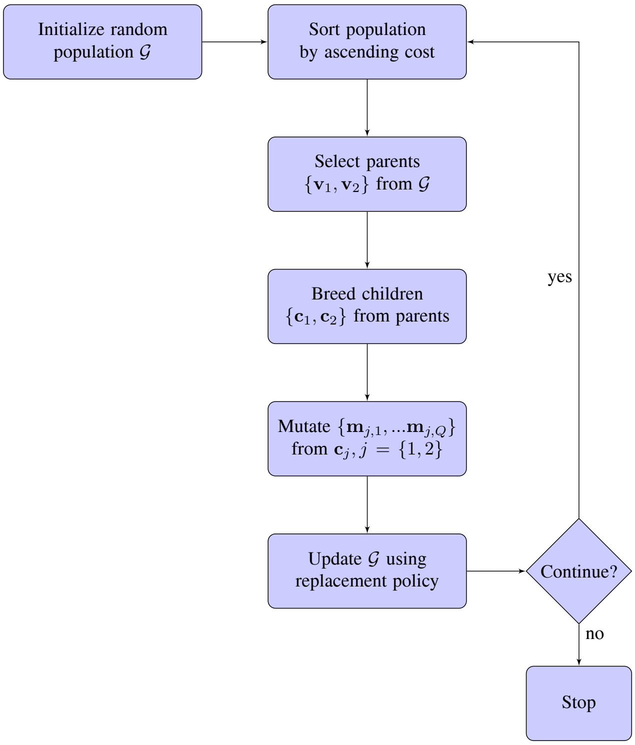
where $E\left[\mathbf{n}_1 \mathbf{n}_1^T\right] = N_0 ||\mathbf{h}||^{-2} \mathbf{I}$. The maximum-likelihood decision for d_n is separately obtained by finding the value of d_n that minimizes $|y_n - d_n|$, $n = 1, 2, \ldots, 2N$, and hence is decoupled from the other component decisions.

For *decoupled decoding* whether or not the orthogonality condition is satisfied, the receiver computes $\mathbf{y} = \mathbf{H}^T\mathbf{y}_r$ and then separately finds the value of d_n that minimizes $|y_n - D_{n,n}d_n|$, $n = 1, 2, \ldots, 2N$, where $\mathbf{D} = \mathbf{H}^T\mathbf{H}$. Equivalently, if $D_{n,n} \neq 0$, for all n, the receiver computes the $2N \times 1$ vector $\mathbf{y} = \mathbf{\Delta}\mathbf{H}^T\mathbf{y}_r = \mathbf{d} + \mathbf{z} + \mathbf{n}_1$, where $\mathbf{\Delta}$ denotes a diagonal matrix with diagonal elements equal to $1/D_{n,n}$, $n = 1, 2, \ldots, 2N$, $E\left[\mathbf{n}_1\mathbf{n}_1^T\right] = N_0\mathbf{\Delta}\mathbf{D}\ \mathbf{\Delta}^T$, and $\mathbf{z}$ is the intersymbol interference that reduces the diversity gain. The decision for d_n is separately obtained by finding the value of d_n that minimizes $|y_n - d_n|$, $n = 1, 2, \ldots, 2N$. The intersymbol interference diminishes as $\mathbf{H}$ approaches orthogonality.

The evolved codes are decoded using the same decoupled decoder used with orthogonal codes, and therefore enjoy the same low complexity, as the burden of calculating and multiplying by each D_n is minor. As an example, consider a rate-1 code for N_T antennas using square quadrature amplitude modulation (QAM) with M symbols. The decoupled decoder independently detects each of the $2L$ real-valued symbols, and each decision requires the comparison of a real value against $\sqrt{M}$ hypotheses. The decoder for a quasi-orthogonal code [3], [4] independently detects two pairs of complex symbols, and each decision requires the comparison of a pair of complex values against M^2 hypothesis. Thus, quasi-orthogonal codes requires a factor of $M\sqrt{M}$ more real-valued comparisons than the proposed evolved codes. The MDC-QO codes [5], [6] allow the L complex symbols to be independently detected, and each decision requires the comparison of a complex value against M hypothesis. Thus, the MDC-QO code requires a factor of $\sqrt{M}$ more real-valued comparisons than the evolved codes.

III. Description of the Genetic Algorithm

In a genetic algorithm, a population of *individuals*, each containing a collection of *genes*, evolves through the process of *breeding* and *mutation* [9]. The definition of an individual and its genes is application specific. When evolving linear dispersion codes, each individual corresponds to the set of $2N$ dispersion matrices associated with a particular code design. We consider linear dispersion codes with entries drawn from a finite alphabet and require that the columns of all dispersion matrices have unit norms[1]. Although the genetic algorithm was run with several different discrete alphabets, the best results were generated using ternary alphabets with either entries $\{0, \pm 1\}$ or entries $\{0, \pm 1/\sqrt{2}\}$. For such codes, the number of permissible unit-norm columns is a finite value J. For the alphabet $\{0, \pm 1\}$, $J = 2L$, and for the alphabet $\{0, \pm 1/\sqrt{2}\}$, $J = 2L(L-1)$. The palette of columns can be represented as $\mathcal{Z} = \{\mathbf{b}_n, 1 \leq n \leq J\}$ where $\mathbf{b}_n$ is the n^{th} possible column. In our algorithm, each gene represents one column of a dispersion matrix. As the $2N$ dispersion matrices each have N_T columns, each individual has $N_G = 2N_TN$ genes, and the number of distinct individuals is J^{N_G}. The vector $\mathbf{g}_k = [g_{k,1}, ..., g_{k,N_G}]$ contains the genes of the k^{th} individual. Gene $g_{k,j}$ can represent column $\mathbf{b}_n \in \mathcal{Z}$ by simply setting $g_{k,j} = n$, i.e. the index of the column within set $\mathcal{Z}$.



Initialize random population $\mathcal{G}$
Sort population by ascending cost
Select parents $\{\mathbf{v}_1, \mathbf{v}_2\}$ from $\mathcal{G}$
Breed children $\{\mathbf{c}_1, \mathbf{c}_2\}$ from parents
Mutate $\{\mathbf{m}_{j,1}, ...\mathbf{m}_{j,Q}\}$ from $\mathbf{c}_j, j = \{1, 2\}$
Update $\mathcal{G}$ using replacement policy
Continue?
yes
no
Stop

Fig. 1. Flowchart describing the genetic algorithm.

Each individual is assigned a cost, and the objective of the genetic algorithm is to evolve individuals with low cost. For our goal of near-orthogonality, the cost of individual $\mathbf{g}$ is defined as

$$\begin{aligned} C(\mathbf{g}) \quad = \quad & w_1 \sum_{n=1}^{N} \left[\left\| \mathbf{A}_n^T\mathbf{A}_n - I \right\|^2 + \left\| \mathbf{B}_n^T\mathbf{B}_n - I \right\|^2 \right] \\ + w_2 \sum_{n=1}^{N} & \sum_{\substack{\ell=1 \\ \ell \neq n}}^{N} \left[\left\| \mathbf{A}_n^T\mathbf{A}_\ell + \mathbf{A}_\ell^T\mathbf{A}_n \right\|^2 + \left\| \mathbf{B}_n^T\mathbf{B}_\ell + \mathbf{B}_\ell^T\mathbf{B}_n \right\|^2 \right] \\ + w_3 \sum_{n=1}^{N} & \sum_{\ell=1}^{N} \left\| \mathbf{A}_n^T\mathbf{B}_\ell - \mathbf{B}_\ell^T\mathbf{A}_n \right\|^2 \end{aligned} \tag{15}$$

where $\{\mathbf{A}_n, \mathbf{B}_n, 1 \leq n \leq N\}$ is the set of dispersion matrices represented by the vector $\mathbf{g}$, $\| \cdot \|$ denotes the Frobenius norm, and $\mathcal{W} = \{w_1, w_2, w_3\}$ is a set of weights. This cost function penalizes STBC designs with matrices that do not satisfy the orthogonality conditions of $(10) - (12)$. The values of $\mathcal{W}$ allow some constraints to be emphasized more than others. We considered several different values for $\mathcal{W}$ but found that $\mathcal{W} = \{1, 1, 1\}$ produced the most effective designs. Note that the ordering of the matrices $\mathbf{A}_n$ or the ordering of the matrices $\mathbf{B}_n$ may be changed without changing the cost.

Fig. 1 is a flowchart describing the operation of the genetic algorithm. The first step of the algorithm is to create an initial population of K individuals, $\mathcal{G} = \{\mathbf{g}_1, ..., \mathbf{g}_K\}$, where $K << J^{N_G}$. The initial $\mathbf{g}_k$'s are generated by randomly selecting each entry from the integers 1 through J with uniform

[1] We also evolved linear dispersion codes with entries drawn from a continuous alphabet [14] but did not discover any such codes that performed better than the best discrete-alphabet codes.

probability. This is equivalent to selecting each column of the corresponding dispersion matrices from the set $\mathcal{Z}$ with equal probability. In the very unlikely event that the population contains a pair of identical individuals, the duplicate is discarded and a new individual is randomly generated. For each individual in the population, its cost is calculated using (15). The population is sorted in ascending order of cost, so that $C(\mathbf{g}_k) \geq C(\mathbf{g}_j)$ when $k > j$.

Once the initial population is seeded, the algorithm runs recursively. Each recursion is called a *generation*. During each generation, two individuals are picked at random from the population. These two individuals are *parents* and breed two children. Next, the children *mutate* by having some of their genes changed at random. Finally, the population is updated by replacing up to two individuals with children or mutated children. The whole process continues until a fixed number of generations has been run or some convergence criteria satisfied, such as the population not changing for a certain number of generations. The details of the algorithm are embodied in the application-specific policies for parent selection, breeding, mutation, and replacement, as described below.

Let $\{\mathbf{v}_1, \mathbf{v}_2\}$ be the parents. We considered four parent-selection strategies: (1) *random selection*, (2) *preferred parenting*, (3) *eugenic selection*, and (4) *alpha-male selection*. With random selection, two distinct individuals are picked at random from the population. With preferred parenting, $\mathbf{v}_2$ is picked at random from the entire population excluding the best individual: $\mathbf{v}_2 \in \{\mathbf{g}_k, 2 \leq k \leq K\}$, while $\mathbf{v}_1$ is selected at random from those individuals that are better than the first: $\mathbf{v}_1 \in \{\mathbf{g}_j, 1 \leq j < k\}$. With eugenic selection, the best two individuals are selected: $\mathbf{v}_1 = \mathbf{g}_1$ and $\mathbf{v}_2 = \mathbf{g}_2$. With alpha-male selection, the first parent is the best individual $\mathbf{v}_1 = \mathbf{g}_1$, while the second parent is selected at random from the rest of the population: $\mathbf{v}_2 \in \{\mathbf{g}_k, 2 \leq k \leq K\}$. In our experiments, we found that preferred parenting provided the most effective combination of rapid convergence and low final cost.

The two selected parents *breed* a pair of children $\{\mathbf{c}_1, \mathbf{c}_2\}$. Together, the two children have the same genes as the two parents. However, each child contains some genes from one parent and the remaining genes from the other parent. The genes are selected by first generating a *cross-over mask* $\mathbf{e} = [e_1, ..., e_{N_G}]$ with binary elements, $e_n \in \{0, 1\}$. The vector $\mathbf{e}$ is generated at random such that its entries are i.i.d. Bernoulli variables with $P[e_n = 1] = p_x$. The value p_x is called the *crossover probability*. The two children are generated according to

$$\begin{aligned} c_{1,k} &= (1 - e_k)v_{1,k} + e_k v_{2,k} \\ c_{2,k} &= e_k v_{1,k} + (1 - e_k)v_{2,k}. \end{aligned} \tag{16}$$

Thus, child $\mathbf{c}_j, j \in \{1, 2\}$, will contain the genes of parent $\mathbf{v}_j$ in those positions that $\mathbf{e}$ is a zero, and will contain the genes of the other parent in those positions that $\mathbf{e}$ is a one.

After breeding, the children are *mutated*. Each child $\mathbf{c}_j$ is transformed into a set of Q mutants, $\{\mathbf{m}_{j,1}, ..., \mathbf{m}_{j,Q}\}$. Mutant $\mathbf{m}_{j,q}$ is derived from $\mathbf{c}_j$ by replacing a few entries of $\mathbf{c}_j$ with new genes selected randomly from the set of possibilities. The choice of which genes to replace is also random, with each gene being mutated with probability p_m. The value p_m is called the *mutation probability,* and genes mutate independently.

The last step of the generation is to update the population by replacing individuals. The number of individuals that are replaced is either zero, one, or two, and the size of the population remains at the fixed value of K after each generation. Two replacement strategies may be used: *normal replacement* or *culling*. With normal replacement, a set $\mathcal{X}_j = \{\mathbf{v}_j, \mathbf{c}_j, \mathbf{m}_{j,1}, ..., \mathbf{m}_{j,Q}\}$ is formed containing a parent and its offspring (its child and the child's Q mutants). Let $\mathbf{g}_j = \mathbf{v}_j$ be the individual in the population that was selected to be the parent. This individual is replaced by

$$\mathbf{g}_j = \arg \min_{\mathbf{x} \in \mathcal{X}_j} C(\mathbf{x}). \tag{17}$$

With this replacement policy, a parent will be replaced by its offspring that has the lowest cost. However, if the parent has a cost lower than its offspring, then it will not be replaced. Equation (17) is run for both individuals that were selected as parents.

With the culling replacement strategy, a set $\mathcal{X} = \{\mathbf{g}_K, \mathbf{c}_1, \mathbf{m}_{1,1}, ..., \mathbf{m}_{1,Q}, \mathbf{c}_2, \mathbf{m}_{2,1}, ..., \mathbf{m}_{2,Q}\}$ is formed containing the worst member of the population ($\mathbf{g}_K$) and all the offspring of both parents. The worst element of the population is replaced by

$$\mathbf{g}_K = \arg \min_{\mathbf{x} \in \mathcal{X}} C(\mathbf{x}). \tag{18}$$

With this replacement strategy, both parents remain in the population, and the best of their offspring is used to replace the worst member of the population (if the best offspring has the lowest cost). Culling after some (but not all) generations is beneficial because it allows unfit individuals to be periodically purged from the population. Rather than always using one policy or the other, our algorithm randomly picks between the two at the conclusion of each generation, with p_c defined to be the probability of using the culling strategy. The value of p_c should be small so that most generations use normal replacement.

After the population has been updated, it is resorted in ascending order of cost. The algorithm then either moves on to the next generation or terminates if the halting condition has been achieved.

IV. Optimization Results

The genetic algorithm described in Section III was used to produce rate-1 designs for $N_T = 3$, 4, and 5 antennas. We experimented with a wide range of parameters for the genetic algorithm and found that the most effective designs were achieved using the following parameters and conditions.

- **Alphabet:** One of two discrete alphabets was used: $\{0, \pm 1\}$ or $\{0, \pm 1/\sqrt{2}\}$.
- **Weights:** The set of weights used in (15) was $\mathcal{W} = [1, 1, 1]$.
- **Population size:** The population contained $K = 400$ individuals.
- **Parent selection:** The *preferred parenting* selection policy was used.

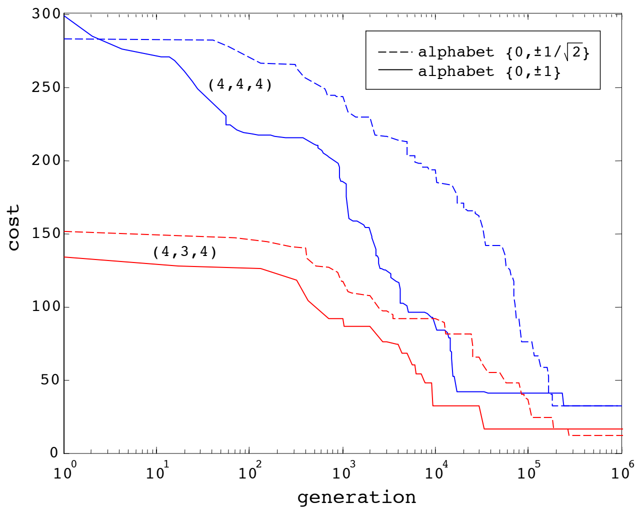


Fig. 2. Cost as a function of generation of evolved (4,3,4) and (4,4,4) codes defined over two discrete alphabets.

- **Number of mutants:** $Q = 2$ mutants were generated per child.
- **Crossover probability:** $p_x = 1/N_G$.
- **Mutation probability:** $p_m = 1/N_G$.
- **Culling probability:** $p_c = 0.01$.

The Appendix lists the dispersion matrices of the best designs found for each of $N_T = 3$, 4, and 5 antennas. For all evolved codes, we use the mapping with each $a_n = x_n^r$ and each $b_n = x_n^i$ in (2). The notation "(a, b, c)" denotes a code with the parameter values $a = N$, $b = N_T$, and $c = L$. The codes for 3 and 4 antennas have $N = L = 4$. For each of the two alphabets, Fig. 2 shows the cost per generation of the best design in the population. After 1 million generations, the cost of the $(4, 3, 4)$ code converges to 12 with alphabet $\{0, \pm 1/\sqrt{2}\}$, and 16 with alphabet $\{0, \pm 1\}$, while cost of the $(4, 4, 4)$ code converges to 32 with both alphabets. The code for 5 antennas has $N = L = 8$ and converges to a cost of 128 after 1.7 million generations. Simulation results confirm that the bit error rate (BER) always improves as the cost decreases.

Fig. 3 shows the BER versus $\mathcal{E}_b/N_0$ of the $(4, 3, 4)$ code with alphabet $\{0, \pm 1/\sqrt{2}\}$ that is obtained by a simulation over a Rayleigh fading channel with quadriphase-shift keying (QPSK). The simulated fading coefficients are constant for blocks of L symbols, but independent from block to block and have zero-mean, unit-variance, complex Gaussian distributions. The figure shows the BER of both maximum-likelihood and decoupled decoding for the best designs obtained after 10 thousand and 1 million generations.

Also shown in Fig. 3 is the performance of the $(4, 3, 4)$ linear dispersion code presented in [8]. As can be seen, the reference code outperforms the evolved code when maximum-likelihood decoding is used. The reason is that the cost function of (15) is not designed to optimize maximum-likelihood performance. However, when the reference code is detected with the simple decoupled decoder, performance is poor and exhibits a relatively high error floor. The reason is that the cost of the reference code is $64w_2$, which is significantly higher than the cost of the evolved code. The designs obtained during early generations of the genetic algorithm exhibit an error floor with decoupled decoding. However, as the design becomes more highly evolved, the decoupled-decoding error floor is lowered. After 1 million generations, the performances of maximum-likelihood and decoupled decoding are identical down to at least a BER of 10^{-6}. While the decoupled-decoding performance improves as the design becomes more highly evolved, the maximum-likelihood performance actually degrades as the design evolves.

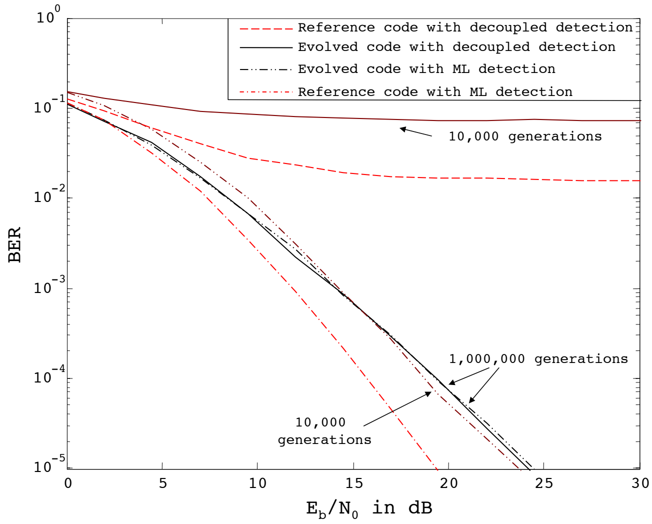


Fig. 3. BERs over a Rayleigh fading channel of maximum-likelihood and decoupled decoding of evolved and reference (4, 3, 4) codes. The evolved code is defined over the discrete alphabet $\{0, \pm 1/\sqrt{2}\}$, and evolved for 10 thousand and 1 million generations. The "reference code" is the (4, 3, 4) code from [8]. QPSK modulation is used.

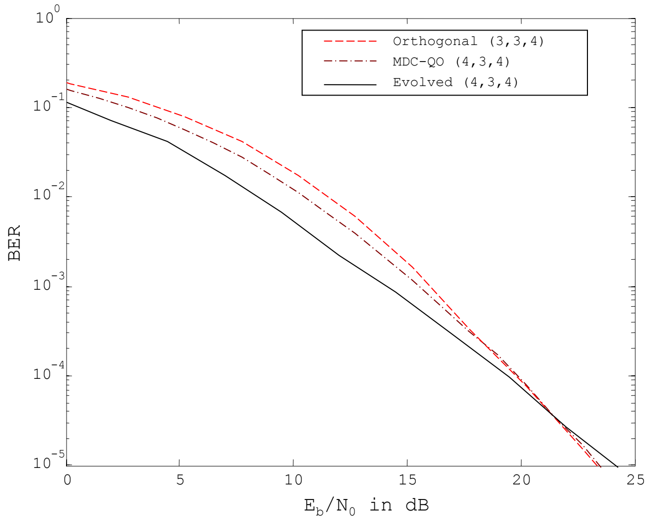


Fig. 4. BERs over a Rayleigh fading channel of (4, 3, 4) codes evolved over the alphabet $\{0, \pm 1/\sqrt{2}\}$ and the alphabet $\{0, \pm 1\}$, the (4, 3, 4) MDCQO code, and the (3, 3, 4) orthogonal code. 8-PSK modulation is used with the (4, 3, 4) codes, and 16-QAM is used with the (3, 3, 4) code.

Fig. 4 compares the BERs over a Rayleigh fading channel of both evolved $(4, 3, 4)$ codes against a rate-3/4 orthogonal $(3, 3, 4)$ code [2] and the MDC-QO $(4, 3, 4)$ code created by deleting the last columns of the dispersion matrices given

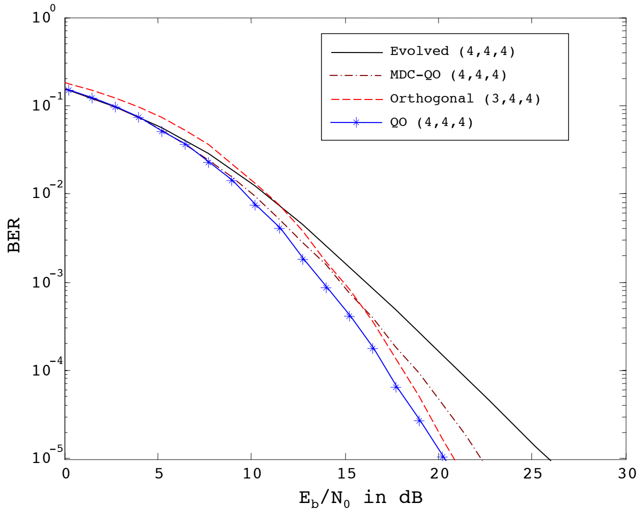


Fig. 5. BERs over a Rayleigh fading channel of the (4, 4, 4) code evolved over the alphabet $\{0, \pm1\}$, the (4, 4, 4) QO code, the (4, 4, 4) MDC-QO code, and the (3, 4, 4) orthogonal code. 8-PSK modulation is used with the (4, 4, 4) codes and 16-QAM is used with the (3, 4, 4) code.

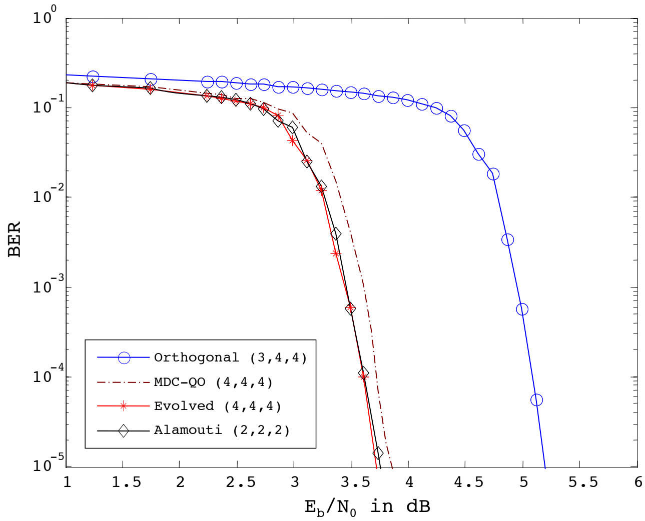


Fig. 6. BERs of the turbo-coded (4, 4, 4) code evolved over the alphabet $\{0, \pm1\}$, (4, 4, 4) MDC-QO code, (2, 2, 2) Alamouti code, and (3, 4, 4) orthogonal code. 8-PSK modulation is used with the full-rate codes and 16-QAM is used with the (3, 4, 4) code. The turbo code's rate is 1/2, and message length is 4500 bits. The fading is Rayleigh.

in Example 1. In order to provide a fair comparison, the spectral efficiency is maintained at 3 bits/s/Hz by using 8-phase-shift keying (8-PSK) for the rate-1 codes and 16-QAM for the rate-3/4 orthogonal code. For the MDC-QO code, the 8-PSK constellation is rotated by 4.9 degrees, which provides full diversity [6]. As can be seen, the evolved rate-1 codes with 8-PSK are better than both MDC-QO with 8-PSK and the orthogonal code with 16-QAM for $\mathcal{E}_b/N_0 < 20$ dB and $BER > 10^{-4}$. Beyond this point, the evolved codes do not perform as well as the other two codes at high $\mathcal{E}_b/N_0$, the signal-to-noise ratio (SNR), indicating that full diversity is advantageous primarily for high SNRs.

Fig. 5 shows the performance over a Rayleigh fading channel of the evolved $(4, 4, 4)$ code with 8-PSK and alphabet $\{0, \pm1\}$, the $(4, 4, 4)$ quasi-orthogonal (QO) code of [3] with 8-PSK and the $\pi/8$ radian constellation rotation described in [4], the $(4, 4, 4)$ MDC-QO code from Example 1 with 4.9-degree rotated 8-PSK, and the rate-3/4 $(3, 4, 4)$ orthogonal code from [2] with 16-QAM. The evolved code shows a loss relative to the QO and MDC-QO codes for $\mathcal{E}_b/N_0 > 5$ dB and $BER < 10^{-1}$. However, the evolved codes are still beneficial even with four transmit antennas for two reasons: First, the evolved code is less complex because, after linear processing, each of the real and imaginary components of each symbol may be demodulated independently (single-component decodable), whereas the MDC-QO code requires joint detection of the real and imaginary components of each symbol (single-symbol decodable) and the QO code requires joint decoding of two (pairwise decodable). Second, the error performance of the evolved code is actually slightly better than MDC-QO code at very low SNR and BERs above 10^{-1} and therefore, as we demonstrate next, will actually perform better than MDC-QO code when both are concatenated with a strong outer channel code.

An MDC-QO code has a special structure that causes a specific constellation rotation to improve its BER at the cost of an increased transmitted peak-to-average power ratio (PAPR) [16]. The evolved codes lack structures that would suggest that rotations would be useful. Simulation results confirm that the best constellation rotations only slightly improve the BERs of the evolved codes at high SNRs and have a negligible effect when outer channel codes are used. Since rotated constellations tend to increase the PAPR, they are not considered further for evolved codes. For constellations in which the symbol components range over the same values, such as PSK and square QAM constellations, the PAPR is the same for all one-to-one coefficient mappings.

When an outer channel code is used, the decoder must be given proper soft input values. Prior to outer decoding, the vector $\mathbf{y} = \mathbf{D}^{-1}\mathbf{H}^T\mathbf{y}_r$ is computed and may be expressed as $\mathbf{y} = [y_1^r, ..., y_N^r, y_1^i, ..., y_N^i]$. For orthogonal and MDC-QO codes, a log-likelihood is produced for each hypothetical group of $2N$ symbols by using the conditional pdf of $\mathbf{y}$ given $\mathbf{d}$, which is Gaussian and given by

$$\log p[\mathbf{y}|\mathbf{d}] \propto -\frac{1}{2N_0}(\mathbf{y}-\mathbf{d})^T\mathbf{D}(\mathbf{y}-\mathbf{d}). \qquad (19)$$

For orthogonal codes, $\mathbf{D}$ is diagonal, and therefore each component of $\mathbf{d}$ may be independently detected. For MDC-QO codes, each real component is correlated with the corresponding imaginary component, and therefore the two components must be jointly detected. For the evolved codes, $\mathbf{D}$ is not diagonal but is almost so. The soft-output demodulator for the evolved codes neglects the correlation among components caused by the nondiagonal $\mathbf{D}$ and detects each component of the vector $\mathbf{d}$ independently. Once the log-likelihoods of the symbols are computed, then they are transformed into the bit log-likelihood ratios that are required by the decoder (see, for instance, equation (6) in [17]).

Fig. 6 shows the performance when an outer code is combined with the evolved full-rate $(4, 4, 4)$ STBC with 8-PSK modulation and alphabet $\{0, \pm1\}$, the $(4, 4, 4)$ MDC-QO

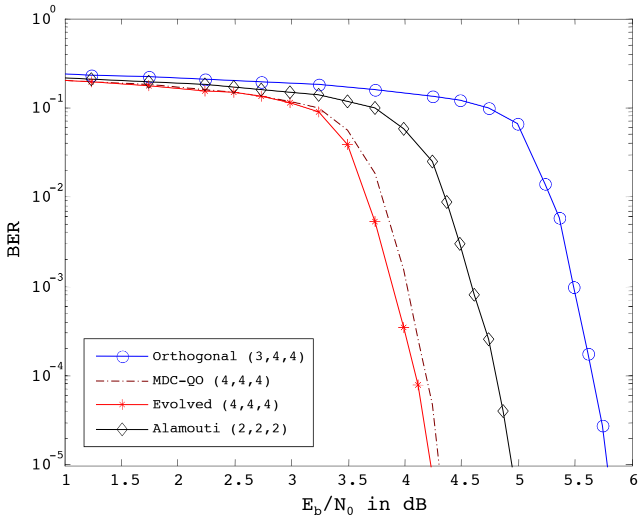


Fig. 7. BERs of the turbo-coded space-time codes described in the caption of Fig. 6 but in Nakagami fading with m = 1/2.

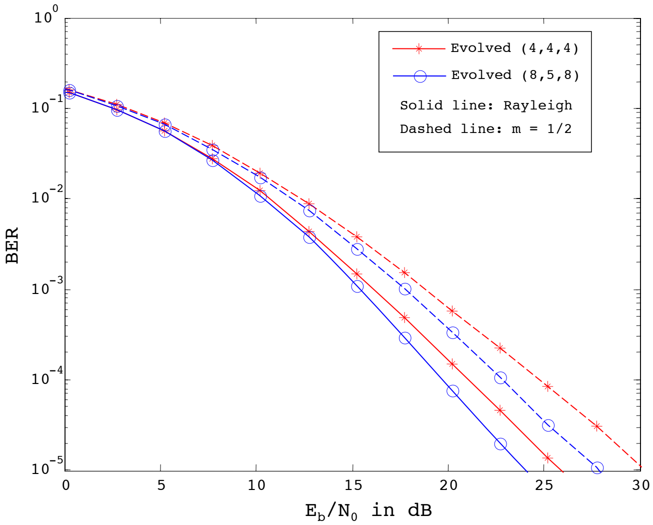


Fig. 8. BERs over Rayleigh and Nakagami-1/2 fading channels of the evolved (8, 5, 8) and (4, 4, 4) codes with 8-PSK. Both codes are evolved over the alphabet $\{0, \pm 1\}$.

code with 4.9-degree rotated 8-PSK, the rate-3/4 orthogonal $(3, 4, 4)$ STBC with 16-QAM, and the full-rate orthogonal Alamouti $(2, 2, 2)$ STBC with 8-PSK modulation. The outer code is the turbo code specified by the UMTS third-generation cellular standard [18]. The code rate is set to $1/2$, and the message length is set to 4500 bits. The fading is Rayleigh and the channel gains are held constant for blocks of 4 consecutive signaling intervals, corresponding to a single $(4, 4, 4)$ or $(3, 4, 4)$ space-time codeword or a pair of Alamouti space-time codewords, but independent from one block to the next. The turbo code is decoded using 14 iterations of the log-MAP algorithm [1]. In the waterfall region, the evolved code provides a gain of 0.1 dB over the MDC-QO and 1.5 dB over the $(3, 4, 4)$ code. The improved performance relative to MDC-QO can be attributed to the evolved code's superior performance at low SNR and the degradation caused by correlated real and imaginary components. Interestingly, the Alamouti code has approximately the same turbo-coded performance as the evolved code in Rayleigh fading.

It has been shown in [19] that often fading is more severe than Rayleigh. While the evolved space-time codes cannot exploit the potential added diversity when the fading is Rayleigh and a turbo code is used, they are able to outperform the Alamouti code when the fading is more severe than Rayleigh. To simulate severe fading, each fading coefficient is assumed to have a uniform phase distribution and a fading amplitude α with a Nakagami-m density [1], $E\left[\alpha^2\right] = 1$, and $1/2 \leq m \leq 1$, where $m = 1$ corresponds to Rayleigh fading. Fig. 7 repeats the simulations used to generate Fig. 6, but instead using a Nakagami-m fading channel with $m = 1/2$. As in the Rayleigh fading case, the evolved code is about 0.1 dB better than MDC-QO and 1.5 dB better than orthogonal in the waterfall region. However, now the evolved code is better than the Alamouti code by about 0.7 dB, suggesting that the additional levels of diversity are exploited in fast Nakagami-m fading.

Fig. 8 shows BER curves for the $(8, 5, 8)$ code listed in the Appendix with 8-PSK modulation and no outer code in both Rayleigh fading and Nakagami-m fading with $m = 1/2$. The performance of the $(4, 4, 4)$ code listed in the Appendix is also shown for comparison. It is apparent that the fifth antenna can provide a diversity benefit when the $(8, 5, 8)$ code is used. Because of the greatly increased search space, the $(8, 5, 8)$ code was evolved in parallel on multiple computers. Each computer was seeded with a different initial population and was allowed to evolve using its own gene pool. Periodically, the populations were compared, and if there was one individual that was superior to the best designs attained on the other computers, then that individual was copied to the other computers to become part of the other gene pools. This process of *cloning and immigration* from one gene pool to another was a key to attaining a low cost when $L \geq 6$.

The genetic algorithm failed to generate (3,3,3) and (5,5,5) codes that performed well, probably because they do not exist. Evolved (4,5,4) and (6,5,6) codes do not perform as well as the evolved (8,5,8) code. Evolved (6,3,6), (6,4,6), and (8,4,8) codes perform as well as but no better than shorter evolved codes, and thus the shorter evolved codes are deemed more practical because they require less processing, incur less latency, and can be used over a more rapidly fading channel.

V. Conclusions

A genetic algorithm has been designed to produce rate-1 space-time block codes with decoupled decoding in the receiver. Although the evolved codes do not have orthogonal transmission matrices and the decoding is simple and sub-optimal, good performance at practical signal-to-noise ratios is obtained in a fading environment. Simulation results show that evolved codes combined with efficient outer codes provide better performance over fading channels than minimum-decoding-complexity quasi-orthogonal codes at typical operating signal-to-noise ratios. When the fading is more severe than Rayleigh fading, the spectral efficiency is specified, and an efficient outer code is used, evolved codes outperform orthogonal space-time block codes.

APPENDIX

The dispersion matrices for the $(4,3,4)$ code evolved over the alphabet $\{0, \pm 1/\sqrt{2}\}$ are

$$\mathbf{A}_1 = \frac{1}{\sqrt{2}}\begin{bmatrix} 0 & +1 & +1 \\ 0 & 0 & 0 \\ +1 & 0 & +1 \\ +1 & -1 & 0 \end{bmatrix}, \mathbf{B}_1 = \frac{1}{\sqrt{2}}\begin{bmatrix} 0 & 0 & 0 \\ 0 & +1 & +1 \\ -1 & 0 & -1 \\ +1 & -1 & 0 \end{bmatrix}$$

$$\mathbf{A}_2 = \frac{1}{\sqrt{2}}\begin{bmatrix} -1 & 0 & -1 \\ -1 & +1 & 0 \\ 0 & +1 & +1 \\ 0 & 0 & 0 \end{bmatrix}, \mathbf{B}_2 = \frac{1}{\sqrt{2}}\begin{bmatrix} +1 & -1 & 0 \\ +1 & 0 & +1 \\ 0 & +1 & +1 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{A}_3 = \frac{1}{\sqrt{2}}\begin{bmatrix} 0 & 0 & 0 \\ 0 & +1 & +1 \\ +1 & -1 & 0 \\ -1 & 0 & -1 \end{bmatrix}, \mathbf{B}_3 = \frac{1}{\sqrt{2}}\begin{bmatrix} 0 & +1 & +1 \\ 0 & 0 & 0 \\ -1 & +1 & 0 \\ -1 & 0 & -1 \end{bmatrix}$$

$$\mathbf{A}_4 = \frac{1}{\sqrt{2}}\begin{bmatrix} +1 & -1 & 0 \\ -1 & 0 & -1 \\ 0 & 0 & 0 \\ 0 & -1 & -1 \end{bmatrix}, \mathbf{B}_4 = \frac{1}{\sqrt{2}}\begin{bmatrix} +1 & 0 & +1 \\ -1 & +1 & 0 \\ 0 & 0 & 0 \\ 0 & +1 & +1 \end{bmatrix}.$$

The best $(4,4,4)$ and $(8,5,8)$ codes were evolved over the discrete alphabet $\{0, \pm 1\}$. With this alphabet, an opportunity exists for a more compact notation based on the fact that each column will have a single non-zero entry of either $+1$ or -1. In the compact notation, the set of matrices $\{\mathbf{A}_1, ...\mathbf{A}_N\}$ are represented by a single N by N_T matrix $\mathbf{A}$. The magnitude of the $(i,j)^{th}$ entry of $\mathbf{A}$ indicates the row index of the nonzero entry in the j^{th} column of matrix $\mathbf{A}_i$, while the sign indicates whether the entry is positive or negative. The set of matrices $\{\mathbf{B}_1, ...\mathbf{B}_N\}$ are similarly represented by a N by N_T matrix $\mathbf{B}$. As an example of this notation, the dispersion matrices of Example 1 may be expressed as

$$\mathbf{A} = \begin{bmatrix} +1 & +2 & +3 & +4 \\ -2 & +1 & -4 & +3 \\ -3 & -4 & -1 & -2 \\ +4 & -3 & +2 & -1 \end{bmatrix}$$

and

$$\mathbf{B} = \begin{bmatrix} +1 & -2 & +3 & -4 \\ +2 & +1 & +4 & +3 \\ +3 & -4 & +1 & -2 \\ +4 & +3 & +2 & +1 \end{bmatrix}.$$

The consolidated dispersion matrices for the $(4,4,4)$ code evolved over alphabet $\{0, \pm 1\}$ are

$$\mathbf{A} = \begin{bmatrix} -2 & +1 & -1 & +2 \\ -1 & -2 & +2 & +1 \\ +4 & -3 & -4 & +3 \\ +3 & +4 & -3 & -4 \end{bmatrix}$$

and

$$\mathbf{B} = \begin{bmatrix} -2 & -1 & +1 & +2 \\ +1 & -2 & +2 & -1 \\ -4 & -3 & +4 & +3 \\ -3 & +4 & +3 & -4 \end{bmatrix}.$$

The consolidated dispersion matrices for the $(8,5,8)$ code evolved over alphabet $\{0, \pm 1\}$ are

$$\mathbf{A} = \begin{bmatrix} -5 & -1 & +5 & -1 & -1 \\ -4 & +7 & -7 & +7 & -4 \\ -1 & +5 & +1 & +5 & +5 \\ +3 & +3 & -2 & -2 & -2 \\ +2 & +2 & +3 & +3 & +3 \\ -8 & -6 & -8 & -8 & +6 \\ -7 & -4 & +4 & -4 & -7 \\ +6 & -8 & +6 & +6 & +8 \end{bmatrix}$$

and

$$\mathbf{B} = \begin{bmatrix} -8 & +6 & -8 & -8 & -6 \\ +4 & +7 & -7 & +7 & +4 \\ -2 & -2 & +3 & +3 & +3 \\ -1 & -5 & +1 & -5 & -5 \\ -6 & -8 & -6 & -6 & +8 \\ +7 & -4 & +4 & -4 & +7 \\ +5 & -1 & -5 & -1 & -1 \\ +3 & +3 & +2 & +2 & +2 \end{bmatrix}.$$

REFERENCES

[1] J. G. Proakis and M. Salehi, *Digital Communications, 5th ed.* New York: McGraw-Hill, 2008.

[2] E. G. Larsson and P. Stoica, *Space-Time Block Coding for Wireless Communications.* New York: Cambridge Univ. Press, 2003.

[3] H. Jafarkhani, "A quasi-orthogonal space–time block code," *IEEE Trans. Commun.*, vol. 49, pp. 1-4, Jan. 2001.

[4] W. Su and X.-G. Xia, "Signal constellations for quasi-orthogonal space-time block codes with full diversity," *IEEE Trans. Inf. Theory*, vol. 50, pp. 2331-2347, Oct. 2004.

[5] D. N. Dao and C. Tellambura, "Quasi-orthogonal STBC with minimum decoding complexity: performance analysis, optimal signal transformations, and antenna selection diversity," *IEEE Trans. Commun.*, vol. 56, pp. 849-853, June 2008.

[6] C. Yuen, Y. L. Guan, and T. T. Tjhung, "Quasi-orthogonal STBC with minimum decoding complexity," *IEEE Trans. Wireless Commun.*, vol. 4, pp. 2089-2094, Sept. 2005.

[7] D. N. Dao and C. Tellambura, "Decoding, performance analysis, and optimal signal designs for coordinate interleaved orthogonal designs," *IEEE Trans. Wireless Commun.*, vol. 7, pp. 48-53, Jan. 2008.

[8] B. Hassibi and B. M. Hochwald, "High-rate codes that are linear in space and time," *IEEE Trans. Inf. Theory*, vol. 48, pp. 1804-1824, July 2002.

[9] K. A. DeJong, *Evolutionary Computation: A Unified Approach.* Cambridge, MA: MIT Press, 2006.

[10] A. Ghaderipoor, M. Hajiaghayi, and C. Tellambura, "Unitary matrix design via genetic search for differential space-time modulation and limited feedback precoding," *IEEE Intern. Symp. Personal, Indoor Mobile Radio Commun.*, Sept. 2006.

[11] A. Ghaderipoor, M. Hajiaghayi, and C. Tellambura, "On the design of 2x2 full-rate full-diversity space-time block codes," pp. 3406-3410, *IEEE Global Telecommun. Conf.*, Nov. 2007.

[12] S. Qinghua and Q. T. Zhang, "Simple nonorthogonal 4×4 space-time block codes with rate one and full diversity," in *Proc. IEEE Veh. Technol. Conf.*, vol. 4, pp. 2449-2452, Sept. 2004.

[13] T. Koike and S. Yoshida, "Genetic designing of near-optimal training sequences for spatial multiplexing transmissions," in *Proc. IEEE Intern. Symp. Multi-Dimensional Mobile Commun.*, vol. 1, pp. 474-478, Aug. 2004.

[14] D. Torrieri and M. C. Valenti, "Efficient space-time block codes designed by a genetic algorithm," *IEEE MILCOM Conf.*, Nov. 2008.

[15] C. Yuen, Y. L. Guan, and T. T. Tjhung, "Algebraic relationship between amicable orthogonal designs and quasi-orthogonal STBC with minimum decoding complexity," in *Proc. IEEE Intern. Conf. on Commun.*, pp. 4882-4887, June 2006.

[16] M. O. Sinnokrot and J. R. Barry, "A single-symbol-decodable space-time block code with full rate and low peak-to-average power ratio," *IEEE Trans. Wireless Commun.*, vol. 8, pp. 2242-2246, May 2009.

[17] M. C. Valenti and S. Cheng, "Iterative demodulation and decoding of turbo coded M-ary noncoherent orthogonal modulation," *IEEE J. Sel. Areas Commun.*, vol. 23, pp. 1738-1747, Sept. 2005.

[18] European Telecommunications Standards Institute, "Universal mobile telecommunications system (UMTS): Multiplexing and channel coding (FDD)," ETSI TS 25.212 version 6.6.0, Sept. 2005.

[19] J. Frolik, "On appropriate models for characterizing hyper-Rayleigh fading," *IEEE Trans. Wireless Commun.*, vol. 7, pp. 5202-5207, Dec. 2008.

Don Torrieri is a research engineer and Fellow of the US Army Research Laboratory. His primary research interests are communication systems, adaptive arrays, and signal processing. He received the Ph. D. degree from the University of Maryland. He is the author of many articles and several books including *Principles of Spread-Spectrum Communication Systems* (Springer, 2005). He teaches graduate courses at Johns Hopkins University and has taught many short courses. In 2004, he received the Military Communications Conference achievement award for sustained contributions to the field.

Matthew C. Valenti has been with West Virginia University since 1999, where he is currently an Associate Professor in the Lane Department of Computer Science and Electrical Engineering. He holds BS and Ph.D. degrees in Electrical Engineering from Virginia Tech and a MS in Electrical Engineering from the Johns Hopkins University. From 1992 to 1995 he was an electronics engineer at the US Naval Research Laboratory. He serves as an associate editor for IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, and has served as a co-chair for the Wireless Communications Symposium at ICC-2009 (Dresden, Germany) and as co-chair for the Communication Theory Symposium at ICC-2011 (Kyoto, Japan). His research interests are in the areas of communication theory, error correction coding, applied information theory, wireless networks, simulation, and grid computing. His research has been funded by the National Research Foundation and the Department of Defense.