IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 20, 2017, accepted September 8, 2017, date of publication October 17, 2017, date of current version November 7, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2763614

Polar Code Design for Irregular Multidimensional Constellations

PHILIP R. BALOGUN, IAN D. MARSLAND, (Member, IEEE), RAMY H. GOHARY, (Senior Member, IEEE), AND HALIM YANIKOMEROGLU, (Fellow, IEEE)

Department of Systems and Computer Engineering, Carleton University, Ottawa, ON K1S 5B6, Canada

Corresponding author: Ian D. Marsland (ianm@sce.carleton.ca)

This work was supported in part by the Huawei Technologies Canada and in part by the Ontario Ministry of Economic Development and Innovations Ontario Research Fund–Research Excellence program.

ABSTRACT Polar codes, ever since their introduction, have been shown to be very effective for various wireless communication channels. This, together with their relatively low implementation complexity, has made them an attractive coding scheme for wireless communications. Polar codes have been extensively studied for use with binary-input symmetric memoryless channels but little is known about their effectiveness in other channels. In this paper, a novel methodology for designing multilevel polar codes that works effectively with arbitrary multidimensional constellations is presented. In order for this multilevel design to function, a novel set merging algorithm, able to label such constellations, is proposed. We then compare the error rate performance of our design with that of existing schemes and show that we were able to obtain unprecedented results in many cases over the previously known best techniques at relatively low decoding complexity.

INDEX TERMS Set merging, set partitioning, bit labeling, multilevel polar codes, code design.

I. INTRODUCTION

Multidimensional constellations are useful for improving the SNR efficiency of systems and representing fractional numbers of bits per two dimensions in comparison to 1 or 2 dimensional constellations [2]. A well known example of an irregular multidimensional constellation is the Golden code. It is a full rate, full diversity space-time block code with arguably the best performance for coherent MIMO channels and is based on the Golden number. Another example of irregular multidimensional constellations are codebooks of unitary matrices that are isotropically distributed on the (compact) Grassmann manifold, specifically designed for noncoherent communication over block-fading channels [3]. These multidimensional constellations work with the observation that the distortion caused by a fading channel does not change the subspace in which a transmitted signal resides, it only rotates and scales the bases of the subspace [4]. These constellations exploit such characteristics and consider orthogonal subspaces to detect the transmitted symbols at the receiver [5]. It was shown in [4] that these constellations are able to approach the ergodic capacity at high signal-to-noise ratios (SNRs).

A natural way to improve the error rate performance of a communication system is to use coded modulation which involves combining error correcting codes with a signal constellation. The field of channel coding is one that has existed since the 1950s when Richard Hamming pioneered the first error-correcting code, the Hamming (7,4) code [6]. Ever since then, more powerful codes have been created such as the capacity approaching low density parity check (LDPC) and turbo codes. However, the decoding of these codes can be computationally intensive.

Polar codes, a recent invention by Erdal Arikan, are a class of error-correcting codes with the proven ability to achieve the capacity of binary input, memoryless output symmetric channels [7]. In addition, they require relatively low encoding and decoding complexity. Although polar codes have been used with bit interleaved coded modulation (BICM), generally speaking, they perform worse than other codes such as LDPC and turbo codes [8]. However, it has been observed [9] that polar codes perform better when used with multilevel coding (MLC) [10] than with BICM [11].

In the literature, multidimensional constellations have been used with different coded modulation techniques.

Golden codes have been combined with trellis coded modulation in [12]. Grassmannian constellations have been combined in a BICM fashion with turbo codes [13]. It was shown that at a high data rate, this combination was able to outperform training based methods [14].

In this paper, we propose a novel methodology for designing multilevel polar codes that works effectively with arbitrary, not necessarily structured, multidimensional signalling schemes. For improved performance, MLC requires that the signalling constellation uses a type of labelling in which constellation points that are far in a Euclidean sense are assigned labels with small Hamming distances. One such labelling is the set partitioning (SP) one [15]. However, the current set partitioning algorithms can only be used for regular constellations, and so are unsuitable for irregular multidimensional constellations. To alleviate this drawback, we will propose an alternate algorithm that is based on a set merging philosophy, rather the traditional set partitioning one.

The paper is arranged as follows. In Sections II and III, the system models for the Golden code and Grassmannian constellations are given, respectively. Although our proposed polar code design methodology works well for a wide variety of constellations, we focus here on these two. In Section IV, the multilevel polar coded system set-up is given. The proposed set merging algorithm is expounded upon in Section V. In Section VI, the system design methodology for polar codes and the bit error rate (BER) performance of the system is provided. In Section VII, simulation results that show the benefits of using our design methodology and algorithm are given. The complexity analysis for our system in comparison with other coding methods is given in Section VIII. Finally, Section IX concludes the paper.

II. SYSTEM MODEL: GOLDEN CODES

Golden codes are full rate, full diversity space time block codes for the coherent MIMO channel [16]. The transmitted symbols are 2×2 complex matrices of the form

$$\mathbf{X} = \frac{1}{\sqrt{5}} \begin{bmatrix} \alpha(a + b\theta) & \alpha(c + d\theta) \\ \gamma\bar{\alpha}(c + d\bar{\theta}) & \bar{\alpha}(a + b\bar{\theta}) \end{bmatrix}, \tag{1}$$

where $\theta = \frac{1+\sqrt{5}}{2}$ is the Golden number, $\bar{\theta} = 1 - \theta$, $\alpha = 1 + i(1 - \theta)$, $\bar{\alpha} = 1 + i(1 - \bar{\theta})$, and a, b, c, and d are M-QAM symbols, which are normalized to a symbol energy of 1. The total number of points in the Golden code signal space is M^4. The average signal energy $E_{\mathbf{X}} = E[\|\mathbf{X}\|^2] = \sum_i \sum_j E[|x_{i,j}|^2] = 4$ for any size of the underlying M-QAM constellation. The $\frac{1}{\sqrt{5}}$ term normalizes the matrix $\mathbf{X}$ to a unitary matrix. The received signal is given as

$$\mathbf{Y} = \mathbf{XH} + \mathbf{W}, \tag{2}$$

where $\mathbf{H}$ is a $2 \times N_r$ fading channel matrix with independent elements which change independently after every block of $T = 2$ channel uses. Each channel coefficient, $h_{i,j}$, has a complex Gaussian distribution, $\mathcal{CN}(0, 1)$ with a mean of 0 and a variance of $E[|h_{i,j}|^2] = 1$, and the real and imaginary parts are independent. Similarly, $\mathbf{W}$ is a $2 \times N_r$ additive white Gaussian noise matrix where its elements are independent and have complex Gaussian distribution $\mathcal{CN}(0, \sigma_W^2)$ with variance $E[|w_{i,j}|^2] = \sigma_W^2$ and the real and imaginary parts are independent.

Upon receiving $\mathbf{Y}$, the receiver used in this system uses maximum likelihood (ML) detection to maximize $\Pr\{\mathbf{Y}|\mathbf{X}\}$ which is given as

$$\Pr\{\mathbf{Y}|\mathbf{X}\} \propto \exp\left\{\frac{-1}{\sigma_W^2}\|\mathbf{Y} - \mathbf{XH}\|^2\right\}, \tag{3}$$

where $\|\cdot\|$ is the Euclidean distance.

III. SYSTEM MODEL: GRASSMANNIAN CONSTELLATIONS

Within the realm of noncoherent communication, Grassmannian signalling has been shown to approach the high-SNR ergodic capacity of frequency-flat block fading channels. It was shown in [4] and [5] that at high SNRs, the ergodic capacity of the channel can be achieved by codebooks of unitary matrices that are isotropically distributed on the (compact) Grassmann manifold. It was also shown that Grassmannian signalling achieves high SNR capacity when

$$T \geq \min\{N_t, N_r\} + N_r, \tag{4}$$

Given T and N_r that satisfy (4), to attain the maximum number of independent channels exploited by the transmitter (the communication degrees of freedom) for such a system, the number of transmit antennas, N_t, should be

$$N_t = \min\left\{\left\lfloor \frac{T}{2} \right\rfloor, N_r\right\}. \tag{5}$$

The received signal is given as

$$\mathbf{Y} = \mathbf{XH} + \mathbf{W}, \tag{6}$$

where $\mathbf{H}$ is an $N_t \times N_r$ fading channel matrix where each element has a complex Gaussian distribution, $\mathcal{CN}(0, 1)$, and can change independently every block of T channel uses. The random $T \times N_r$ noise matrix, $\mathbf{W}$, is assumed to have a complex Gaussian distribution, $\mathcal{CN}(0, \sigma_W^2)$. The transmitted $T \times N_t$ matrix, $\mathbf{X}$, is assumed to be unitary in the sense that $\mathbf{X}^\dagger\mathbf{X} = \mathbf{I}_{N_t}$. The superscript † denotes the conjugate transpose and $\mathbf{I}_{N_t}$ is the $N_t \times N_t$ identity matrix. The matrices $\{\mathbf{X}\}$ are isotropically distributed, and $E_{\mathbf{X}} = \|\mathbf{X}\|^2 = N_t$ where $\|\cdot\|$ is the Frobenius norm which is defined as $\sqrt{\mathrm{Tr}(\mathbf{X}^\dagger\mathbf{X})}$. The average SNR can then be calculated as

$$\rho = \frac{E[\|\mathbf{XH}\|^2]}{E[\|\mathbf{W}\|^2]}, \tag{7}$$

where

$$E[\|\mathbf{XH}\|^2] = E[\mathrm{Tr}\{\mathbf{H}^\dagger\mathbf{X}^\dagger\mathbf{XH}\}] = E[\mathrm{Tr}\{\mathbf{H}^\dagger\mathbf{H}\}] = N_tN_r, \tag{8}$$

and

$$E[\|\mathbf{W}\|^2] = TN_r\sigma_W^2, \tag{9}$$

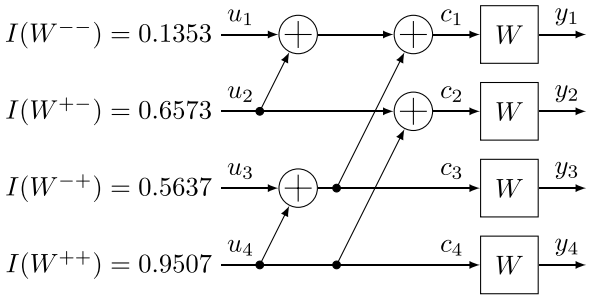

FIGURE 1. Sample 4 bit channel polar encoder for a binary AWGN channel with capacities shown. $\sigma = 1$.

so that

$$\rho = \frac{N_t N_r}{T N_r \sigma_W^2} = \frac{N_t}{T \sigma_W^2}. \tag{10}$$

For uncoded communication, the receiver uses ML detection by searching the entire constellation to find the signal which maximizes the likelihood function $\Pr\{\mathbf{Y}|\mathbf{X}\}$, where [4]

$$\Pr\{\mathbf{Y}|\mathbf{X}\} = \frac{\exp\left\{\frac{-1}{\sigma_W^2}\mathrm{Tr}\left(\mathbf{Y}^\dagger\left(\mathbf{I}_T - \frac{1}{1+\sigma_W^2}\mathbf{X}\mathbf{X}^\dagger\right)\mathbf{Y}\right)\right\}}{(\pi\sigma_W^2)^{TN_r}(1+\frac{1}{\sigma_W^2})^{N_t N_r}}$$
$$\propto \exp\left\{\frac{\|\mathbf{X}^\dagger\mathbf{Y}\|^2}{\sigma_W^2(1+\sigma_W^2)}\right\}. \tag{11}$$

IV. MULTILEVEL POLAR CODES

As stated previously, polar codes have the ability to achieve the capacity of binary input memoryless output symmetric channels. These codes are able to achieve this by using an effect known as channel polarization where channels are transformed into good and bad ones (see Fig. 1). By recursively applying such polarization transformation over the resulting channels, the reliabilities of the synthesized channels will show significant difference: the "good ones get better and the bad get worse" [8]. The channels get more distinctly polarized as the code length is increased and the good channels can be chosen to transmit information bits over while the others are frozen (set to a zero).

The transmitter uses multilevel polar coding [17] as shown in Fig. 2. A bank of m separate polar encoders are used, one for each bit position in the signal constellation, where $m = \log_2 M$ is the number of bits per symbol and M is the constellation size. Each component polar code has a length of N' and thus, the encoders output a total of $N = mN'$ bits. Each component code has a code rate of R_i that is chosen in such a way that the overall rate is $R = \frac{1}{m}\sum_{i=1}^{m} R_i$ [17]. Each code bit from a polar code is transmitted in a different symbol, and each symbol depends on m code bits, one from each encoder.

It was shown in [17] and [18] that labelling generated using the SP philosophy appear to yield favourable performance. Therefore, SP should be used to define the labelling between code bits and constellation points because this leads to large bit level variances compared to other types of labelling schemes. A novel set merging algorithm that yields labels that resemble SP ones is presented in Section V. Unlike currently available algorithms for generating SP labels, the proposed algorithm can work with arbitrary multidimensional signal constellations.

The receiver consists of a bank of m demapper/decoder pairs, with one pair per polar subcode. The demapper used in this system is a soft demapper that calculates the log likelihood ratio (LLR) of each bit given the received signal matrix as:

$$\lambda_l = \ln\frac{\Pr\{c_l = 0|\mathbf{Y}, c_1, \ldots, c_{l-1}\}}{\Pr\{c_l = 1|\mathbf{Y}, c_1, \ldots, c_{l-1}\}}$$
$$= \ln\frac{\sum_{X\in\chi_{l,0}}\Pr\{\mathbf{Y}|\mathbf{X}\}}{\sum_{X\in\chi_{l,1}}\Pr\{\mathbf{Y}|\mathbf{X}\}}, \tag{12}$$

where $\mathbf{Y}$ is the received signal matrix, c_l is a transmitted bit at the l^{th} level and the set $\chi_{l,k}$ contains all the possible matrices in the constellation that can be received at level l which have bit k at that level. The expression $\Pr\{\mathbf{Y}|\mathbf{X}\}$ is given in (11).

As (12) shows, the de-mapping of the bit at level l, c_l, relies on knowledge of the originally transmitted bits of the bit levels before it, $(c_1, \ldots, c_{l-1})$. When the LLRs for a particular level are calculated and the bits are decoded (using a standard successive cancellation (SC) polar decoder), the receiver passes those bits back into a polar encoder to obtain an approximation of the bits that were transmitted. These approximate codeword bits are then used in the de-mapping of the next bit level and so on (Fig. 2).

V. SET MERGING FOR IRREGULAR MULTIDIMENSIONAL CONSTELLATIONS

Ungerboeck proposed an SP methodology [18] used for regular constellations, such as PSK or QAM, that involves dividing the signal points into two subsets in such a way that the minimum Euclidean distance between any two points in a subset is greater than the minimum distance in the whole constellation (Fig. 3). All the points in one subset are assigned a bit value of 0 in the first bit position, while all the points in the other subset are assigned a 1. Each subset is in turn divided into two subsets, again with increasing minimum distance between points within a subset, and all points within one of the new subsets are assigned a bit value of 0 in the second position, while the points within the other new subset are assigned a 1. This process is repeated until each subset contains only one point, and all points have been assigned a unique value of $m = \log_2 M$ bits, where M is the number of points in the constellation. Forney later provided a formalized algorithm for partitioning constellations when the signal points fall on a regular multidimensional lattice [19], but there are no general-purpose algorithms that work with irregular constellations.

Ungerboeck's method works well for regular constellations, and could also be used with small irregular ones, but implementation becomes problematic for large irregular

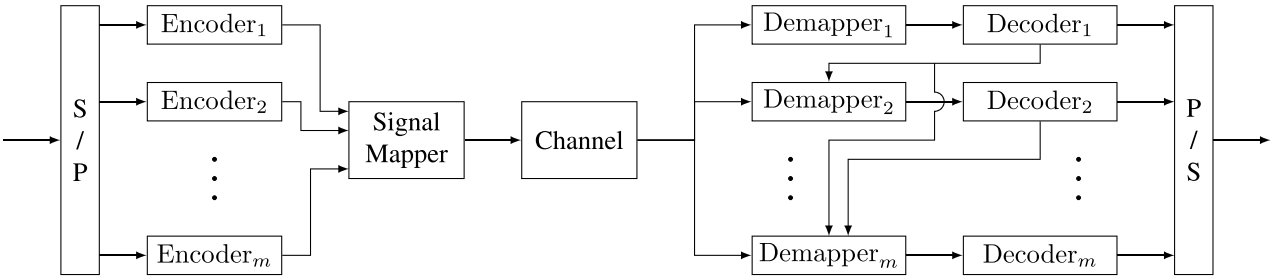

FIGURE 2. Multilevel polar coding system model.

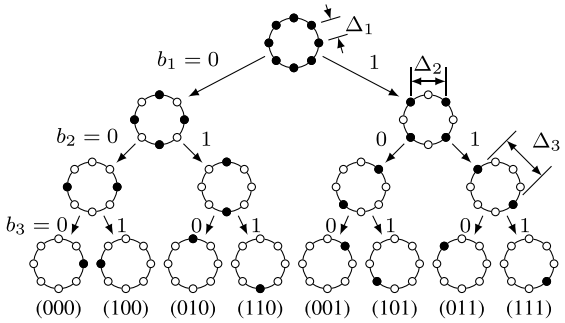

FIGURE 3. Ungerboeck's method for set partitioning an 8-PSK constellation.

multidimensional constellations. To alleviate this drawback, we will now propose a novel low-complexity set merging algorithm that generates SP-like labellings for arbitrary constellations. As implied by its name, this algorithm works in the opposite order from Ungerboeck's, starting with M subsets containing just one point each that are paired together to form $M/2$ subsets of two points, which are in turn combined, and so on, until there is one subset of M points.

Given a constellation $\mathcal{C} = \{X_i | i = 1, \ldots, M\}$, the ultimate objective all labelling algorithms is to create a particular mapping between sequences of $\log_2 M$ bits and points in $\mathcal{C}$. In this paper we propose a novel labelling algorithm based on set-merging. Let $B_{i,l}$ be the l^{th} bit of the mapping for X_i. Furthermore, let $S_{l,i}$ be the set containing the indices of the constellation points in the i^{th} subset at level l. Note that $|S_{l,i}| = 2^{l-1}$ for $i \in \{1, \ldots, M/2^{l-1}\}$ and $l \in \{1, \ldots, \log_2 M\}$. Starting with $S_{1,i} = i$, the algorithm combines subsets at level $l-1$ to produce subsets at level l, based on the inter-subset distance table, $\mathcal{D}_l(i,j) = \min_{a \in S_{l,i}, b \in S_{l,j}} d(X_a, X_b)$, where $d(X_a, X_b)$ is the distance between constellation points X_a and X_b. That is, $\mathcal{D}_l(i,j)$ contains the distance between the two closest points in subsets i and j at level l. For coherent detection schemes, the Euclidean distance is normally used. Hence, the distance between $\mathbf{X}_a$ and $\mathbf{X}_b$ is given by

$$d(\mathbf{X}_a, \mathbf{X}_b) = \|\mathbf{X}_a - \mathbf{X}_b\|. \tag{13}$$

It was shown in [4] that, for noncoherent detection of Grassmannian constellations, the chordal Frobenius norm is a more appropriate metric. Hence, the distance between $\mathbf{X}_a$ and $\mathbf{X}_b$ is given by

$$d(\mathbf{X}_a, \mathbf{X}_b) = \sqrt{2N_t - 2\sum_{k=1}^{N_t} \sigma_k}, \tag{14}$$

where σ_k are the singular values of $\mathbf{X}_a^\dagger \mathbf{X}_b$.

When combining subsets it is desirable to pair subsets that are as far apart as possible. That is, a greedy algorithm could pair subset i with subset $\arg\max_j \mathcal{D}_l(i,j)$. We note, however, that at level l there will always be a pairing with at least distance $\Delta_l = \min_i \max_j \mathcal{D}_l(i,j)$. Since the system performance depends mostly on this minimum distance, Δ_l, and making the most greedy choices for each i tends to make Δ_{l+1} smaller, we have found it better to pair subset i with the subset that is closest to i but with distance no less than Δ_l. This will ensure that the minimum distance is still Δ_l, without being needlessly greedy.[1] For some constellations, however, there may be situations in which it is not possible to pair a subset i with another subset j that is a distance of at least Δ_l away simply because of a 'poor' pairing that was made previously. To mitigate this problem, the algorithm simply keeps track of each possible index j that could be paired to each subset i in a tree, i.e. all indices $J_{i,l}$ such that the distance between i and any subset in $J_{i,l}$ is at least Δ_l. The points in $J_{i,l}$ are sorted from lowest to highest distance away from i. If a proper pairing cannot be made for a given i, the algorithm can then backtrack to the previous paired subsets, uncouple them, and pair i with another subset j that is in $J_{i,l}$. The algorithm can then move through that branch of the tree until either all the subsets are paired or a subset cannot be paired. The algorithm will recursively backtrack through the tree until it finds a path in the tree that satisfies the Δ_l condition. The proposed set merging algorithm is formally given by the pseudo-code in Algorithm 1. An example of the use of the algorithm is given in Appendix A.

This algorithm aims to solve a difficult problem of pairing points together in a constellation. For larger constellations,

[1] Because of numerical rounding errors in calculating the chordal Frobenius norm, a more robust approach is to include a slight error tolerance and accept pairs with distances greater than $\Delta_l - \epsilon$ for some small ϵ (e.g., $\epsilon = 10^{-3}$).

Algorithm 1 The Set Merging Algorithm

Initialize $S_{1,i} \leftarrow \{i\}, M_1 \leftarrow M, \mathcal{D}_1(i,j) \leftarrow d(X_i, X_j)$
for $l = 1, 2, \ldots, \log_2 M$ **do**
  $\Delta_l \leftarrow \min_{i \in \{1,\ldots,M_l\}} \max_{j \in \{1,\ldots,M_l\}} D_l(i,j)$
  $\alpha_{i,l} \leftarrow 1 \; \forall \, i \in \{1, 3, \ldots, M_l - 1\}$
  for $i = 1, 3, \ldots, M_l - 1$ **do**,
    $J_{i,l} \leftarrow \arg\text{sort}(D_l(i,j))$ such that $D_l(i,j) > \Delta_l - \epsilon, j \in \{i+1, \ldots, M_l\}$
    if $\alpha_i > |J_{i,l}|$ or $|J_{i,l}| = 0$ **then**
      stepBack$(\alpha_l, i, J_{i,l})$
    else
      $j \leftarrow J_{i,l}(\alpha_{i,l})$
      $B_{a,l} \leftarrow 0, \; \forall \, a \in S_{l,i}$
      $B_{b,l} \leftarrow 1, \; \forall \, b \in S_{l,j}$
      $S_{l+1,(i+1)/2} \leftarrow S_{l,i} \cup S_{l,j}$
      swap row j of $\mathcal{D}_l$ with row $i+1$
      swap column j of $\mathcal{D}_l$ with column $i+1$
      $\alpha_{i,l} \leftarrow 1 \; \forall \, i \in \{i+2, \ldots, M_l - 1\}$
    end if
  end for
  $M_{l+1} \leftarrow M_{l/2}$
  $\mathcal{D}_{l+1}(i,j) \leftarrow \min(\mathcal{D}_l(2i-1, 2j-1), \mathcal{D}_l(2i-1, 2j), \mathcal{D}_l(2i, 2j-1), \mathcal{D}_l(2i, 2j))$
  $\forall i, j \in \{1, 2, \ldots, M_{l+1}\}$
end for
function stepBack$(\alpha_l, i, J_{i,l})$
  $i \leftarrow i - 2$
  if $\alpha_{i,l} < |J_{i,l}|$ **then**
    $\alpha_{i,l} \leftarrow \alpha_{i,l} + 1$
  else
    stepBack$(\alpha_l, i, J_{i,l})$
  end if
end function

the algorithm may take an unreasonable amount of time to backtrack through the possible pairs for each subset i, therefore, a simpler, albeit sub optimal, set merging algorithm is proposed. If a pair cannot be found for a subset i that satisfies the Δ_l constraint, a compromise is made and i is paired with a subset that is the closest distance to Δ_l away. This ensures that the algorithm is not greedy in trying to look for the perfect pairs but is able to produce labels in a single pass.

Figures 4 and 5 show the comparisons between the performance of two constellations partitioned using both the simple and the complex set merging algorithms. The backtracking algorithm makes a significant difference when the size of the constellation is small, but, this advantage fades away as the size of the constellation increases. Therefore, for the larger constellations used in the results section of this paper, the simpler algorithm is used to generate the labellings.

VI. SYSTEM DESIGN METHODOLOGY

Due to the polarization effect that occurs in the polar codes, data should be transmitted over the bit channels that have the best channel capacity while the others are not used. With a

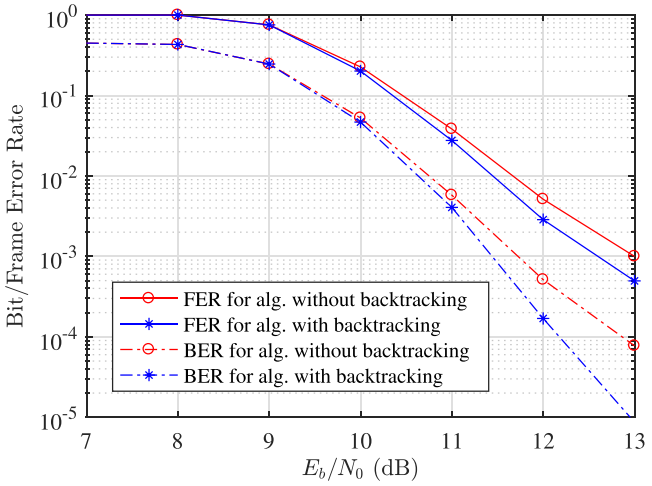

FIGURE 4. Comparing the two algorithms with a circular 8 QAM constellation.

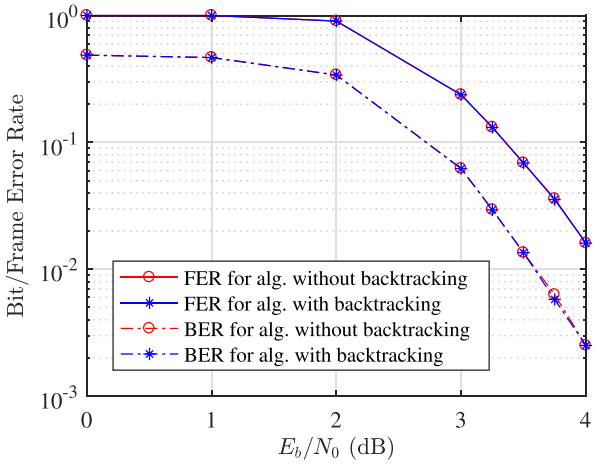

FIGURE 5. Comparing the two algorithms with a 256 point Golden code constellation.

simple binary erasure channel or a binary AWGN channel, it is easy to calculate the capacities of each bit channel. However, with our system, this computation is not trivial. Therefore, code design was carried out using MATLAB simulations. The code design component of the system is similar to the set-up described above where an M point signal constellation is combined with $m = \log_2 M$ component polar encoders each with length N' (where $mN' = N$). Because the positions of the frozen bits are not yet known however, the component codes all operate at a rate $R_i = 1$.

The design methodology involves simulating the transmission of a large number of message words through the system at a specified design SNR, and recording the bit positions where the first errors occur. When polar codes are decoded using the successive cancellation decoder, the message bits are recovered one at a time, in order from the first to the last message bit. For the purpose of code design, the "first error" bit is defined as the first message bit to be decoded incorrectly in a message word (if a message word is decoded

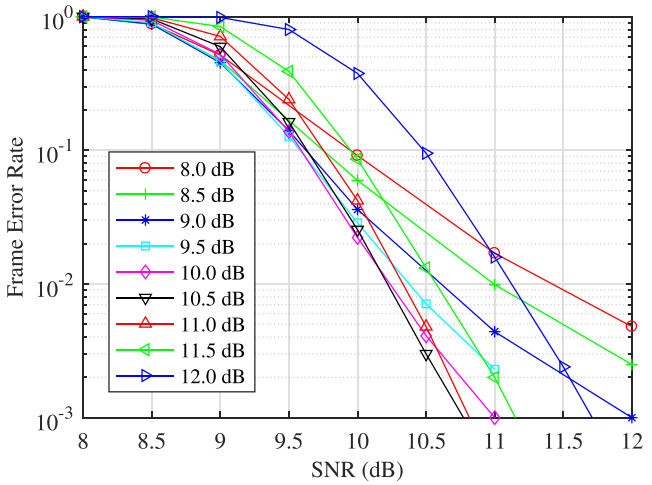

FIGURE 6. Effect of design SNR on the frame error rate (FER) performance of our system. $N' = 512$, rate $R = 4/5$, $M = 4096$.

correctly then there is no first error bit for that word). The first error probability for a bit is the probability of error for a bit, given that all previous bits are known (i.e. are either frozen or have been decoded correctly). This error represents the error of the bit channel and does not include propagated errors while decoding. In order to record only first errors, the receiver would need to stop decoding a frame as soon as the first error is detected and keep track of how many times each bit position in the frame was reached as well as how many times an error occurred there. With this method, the first error probabilities would likely not be accurately estimated in the higher level codes without simulating an extremely large number of message words. Therefore, we use a system where the receiver knows the bits that were transmitted and keeps track of every bit position in the received frame where an error occurred. Once an error is detected and registered, the receiver corrects the error, thereby preventing that error from propagating. In this way, the next error detected can be counted as a 'first' error for that particular bit position in the frame, so, the first errors can be calculated wherever they occur in every received frame. Eventually, with enough simulated bits, the total number of errors that occurred for each bit channel will be recorded and their respective first error probabilities can be calculated. This method of calculating first errors is necessary because our system uses a multilevel design where the de-mapping of each successive level depends on the correct detection of all the previous levels. Depending on what coding rate was required, the channels with the smallest first error probabilities would be chosen to transfer our data bits over.

The performance of the system can then be evaluated using Monte Carlo simulations. However, as shown in Fig. 6, the system performance is heavily dependent on the design SNR – although the designed code tends to work well at the design SNR, its performance can be quite poor at other SNRs. It is often preferable to design a code with a given rate that achieves a given target frame error rate (FER) at the lowest possible SNR. For example, from Fig. 6, to achieve a target FER of 10^{-2}, a design SNR of between 10 and 10.5 dB should be used. Since the optimal design SNR is not known in advance, we use a bisection search to find it. This search, which is known for its rapid convergence, works with the observation that a code designed at a particular SNR will perform the best at that SNR. That is, for example, as shown in Fig. 6, the code designed at 9.5 dB has the best FER performance at that SNR while the code designed at 10.5 dB performs the best at that SNR. Note also that the two SNRs produce different optimal FERs (e.g. the 10.5 dB code is optimal at a FER of 3×10^{-3}). Therefore, the bisection search starts by choosing two SNRs, one low and one high, as our starting points such that when the code designed at the lower SNR ($\mathbf{SNR}_L$) is simulated at that SNR, it produces an optimal FER greater than our target FER. Conversely, when the code designed at the higher SNR ($\mathbf{SNR}_H$) is simulated at that SNR, the resulting FER is less than our target FER. A design SNR ($\mathbf{SNR}_D$) is chosen such that $\mathbf{SNR}_D = (\mathbf{SNR}_D + \mathbf{SNR}_h)/2$. The code is designed and simulated at $\mathbf{SNR}_D$ and its optimal FER is obtained. If the FER is equal to our target FER (with a small tolerance of $\pm\epsilon$), $\mathbf{SNR}_D$ is our target design SNR. If the FER is greater than our target FER, $\mathbf{SNR}_L = \mathbf{SNR}_D$, else, $\mathbf{SNR}_H = \mathbf{SNR}_D$. The algorithm repeats by calculating a new $\mathbf{SNR}_D$ with the new values of $\mathbf{SNR}_L$ and $\mathbf{SNR}_H$ until a design SNR produces an FER approximately equal to our target FER. In order to prevent a possible infinite loop, a maximum number of iterations, *imax*, is included and once this number is reached, the algorithm ends with an error message. The algorithm is formally given by the pseudo-code in Algorithm 2. This method differs from what is normally done where the FER is optimised for a chosen SNR [20].

Algorithm 2 Bisection Algorithm

$\mathcal{F}(\mathbf{SNR})$ is the function that designs and simulates at a given $\mathbf{SNR}$, producing the optimal FER. $\mathbf{SNR}_H$ and $\mathbf{SNR}_H$ are chosen such that $\mathcal{F}(\mathbf{SNR}_L)$>target FER> $\mathcal{F}(\mathbf{SNR}_H)$.
for $i = 1, 2, \ldots, i_{max}$, **do**
  $\mathbf{SNR}_D = (\mathbf{SNR}_L + \mathbf{SNR}_H)/2$
  if $|\mathcal{F}(\mathbf{SNR}_D)$ - target FER$| < \epsilon$ **then**
    target SNR $= SNR_D$. stop program
  end if
  if $\mathcal{F}(\mathbf{SNR}_D) >$ target FER **then**
    $\mathbf{SNR}_L = \mathbf{SNR}_D$
  else
    $\mathbf{SNR}_H = \mathbf{SNR}_D$
  end if
end for
output error

VII. SIMULATION RESULTS

In order to investigate the performance of the proposed system with arbitrary multidimensional constellations, we use two constellations: the noncoherent Grassmannian

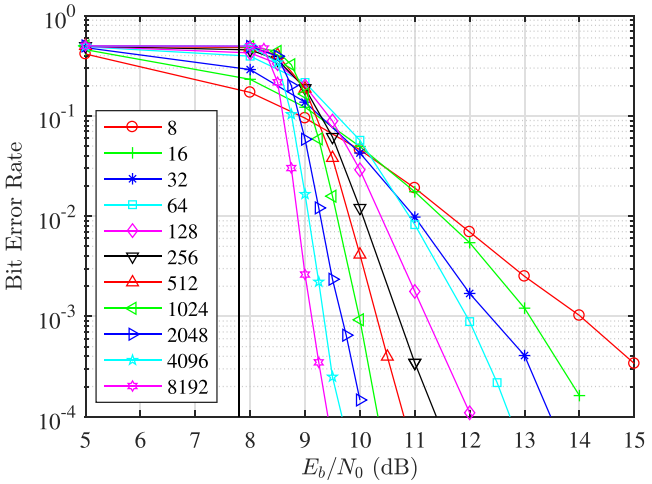

FIGURE 7. 4096 point Grassmannian constellation with multilevel polar codes of different sub-code lengths. The overall code rate $R = 4/5$. The SNR threshold is shown at an E_b/N_o of about 7.8 dB.

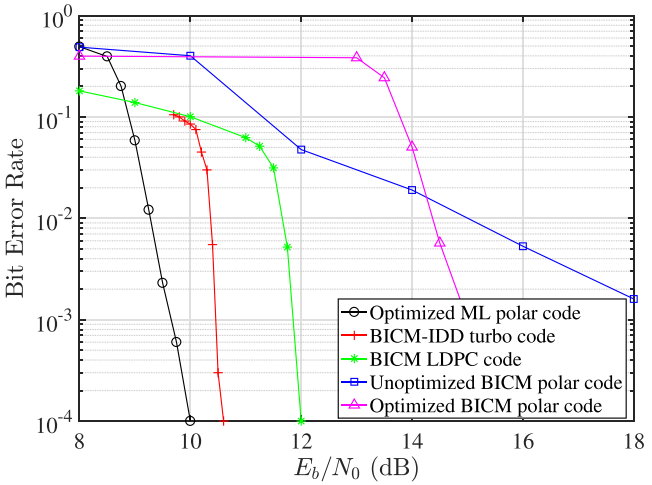

FIGURE 8. Different codes running with 4096 point Grassmannian constellation with rate $R = 4/5$. All BICM figures use quasi-Gray labelling for the constellation. Multilevel code uses the proposed set merging labelling. Unoptimized BICM codes are not optimized for the multidimensional constellation channel but for a BPSK AWGN channel only.

constellation and the coherent Golden codes. An $M = 4096$ point Grassmannian constellation ($m = 12$), designed as described in [4], for a 2×2 MIMO channel with a coherence time of $T = 4$ is used. An $M = 256$ point Golden code, designed as described in [16], for a 2×2 MIMO channel with coherence time of $T = 2$ is also used. To show the bit error curves with respect to the energy per bit, Eb/No was used instead of SNR.

As one might expect, increasing the frame size improves the bit error rate, but not above the SNR threshold set by the channel capacity. In Fig. 7 we show the effect of increasing the frame size when the code rate is 4/5. The figure shows a 4096 point Grassmannian constellation combined with polar codes of varying sub-code lengths ranging from $N' = 8$ to $N' = 8192$ which corresponds to total code lengths of $N = 96$ to $N = 98304$ bits respectively. As expected, the longer the length of the polar codes, the better the BER performance; However, this improvement is limited by the channel capacity. The SNR limit at capacity was calculated using the expression computed for Grassmannian constellations with equal number of transmit and receive antennas found in [3].

The SNR threshold for the constellation at rate 4/5 (2.4 bits per channel use) is calculated to be 11.6 dB and this corresponds to an E_b/N_o value of about 7.8 dB. As Fig. 7 shows, with a sub-code length of $N' = 8192$, this design is able to operate within 1.6 dB of the approximate noncoherent ergodic capacity at a BER of 10^{-4}. This is the closest to the channel capacity for a noncoherent system that has been reached as far as we are aware.

Thus far, the results seem to indicate that our new system performs quite well. However, it is useful to compare it with other well known design methodologies and codes. Fig. 8 shows the comparison between our polar code design and other polar, turbo and LDPC designs with a 4096 point Grassmannian constellation at a rate of 4/5. The figure shows the BER performance of our design using $m = 12$ component codes and choosing each to have $N' = 2048$ and $N = 24576$. Also shown is the performance of a turbo coded system as designed in [14] with $N = 32016$, using bit interleaved coded modulation with iterative detection and decoding (BICM-IDD), and quasi-Gray labelling for the constellation. Also, the performance of a BICM system using the standard single level DVB-S2 LDPC code also with quasi-Gray labelling for the constellation and $N = 64800$ is shown in the figure. The performance of an unoptimized (that is, not optimized for the Grassmannian channel) BICM polar code system with $N = 32768$ is shown as well. Finally we see in the figure the same BICM polar code system with $N = 32768$ but with polar codes designed for the Grassmannian channel using our design methodology. As the figure shows, the polar coding technique outperforms all the other BICM based techniques even those of turbo and LDPC codes with longer code lengths. It is also of note that the polar encoder/decoder has significantly less complexity compared with the other two codes. Therefore, the polar scheme not only provides better error rate performance, but it does so with a much lower complexity.

Fig. 9 shows the performance comparison between our system and other design methods using a 256-point Golden code constellation with $N = 8192$ over a MIMO fading channel. We compare the performance of the Golden code constellation, labelled with either our set merging algorithm or with Gray labelling, when used with an optimized multilevel polar code designed for this constellation and labelling with a MIMO fading channel. We see more than 2 dB of gain at a BER of 10^{-4} when set merging labelling is used instead of Gray labelling. To illustrate the importance of designing the polar code for the intended channel, we also show the performace of the set-merging labelled constellation when

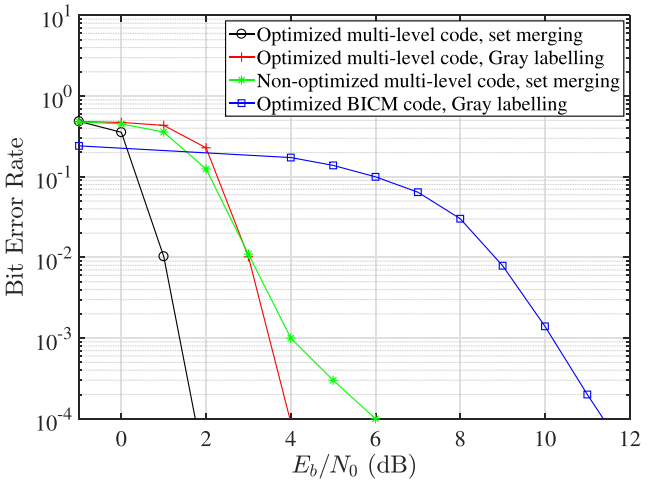

FIGURE 9. Polar codes with 256 point coherent Golden codes with $R = 1/2$ and $N = 8192$.

used with a multilevel polar code designed for use over an AWGN channel. By optimizing the polar codes for the MIMO fading channel, we are able to achieve more than a 4 dB gain compared with the unoptimized case at a BER of 10^{-4}. For a final comparision, the performance of a standard single-level BICM polar code, optimized for the Golden code constellation with Gray labelling for MIMO fading is shown. The optimized multilevel polar code achieves a gain of about 9.5 dB a BER of 10^{-4} over the optimized BICM method.

VIII. COMPLEXITY ANALYSIS

So far, we have seen that our system is able to produce significantly better error rate performances compared with other coding and mapping systems. Therefore, an important metric to consider is the receiver complexity, especially compared with the other coded schemes. Precise comparisons of relative complexity between different algorithms is subjective without a standard definition of complexity. For software implementations, the execution speed is a useful metric whereas for hardware implementations, the chip area and power draw may be more relevant. In this paper, we will use the number of floating point operations (flop) required to de-map and decode one message. Although the flop count is a useful and widely used metric, it does not distinguish between the relative costs of performing different operations, (for example, modern Pentium processors can perform additions twice as quickly as multiplications which in turn are 20 times faster than division [21]), which can also vary greatly depending on the type of processor being used and how carefully optimized the algorithm is to fully exploit the operator pipeline. Since these finer details are particularly subjective, in the following we restrict our attention to the flop count. Each complex addition requires two flop and each complex multiplication requires six flop. Logarithmic and exponential operations, which are typically calculated using rational polynomial interpolation, require 20 flop.

TABLE 1. De-mapping complexity comparison of Grassmannian constellation and Golden codes.

Method	Number of flop
Grassmannian	$N'M(8N_tN_rT + 2N_tN_r + 20)$
Golden Code	$N'M(8N_t^2N_r + 4N_tN_r + 20)$

A. DE-MAPPING COMPLEXITY ANALYSIS

We first consider the de-mapping complexities of Grassmannian constellations and Golden code signals. As both schemes use the same multilevel polar coding technique, the encoding and decoding complexities, described in the next subsection, are therefore the same.

After the signals $\mathbf{Y}$ are received, the de-mapper works out the ML probabilities as follows:

$$\Pr\{\mathbf{Y}|\mathbf{X}\} \propto \begin{cases} \exp\left\{\dfrac{\|\mathbf{X}^\dagger\mathbf{Y}\|^2}{\sigma_W^2(1+\sigma_W^2)}\right\} & \text{for the Grassmannian,} \\ \exp\left\{\dfrac{\|\mathbf{Y}-\mathbf{X}\mathbf{H}\|^2}{\sigma_W^2}\right\} & \text{for the Golden code.} \end{cases} \tag{15}$$

The de-mapper calculates these values for every received block (i.e. after T channel uses). For the Grassmannian de-mapper, the value $\mathbf{X}^\dagger\mathbf{Y}$ is calculated first. Since $\mathbf{X}^\dagger$ is an $N_t \times T$ matrix and $\mathbf{Y}$ is a $T \times N_r$ matrix, the calculation has $N_tN_r(T-1)$ complex additions and N_tN_rT complex multiplications, for a total of $8N_tN_rT - 2N_tN_r$ flop. Next, the $\|\cdot\|^2$ operation is performed, where $\|\mathbf{A}\|^2 = \sum_{i=1}^{N_r}\sum_{j=1}^{N_t}|a_{i,j}|^2$ with $a_{i,j}$ being the element in row i and column j of $\mathbf{A}$. This requires $6N_tN_r$ flop to calculate all $|a_{i,j}|^2$, and $N_tN_r - 1$ flop to sum all the elements, for a total of $7N_tN_r - 1$ flop. To finally calculate $\Pr\{\mathbf{Y}|\mathbf{X}\}$, one division by the constant $\sigma_W^2(1+\sigma_W^2)$ and one exponent operation (20 flop) are also needed. Therefore, calculation of $\Pr\{\mathbf{Y}|\mathbf{X}\}$ requires a total of $8N_tN_rT + 5N_tN_r + 20$ flop. This must be repeated M times, for each $\mathbf{X}$ in the constellation. Furthermore, since the transmission of one message word requires the transmission of N' Grassmannian symbols, the flop count must be multiplied by N', where $N' = N/m$. The total flop count for the Grassmannian de-mapper is shown in Table 1.

Similarly, for the coherent Golden code de-mapper, $\|\mathbf{Y}-\mathbf{X}\mathbf{H}\|^2$ is calculated for each $\mathbf{X}$ and for each transmission block. The $N_t \times N_t$ matrix, $\mathbf{X}$, is multiplied by the $N_t \times N_r$ matrix, $\mathbf{H}$, requiring $N_tN_rN_t$ complex multiplications, and $N_tN_r(N_t-1)$ complex additions, or $8N_t^2N_r - 2N_tN_r$ flop. Subtracting the result from $\mathbf{Y}$ requires an additional $2N_tN_r$ flop, and calculating $\|\cdot\|^2$ needs another $4N_tN_r - 1$ flop. Calculation of $\Pr\{\mathbf{Y}|\mathbf{X}\}$ requires an additional 21 flop for the exponent and division by σ_W^2. The total flop count for all M possibilities of $\mathbf{X}$ and for all N' blocks is given in Table 1.

As an example using our set-up with $N_t = N_r = 2, T = 4$, and using 4096 point constellations and a sub-code length $N' = 512$, we see the number of operations required to de-map both a Grassmannian and a Golden code constellation

TABLE 2. Example of de-mapping complexity comparison of Grassmannian constellation and Golden codes for $N_t = N_r = 2$, $T = 4$, $M = 4096$, and $N' = 512$.

Method	Number of flop
Grassmannian	$327,155,712$
Golden Code	$209,715,200$

in Table 2. As expected, the table shows, the Grassmannian constellation method is more computationally intensive than the Golden code method and this is due to the fact that while information bits are transmitted over T timeslots for a Grassmannian constellation, the Golden code only transmits information bits over N_t time slots. Therefore, while the Grassmannian approach has been shown to produce better error rate performances than its noncoherent Golden code counterpart under block fading conditions, there is a trade-off between complexity and performance. However, in both cases, there are between two hundred to three hundred and thirty million flop, which may not amount to much of a difference depending on the coding used in the system. As we will soon see, a large part of a system's complexity could potentially come from the decoding process, therefore, choosing a good coding scheme is necessary.

B. BIT LLR CALCULATION COMPLEXITY ANALYSIS

The decoders are fed soft symbols (the LLRs), which are calculated from $\Pr\{\mathbf{Y}|\mathbf{X}\}$. As shown previously in (12), the LLRs for the polar MLC are calculated as

$$\lambda_l = \ln \frac{\sum_{\mathbf{X}\in\chi_{l,0}} \Pr\{\mathbf{Y}|\mathbf{X}\}}{\sum_{\mathbf{X}\in\chi_{l,1}} \Pr\{\mathbf{Y}|\mathbf{X}\}}, \quad \forall\, 1 \le l \le m. \tag{16}$$

We see from (16) that for each bit level one, there is one log operation and one divide operation for a total of m log operations and m divide operations per received symbol/block. When $l = 1$, the sets $\chi_{l,1}$ and $\chi_{l,0}$ are each half the size of $\Pr\{\mathbf{Y}|\mathbf{X}\}$ which has M elements. The summation operator adds these elements in the numerator and denominator, requiring $\frac{M}{2} - 1$ additions each. With $l = 2$, the sets $\chi_{l,k}$, $k = \{0, 1\}$, which depend on the knowledge of the decoded bits of the level $l = 1$, are half the size of the sets $\chi_{1,k}$. Thus the numerator and denominator each have $\frac{M}{4} - 1$ additions. Similarly, the sets $\chi_{3,k}$, which depend on the knowledge of the decoded bits of levels $l = 1$ and $l = 2$, are half the size of the sets $\chi_{2,k}$ and the numerator and denominator each have $\frac{M}{8} - 1$ additions each. When $l = m$, the sets $\chi_{l,k}$ have only one element each and therefore no additions. There are therefore a total of $2\sum_{l=1}^{m}\left(\frac{M}{2^l} - 1\right) = 2(M - 1 - m)$ additions per block. Taking into account the division and logarithm needed in (16) for each bit increases the flop count per block to $2(M - 1) + 19m$, which must be multiplied by N' to give the count per frame as shown in Table 3.

When BICM is used instead of MLC, knowledge of the previously decoded bits cannot be exploited since the LLRs of all bits are calculated at the same time. Therefore the sets $\chi_{l,k}$ in (16) are always of size $M/2$, and the total flop count to calculate the LLRs is $2m(\frac{M}{2} - 1) + 21m = m(M + 19)$ per block.

TABLE 3. LLR Calculation, b = number of de-mapping iterations for $m = \log_2 M$.

MLC	$N'(2M - 2 + 19m)$
BICM	$N(M + 19)$
BICM-IDD	$N(M + 19)$ $+(b-1)N'[(22 + m)M - 2 + 19m]$

TABLE 4. Decoding complexity of turbo and multi-level polar codes for $n' = \log_2 N'$, k = constraint length, N = Frame length, i = total number of decoder iterations, d = number of decoders, b = number of de-mapping iterations.

Component	Number of Flop
Polar Decoder	$32.5Nn'$
Turbo Decoder	$diN(11(2^{k+1}) + 2^{k+3}$ $+2^{2k+1} + 2^{k+2} + 19)$

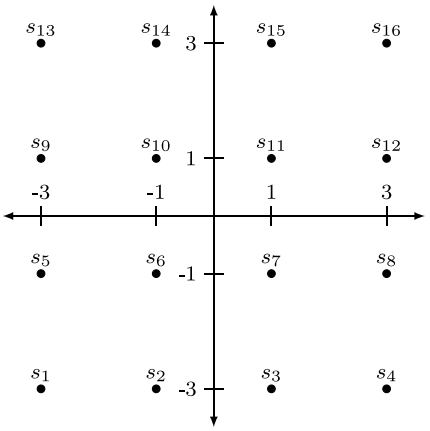

FIGURE 10. Sample 16-QAM Constellation.

The use of BICM-IDD, as in [13], further increases the complexity of the LLR calculations. Instead of (16) we use

$$\lambda_l = \ln \frac{\sum_{\mathbf{X}\in\chi_{l,0}} \Pr\{\mathbf{Y}|\mathbf{X}\}\mu_X}{\sum_{\mathbf{X}\in\chi_{l,1}} \Pr\{\mathbf{Y}|\mathbf{X}\}\mu_X}, \quad \forall\, 1 \le l \le m. \tag{17}$$

where μ_X is the a priori probability that $\mathbf{X}$ was transmitted, based on feedback from the decoder during the previous iteration. Calculation of μ_X for all M values of $\mathbf{X}$ can be performed efficiently using $M-2$ additions and M exponents, or $21M-2$ flop. A further M multiplications are needed when multiplying by $\Pr\{\mathbf{Y}|\mathbf{X}\}$, so BICM-IDD requires $(22+m)M-2+19m$ flop for each block for all but the first iteration, when only $m(M + 19)$ are needed.

C. DECODING COMPLEXITY ANALYSIS

We now consider the total decoding complexity of the polar MLC system. Recall from Fig. 2 that the decoding process requires a multi-stage decoding method in which bits from

$\mathcal{D}_1$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0	$\sqrt{4}$	$\sqrt{16}$	$\sqrt{36}$	$\sqrt{4}$	$\sqrt{8}$	$\sqrt{20}$	$\sqrt{40}$	$\sqrt{16}$	$\sqrt{20}$	$\sqrt{32}$	$\sqrt{52}$	$\sqrt{36}$	$\sqrt{40}$	$\sqrt{52}$	$\sqrt{72}$
2	$\sqrt{4}$	0	$\sqrt{4}$	$\sqrt{16}$	$\sqrt{8}$	$\sqrt{4}$	$\sqrt{8}$	$\sqrt{20}$	$\sqrt{20}$	$\sqrt{16}$	$\sqrt{20}$	$\sqrt{32}$	$\sqrt{40}$	$\sqrt{36}$	$\sqrt{40}$	$\sqrt{52}$
3	$\sqrt{16}$	$\sqrt{4}$	0	$\sqrt{4}$	$\sqrt{20}$	$\sqrt{8}$	$\sqrt{4}$	$\sqrt{8}$	$\sqrt{32}$	$\sqrt{20}$	$\sqrt{16}$	$\sqrt{20}$	$\sqrt{52}$	$\sqrt{40}$	$\sqrt{36}$	$\sqrt{40}$
4	$\sqrt{36}$	$\sqrt{16}$	$\sqrt{4}$	0	$\sqrt{40}$	$\sqrt{20}$	$\sqrt{8}$	$\sqrt{4}$	$\sqrt{52}$	$\sqrt{32}$	$\sqrt{20}$	$\sqrt{16}$	$\sqrt{72}$	$\sqrt{52}$	$\sqrt{40}$	$\sqrt{36}$
5	$\sqrt{4}$	$\sqrt{8}$	$\sqrt{20}$	$\sqrt{40}$	0	$\sqrt{4}$	$\sqrt{16}$	$\sqrt{36}$	$\sqrt{4}$	$\sqrt{8}$	$\sqrt{20}$	$\sqrt{40}$	$\sqrt{16}$	$\sqrt{20}$	$\sqrt{32}$	$\sqrt{52}$
6	$\sqrt{8}$	$\sqrt{4}$	$\sqrt{8}$	$\sqrt{20}$	$\sqrt{4}$	0	$\sqrt{4}$	$\sqrt{16}$	$\sqrt{8}$	$\sqrt{4}$	$\sqrt{8}$	$\sqrt{20}$	$\sqrt{20}$	$\sqrt{16}$	$\sqrt{20}$	$\sqrt{32}$
7	$\sqrt{20}$	$\sqrt{8}$	$\sqrt{4}$	$\sqrt{8}$	$\sqrt{16}$	$\sqrt{4}$	0	$\sqrt{4}$	$\sqrt{20}$	$\sqrt{8}$	$\sqrt{4}$	$\sqrt{8}$	$\sqrt{32}$	$\sqrt{20}$	$\sqrt{16}$	$\sqrt{20}$
8	$\sqrt{40}$	$\sqrt{20}$	$\sqrt{8}$	$\sqrt{4}$	$\sqrt{36}$	$\sqrt{16}$	$\sqrt{4}$	0	$\sqrt{40}$	$\sqrt{20}$	$\sqrt{8}$	$\sqrt{4}$	$\sqrt{52}$	$\sqrt{32}$	$\sqrt{20}$	$\sqrt{16}$
9	$\sqrt{16}$	$\sqrt{20}$	$\sqrt{32}$	$\sqrt{52}$	$\sqrt{4}$	$\sqrt{8}$	$\sqrt{20}$	$\sqrt{40}$	0	$\sqrt{4}$	$\sqrt{16}$	$\sqrt{36}$	$\sqrt{4}$	$\sqrt{8}$	$\sqrt{20}$	$\sqrt{40}$
10	$\sqrt{20}$	$\sqrt{16}$	$\sqrt{20}$	$\sqrt{32}$	$\sqrt{8}$	$\sqrt{4}$	$\sqrt{8}$	$\sqrt{20}$	$\sqrt{4}$	0	$\sqrt{4}$	$\sqrt{16}$	$\sqrt{8}$	$\sqrt{4}$	$\sqrt{8}$	$\sqrt{20}$
11	$\sqrt{32}$	$\sqrt{20}$	$\sqrt{16}$	$\sqrt{20}$	$\sqrt{20}$	$\sqrt{8}$	$\sqrt{4}$	$\sqrt{8}$	$\sqrt{16}$	$\sqrt{4}$	0	$\sqrt{4}$	$\sqrt{20}$	$\sqrt{8}$	$\sqrt{4}$	$\sqrt{8}$
12	$\sqrt{52}$	$\sqrt{32}$	$\sqrt{20}$	$\sqrt{16}$	$\sqrt{40}$	$\sqrt{20}$	$\sqrt{8}$	$\sqrt{4}$	$\sqrt{36}$	$\sqrt{16}$	$\sqrt{4}$	0	$\sqrt{40}$	$\sqrt{20}$	$\sqrt{8}$	$\sqrt{4}$
13	$\sqrt{36}$	$\sqrt{40}$	$\sqrt{52}$	$\sqrt{72}$	$\sqrt{16}$	$\sqrt{20}$	$\sqrt{32}$	$\sqrt{52}$	$\sqrt{4}$	$\sqrt{8}$	$\sqrt{20}$	$\sqrt{40}$	0	$\sqrt{4}$	$\sqrt{16}$	$\sqrt{36}$
14	$\sqrt{40}$	$\sqrt{36}$	$\sqrt{40}$	$\sqrt{52}$	$\sqrt{20}$	$\sqrt{16}$	$\sqrt{20}$	$\sqrt{32}$	$\sqrt{8}$	$\sqrt{4}$	$\sqrt{8}$	$\sqrt{20}$	$\sqrt{4}$	0	$\sqrt{4}$	$\sqrt{16}$
15	$\sqrt{52}$	$\sqrt{40}$	$\sqrt{36}$	$\sqrt{40}$	$\sqrt{32}$	$\sqrt{20}$	$\sqrt{16}$	$\sqrt{20}$	$\sqrt{20}$	$\sqrt{8}$	$\sqrt{4}$	$\sqrt{8}$	$\sqrt{16}$	$\sqrt{4}$	0	$\sqrt{4}$
16	$\sqrt{72}$	$\sqrt{52}$	$\sqrt{40}$	$\sqrt{36}$	$\sqrt{52}$	$\sqrt{32}$	$\sqrt{20}$	$\sqrt{16}$	$\sqrt{40}$	$\sqrt{20}$	$\sqrt{8}$	$\sqrt{4}$	$\sqrt{36}$	$\sqrt{16}$	$\sqrt{4}$	0

$\mathcal{D}_2$	1,11	3,9	5,15	7,13	4,10	2,12	8,14	6,16
1,11	0	$\sqrt{16}$	$\sqrt{4}$	$\sqrt{4}$	$\sqrt{4}$	$\sqrt{4}$	$\sqrt{8}$	$\sqrt{8}$
3,9	$\sqrt{16}$	0	$\sqrt{4}$	$\sqrt{4}$	$\sqrt{4}$	$\sqrt{4}$	$\sqrt{8}$	$\sqrt{8}$
5,15	$\sqrt{4}$	$\sqrt{4}$	0	$\sqrt{16}$	$\sqrt{8}$	$\sqrt{8}$	$\sqrt{4}$	$\sqrt{4}$
7,13	$\sqrt{4}$	$\sqrt{4}$	$\sqrt{16}$	0	$\sqrt{8}$	$\sqrt{8}$	$\sqrt{4}$	$\sqrt{4}$
4,10	$\sqrt{4}$	$\sqrt{4}$	$\sqrt{8}$	$\sqrt{8}$	0	$\sqrt{16}$	$\sqrt{4}$	$\sqrt{4}$
2,12	$\sqrt{4}$	$\sqrt{4}$	$\sqrt{8}$	$\sqrt{8}$	$\sqrt{16}$	0	$\sqrt{4}$	$\sqrt{4}$
8,14	$\sqrt{8}$	$\sqrt{8}$	$\sqrt{4}$	$\sqrt{4}$	$\sqrt{4}$	$\sqrt{4}$	0	$\sqrt{16}$
6,16	$\sqrt{8}$	$\sqrt{8}$	$\sqrt{4}$	$\sqrt{4}$	$\sqrt{4}$	$\sqrt{4}$	$\sqrt{16}$	0

$\mathcal{D}_3$	1,11,3,9	5,15,7,13	4,10,2,12	8,14,6,16
1,11,3,9	0	$\sqrt{4}$	$\sqrt{4}$	$\sqrt{8}$
5,15,7,13	$\sqrt{4}$	0	$\sqrt{8}$	$\sqrt{4}$
4,10,2,12	$\sqrt{4}$	$\sqrt{8}$	0	$\sqrt{4}$
8,14,6,16	$\sqrt{8}$	$\sqrt{4}$	$\sqrt{4}$	0

$\mathcal{D}_4$	1,11,3,9 8,14,6,16	4,10,2,12 5,15,7,13
1,11,3,9 8,14,6,16	0	$\sqrt{4}$
4,10,2,12 5,15,7,13	$\sqrt{4}$	0

FIGURE 11. Distance table for labelling the 16-QAM constellation shown in Fig. 10.

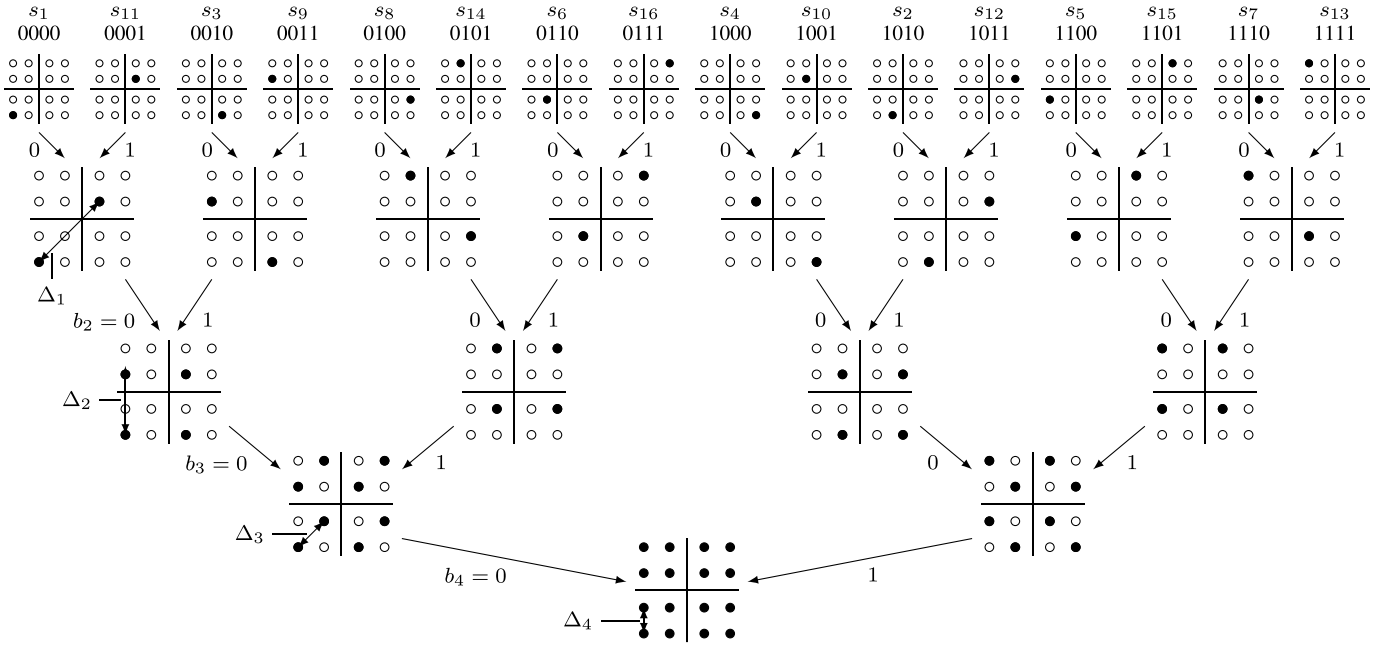

FIGURE 12. Set merging the 16-QAM signal constellation using our algorithm, showing how pairs of signal points are merged into sets based on the distance tables, and then pairs of these sets are in turn merged into larger sets. The signal points, and the resulting bit labelling, are shown at the top. $\Delta_1 = \sqrt{32}$, $\Delta_2 = \sqrt{16}$, $\Delta_3 = \sqrt{8}$, $\Delta_4 = \sqrt{4}$.

lower levels are decoded, re-encoded, and used to decode bits of the higher levels. Table 4 compares the decoding complexity of the polar MLC decoder and that of turbo codes. First, we consider the complexity of a single polar decoder.

The polar decoder makes use of two basic node calculations in order to decode received signals. These are

$$\lambda_{i,j} = \begin{cases} \ln\left[\dfrac{\exp\{\lambda_{i,j+1}\}\exp\{\lambda_{i+2^j,j+1}\}+1}{\exp\{\lambda_{i,j+1}\}+\exp\{\lambda_{i+2^j,j+1}\}}\right], \\ \qquad\qquad \text{if } \left\lfloor \frac{i-1}{2^j} \right\rfloor \equiv 0 \mod 2 \\ \lambda_{i,j+1} + (1-2\hat{v}_{i-2^j,j})\lambda_{i-2^j,j+1}, \quad \text{otherwise} \end{cases} \tag{18}$$

for $1 \leq i \leq N'$ and $0 \leq j \leq n'$, where $n' = \log_2 N'$. The upper node calculation for $\lambda_{i,j}$ requires two exponential computations, two additions, one multiplication, one division, and one logarithm, or 64 flop. The lower node calculation needs either one addition or one subtraction, depending on $\hat{v}_{i-2^j,j}$. For each polar decoder in the MLC, the upper node calculations occur in each column $j < n'$ and in half the rows of i where $\left\lfloor \frac{i}{j-1} \right\rfloor \equiv 0 \pmod 2$, while the lower node calculations also occur in each column and in the other half of the rows. Therefore, the lower and upper node values of $\lambda_{i,j}$ are each calculated a total of $\frac{N'}{2}n'$ times per decoder and there are m decoders in the MLC system.

In [22], the complexity of a turbo decoder per bit was determined. For each code bit, each decoder requires 2^k exponents, one log, $2^{k+3}+2^{2k}$ multiplications, $2^{k+2}+2^{2k}-2$ additions, and $2^{k+1}+1$ divisions for a total of $11(2^{k+1}) + 2^{k+3} + 2^{2k+1} + 2^{k+2} + 19$ flop where k is the constraint length of the turbo encoders. To obtain the number of flop for the entire decoding procedure, this must be multiplied by the frame length N, the number of decoders used d and the total number of decoder iterations performed a.

TABLE 5. Decoding complexity example of turbo and multi-level polar codes for $n' = 9$, $k = 5$, $N = 6144$, $i = 32$, $d = 2$, $b = 4$.

Component	Number of Flop
Polar Decoder	$1,797,120$
Turbo Decoder	$1,240,596,480$

As an example, with a sub-code length of $N' = 512$ and a Grassmannian constellation with $M = 4096$ points, $m = \log_2 4096 = 12$ and $N = 12 \times 512 = 6144$ bits, Table 5 shows the number of operations needed for decoding polar and turbo codes per received frame. The turbo decoder implemented in [13] uses $d = 2$ MAP decoders with $b = 4$ de-mapping/decoding iterations (the LLRs are calculated 4 times) and $a = 8$ BCJR-based turbo iterations within each de-mapping/decoding iteration for a total of $i = 32$ iterations each, and $k = 5$. We see from Table 5 that the turbo decoder is orders of magnitude more complex than the polar MLC decoders.

As for the LDPC decoder, it is difficult to estimate the number of iterations needed for decoding a frame. However, because the LDPC system simulated in this paper uses BICM (Table 3). The LLR calculations, even without taking into consideration the computational cost of the actual LDPC decoder, require 25, 282, 560 flop using our example. In comparison, the polar MLC requires 6, 107, 136 flop for decoding and LLR calculations. The LLR calculations for the BICM LDPC system alone require about 4 times more flop. It is of note however, that the SC polar decoder is a serial process and cannot be parallelized like the decoding process for LDPC codes and this affects the latency. However, even with this taken into account, the overall latency for the polar code system will still be less than that of the LDPC codes. Therefore, we see that the polar decoder system is significantly less complex than both the LDPC and the turbo systems while still providing superior error rate performance.

IX. CONCLUSION

A novel methodology for designing polar codes that work effectively with Grassmannian constellations was proposed in this work. This combination, based on multi-level polar coding, was effectuated by the help of a novel set merging algorithm. This algorithm is a generalised one that works both for regular and irregular constellations. It was shown that this algorithm was able to label both noncoherent (Grassmannian) and coherent (Golden codes) multidimensional constellations. In addition, a different way of designing polar codes which involves finding the codes that achieve a target FER at the lowest possible design SNR was proposed. This is in contrast to the previous design methodologies which seek to minimize the FER at one design SNR, which tends to produce codes that perform poorly at other SNRs. By combining these two propositions, the error rate curves showed that this novel system came the closest to the noncoherent MIMO capacity compared with the other state-of-the-art coding techniques. In addition, this novel method is significantly less computationally intensive than the other BICM based coding methods.

APPENDIX
APPLYING THE PROPOSED SET MERGING ALGORITHM ON A 16-QAM CONSTELLATION

As an illustrative example of the operation of the set merging algorithm, consider the rectangular 16-QAM constellation shown in Fig. 10, which has a distance metric table $\mathcal{D}_1$ as shown in Fig. 11. Here, there are $M_1 = 16$ sets of $\mathbf{S}_{1,i}$ each containing 1 point. The maximum distance from each point, highlighted in the figure, is obtained and the minimum of these is selected. In this example, $\Delta_1 = \sqrt{32}$. Point $i = 1$ is paired with point $j = 11$ then i is assigned a bit value of 0 while j is assigned a bit value of 1 in the first bit position (i.e., the least significant bit). These two points are then concatenated into one set $\mathbf{S}_{2,1}$. Row $i+1 = 2$ is swapped with row $j = 11$ such that the paired sets are next to each other and row 2 is now in the eleventh position (shown on top of the table). The same is done for the columns. In the next iteration, $i = 3$ and is paired with $j = 9$. The bits are assigned, the sets are concatenated and the rows and columns are swapped. This continues till $i = 15$ and the paired sets are $\mathbf{S}_{2,1} = \{1, 11\}$, $\mathbf{S}_{2,2} = \{3, 9\}$, $\mathbf{S}_{2,3} = \{5, 15\}$, $\mathbf{S}_{2,4} =$

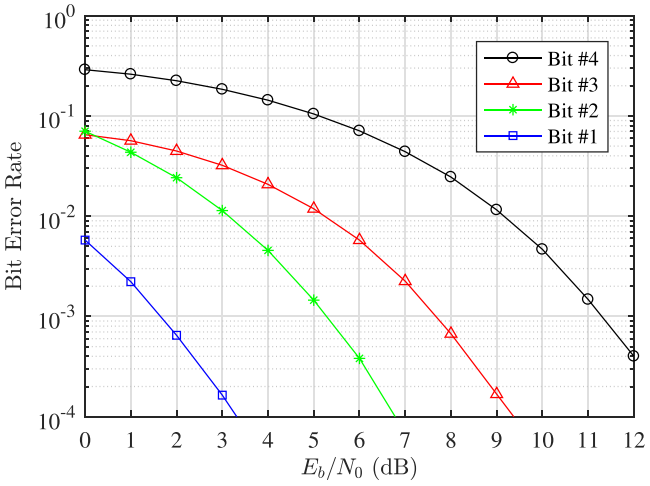

FIGURE 13. Bit error rate per bit of the 16-QAM constellation labelled with set merging.

$\{7, 13\}$, $\mathbf{S}_{2,5} = \{4, 10\}$, $\mathbf{S}_{2,6} = \{2, 12\}$, $\mathbf{S}_{2,7} = \{8, 14\}$ and $\mathbf{S}_{2,8} = \{6, 16\}$. At the end, there are $M_2 = 8$ sets of $\mathbf{S}_{2,i}$ each containing 2 points. The distance table $\mathcal{D}_2$, shown in the figure, is generated by taking the minimum distance between each point in the new sets and all the sets. The maximum distance from each set is obtained and the minimum of these is selected with $\Delta_2 = 4$. Sets $\mathbf{S}_{2,1}$ and $\mathbf{S}_{2,2}$ are concatenated with all the points in the former set assigned a value of 0 and all the points in the latter a value of 1 in the second bit position. No swaps are necessary here because the paired set is already adjacent. In the second iteration, $i = 3$ and the sets $\mathbf{S}_{2,3}$ and $\mathbf{S}_{2,4}$ are concatenated and the bits are assigned. This continues till all the sets are paired and concatenated and the bits are assigned. In the next level, there are $M_3 = 4$ sets where $\mathbf{S}_{3,1} = \{1, 11, 3, 9\}$, $\mathbf{S}_{3,2} = \{5, 15, 7, 13\}$, $\mathbf{S}_{3,3} = \{4, 10, 2, 12\}$, and $\mathbf{S}_{3,4} = \{8, 14, 6, 16\}$. By taking the minimum distance between these sets, $\mathcal{D}_3$ is formed. The minimum of the maximum distance from each set is obtained an at this level, $\Delta_3 = \sqrt{8}$. $\mathbf{S}_{3,1}$ is paired with $\mathbf{S}_{3,4}$ and sets $\mathbf{S}_{3,2}$ and $\mathbf{S}_{3,4}$ are swapped in the distance table. The bits are labelled appropriately and the next two sets are paired, concatenated and labelled accordingly. In the last level with $M_4 = 2$ sets where $\mathbf{S}_{4,1} = \{1, 11, 3, 9, 8, 14, 6, 16\}$ and $\mathbf{S}_{4,2} = \{4, 10, 2, 12, 5, 15, 7, 13\}$, $\Delta_4 = 2$. The two sets are paired, all points in $\mathbf{S}_{4,1}$ are labelled 0 and $\mathbf{S}_{4,2}$ are labelled 1 in the fourth bit position, and these are concatenated giving a final set $\mathbf{S}_{5,1}$ which contains all of the points in the constellation. Thus, the set merging is completed and a visual example can be seen in Fig. 12.

Fig. 13 shows the BER per bit level of the 16-QAM constellation labelled using the proposed set merging algorithm. This bit variance can be obtained in simulation when the receiver detects the signals one bit at a time, assuming the previous bits have been detected correctly. Because the bits at the lower levels have a greater distance apart in the signal space than those at the higher levels, i.e. $\Delta_l > \Delta_{l+1}$, those bits show a better performance.

We see therefore that our set merging algorithm converges to the Ungerboeck SP solution when used on a 16-QAM constellation. Our numerical experiments suggest that the proposed set merging algorithm yields close-to-optimal labellings in the majority of cases.

Acknowledgement

This paper was presented at the IEEE International Conference on Communications, Kuala Lumpur, Malaysia, May 2016 [1].

REFERENCES

[1] P. R. Balogun, I. D. Marsland, R. H. Gohary, and H. Yanikomeroglu, "Polar codes for noncoherent MIMO signalling," in *Proc. IEEE Int. Conf. Commun.*, Kuala Lumpur, Malaysia, May 2016, pp. 1–6.

[2] G. D. Forney, Jr., and L.-F. Wei, "Multidimensional constellations. I. Introduction, figures of merit, and generalized cross constellations," *IEEE J. Sel. Areas Commun.*, vol. 7, no. 6, pp. 877–892, Aug. 1989.

[3] L. Zheng and D. N. C. Tse, "Communication on the Grassmann manifold: A geometric approach to the noncoherent multiple-antenna channel," *IEEE Trans. Inf. Theory*, vol. 48, no. 2, pp. 359–383, Feb. 2002.

[4] R. H. Gohary and T. N. Davidson, "Noncoherent MIMO communication: Grassmannian constellations and efficient detection," *IEEE Trans. Inf. Theory*, vol. 55, no. 3, pp. 1176–1205, Mar. 2009.

[5] S. Roger *et al.*, "Non-coherent MIMO communication for the 5th generation mobile: Overview and practical aspects," *Waves J.*, vol. 6, pp. 5–13, 2014.

[6] R. W. Hamming, "Error detecting and error correcting codes," *Bell Syst. Tech. J.*, vol. 29, no. 2, pp. 147–160, Apr. 1950.

[7] E. Arikan, "Channel polarization: A method for constructing capacity-achieving codes," in *Proc. IEEE Int. Symp. Inf. Theory*, Toronto, ON, Canada, Jul. 2008, pp. 1173–1177.

[8] K. Nui, K. Chen, J. Lin, and Q. Zhang, "Polar codes: Primary concepts and practical decoding algorithms," *IEEE Commun. Mag.*, vol. 52, no. 7, pp. 192–203, Jul. 2014.

[9] M. Seidl, A. Schenk, C. Stierstorfer, and J. B. Huber, "Aspects of polar-coded modulation," in *Proc. 9th Int. ITG Conf. Sys., Commun. Coding*, Munich, Germany, Jan. 2013, pp. 1–6.

[10] H. Imai and S. Hirakawa, "A new multilevel coding method using error-correcting codes," *IEEE Trans. Inf. Theory*, vol. IT-23, no. 3, pp. 371–377, May 1977.

[11] E. Zehavi, "8-PSK trellis codes for a Rayleigh channel," *IEEE Trans. Commun.*, vol. 40, no. 5, pp. 873–884, May 1992.

[12] E. Viterbo and Y. Hong, "Applications of the golden code," in *Proc. Inf. Theory Appl. Workshop*, La Jolla, CA, USA, Jan/Feb. 2007, pp. 393–400.

[13] M. A. El-Azizy, R. H. Gohary, and T. N. Davidson, "A BICM-IDD scheme for non-coherent MIMO communication," *IEEE Trans. Wireless Commun.*, vol. 8, no. 2, pp. 541–546, Feb. 2009.

[14] G. W. K. Colman, R. H. Gohary, M. A. El-Azizy, T. N. Willink, and T. Davidson, "Quasi-Gray labelling for Grassmannian constellations," *IEEE Trans. Wireless Commun.*, vol. 10, no. 2, pp. 626–636, Feb. 2011.

[15] U. Wachsmann, R. F. H. Fischer, and J. B. Huber, "Multilevel codes: Theoretical concepts and practical design rules," *IEEE Trans. Inf. Theory*, vol. 45, no. 5, pp. 1361–1391, Jul. 1999.

[16] J.-C. Belfiore, G. Rekaya, and E. Viterbo, "Golden code: A 2×2 full-rate space-time code with non-vanishing determinants," *IEEE Trans. Inf. Theory*, vol. 51, no. 4, pp. 1432–1436, Apr. 2005.

[17] M. Seidl, A. Schenk, C. Stierstorfer, and J. B. Huber, "Multilevel polar-coded modulation," in *Proc. IEEE Int. Symp. Inf. Theory*, Istanbul, Turkey, Jul. 2013, pp. 1302–1306.

[18] G. Ungerboeck, "Trellis-coded modulation with redundant signal sets Part I: Introduction," *IEEE Commun. Mag.*, vol. 25, no. 2, pp. 5–11, Feb. 1987.

[19] G. D. Forney, "Coset codes. I. Introduction and geometrical classification," *IEEE Trans. Inf. Theory*, vol. IT-34, no. 5, pp. 1123–1151, Sep. 1988.

[20] H. Vangala, E. Viterbo, and Y. Hong. (Jan. 2015). "A comparative study of polar code constructions for the AWGN channel." [Online]. Available: https://arxiv.org/abs/1501.02473

[21] A. Fog. (Jan. 2016). Instruction Tables, Technical University of Denmark. Accessed: Jan. 14, 2016. [Online]. Available: http://www.agner.org/optimize/instruction_tables.pdf
[22] L. Hebbes, R. R. Malyan, and P. Bali, "Computational complexities and the relative performance of turbo codes," in *Proc. EUROCON*, vol. 1. Bratislava, Slovakia, Jul. 2001, pp. 37–40.

PHILIP R. BALOGUN received the B.Eng. degree (Hons.) in communications engineering, the M.A.Sc degree in electrical and computer engineering from Carleton University, Ottawa, Canada, in 2013 and 2016. His master's research produced an IEEE conference paper and a U.S. patent. He is currently a Software Engineer with Ciena Corporation, Kanata, Canada.

IAN D. MARSLAND (M'99) received the B.Sc. Eng. degree (Hons.) in mathematics and engineering from Queens University, Kingston in 1987, the M.A.Sc. and the Ph.D. degrees in electrical engineering from the University of British Columbia in 1994 and 1999, respectively. From 1987 to 1990, he was with Myrias Research Corp., Edmonton, and CDP Communications Inc., Toronto, where he was a Software Engineer. He is currently an Associate Professor with the Department of Systems and Computer Engineering, Carleton University. His research interests fall in the area of wireless digital communication.

RAMY H. GOHARY (S'02–M'06–SM'13) received the B.Sc. degree (Hons.) from Assiut University, Egypt, in 1996, the M.Sc. degree from Cairo University, Egypt, in 2000, and the Ph.D. degree from McMaster University, Ontario, Canada, in 2006, all in electronics and communications engineering. He is currently an Assistant Professor with the Systems and Computer Engineering Department, Carleton University. He is a registered Limited Engineering Licensee in the province of Ontario.

Dr. Gohary is the co-inventor of five U.S. patents, and the author of over forty well-cited IEEE journal papers. He is also the referee for over ten scientific IEEE journals, and a member of the technical program committees of seven international IEEE conferences. His research interests include analysis and design of MIMO and cooperative wireless communication systems, applications of optimization and geometry in signal processing and communications, information theoretic aspects of multiuser communication systems, and applications of iterative detection and decoding techniques in multiple antenna, and multiuser systems.

HALIM YANIKOMEROGLU (F'17) received the B.Sc. degree in electrical and electronics engineering from the Middle East Technical University, Ankara, Turkey, and the M.A.Sc. and Ph.D. degrees in electrical and computer engineering from the University of Toronto, Canada. Since 1998, he has been with the Department of Systems and Computer Engineering, Carleton University, Ottawa, Canada, where he is currently a Full Professor. His research interests cover many aspects of wireless networks and technologies. In recent years, his research has been funded by Huawei, Telus, Allen Vanguard, Blackberry, Samsung, Industry Canada, Communications Research Centre of Canada, DragonWave, Mapsted, and Nortel. This collaborative research resulted in OVER 30 patents.

Dr. Yanikomeroglu is a Distinguished Lecturer for the IEEE Communications Society and a Distinguished Speaker of the IEEE Vehicular Technology Society in 5G wireless technologies. He has been involved in the organization of the IEEE Wireless Communications and Networking Conference from its inception in 1998 in various capacities including serving as a Steering Committee member, Executive Committee member and the Technical Program Chair or Co-Chair of WCNC 2004 (Atlanta), WCNC 2008 (Las Vegas), and WCNC 2014 (Istanbul). He was the General Co-Chair of the IEEE 72nd Vehicular Technology Conference (VTC 2010-Fall) held in Ottawa, and is currently serving as the General Chair of the IEEE 86th Vehicular Technology Conference (VTC 2017-Fall) to be held in Toronto. He has served in the editorial boards of the several IEEE journals. He was the Chair of the IEEE's Technical Committee on Wireless Technical Committee.