



Missouri University of Science and Technology
Scholars' Mine

Electrical and Computer Engineering Faculty
Research & Creative Works

Electrical and Computer Engineering

01 Nov 2007

A Gray-Code Type Bit Assignment Algorithm for Unitary Space-Time Constellations

Adam Panagos

Kurt Louis Kosbar

Missouri University of Science and Technology, kosbar@mst.edu

Follow this and additional works at: https://scholarsmine.mst.edu/ele_comeng_facwork

 Part of the [Electrical and Computer Engineering Commons](#)

Recommended Citation

A. Panagos and K. L. Kosbar, "A Gray-Code Type Bit Assignment Algorithm for Unitary Space-Time Constellations," *Proceedings of the 2007 IEEE Global Telecommunications Conference (2007, Washington, DC)*, pp. 4005-4009, Institute of Electrical and Electronics Engineers (IEEE), Nov 2007. The definitive version is available at <https://doi.org/10.1109/GLOCOM.2007.761>

This Article - Conference proceedings is brought to you for free and open access by Scholars' Mine. It has been accepted for inclusion in Electrical and Computer Engineering Faculty Research & Creative Works by an authorized administrator of Scholars' Mine. This work is protected by U. S. Copyright Law. Unauthorized use including reproduction for redistribution requires the permission of the copyright holder. For more information, please contact scholarsmine@mst.edu.

A Gray-Code Type Bit Assignment Algorithm for Unitary Space-Time Constellations

Adam Panagos
 Dynetics, Inc.
 Huntsville, Alabama 35806
 adam.panagos@dynetics.com

Kurt Kosbar
 University of Missouri - Rolla
 Rolla, Missouri 65401
 kosbar@umr.edu

Abstract—Many techniques for constructing unitary space-time constellations have been proposed. To minimize bit-error rate (BER) in a wireless communication system, constellations constructed using these techniques should be given a Gray-code type bit assignment, where symbols which are close in signal space have bit assignments which have small Hamming distance. To the authors' knowledge, no efficient general strategy for making this bit assignment has been suggested. This work proposes a prioritized distance (PD) algorithm for making this assignment in an optimal manner by minimizing the probability of bit error union bound. The algorithm can be used on constellations constructed using any technique. Simulation results show this algorithm significantly outperforms random searches and achieves near globally optimum results with moderate complexity.

I. INTRODUCTION

Unitary space-time codes are a coding technique for multiple-input multiple-output (MIMO) channels where neither the transmitter nor receiver have channel state information [1]. The two design criteria of primary interest are the *diversity product* and *diversity sum*. Constellations designed for high signal-to-noise ratio (SNR) seek to maximize the diversity product, while constellations designed for low SNR seek to maximize the diversity sum.

Many techniques for constructing unitary space-time constellations have been proposed [2]–[9]. The goal of these techniques is to construct a set of L unitary matrices denoted $\mathcal{V} = \{V_0, V_1, \dots, V_{L-1}\}$ with large diversity product or diversity sum. For M transmitting antennas, V_i is an $M \times M$ unitary matrix, i.e. $V_i^\dagger V_i = I_M$ for $i = 0, \dots, L-1$.

One of the first construction techniques proposed were *cyclic codes* [2]. Other techniques and results can be found in [3]–[9]. To the authors' knowledge, Weak Group Givens (WGG) codes currently have the best known diversity product and diversity sum for most constellation sizes and number of transmitting antennas [10].

The goal of this paper is to develop a low complexity algorithm that works with any unitary constellation construction technique, and that yields optimal bit assignments in terms of optimizing a design metric $f(\cdot)$. The problem of assigning bits to unitary constellations has only been briefly addressed in the open literature [11], [12]. We initially investigate a specific form of $f(\cdot)$ given in (7), and later consider another metric for comparison with other published results. The simulation results presented in [11], [12] sufficiently demonstrate the

significant impact the bit assignment selection can have on the performance of iterative decoding systems.

Section II gives a mathematical formulation for assigning bits to unitary constellations in an optimal manner. Strategies for accomplishing this task are also discussed. Notation, required functions, and general strategy of the PD algorithm are discussed next. Numerical results demonstrating the performance of the algorithm, and a comparison with existing results is provided in Section IV.

II. PROBLEM FORMULATION

Throughout this work, constellations of size $L = 2^b$ where b is an integer greater than one are assumed. Let \mathbf{b}_a denote the $L \times 1$ *bit assignment vector* containing the bit assignments for the unitary constellation. Each element of \mathbf{b}_a is a b -bit sequence, and $\mathbf{b}_a(i)$ represents the bits assigned to signal V_i . There are L unique b -bit sequences, and each can be used only once. Thus, for a unitary constellation with L signals there are $L!$ unique choices for \mathbf{b}_a .

For a unitary constellation, the union bound of the bit-error rate (BER) is well approximated by [13]

$$P_{\text{bit}}(\rho) \approx \rho^{-MN} \frac{\sum_{i \neq j} d^H(\mathbf{b}_a(i), \mathbf{b}_a(j)) \left(\frac{\det(V_i - V_j)}{\sqrt{8^M}} \right)^{-2N}}{2L \log_2 L}, \quad (1)$$

where $d^H(\mathbf{b}_a(i), \mathbf{b}_a(j))$ is the Hamming distance between bits $\mathbf{b}_a(i)$ and $\mathbf{b}_a(j)$, M is the number of transmitting antennas, N is the number of receiving antennas, and ρ is the SNR at the receiving antennas.

Taking the logarithm of $P_{\text{bit}}(\rho)$ yields the quantity $\zeta(\mathbf{b}_a)$ defined as

$$\zeta(\mathbf{b}_a) \triangleq \log_{10} \left(\frac{\sum_{i \neq j} d^H(\mathbf{b}_a(i), \mathbf{b}_a(j)) \left(\frac{\det(V_i - V_j)}{\sqrt{8^M}} \right)^{-2N}}{2L \log_2 L} \right), \quad (2)$$

which is a function of \mathbf{b}_a and is independent of ρ . For a unitary constellation \mathcal{V} constructed for an $M \times N$ MIMO system, we wish to find \mathbf{b}_a to minimize $\zeta(\cdot)$. Specifically, let \mathcal{S} be the set of all possible bit assignment vectors. Then $|\mathcal{S}| = L!$ and we seek $\mathbf{b}_a^{\text{opt}} \in \mathcal{S}$ where

$$\mathbf{b}_a^{\text{opt}} = \arg \min_{\mathbf{b}_a \in \mathcal{S}} \zeta(\mathbf{b}_a). \quad (3)$$

One approach to finding $\mathbf{b}_a^{\text{opt}}$ is exhaustive search. For constellations of size $L \leq 8$ this works well. For constellations of size $L \geq 16$, it is currently impractical to perform an exhaustive search on typical desktop computers due to the size of \mathcal{S} .

Another approach to finding $\mathbf{b}_a^{\text{opt}}$ is random search. Results in Section IV show this technique can be effective for $L \leq 16$. However, our results show that random searches do not work well for larger constellations.

Cyclic unitary space-time codes have the special property that signals offset by $L/2$ are maximally separated. Thus, signals offset by $L/2$ are given complementary bit assignments in [14]. Note that this strategy is valid only for the codes of [14], and is not a general strategy for unitary constellation bit assignment. The bit assignment strategy of [14] assigns bits with large Hamming distance to dissimilar signals. This strategy is reasonable, but results presented here show the PD algorithm yields better results in terms of minimizing (2). Unitary constellations constructed using other techniques do not have this structure, and no bit assignment strategy for them currently exists to our knowledge.

A novel approach to the bit assignment problem was considered in [13]. Bits were pre-assigned to signal indices before constellation design. The constellation was then constructed to minimize a function of the BER union bound. The constellations constructed have diversity product and sum of zero (which indicates poor performance in *asymptotic* SNR regions), but results presented in [13] show this technique can yield good performance at *typical* SNR values. While this is an interesting approach to the bit assignment problem, a large collection of constellations already exist with optimized diversity product and diversity sum. These constellations were designed without thought for how final constellation bit assignments should be made. The PD algorithm solves this problem.

III. THE PRIORITIZED DISTANCE ALGORITHM

This section gives details of the prioritized distance algorithm provided in block diagram form in Figure 1.

A. Notation and Initialization

Let \mathcal{D} denote the $L \times L$ *difference matrix*. The (i, j) th element of \mathcal{D} contains the quantity

$$d_{i,j} \triangleq |\det(V_i - V_j)|, \quad (4)$$

which is a measure of the similarity, or distance, between signals V_i and V_j .

Let \mathbf{b}_a denote the $L \times 1$ bit assignment vector. This vector is initialized to all zeros. Upon completion of the PD algorithm, the signal V_i is assigned the bits $\mathbf{b}_a(i)$.

The dimensions of vector quantities \mathbf{g} and \mathbf{v} are modified during execution of the the PD algorithm. The initial vector dimensions are listed below.

Let \mathbf{v} denote the $L^2 \times 3$ *priority vector*. This vector is constructed by sorting \mathcal{D} in order of increasing values of $d_{i,j}$. The first column of \mathbf{v} contains the distance value $d_{i,j}$, the second column contains the signal index i , and the last column contains the signal index j . This vector establishes the order in which signals of the unitary constellation will be assigned bits.

Let \mathbf{g} denote the $L \times 1$ *Gray-code vector*. It contains a b -bit Gray-code sequence. This vector represents the bits that still need to be assigned to signals in the unitary constellation. Upon completion of the PD algorithm, this vector is empty.

Let \mathbf{u} denote the $L \times 1$ *assignment vector*. The assignment vector is initialized to all zeros, i.e. $\mathbf{0}_L$, indicating that no bit assignments have been made. If signal i is assigned a bit string, set $\mathbf{u}(i) = 1$. When $\sum_{k=1}^L \mathbf{u}(k) = L$ the algorithm is completed.

B. Functions

The function `deleterows(a, r)` deletes rows \mathbf{r} (e.g. $\mathbf{r} = [1, 2, 5]$) of vector \mathbf{a} . The function `calcpairs(g)` calculates all pairs of bits stored in \mathbf{g} that have the smallest Hamming distance. The indices of the pairs are stored in the matrix \mathbf{P} . As an example let

$$\mathbf{g} = \begin{pmatrix} 00 \\ 01 \\ 11 \\ 10 \end{pmatrix}. \quad (5)$$

Then,

$$\mathbf{P} = \begin{pmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{pmatrix} = \begin{pmatrix} 1 & 2 \\ 1 & 4 \\ 2 & 3 \\ 3 & 4 \end{pmatrix} \quad (6)$$

since all pairs of bits p_i listed in \mathbf{P} have Hamming distance one.

C. Strategy

The PD algorithm constructs a value of \mathbf{b}_a in an attempt to minimize $\zeta(\cdot)$. The bit assignment vector constructed is not necessarily $\mathbf{b}_a^{\text{opt}}$, but the value of \mathbf{b}_a obtained from the algorithm does yield a value of ζ that is close to a global minimum. See section IV for results that substantiate this claim.

The PD algorithm seeks to minimize $\zeta(\cdot)$ by ensuring that signal indices i and j with small values of $d_{i,j}$ are assigned bits with small Hamming distance.

More specifically, bits are assigned to the unitary constellation based on the order in which signal indices appear in the priority vector. The top row of the priority vector contains signal indices i and j that have the closest distance $d_{i,j}$. If both signals i and j have yet to be assigned, $\mathbf{u}(i)$ and $\mathbf{u}(j)$ are set equal to one. The pairs matrix \mathbf{P} is calculated based on the current value of \mathbf{g} . The algorithm then searches over all pairs of possible bit assignments stored in \mathbf{P} and finds the bit assignment that minimizes $f(\mathbf{b}_a)$ where

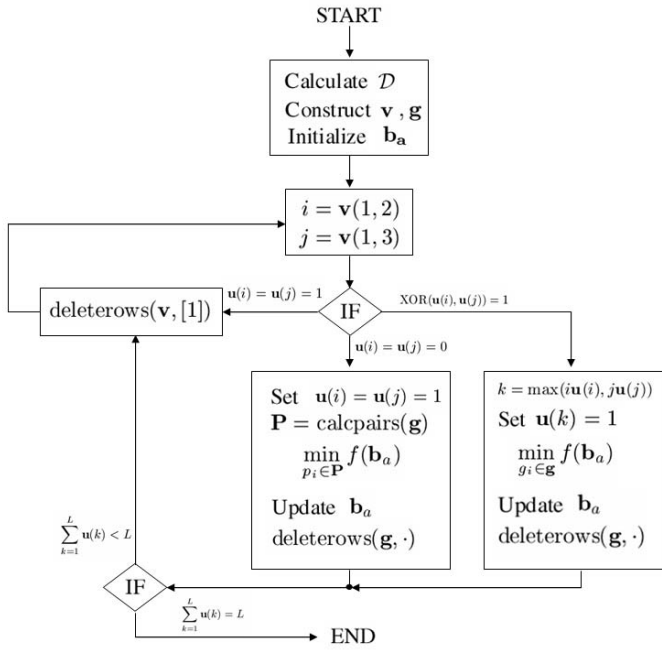


Fig. 1. Block diagram of prioritized distance algorithm.

$$f(\mathbf{b}_a) = \frac{\sum_{i \neq j} \left(d^H(\mathbf{b}_a(i), \mathbf{b}_a(j)) \left(\frac{\det(V_i - V_j)}{\sqrt{8^M}} \right)^{-2N} \mathbf{u}(i)\mathbf{u}(j) \right)}{2L \log_2 L} \quad (7)$$

Use of the assignment vector \mathbf{u} in (7) ensures only signals that have been assigned bits are included in the summation. Upon finding the bit assignments for signals i and j that minimize $f(\mathbf{b}_a)$, these values are stored in \mathbf{b}_a and the corresponding rows are deleted from \mathbf{g} .

If signal index i has already been assigned, but not signal index j , a search over the values in \mathbf{g} to minimize \mathbf{b}_a is performed. As before, the best bit assignment is stored in \mathbf{b}_a and deleted from \mathbf{g} .

Once an assignment is made, the algorithm deletes the top row of the priority vector, examines the priority vector to find the next indices to work with, and repeats the above process until the algorithm end condition

$$\sum_{k=1}^L \mathbf{u}(k) = L, \quad (8)$$

is satisfied.

A block diagram of the algorithm can be seen in Figure 1.

Upon completion of the PD algorithm, the bit assignment vector \mathbf{b}_a contains the bits assigned to each signal in the unitary constellation.

D. Complexity

The PD algorithm iterates until all signals have been assigned a bit sequence. The primary complexity during each

iteration is performing the minimization $\min_{p_i \in \mathbf{P}} f(\mathbf{b}_a)$ or $\min_{g_i \in \mathbf{g}} f(\mathbf{b}_a)$ as this requires numerous evaluations of the $f(\mathbf{b}_a)$ function. While the algorithm is running, at any given time the number of signals that have already been assigned bits is $\sum_{k=1}^L \mathbf{u}(k)$. The minimizations are thus performed over spaces whose sizes are

$$|\mathbf{g}| = L - \sum_{k=1}^L \mathbf{u}(k), \quad (9)$$

and

$$|\mathbf{P}| \leq \binom{L - \sum_{k=1}^L \mathbf{u}(k)}{2}. \quad (10)$$

Since $|\mathbf{g}| < |\mathbf{P}|$ when $\sum_{k=1}^L \mathbf{u}(k) > 3$, the algorithm runs quickest when optimizations are performed over \mathbf{g} (i.e. when the algorithm takes the $\text{XOR}(\mathbf{u}(i), \mathbf{u}(j)) = 1$ branch of the block diagram). Let I_{\min} be the lower bound on the total number of $f(\mathbf{b}_a)$ function evaluations. Since the first two signals from the priority vector can be assigned from the first two elements of the Gray-code vector, there are only $L - 2$ assignments to make. The lower bound is thus

$$\begin{aligned} I_{\min} &= (L - 2) + (L - 3) + \dots + 2 \\ &= \sum_{k=2}^{L-2} k = \frac{L^2 - 3L}{2}. \end{aligned} \quad (11)$$

The algorithm runs slowest when optimizations are performed over \mathbf{P} (i.e. when the algorithm takes the $\mathbf{u}(i) = \mathbf{u}(j) = 0$ branch of the block diagram). Let I_{\max} be an upper bound on the total number of $f(\mathbf{b}_a)$ function evaluations defined as

$$I_{\max} = \sum_{i=0}^{L/2-1} \binom{L - 2i}{2} = \frac{2L^3 + 3L^2 - 2L}{24}. \quad (12)$$

A plot of the upper and lower bounds on the total number of $f(\mathbf{b}_a)$ function evaluations, along with the actual number of calculations required for the constellations investigated here, can be seen in Figure 2. For the same value of L , different constellations had slightly different numbers of function f evaluations required. However, in all cases, the actual complexity is closer to the lower bound. Thus, the complexity of the PD algorithm proposed here is approximately $\mathcal{O}(L^2)$.

IV. RESULTS

A. Design Metric (7)

The results presented in this section use one receive antenna for simplicity although the algorithm works for any N . Cyclic unitary space-time constellations [2] and Weak Group Givens codes [10] were used to investigate the performance of the proposed PD bit assignment algorithm. Recall that regardless of the construction technique, the goal is to find a bit assignment vector \mathbf{b}_a to minimize $\zeta(\mathbf{b}_a)$.

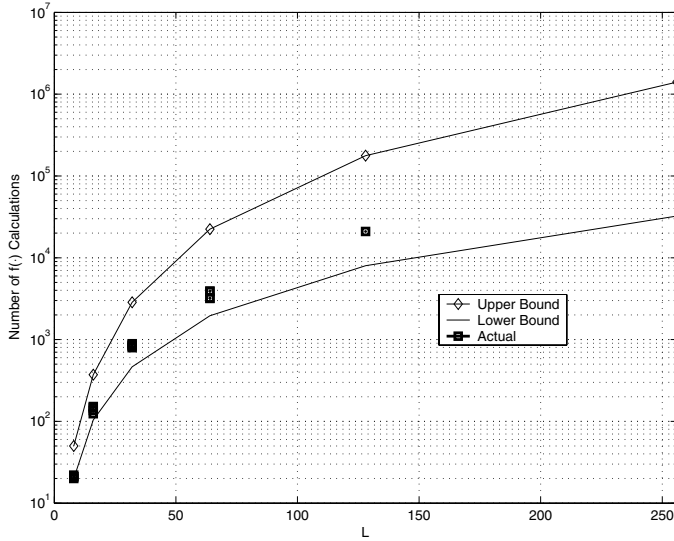


Fig. 2. Complexity Analysis

Table I summarizes the performance of the PD bit assignment algorithm, the performance of a random search, and in the case of the cyclic codes, the performance of the $L/2$ assignment strategy. Different numbers of transmit antennas and different constellations sizes were examined. The numbers tabulated in the table are the values of ζ obtained by each respective method. For $L = 8$ it is possible to perform an exhaustive search to find the global minimum. Thus, the value of ζ listed in the “Random” column is the global minimum obtained from an exhaustive search. The P% column lists the percentile of the value of ζ found. For $L = 8$ and 16 the percentile was calculated based on empirical data obtained during the random search. For $L = 32$ and 64 the percentile value found using the PD algorithm is much smaller than any value found with the random search. Figure 3 shows that the distribution of ζ from the random search is approximately Gaussian. The percentile was approximated using the tail of the appropriate Gaussian distribution.

Results in Table I suggest that a random search works well for $L \leq 16$, while the PD algorithm works well for $L \geq 32$. For example, with $M = 3$, $L = 32$, and a cyclic code, the PD algorithm finds a bit assignment vector that yields a value of ζ that is approximately in the 3×10^{-22} percentile of a random search. This suggests that using a random search to find an optimal bit assignment vector for this constellation would be very time consuming, if not impractical. It is also interesting to note that in the case of cyclic unitary codes, the PD algorithm yields bit assignment vectors that yield smaller values of ζ than the $L/2$ assignment strategy originally suggested.

B. Another Design Metric

A similar bit assignment metric proposed in [11] is

$$\gamma(\mathbf{b}_a) = \frac{1}{Lb} \sum_{\Psi_l \in \Psi} \sum_{k=1}^b \left[\prod_{m=1}^M (1 - \delta_{m,l,k}^2) \right] \quad (13)$$

TABLE I
PD ALGORITHM OUTPERFORMS RANDOM SEARCH FOR $L \geq 32$

M	L	Construction	PD Algorithm	Random	L/2	P%
2	8	Cyclic	1.2583	1.2583	1.2583	0
	8	WGG	0.8293	0.7871	NA	7
	16	Cyclic	2.5943	2.5550	2.6001	0.008
	16	WGG	1.7156	1.7021	NA	3
	32	Cyclic	3.9664	4.0195	4.0046	2e-12
	32	WGG	2.6686	2.6858	NA	4e-10
	64	Cyclic	4.9311	5.0735	5.0236	7e-79
	64	WGG	3.6305	3.7248	NA	3e-91
3	8	Cyclic	2.0329	2.0329	2.1274	0
	8	WGG	0.2539	0.2407	NA	6
	16	Cyclic	3.3583	3.3067	3.4526	0.04
	16	WGG	1.7865	1.7787	NA	0.08
	32	Cyclic	5.4788	5.5861	5.7681	3e-22
	32	WGG	3.4901	3.5103	NA	2e-9
	64	Cyclic	7.1426	7.2889	7.2753	3e-74
	64	WGG	4.3155	4.3943	NA	2e-53

where $\Psi = \{\Psi_0, \Psi_1, \dots, \Psi_{L-1}\}$ is the size L constellation of, $T \times M$ unitary matrices, and $\delta_{m,l,k}$ is the m th singular value of $\hat{\Psi}_l^\dagger \Psi_l$ in which the bit assignment labels of $\hat{\Psi}_l$ and Ψ_l differ in only the k th bit position. When performing the sum of equation (13) for a given signal Ψ_l and bit position k , there is only one other signal that differs from Ψ_l in only the k th bit position. This signal is denoted $\hat{\Psi}_l$. These are the M singular values that are calculated and used in the product of equation (13). The optimal bit assignment is one that *maximizes* this design metric.

This bit labeling metric can be easily used in the PD algorithm by letting

$$f(\mathbf{b}_a) = \gamma(\mathbf{b}_a) \mathbf{u}(l) \quad (14)$$

and

$$d_{i,j} = \left(\prod_{m=1}^M (1 - \sigma_{m,i,j}^2) \right)^{-1} \quad (15)$$

where $\sigma_{m,i,j}$ for $m = 1, \dots, M$ are the singular values of $\Psi_i^\dagger \Psi_j$.

The authors of [11] derive an optimal bit mapping for a unitary constellation derived from an orthogonal design, namely the unitary constellation

$$\Psi_l = \frac{1}{2} \begin{pmatrix} 1 & -1 & \exp\left(j\frac{2\pi}{Q}k\right) & -\exp\left(-j\frac{2\pi}{Q}p\right) \\ 1 & 1 & \exp\left(j\frac{2\pi}{Q}p\right) & \exp\left(-j\frac{2\pi}{Q}k\right) \end{pmatrix}^T \quad (16)$$

where $1 \leq l \leq L$, Q is a positive integer such that $Q^2 = L$, and the integers k and p satisfy $k = (l-1)\text{div } Q$ and $p = (l-1) \bmod Q$.

For the special case of $L = 16$, $M = 2$, they take advantage of similarities between this constellation and a 4-dimensional hypercube to derive an upper bound of $\gamma(\mathbf{b}_a) \leq 43/64$. They then devise a bit labeling scheme that meets this upper bound, proving the optimality of this mapping.

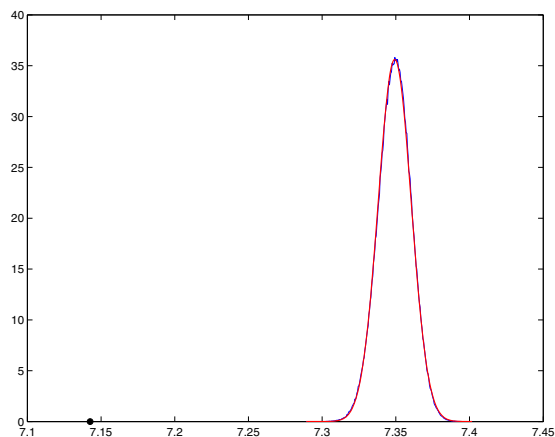


Fig. 3. Distribution of $\zeta(\mathbf{b}_a)$ using random search for $M = 3$, $L = 64$ and cyclic construction. The red line is a Gaussian approximation to the empirical distribution (blue line). The value of $\zeta(\mathbf{b}_a)$ obtained using the PD algorithm is the black dot at ≈ 7.14 and is significantly better than values found with the random search.

The PD algorithm also generates an optimal bit mapping for this constellation. The mapping is different from that provided in [11], but still meets the upper bound. The mapping for this constellation obtained from the PD algorithm is $\{\Psi_0, \Psi_1, \dots, \Psi_{15}\} \rightarrow \{0000, 0011, 0110, 0101, 1001, 1010, 1111, 1100, 0111, 0100, 0001, 0010, 1110, 1101, 1000, 1011\}$.

For $L = 64$, $M = 4$, the unitary constellation shows strong similarities to a 6-dimensional hypercube [15]. The optimal mapping in this case gives $\gamma(\Psi, \mathbf{b}_a) = 0.568544$. Use of the PD algorithm yields a mapping with $\gamma(\Psi, \mathbf{b}_a) = 0.522055$, or 92% of the optimal value.

Exploiting the geometric structure of a code to devise optimal bit mappings is clearly a preferred technique. However, most unitary constellations do not have an easily identifiable structure (or any organized structure at all).

In this case, the past approach [11], [12] has been to use numerous trials of a greedy binary switching algorithm (BSA). For the systematic code $U = [1, 2, 5, 12]$ a mapping is found via the BSA algorithm with $\gamma(\Psi, \mathbf{b}_a) = 25.701/64 \approx 0.401578$ [11]. Using the PD algorithm we attain a value of $\gamma(\Psi, \mathbf{b}_a) = 0.379576$, which although slightly less, was obtained using a deterministic algorithm that requires only a few seconds to run, significantly faster than performing 10^5 trials of the BSA algorithm.

V. CONCLUSION

A *prioritized distance* algorithm for assigning bits to unitary space-time constellations in an optimal manner has been proposed, and its performance has been investigated through various examples. The PD algorithm returns a bit assignment vector \mathbf{b}_a that minimizes a design metric $f(\cdot)$. A probability of bit error union bound design metric was chosen initially, but results of Section IV show the PD algorithm can be easily changed to accommodate other design metrics. These results also show the algorithm can achieve a significant fraction of the globally optimum solution. This algorithm can

be used regardless of the technique used to construct the unitary constellation. A block diagram of the algorithm can be seen in Figure 1. Numerical results show that a random search for the optimal bit assignment vector works well when $L \leq 16$. However, for $L \geq 32$, the PD algorithm significantly outperforms a random search. For the case of cyclic unitary constellations, the PD algorithm also outperforms a previously proposed $L/2$ assignment strategy. A complexity analysis was performed and the algorithm has a reasonable complexity on the order of $\mathcal{O}(L^2)$.

REFERENCES

- [1] B. Hochwald and T. Marzetta, "Unitary space-time modulation for multiple-antenna communications in rayleigh flat fading," *IEEE Trans. Inform. Theory*, vol. 46, pp. 543–564, Mar. 2000.
- [2] B. Hochwald, T. Marzetta, T. Richardson, W. Sweldens, and R. Urbanke, "Systematic design of unitary space-time constellations," *IEEE Trans. Inform. Theory*, vol. 46, no. 6, pp. 1962–1973, Sept. 2000.
- [3] X.-B. Liang and X.-G. Xia, "Unitary signal constellations for differential space-time modulation with two transmit antennas: parametric codes, optimal designs, and bounds," *IEEE Trans. Inform. Theory*, vol. 48, pp. 2291–2322, Aug. 2002.
- [4] C. Shan, A. Nallanathan, and P.Y.Kam, "A new class of signal constellations for differential unitary space-time modulation (dustm)," *IEEE Commun. Lett.*, vol. 8, pp. 1–3, Jan. 2004.
- [5] T. Soh, C. Ng, and P. Kam, "Improved signal constellations for differential unitary space-time modulations with more than two transmit antennas," *IEEE Commun. Lett.*, vol. 9, pp. 7–9, Jan. 2005.
- [6] T. Konishi, "Unitary subgroup space-time codes using bruhat decomposition and weyl groups," *IEEE Trans. Inform. Theory*, vol. 49, pp. 2713–2717, Oct. 2003.
- [7] T. Niyomsataya, A. Miri, and M. Nevins, "Full diversity unitary space-time bruhat constellations," *Proc. IEEE Information Theory Workshop (ITW'04)*, pp. 370–374, Oct. 2004.
- [8] I. Kammoun and J.-C. Belfiore, "A new family of grassmann space-time codes for non-coherent mimo systems," *IEEE Commun. Lett.*, vol. 6, pp. 528–530, Nov. 2003.
- [9] G. Han and J. Rosenthal, "Geometrical and numerical design of structured unitary space time constellations," *Submitted to IEEE Trans. Info Theory. Available at <http://www.math.ubc.ca/~ghan/projectindex.html>*, Sept. 2005.
- [10] A. Panagos, K. Kosbar, and A. Mohammad, "Weak group unitary space-time codes," *To appear: IEEE Global Telecommunications Conference (GLOBECOM'06)*, Nov. 2006.
- [11] N. H. Tran, H. H. Nguyen, and T. Le-Ngoc, "Tight error bound for coded unitary space-time modulation," *Proc. IEEE Wireless Communications and Networking Conference (WCNC'06)*, vol. 4, pp. 1893–1898, Apr. 2006.
- [12] Y. Li and X.-G. Xia, "Constellation mapping for space-time matrix modulation with iterative demodulation/decoding," *IEEE Trans. Commun.*
- [13] X. Chen, K. Zhou, and J. Aravena, "A new family of unitary space-time codes with a fast parallel sphere decoder algorithm," *IEEE Trans. Inform. Theory*, vol. 52, pp. 115–140, Jan. 2006.
- [14] B. Hochwald and W. Sweldens, "Differential unitary space-time modulation," *IEEE Trans. Commun.*, vol. 48, pp. 2041–2051, Dec. 2000.
- [15] N. H. Tran, H. H. Nguyen, and T. Le-Ngoc, "Coded unitary space-time modulation with iterative decoding: Error performance and mapping design," *submitted to IEEE Trans. on Commun.*, Oct. 2005.