Pairwise Optimization of Modulation Constellations for Non-Uniform Sources

Brendan Moore, Glen Takahara and Fady Alajaji [*]

The design of two-dimensional signal constellations for the transmission of binary non-uniform memoryless sources over additive white Gaussian noise channels is investigated. The main application of this problem is the implementation of improved constellations where transmitted data is highly non-uniform. A simple algorithm, which optimizes a constellation by re-arranging its points in a pairwise fashion (i.e., two points are modified at a time, with all other points remaining fixed), is presented. In general, the optimized constellations depend on both the source statistics and the signal-to-noise ratio (SNR) in the channel. We show that constellations designed with source statistics considered can yield symbol error rate (SER) performance that is substantially better than rectangular quadrature amplitude modulation signal sets used with either Gray mapping or more recently developed maps. SER gains as high as 5 dB in E_b/N_0 SNR are obtained for highly non-uniform sources. Symbol mappings are also developed for the new constellations using a similar pairwise optimization method whereby we assign and compare a weighted score for each pair. These maps, when compared to the mappings used in conjunction with the standard rectangular QAM constellation, again achieve considerable performance gains in terms of bit error rate (BER). Gains as high as $4\ dB$ were achieved over rectangular QAM with Gray mapping, or more than $1\ dB$ better than previously improved mappings. Finally, the uncoded pairwise optimized system is compared to a standard tandem (separate) source and channel coding system. Although neither system is universally better, the uncoded system with optimized constellations outperforms the tandem coding system for low-to-mid SNRs. Performance/complexity trade-offs between the two systems are also discussed.

I Introduction

For uniformly distributed sources, rectangular quadrature amplitude modulation (QAM) using Gray mapping is known to perform well, and is shown as optimal in terms of bit error rate (BER) for high enough signal-to-noise ratios (SNR) [1]. As noted in [13], however, there are many real-world examples of data sources which are highly non-uniform, such as text (email and instant/short messages), medical images and encoded voice data [2]. Compression will often have residual redundancy in the output due to non-ideal coding methods [3]. Rather than using traditional separate source and channel coding (which may be sensitive to noise-related errors in decoding if optimal variable-length source coding is used, as we later illustrate in Section VI), we can choose instead to directly exploit the non-uniformity of the source via the modulation scheme, while gaining noise-resiliency in many cases and significantly reducing system complexity and delay [3]. Such an approach, which can be characterized as a *joint* source-channel coding approach, is quite attractive for complexity-constrained and delay-sensitive applications such as wireless sensor networks. In these non-uniform situations, the performance of Gray mapped M-ary rectangular QAM is sub-optimal. One simple improvement is to exploit the knowledge of symbol probability by implementing (optimal) maximum *a posteriori* (MAP) decoding (instead of maximum-likelihood decoding) at the receiver. In [13], new $M1$-mappings were developed to improve performance of M-ary rectangular QAM and phase-shift keying constellations. It is also noted in [13] that performance can be improved by translating each mapped constellation so that it has zero mean. Here we consider making further modifications to the constellations in order to achieve lower symbol error rate (SER). In [5], such a constellation design problem was considered for uniform sources under additive white Gaussian noise (AWGN).

In this paper, we propose a pairwise optimization (PO) method for redesigning M-ary constellations to better exploit the non-uniformity in the data, for large values of M, under AWGN and MAP decoding. The method, which is simple to implement, consists of iteratively improving the performance of a constellation by re-arranging its points two at a time, while keeping the other points fixed. We verify our work by comparing it to the known optimal constellations in [7] for $M = 2$, and in [10] for $M = 4$, before considering larger constellations. Other related works on constellation design include [4, 6, 12, 14].

We next introduce a similar PO method for designing good maps for the asymmetric and irregular constellations created by the PO algorithm. Performance in terms of BER is again compared to standard modulation constellations and maps, as well as some previously improved maps. Finally, trade-offs, in terms of both performance and complexity, between our uncoded transmission scheme and a tandem (separate) source-channel coding system are investigated.

The remainder of this paper is organized as follows. After formulating the problem in Section II, we develop our PO process for designing constellations for non-uniform sources in Section III. Before considering larger constellations, we compare our findings to the existing literature for small constellations. From there, we evaluate the performance of our system in terms of SER. In Sections IV and V, we describe an iterative method for designing symbol mappings for the PO constellations, and we compare the results to existing maps for conventional constellations. In Section VI, we compare our uncoded PO system to a coded system which employs a tandem source and channel coding scheme (separately, but simultaneous). Finally, we draw our conclusions in Section VII.

II Problem Statement

We consider a memoryless source $\{X_n\}$ which generates independent binary symbols $\{0, 1\}$ non-uniformly with $p = Pr\{X_n = 0\} > \frac{1}{2}$. We wish to transmit this data over an AWGN channel with noise variance of $\frac{N_0}{2}$ per dimension. We assume that an M-ary two-dimensional (2-D) modulation scheme is to be used, and that it is desirable to maxi-

[*]Mathematics and Engineering, Department of Mathematics and Statistics, Jeffrey Hall, Queen's University, Kingston, Ontario K7L 3N6, Canada; email: {brendan,takahara,fady}@mast.queensu.ca.
This work was supported in part by funding provided by the R. Samuel McLaughlin Fellowship and Herman K. Walter Award from Queen's University and NSERC of Canada.
Parts of this work were presented at the IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, Victoria, BC, August 2009.

mize data throughput per transmission while achieving the lowest possible SER. For convenience, we assume M to be a power of two. Binary symbols are grouped into sequences of $\log_2 M$ bits, forming a new symbol sequence $\{Y_n\}$ having M distinct values $\{s_1, s_2, ..., s_M\}$ with probabilities $\{p_1, p_2, ..., p_M\}$. The probabilities are defined by the number of zeros in the bit sequence. If sequence s_i has n_i zeros, then $p_i = p^{n_i}(1-p)^{\log_2 M - n_i}$. (In the constellation diagrams that come later, we refer to equiprobable symbols by the number of zeros, n, they have in their corresponding binary sequence.) Each channel symbol is then mapped to a signal point, $\vec{s_i}$, in some initial M-ary constellation, where $\vec{s_i} = (s_{i,x}, s_{i,y})$. Our objective is then to change the arrangement of the points in that constellation to achieve the lowest SER possible at a given SNR E_b/N_0, where E_b is the average energy per bit.

The search space to be considered is continuous and consists of all collections of points $\{\vec{s_1}, \vec{s_2}, ..., \vec{s_M}\}$ satisfying

(i) a zero mean constraint: $\sum_{i=1}^{M} p_i \vec{s_i} = 0$; and

(ii) an average power constraint: $\sum_{i=1}^{M} p_i \|\vec{s_i}\|^2 = E$,

where the average energy per symbol, E, is given. Note that E and E_b are related by $E_b = \frac{E}{\log_2 M}$. Our objective function is the SER. For $M = 2$, the optimal constellation was found analytically in [7], but as the constellation size grows, this quickly becomes difficult. In [10], the authors design optimal constellations for $M = 4$ by numerically evaluating tight error bounds developed in [8]. Our goal is to design signal point arrangements that are near-optimal for larger constellation sizes, such as $M = 16, 64, 256$, under MAP decoding.

III Pairwise Optimization of M-ary Constellations

III.A PO Design

In this section, we consider a new method for developing improved signal constellations for 2-D transmission. While the search space is continuous, the zero mean and average power constraints may be used to reduce the search complexity. The zero mean constraint is a necessary property of any optimal (in terms of minimal SER) constellation with constrained average energy, since SER performance under MAP decoding is not affected by translation or rotation of the constellation; it is only affected by changing the relative distances between points. It is of note that for non-uniform sources, rectangular (symmetric) constellations such as 16-, 64- and 256-QAM are not zero mean. It is trivial to improve such constellations slightly by translating them to be zero mean, and scaling them up to their original average energy (which will increase the separation between all points).

For a given initial constellation, it is not possible to adjust the position of a single point while adhering to the above two constraints. Taking any pair of points, however, allows us to move those points around while still adhering to the constraints. If $\vec{s_1}$ and $\vec{s_2}$ are the selected points, then the zero mean constraint implies that

$$p_1\vec{s_1} + p_2\vec{s_2} = -\sum_{i=3}^{M} p_i \vec{s_i}$$

so, if we let $\vec{b} = \sum_{i=3}^{M} p_i \vec{s_i}$, then

$$\vec{s_1} = \frac{1}{p_1}(-\vec{b} - p_2\vec{s_2})$$

or

$$\vec{s_1} = \vec{a} - c\vec{s_2}$$

where $\vec{a} = -\frac{\vec{b}}{p_1}$ and $c = \frac{p_2}{p_1}$. Thus

$$s_{1,x} = a_x - c \cdot s_{2,x} \text{ and } s_{1,y} = a_y - c \cdot s_{2,y}. \quad (1)$$

The average energy constraint implies the following:

$$p_1\|\vec{s_1}\|^2 + p_2\|\vec{s_2}\|^2 = E - \sum_{i=3}^{M} p_i \|\vec{s_i}\|^2. \quad (2)$$

Letting the constant $d = \sum_{i=3}^{M} p_i \|\vec{s_i}\|^2$ and substituting (1) in (2) yields

$$\begin{aligned} p_1\left((a_x - c \cdot s_{2,x})^2 + (a_y - c \cdot s_{2,y})^2\right) & \\ + p_2(s_{2,x}^2 + s_{2,y}^2) = E - d. & \quad (3) \end{aligned}$$

Expanding and completing the square gives us

$$\left(s_{2,x} - \frac{p_1 a_x}{p_1 + p_2}\right)^2 + \left(s_{2,y} - \frac{p_1 a_y}{p_1 + p_2}\right)^2 = r^2 \quad (4)$$

where $r^2 = \frac{p_1(E-d)}{p_2(p_1+p_2)} - \frac{p_1^3}{p_2(p_1+p_2)^2}\left(a_x^2 + a_y^2\right)$. Under the constraints, Eqn. (4) gives us a circle, centered at $\left(\frac{p_1 a_x}{p_1+p_2}, \frac{p_1 a_y}{p_1+p_2}\right)$ with radius r, on which $\vec{s_2}$ may travel, and the relationship given by Eqn. (1) defines a corresponding circle for $\vec{s_1}$. With (4), for each pair of signals $(\vec{s_1}, \vec{s_2})$, the problem of searching over four variables $(s_{1,x}, s_{1,y}, s_{2,x}, s_{2,y})$ is effectively reduced to searching over a single variable, θ, which is the angle parametrizing this circle for $\vec{s_2}$, measured counterclockwise relative to the positive x-axis for the center of the circle. For a given value of θ, $\vec{s_2}$ is defined, and $\vec{s_1}$ has a corresponding position. It is over this parameter θ that each pair of points can be optimized for performance.

With regards to the performance for a potential constellation, we consider the union upper bound[1] on the SER P_s, which is fairly tight for medium to high SNRs:

$$\begin{aligned} P_s &= \sum_{u=1}^{M} P(\epsilon|\vec{s_u})P(\vec{s_u}) \\ &= \sum_{u=1}^{M} P\left(\bigcup_{i\neq u} \epsilon_{iu}\right) P(\vec{s_u}) \\ &\leq \sum_{u=1}^{M}\sum_{i\neq u} P(\epsilon_{iu})P(\vec{s_u}) \quad (5) \end{aligned}$$

where

$$P(\epsilon_{iu}) = Q\left(\frac{\|\vec{s_i} - \vec{s_u}\|}{\sqrt{2N_0}} + \frac{\sqrt{2N_0}\ln\frac{P(\vec{s_u})}{P(\vec{s_i})}}{2\|\vec{s_i} - \vec{s_u}\|}\right)$$

and $Q(x) = \frac{1}{\sqrt{2\pi}}\int_x^\infty e^{-y^2/2}dy$ is the Gaussian Q-function. Note that $P(\epsilon_{iu})$ is the probability that $\vec{s_i}$ has a larger MAP decoding metric than $\vec{s_u}$ given that $\vec{s_u}$ was sent [8].

When considering only the pair of points $\vec{s_1}$ and $\vec{s_2}$, we can ignore the terms in Eqn. (5) for $u \neq 1, 2$ and $i \neq 1, 2$ as they will remain constant. The remaining terms we use as an upper bound are

$$\begin{aligned} F_{12} &= \textstyle\sum_{i\neq 1} P(\epsilon_{i1})P(\vec{s_1}) + \sum_{i\neq 2} P(\epsilon_{i2})P(\vec{s_2}) \\ &\quad + \textstyle\sum_{u=3}^{M} P(\vec{s_u})\left(P(\epsilon_{1u}) + P(\epsilon_{2u})\right) \quad (6) \end{aligned}$$

which is the objective function to be minimized for each pair.

III.B Algorithm

The implemented algorithm is as follows:

[1]To keep things simple, we herein employ the union bound which may be inaccurate for low SNRs. However, the tight upper and lower bounds of [8] can also be used to further improve system performance.

1. Configure some initial constellation, ensuring it adheres to the zero mean and average energy constraints.
2. Randomly (uniformly) select a pair of points $(\vec{s_1}, \vec{s_2})$.
3. Calculate the constrained circles from (4) and (1).
4. Find the positions of $(\vec{s_1}, \vec{s_2})$ by minimizing (6).
5. Go back to Step 2 and repeat until the constellation stabilizes.

The initial constellation used in Step 1 contains the source information implicitly through the symbol probabilities. Tests using different initial constellations (rectangular, circular, asymmetric) all yielded similar results. In Step 4, we calculate the circle noted in Eqn. (4) and set angle θ to be 0 relative to the x-axis, and take discrete steps counterclockwise. At each step of θ, F_{12} is calculated using the corresponding $\vec{s_1}$ and $\vec{s_2}$ on their respective circles, and the design SNR (E_b/N_0), which is set as a constant. This is a simple and brute-force approach, but it works well enough for our needs. The complexity of the algorithm can be approximated by the number of times we calculate the Gaussian Q-function. For each pair of points being optimized, we calculate F_{12} for 50 steps of θ, each of which requires $4M$ calls to $Q(\cdot)$ as in (6), or $200M$ calls per pair. We need roughly M^2 pairs before good constellations are achieved, for a total of $200M^3$ calls (each call takes approx. 3 μs on our 3.0 GHz AMD hardware). When executed, our algorithm stabilizes in a matter of seconds for sizes up to $M = 16$, and scales up to three or four hours for $M = 256$. Stabilization, as used in Step 5, means visual inspection of the constellation at this point. When considering the speed of convergence, it is difficult to be precise, since we do not know what the optimal constellation looks like, or even the final PO constellation for larger sizes. In general, the more likely symbols settle quickly, but the large number of unlikely symbols in large constellations tend to continue rearranging (with better performance at each step) for much longer. The PO algorithm must converge on a final constellation (possibly a local minimum) since each iteration can only decrease the union upper bound, and SER in a non-negative quantity. Since we have that $UnionBound(i) \geq 0$ for all i and the PO algorithm is such that $UnionBound(i) \geq UnionBound(i+1)$, we must have the union bound converging to some stable value as the number of iterations goes to infinity.

III.C Numerical Results and Discussion

We consider the memoryless non-uniform binary source with distribution p for transmission over an AWGN channel for $M = 2, 4, 16, 64, 256$ and compare the performance (in terms of SER simulations) under symbol-by-symbol MAP decoding of our pairwise optimized constellations (which are denoted by PO2, PO4,$\cdots$, PO256) to existing constellations. The exact coordinates of all PO constellations are available in [9, Appendix A]. We use $p = 0.9$ in the simulations, except for the discussion at the end of the Section III.C.2.

III.C.1 Binary and Quaternary Constellations

We begin by comparing to the known optimal constellation presented in [7] for $M = 2$. Our algorithm directly arrives at the same final constellation as the work in [7], as shown in Eqn. (3) with both $\vec{a} = \vec{0}$ and $d = 0$ (since we have no symbols beyond $\vec{s_1}$ and $\vec{s_2}$):

$$p\left((-c \cdot s_{2,x})^2 + (-c \cdot s_{2,y})^2\right) + (1-p)(s_{2,x}^2 + s_{2,y}^2) = E$$

and we choose the point with $s_{2,y} = 0$, so

$$p(-c \cdot s_{2,x})^2 + (1-p)s_{2,x}^2 = E$$

$$s_{2,x} = \sqrt{\frac{E}{pc^2 + (1-p)}} = \sqrt{\frac{E \cdot p}{(1-p)}}$$

$$s_{1,x} = -c \cdot s_{2,x} = -\sqrt{\frac{E \cdot (1-p)}{p}}$$

which is the result obtained in [7]. Note that for $M = 2$, the union bound in (5) yields the exact SER. While the pairwise algorithm is not limited to one dimension, the results are equivalent after rotation. Simulation confirms an exact SER performance match, as expected. There is no consideration of design SNR for $M = 2$, because the constraints alone fix the relative positions of $\vec{s_1}$ and $\vec{s_2}$, and we have no other points with respect to which we may optimize.

We next consider the constellations found in [10] for $M = 4$. When the pairwise optimization stabilizes, the resulting constellation is very similar to those arrived at in [10] for the given design SNR (in this case $SNR = 0\ dB$), up to a rotation and/or reflection. In Fig. 1, it is clear that the pairwise optimized constellation PO4 performs identically to the optimized $M = 4$ constellation of [10]. Both constellations perform considerably better than quaternary phase shift keying (QPSK) for highly non-uniform sources, with nearly 5 dB gain at any SNR. The above results indicate that the algorithm does in fact tend towards optimal constellations, and we may proceed to apply it to larger modulation constellations, where optimal constellations are not known.

III.C.2 16-ary Constellations and Robustness

Before investigating large constellations, we will examine the performance of the 16 point constellation. The PO constellation is shown in Fig. 2. In Fig. 3, the M1 mapping of [13] already improves the performance of (rectangular) 16-QAM by approximately 1 dB. We can also remark that the pairwise optimized constellation PO16 achieves a further improvement of $2dB$ over the M1 mapping, for a total gain of $3dB$ over Gray mapped 16-QAM. This PO16 constellation was designed for a noise level of $SNR = 1\ dB$, but perhaps overall performance across all noise levels is not as good as it could be. To examine this, also included in Fig. 3 is the performance at each true SNR step of a specialized constellation designed specifically for that noise level. It is clear that this specialized configuration does not provide considerable gains over a constellation designed at a single SNR that is carefully selected (in this case, 1 dB) and used for transmission across all noise levels. This shows that a constellation designed using a single appropriately chosen design SNR can provide robust performance *vis-à-vis* changes in the true SNR, and that it is not necessary to have a set of constellations tuned to every channel noise level.

Fig. 4 indicates that gains over Gray-mapped 16-QAM are also achieved by PO16 for smaller values of p. For $p = 0.5$, the gain achieved was negligible, as expected. With $p = 0.6$, the gain is about $0.25\ dB$. For $p = 0.7$ and $p = 0.8$, more significant gains of $0.5\ dB$ and $1.5\ dB$ are achieved, respectively.

At design $SNR = -10\ dB$, Fig. 5 shows that the resulting constellation is now quite different than the PO16 constellation of Fig. 2. The less likely point clustered at the center is effectively coincident to the most likely symbol. As such, it will never be decoded. We also see that there is clustering among the less likely symbols farther from the center. Instead of being more evenly spread out, they are gathered into small groups that are farther apart. As [14] notes, and we are inclined to agree, this allows the errors between groups to be greatly reduced at the expense of more likely errors within a cluster. Since we are dealing with a highly non-uniform source, we have the advantage of being able to err on the safe side with MAP decoding, and will simply decode the most likely symbol of a given cluster.

When designing constellations for higher SNR, the PO algorithm creates constellations similar to the one shown in Fig. 2 for mid to high SNR values. As we move up to quite high SNR, however, the resulting configuration tends to rely less and less on the source distribution, since all points are very likely to be decoded correctly, regardless of placement. This situation essentially turns into maximum likelihood decoding (since the probabilities do not significantly affect the MAP metric for very small noise), and the resulting constellation reflects this by having points placed equidistant from one another. For design at 10 dB, the constellation is shown in Fig. 6; it and is nearly identical to the constellation in Fig. 6(d) of [5].

We close this subsection with some practical considerations that need to be addressed when designing constellations for a range of both p and SNR values. Since we have shown the PO constellations to be robust over a reasonable range of true SNRs using only a single design SNR, we could arguably provide constellations suited to large steps in noise power. Both the transmitter and receiver would need to know the constellations in advance, and be capable of measuring noise power in the channel (e.g., using a pilot tone) and signaling when a constellation switch should happen. This can be accomplished with a very small overhead scheme which transmits the constellation to be used after every several thousand transmissions, using only a few bits each time (i.e., a single transmission) to do so. Similarly, we can implement multiple constellations to address a range of source distributions for reasonable steps in p. We would thus arrive at a set of constellations for each of a small number of SNRs and perhaps five to ten different p-values. This array of solutions would allow the best performance gains possible for the current source and channel characteristics, using only a small transmission over head to signal the switches. This switching scheme has not been implemented in this work, but is a viable option for real-world implementations.

III.C.3 64-ary and 256-ary Constellations

The result of pairwise optimization of a 64 point constellation using design $SNR = 2dB$ is shown in Fig. 7. Again we remark the tendency of more likely points to lay closer to the origin. This keeps the average energy low, allowing less likely points to sit farther away, thus creating more distance between points overall. In Fig. 8, we compare the performance of this constellation to 64-QAM with the M1 and Gray mappings. The $M1$ 64-QAM mapping developed in [13] already outperforms Gray mapped 64-QAM by approximately 3.5 dB for any given SER. The pairwise optimized constellation we develop here, PO64, outperforms 64-QAM with $M1$ mapping by another 1.5 dB at a given SER, for a total improvement of about 5 dB over 64-QAM with Gray mapping. It is interesting to note that for medium and high SNRs (above 4 dB), the PO64 constellation achieves better SER than the BER of binary phase shift keying (BPSK). It is likely that the BER of PO16 will be lower than that of BPSK for sufficiently high values of p. The performance of the pairwise optimized constellation for $M = 256$ (PO256 in Fig. 9) is better than 64-QAM with Gray map by approximately 2 dB for any SER. Note that PO256 has both a higher data rate and a lower SER than Gray mapped rectangular 64-QAM at all SNRs, thus improving both system performance and throughput.

IV Designing Maps for PO Constellations

In the previous section, we developed a method for designing improved modulation constellations for non-uniform sources, based on the SER performance of those constellations. But, in order to be able to use these constellations in any real systems involving binary streams of data, we must have a direct mapping for each possible $log_2 M$-bit symbol. In this section we wish to design maps for the PO constellation we have designed in order to assign a specific bit pattern to each point in the constellation.

IV.A Initialization and Probability Constraint

By the nature of the PO algorithm, the mapping can be initiated and modified under a fairly strong (and helpful) constraint. For a given symbol, we know from its source probability that we can immediately reduce the possible mappings for that symbol to a subset of the points found in the PO constellation – those corresponding to the same source probability.

For each constellation treated, the map is initialized arbitrarily, but such that it conforms to the probability constraint. We define a *layer* as a set of binary symbols taken together with a set of constellation points which are equiprobable. For our binary non-uniform source with source distribution $Pr\{X_n = 0\} = p$, and an M-ary constellation, the symbols (bit patterns) in layer l will each have l zeros in their binary sequences. Layer l will also have exactly $\binom{m}{l}$ symbols, each with probability $p^l(1-p)^{m-l}$, where $m = log_2 M$ is the number of bits in each symbol. During the initialization of the map optimization procedure (described later), the symbols and points within each layer are assigned randomly.

IV.B Objectives

Our guiding objective is to minimize the BER of the constellation and mapping pair. We aim to achieve a BER much lower than the SER by trying to minimize the number of bit errors that occur, even in the presence of noise which causes a symbol error. We can achieve this by minimizing the Hamming distance of each symbol to its neighbours. For a uniform source, using rectangular QAM, this can be achieved using Gray mapping (e.g., cf. Fig. [13, Fig. 8]). This mapping is arranged such that the Hamming distance of any symbol to any of its neighbours is always one. To exploit the source statistic of the non-uniform source, improved mappings were developed in [13] without modifying the geometry of the underlying constellation. The challenge we face when dealing with the non-uniform source and our PO constellations is that we have neither equiprobable nor equidistant (in terms of the Euclidean distance) neighbours. We still wish to create a "Gray-like" mapping, but it is not so simple as the rectangular QAM case. If not all nearby points are equidistant (Euclidean distance-wise), how do we choose which points we will consider as neighbours? Since, however we select them, those neighbours will not necessarily be equiprobable, we must consider how to measure the Hamming distance to the entire neighbourhood.

IV.B.1 Defining the Neighbourhood and Weighted Hamming Score

We must first decide and define what symbols we will consider as neighbours. We initially considered setting the neighbourhood of a point $\vec{s_u}$ to be all points which lay inside a circle of radius r (*i.e.*, all points $\vec{s_i}$ such that $\|\vec{s_u} - \vec{s_i}\| \leq r$). But how do we choose an appropriate r? Should it change depending on the probability of the symbol being considered? Without a way to determine what r should be, we moved on to the idea of selecting some k points which lay closest (in terms of Euclidean distance) to $\vec{s_u}$ (i.e., the k nearest neighbours). This method provided a natural solution to the flexible radius problem, as we would always include the closest k points, despite those points laying farther away for the less likely symbols towards the outside of the constellation. Looking at the PO constellations we had obtained so far, we selected $k = \sqrt{M}$ as the neighbourhood size. This value provided a good balance between limiting the neighbourhood to those points which would most likely be decoded in error, but which also would include enough points to ensure we indeed had an appropriate neighbourhood "around" the symbol in question.

Now we must define the *Weighted Hamming Score* we will use when considering the suitability of a given symbol for its neighbourhood. To do this, we disregard the Euclidean distance between the symbol and each of its neighbours, and instead consider the Hamming distance and probability of each other point. For symbol $\vec{s_u}$ and its k nearest neighbours (indexed by $\{n_1, ..., n_k\}$), the Weighted Hamming Score,

$WHS(\cdot)$, is then defined as:

$$WHS(\vec{s_u}) = \sum_{i=1}^{k} p_{n_i} d(b_{n_i}, b_u) \tag{7}$$

where p_{n_i} is the probability of $\vec{s_{n_i}}$, $d(\cdot,\cdot)$ is Hamming distance and b_{n_i} is the binary sequence (symbol) currently assigned to $\vec{s_{n_i}}$.

V Map Improvement Algorithm

The Map Improvement algorithm is implemented as follows:

1. Configure some initial mapping (as described above).
2. Loop through set of all layers several times.
   (a) Loop through each layer individually, proceeding outwards.
      i. For current layer, generate many pairs of symbols.
         A. For each pair of symbols, determine both neighbourhoods.
         B. Compare total Weighted Hamming Score with and without switching the symbol assignment, using (8).
         C. If switching the symbol assignment decreases WHS, then switch the mapping of binary sequences to selected points.

As described above, the initial mapping of Step 1 is arbitrary aside from conforming to the probability requirements. For Steps 2 through to 2(a)i, we adjusted the number of times to repeat each loop to achieve what appeared to be the best results. The values used in our code were as follows:
We looped through the set of all layers $4m$ times for Step 2, where $m = log_2 M$ is the number of bits in each sequence (and is also the number of layers). When stepping through each individual layer in Step 2a, we start with the center and proceed outwards. We only “loop” each layer once per overall loop, since repeating a single layer immediately would be equivalent to simply generating more pairs. Finally, for Step 2(a)i, we achieved favourable results when considering $2L^2$ pairs, where $L = \binom{m}{l}$ is the number of symbols in layer l (members of which have l zeros in their binary sequences).

When checking each pair of symbols in Step 2(a)iB to determine whether they should be switched, we calculate the sum of the WHD for each symbol in both neighbourhoods ($(n_{u,1}, ..., n_{u,k})$ is the neighbourhood of $\vec{s_u}$, and $(n_{v,1}, ..., n_{v,k})$ is the neighbourhood of $\vec{s_v}$). We want to know if

$$\sum_{i=1}^{k} \left(p_{n_{u,i}} d(b_{n_{u,i}}, b_u) + p_{n_{v,i}} d(b_{n_{v,i}}, b_v)\right)$$
$$> \sum_{i=1}^{k} \left(p_{n_{u,i}} d(b_{n_{u,i}}, b_v) + p_{n_{v,i}} d(b_{n_{v,i}}, b_u)\right) \tag{8}$$

and, if so, we switch the mappings b_u and b_v for $\vec{s_u}$ and $\vec{s_v}$.

V.A Results and Performance

The results of this procedure are now considered. For small constellation sizes, the mapping is not particularly important. For $M = 2$, there is no map to be considered at all – the symbols are exactly determined by the constellation. For $M = 4$, the only consideration is the placement of the mapping 01 versus 10. Indeed these are different, but the resulting BER is identical, since each configuration has identical Hamming distance to its neighbours. As such, we immediately move to considering 16-ary constellations and larger.

V.A.1 16-ary Constellations

We present the result of the mapping improvement algorithm in Fig. 10 for the PO constellation we developed in the previous section. We immediately note that the Hamming distance between many close neighbours is 2, where the Gray had put all distances to just 1. By the design of the PO constellations, we cannot achieve distances as small as those of the Gray map. This is because we force symbols of equal probability to be near one another. For instance, consider the layer $l = 3$ in Fig. 10. All symbols in this layer have exactly three zeros in their binary sequences (there are four of these symbols). Since they must be a neighbour to at least one other point in their own layer, and we do not have any repeated symbols, the Hamming distance of these close neighbours must be 2. However, you will also find that this mapping also has a *maximum* Hamming distance of 2, aside from the distance involving the symbol 1111 to layer $l = 3$ (which cannot be avoided, again by the design of the PO constellation). Despite having greater Hamming distance between some neighbours, the PO constellation with the associated map performs very well.

Inspecting Fig. 11, the performance improvement on the PO constellation with its designed map is clear. There are considerable gains over the standard rectangular QAM constellation with both maps considered. While the optimized $M1$ maps of [13] achieve approximately $1\,dB$ gain of the Gray map, the PO constellation and map makes a further improvement of more than $2dB$ over the $M1$, where the difference is greatest. This best improvement occurs at mid-range SNRs of 2 to $5\,dB$, and is approximately in line with the gains we saw when considering SER performance. It is also interesting to note that the *SER* performance of PO16 is in fact superior to the BER performance of both rectangular 16-QAM maps for mid-high SNRs, so even without the map improvement procedure, the BER of PO16 would be considerably better than the standard constellations.

We also note that the performance difference between the BER and SER of PO16 shrinks as SNR increase to high levels. This is to be expected, as when the noise is very small compared to the signal, we will make very few errors on the most likely symbols, and this is where reducing bit errors has the greatest impact.

V.A.2 64-ary and 256-ary Constellations

We display the resulting map for the PO64 constellation in Figs. 12. The map for the PO256 constellation can be found in [9, Fig. 4.6]. Examining first at Fig. 12, again we see Hamming distances of close neighbours larger than those seen in a Gray map (as we saw for the PO16 constellation). But we also see again that most symbols (especially the group of more likely symbols near the center) are assigned so that they are quite similar to their neighbours. Many of the most likely symbols have a Hamming distance of only 1 or 2 to their neighbours, but we also have the Hamming distance between *any* neighbours being at most 3. For the 256-ary PO constellation and mapping (cf. [9, Fig. 4.6]), we observe the same situation in general. There are some neighbours with greater Hamming distances, but most are kept quite low (2 or 3), especially when compared to the longer sequence length (8 bits) of symbols in this constellation. The BER performance of the 64- and 256-ary PO constellations is shown in Fig. 13. We remark that the 64-QAM M1-map of [13] provides gains of approximately $3\,dB$ over Gray mapped 64-QAM, and even $0.25\,dB$ over BPSK. The PO64 constellation with the improved map further provides more than $1\,dB$ over the BER performance of the M1-mapped QAM. Again, we find that this is approximately in line with the gains observed when examining SER performance. Interestingly, we notice that the PO256 constellation and map has better BER performance than Gray mapped 64-QAM, despite the constellation density and the larger Hamming distances observed in PO256. As was the case with SER performance, PO256 also simultaneously achieves superior BER performance and higher data throughput.

VI Comparison to Tandem Source and Channel Coding

Thus far we have only compared our PO system to similar transmission schemes: mainly BPSK and rectangular QAM. We have been considering only the case of transmitting directly the modulated source symbols, without trying to compress the source, or protect the message with parity.

We will now consider another possible transmission scheme: using in tandem both source and channel coding on the data (separate source and channel coding) to first compress the message, and then protect it during transmission. Using this tandem scheme has a trade-off relative to our uncoded system, and different performance profiles. We will explore these differences in this section.

VI.A Tandem Coding System

We begin by describing the tandem coding scheme to which we will compare the PO system. The goal is to compare to a system that is reasonably representative of real world systems employing such tandem coding schemes.

As in all of the test cases, we generate bits according to the source distribution. For each of T trials, we generate messages in blocks of N bits at a time to be processed. The message will first be compressed (losslessly) using a fourth-order Huffman code. This means we will look at four bit symbols from our non-uniform binary source, and design a Huffman code for them. For $p = 0.9$, the resulting Huffman code has an average code rate of 0.49 bits/source symbol. This code rate is close to the entropy of our binary source, which is 0.46, so we know this code is appropriate. Given this code rate close to 0.5, the compressed message will be approximately $\frac{N}{2}$ bits long.

The channel code we will be using is a convolutional code with constraint length $k = 3$ and rate $r = 1/2$. The generator functions used for the convolutional code are $G_0 = 101$ and $G_1 = 111$, in binary representation form [11, pp 470-477]. This leads the output to follow the state machine described in Fig. 14. The states represent the two previous input bits, $X_{n-1}X_{n-2}$, and the transition labels indicate the current input bit and the two channel coded parity bits which will be transmitted, X_n/g_0g_1. To allow us to know how the transmission begins, we always reset the initial state to 00 at the beginning of each message block, and record the transitions from there forward. Hence for each compressed message bit to be sent, two coded parity bits are sent representing the state change. The channel coded bits are transmitted using BPSK with AWGN, and the receiver collects the observed voltages in the channel.

Once the entire block (approximately N bits) has been received, the observed voltages are passed to a soft Viterbi decoder to recover the transmitted message. The output of the Viterbi algorithm is a best guess of the bit sequence (by minimizing the sequence error probability) which comprises the encoded message (approximately $\frac{N}{2}$ bits). This binary sequence is then passed to the Huffman decoder to be converted back into the original message (N bits long). Note that the overall rate of this tandem coding system is 1 source bit per channel use.

VI.B Performance Comparison

We will now examine the performance of this tandem coding scheme in comparison to our uncoded PO system. To test the tandem coded system, messages of N source bits were generated and passed through the system described in Section VI.A. Tests were conducted for a range of values of N, as the performance of the Viterbi decoder does indeed depend heavily on the length of the blocks received. Other values were tested, but the values we have selected for interesting performance comparison to the PO systems are $N = 12, 100, 200, 800, 5000$. For smaller values of N, more trials (T) were performed to simulate approximately ten to twenty million bits[2] total for each SNR.

Examining the data presented in Fig. 15, we first notice that performance of the tandem system is quite poor at low SNR for all block sizes, but is slightly worse for larger N. For small block length ($N = 12$) we note that the tandem scheme does not outperform PO2 and PO4 even at relatively high SNR, and in fact only beats out PO16 at a mid-high SNR of $5\ dB$. Keep in mind that at that level of noise, PO16 is communicating four times as much data as the tandem scheme. Stepping the block size up to $N = 100$, we start to get performance approaching PO2 and PO4 for mid-high SNRs. In fact, at $3\ dB$, the tandem scheme with $N = 100$ matches the performance of PO4 (which is already slightly better than PO2), but then fails to overtake it. To beat out PO4 completely, we must further increase block size.

Once we move up to $N = 800$, we note that it is possible to beat the performance of PO4. Here the tandem scheme surpasses the BER performance of PO4 at slightly above $3\ dB$, and remains superior from then on. Considering $N = 5000$, we detect the tandem scheme surpassing the performance of PO4 just beyond $4\ dB$, and subsequently surpassing the tandem scheme for $N = 800$ at approximately $4.5\ dB$.

These results indicate that PO4 is clearly superior in performance for low and mid-range SNRs. At mid-high SNRs and beyond, it is possible for the tandem scheme to surpass the performance of PO4 for sufficiently long block sizes. It should be noted, however, that the gains over PO4 are less than $1\ dB$ at SNR around $5\ dB$, even for a very large block size, and that PO4 is achieving twice as much data throughput for its performance.

There are some additional trade-offs to be considered here. The relative complexity will be discussed in the next section, but there are performance trade-offs to be considered, as well. For instance, the long block length necessary to beat the performance of PO4 entails a considerable amount of decoder delay. The receiver must wait for the entire block to be transmitted before passing the observations to the Viterbi decoder, which then must process the data. The long block length, while resulting in better BER performance overall, is susceptible to long runs of corrupted data in individual messages. That is, when the Viterbi decoder outputs an incorrect bit, the message from that point onward (the tail) is heavily corrupted (high concentration of bit errors) due to desynchronization of the Huffman decoder.

The average corruption run lengths measured during simulation (at the Viterbi decoder output) are shown in Table 1. The average lengths presented are over the number of transmissions where an extended data corruption has occurred (rather than average over all trials). The rate of the appearance of these types of errors is given by the occurrence rate. Thus for $N = 800$, in 0.5% of trials, there were long runs of corrupted data which had an average length of 184 bits. For $N = 5000$, the average corrupted run length was 1285 bits, occurring in 0.2% of trials. While this is a rare occurrences in both cases, it is a considerable portion of the message when it does happen – over 20% of the message (the tail end) is corrupted on average. While the PO constellation has a higher BER at high SNRs, it does not suffer from these long runs of corrupted data, and instead has its bit errors spread more evenly throughout the messages. Whether this concentration of errors is important depends entirely on the specific application, and the tolerance or sensitivity towards different types of failures.

The careful reader will note that manipulation of the average corruption run length and occurrence rate in Table 1 does not yield the same average bit error rate as the data presented in Fig. 15. This is not a data discrepancy, but rather one of presentation. For Table 1, we have a view *inside the machine*, and are able to check when the Viterbi

[2]For $N = 5000$, the total number of bits simulated was considerably higher (approx. 100M rather than 20M), since the results produced were inconsistent using only a few thousand trials.

decoder has made a bit error in the sequence it returns, and consider the message bits decoded from that point onward to be corrupted (high probability of error). The actual result of such a bit error has two possible manifestations. The less likely case is that the error causes the Huffman decoder to decode an incorrect codeword *of the same length* as the intended codeword. It will then continue to decode the rest of the message correctly as if there were no error. The more likely scenario is that the Viterbi bit error will cause the Huffman decoder to decode an incorrect codeword *of a different length* than the intended codeword. This causes the Huffman decoder to become out of synchronization with the true data, but continues to decode incorrect codewords until near the end of the message, when it may find a non-existent codeword. It is likely, given the source distribution under consideration, that the corrupted output will still have many of the bits of the original message correct (the zeros). Since this is essentially the decoder *getting lucky* with its mistakes, we have considered the entire tail of the message to be corrupted when such a Viterbi bit error occurs, as the decoded message is unreliable and contains a higher-than-normal concentration of bit errors, but the measurement of BER for Fig. 15 considers only the individual bits (the lucky bits are counted as correct data).

VI.C Complexity Issues

The most fundamental difference between the two systems is one of hardware complexity versus software complexity. The PO system requires more complicated hardware design (in the transmitter), whereas the tandem source and channel coding needs a processor capable of the calculations required to run the system.

The PO constellation might increase the hardware requirements in terms of needing strongly linear power amplifiers, since the constellation points can be quite far apart (in a constellation designed for a highly non-uniform source) and can thus result in a high peak-to-average power ratio for signals sent over successive modulation intervals. The tandem scheme is simplified in hardware as it employs conventional BPSK.

The tandem scheme is relatively quite complicated in software implementation. The PO system requires only simple instructions to translate the source data to a constellation point, and a fairly straightforward system for MAP demodulation at the receiver. The tandem scheme requires considerable processing, particularly the receiver for performing the Viterbi calculations to detect the encoded message sequence. This processing could be quite taxing for a small wireless node running off limited battery power, such and those found in a wireless sensor network.

VII Conclusions

It is clear that the pairwise optimized constellations offer significant gains over traditional (rectangular QAM) modulation constellations for highly non-uniform sources. This is especially true for high rate constellations where a great deal of energy is "wasted" by placing likely symbols far from the origin. We recognize that asymmetric non-rectangular constellations introduce additional complexity both in the hardware of the transmitter (for modulation) and in the calculations required for demodulation. Smaller improvements can be easily obtained by re-centering the traditional rectangular constellations to be zero mean, and scaling them up to their original average energy.

The gains achieved when considering SER were not only sustained, but improved, as good maps were designed for the PO constellation and BER performance was measured. Despite higher Hamming distances between neighbouring symbols, the BER performance of the PO constellation was significantly better than standard Gray-mapped rectangular QAM.

When comparing the uncoded PO system to a tandem source and channel coded system, neither was universally superior. PO was general better at low SNR, while the tandem scheme (for large enough block size) was superior at high SNRs. Where the BER performance of the two systems matched, it is notable that the PO constellations transmitted at a higher rate. Each system had its advantages and disadvantages, and making a choice between them for implementation would depend entirely on the application.

References

[1] E. Agrell, E. Strom, and T. Ottossom. Gray Coding for Multilevel Constellations in Gaussian Noise. *IEEE Transactions on Information Theory*, IT-53(1):224–235, January 2007.

[2] F. Alajaji, N. Phamdo, and T. Fuja. Channel Codes that Exploit the Residual Redundancy in CELP-Encoded Speech. *IEEE Transactions on Speech Audio Processing*, 4:325–336, Sept. 1996.

[3] F. Behnamfar, F. Alajaji, and T. Linder. MAP Decoding for Multi-Antenna Systems with Non-Uniform Sources: Exact Pairwise Error Probability and Applications. *IEEE Transactions on Communications*, 57(1):242–254, January 2009.

[4] S. Emami and S. L. Miller. Nonsymmetric Sources and Optimum Signal Selection. *IEEE Transactions on Communications*, 44(4):440–447, April 1996.

[5] G. J. Foschini, R. D. Gitlin, and S. B. Weinstein. Optimization of Two-Dimensional Signal Constellations in the Presence of Gaussian Noise. *IEEE Transactions on Communications*, COM-22(1):28–38, January 1974.

[6] J. Huang, S. Meyn, and M. Médard. Error Exponents for Channel Coding With Applications to Signal Constellation Design. *IEEE Journal on Selected Areas of Communications*, 24(8):1647–1661, August 2006.

[7] I. Korn, J. P. Fonseka, and S. Xing. Optimal Binary Communication With Nonequal Probabilities. *IEEE Transactions on Communications*, 51(9):1435–1438, September 2003.

[8] H. Kuai, F. Alajaji, and G. Takahara. Tight Error Bounds for Nonuniform Signaling over AWGN Channels. *IEEE Transactions on Information Theory*, 46(7):2712–2718, November 2000.

[9] B. F. D. Moore. Pairwise Optimization of Modulation Constellations for Non-Uniform Sources. Master's thesis, Queen's University, Kingston, Ontario, Canada, September 2009.

[10] H. Nguyen and T. Nechiporenko. Quarternary Signal Sets for Digital Communications with Nonuniform Sources. *IEEE CCECE/CCGEI*, pages 2085–2088, May 2005.

[11] J. G. Proakis. *Digital Communications*. McGraw-Hill, fourth edition, 2000.

[12] Y. Sun. Stochastic Iterative Algorithms for Signal Set Design for Gaussian Channels and Optimality of the L2 Signal Set. *IEEE Transactions on Information Theory*, 43(5):1574–1587, September 1997.

[13] G. Takahara, F. Alajaji, N. C. Beaulieu, and H. Kuai. Constellation Mappings for Two-Dimensional Signaling of Nonuniform Sources. *IEEE Transactions on Communications*, 51(3):400–408, March 2003.

[14] N. Wei and Y. Wan. Optimal Constellation for General Rectangular PAM/QAM with Arbitrary Code Mapping. In *Proceedings of IEEE International Conference on Communications*, pages 2749–2754, Glasgow, Scotland, June 24-28 2007.

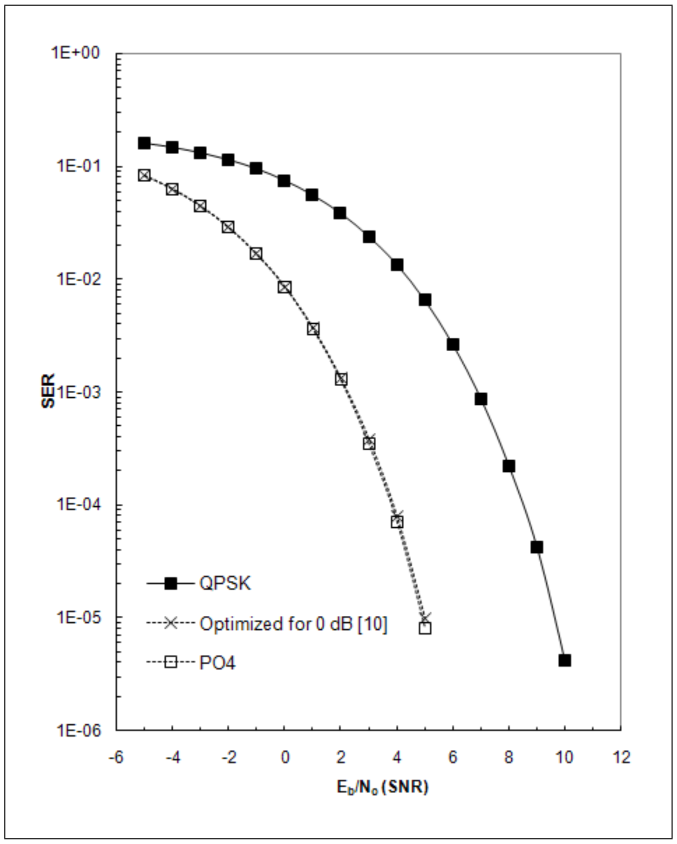

Figure 1: Performance of $M = 4$ constellations for $p = 0.9$. Optimized from [10] and PO4 are both designed for $SNR = 0\ dB$.

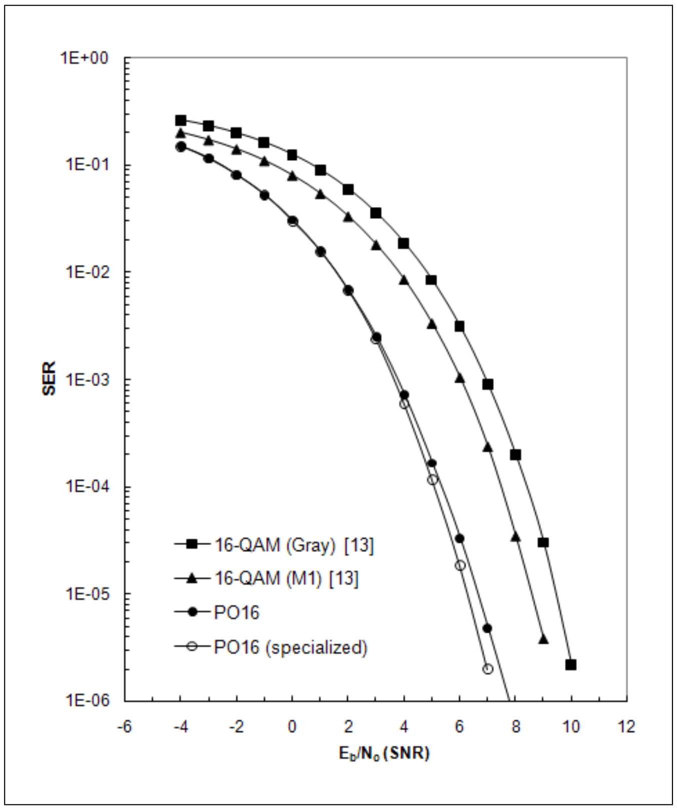

Figure 3: Performance of $M = 16$ constellations for $p = 0.9$ and design $SNR = 1\ dB$. Performance of a specialized constellation (i.e., with design SNR identical to true SNR) also shown.

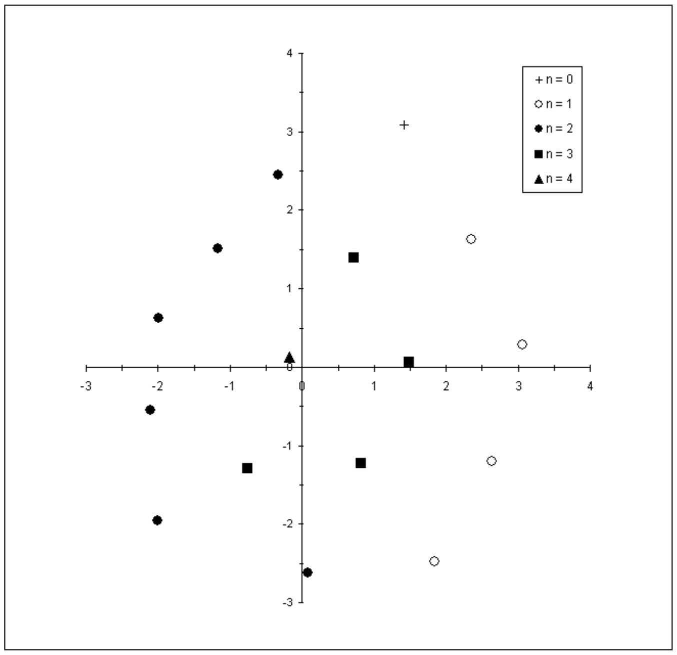

Figure 2: Pairwise optimized constellation for $M = 16$, $p = 0.9$ and design $SNR = 1\ dB$.

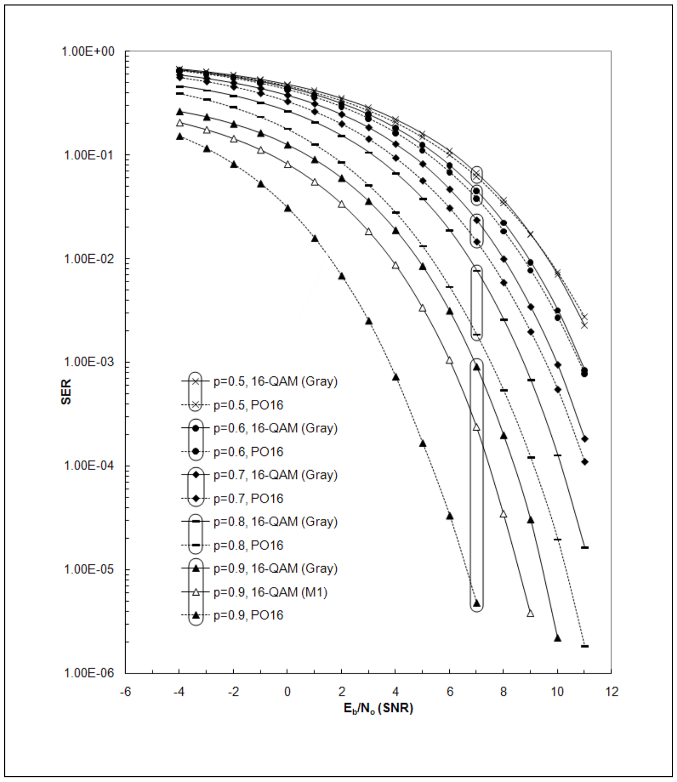

Figure 4: Performance of $M = 16$ constellations for varying values of p and design $SNR = 1\ dB$.

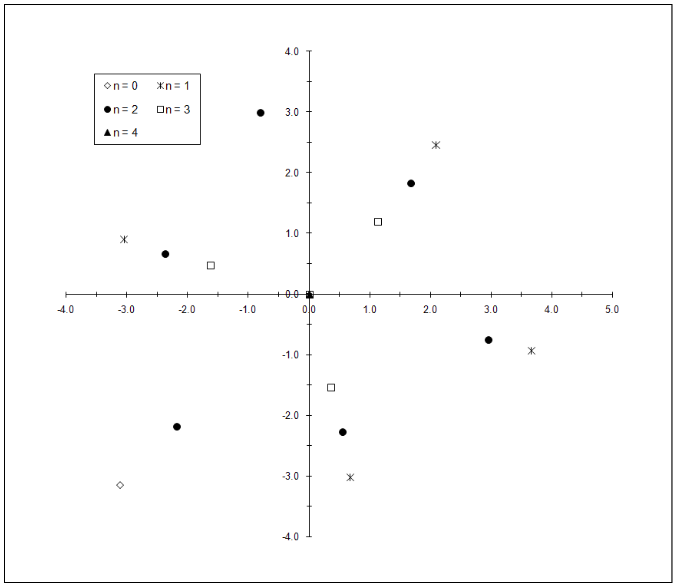

Figure 5: Pairwise optimized constellation for $M = 16$, $p = 0.9$ and design $SNR = -10\ dB$.

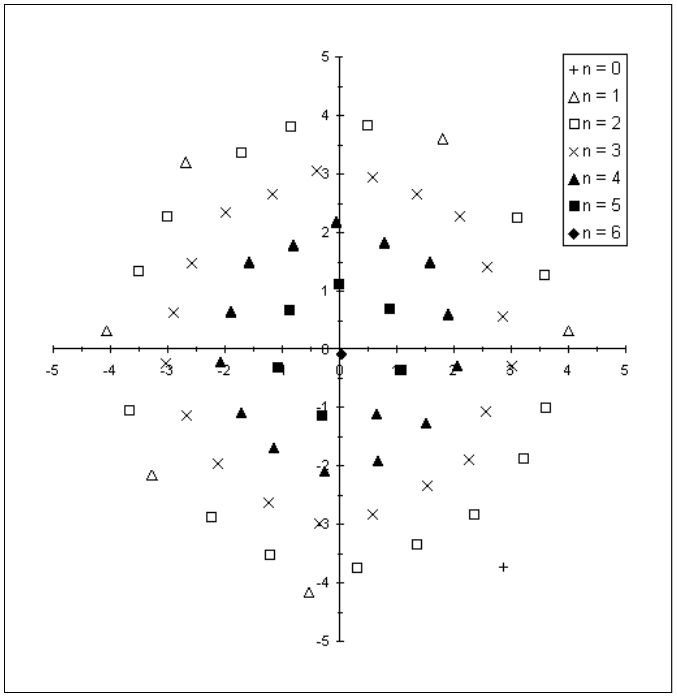

Figure 7: Pairwise optimized constellation for $M = 64$, $p = 0.9$ and design $SNR = 2\ dB$.

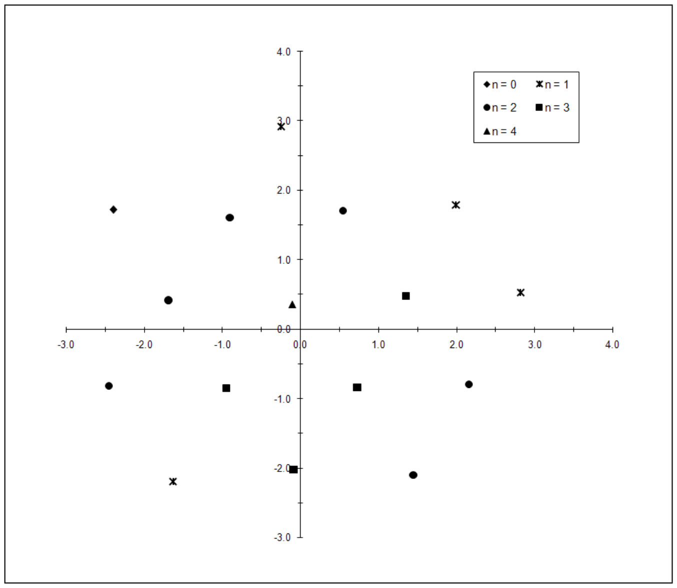

Figure 6: Pairwise optimized constellation for $M = 16$, $p = 0.9$ and design $SNR = 10\ dB$.

Table 1
Tandem Code Errors

Block Size, N	Crossover SNR	ACL (bits)	Freq.
12	$5\ dB$	4	0.02%
100	$3\ dB$	9	0.35%
800	$3\ dB$	184	0.50%
5000	$4\ dB$	1285	0.205%

Table 1: Average corruption length (ACL) of tail in bits and frequency of occurrence (within the Viterbi decoder) for various message block sizes. Results shown only at *Crossover SNR* (the point at which the tandem scheme overall BER performance matches that of PO4). The tandem scheme does not outperform PO4 for block size 12, so this Crossover SNR is where it surpasses PO16.

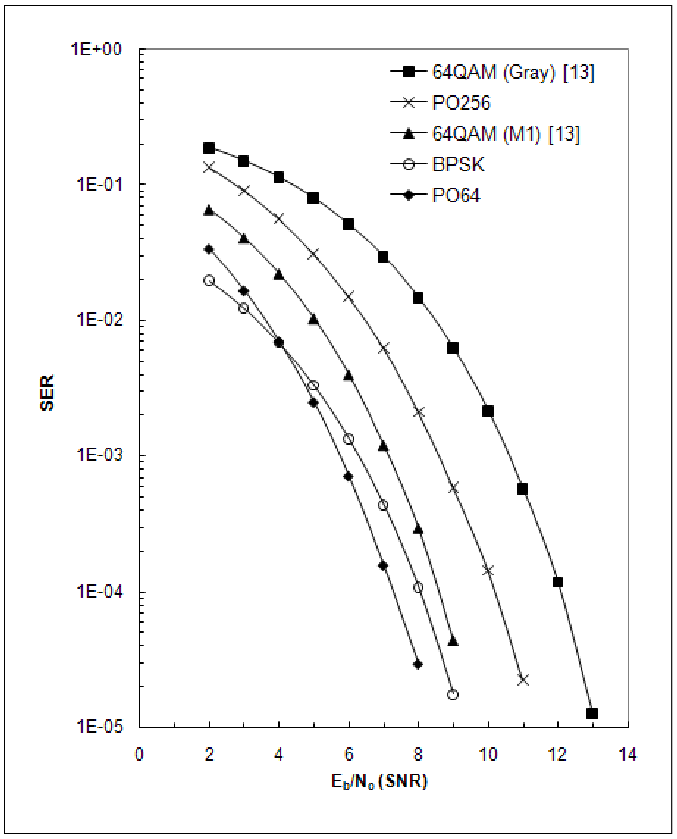

Figure 8: Performance of constellations for $M = 64$, $p = 0.9$ and design $SNR = 2dB$ and the pairwise optimized constellation for $M = 256$ with design $SNR = 4dB$. BPSK also shown as reference point.

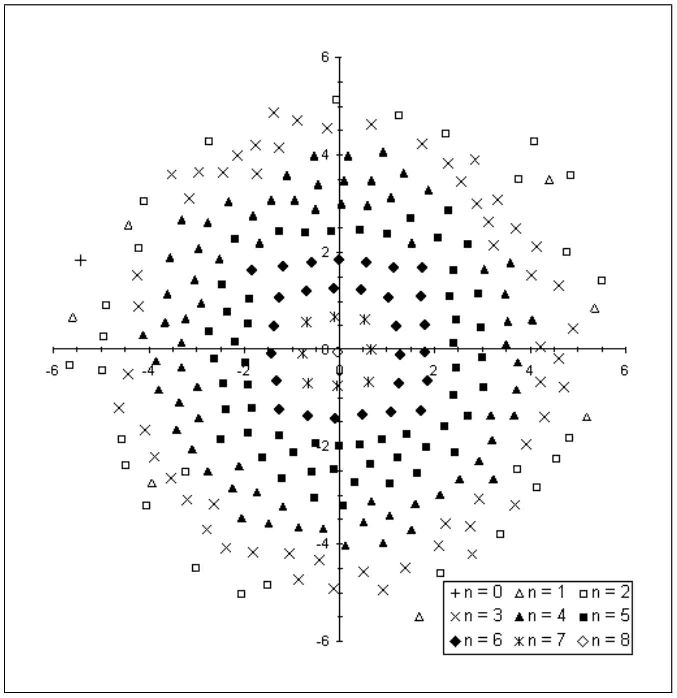

Figure 9: Pairwise optimized constellation for $M = 256$, $p = 0.9$ and design $SNR = 4\, dB$.

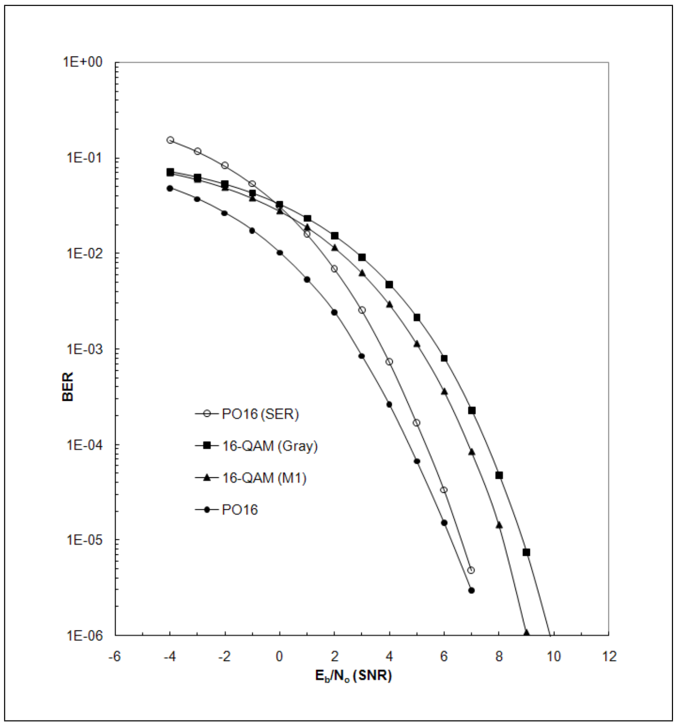

Figure 11: BER Performance of 16-ary constellations. PO constellation simulated with mapping seen in Fig. 10.

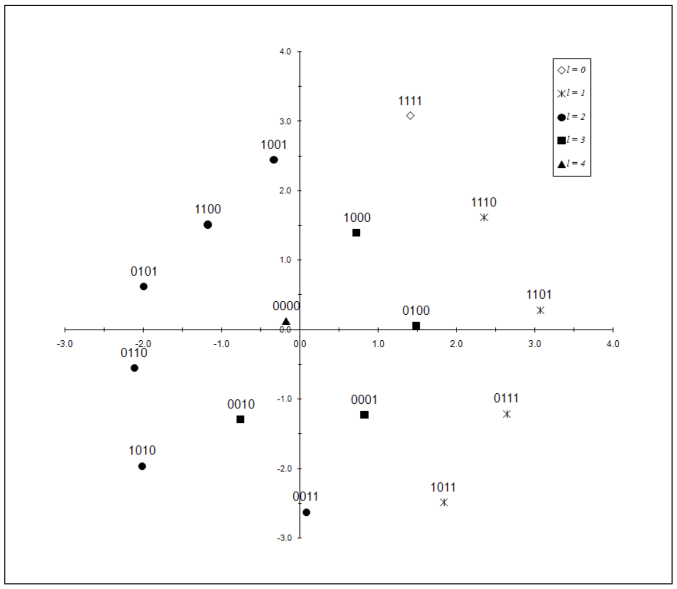

Figure 10: PO constellation for $M = 16$ with improved mapping.

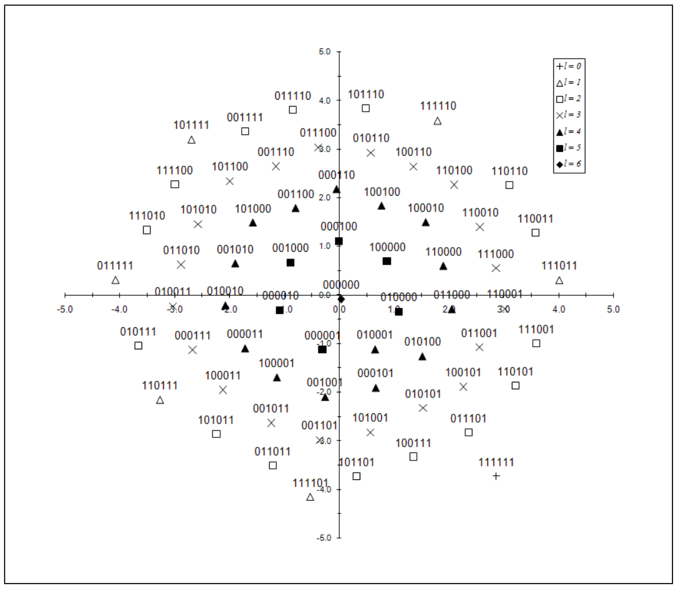

Figure 12: PO constellation for $M = 64$ with improved mapping.

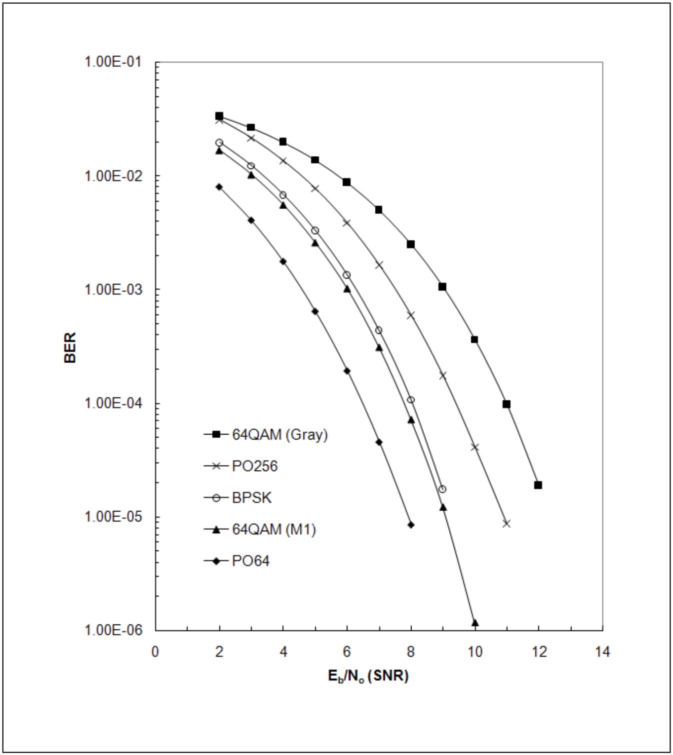

Figure 13: BER Performance of 64-ary constellations and PO256. PO constellations simulated with mappings given in Fig. 12 and [9], Fig.4.6.

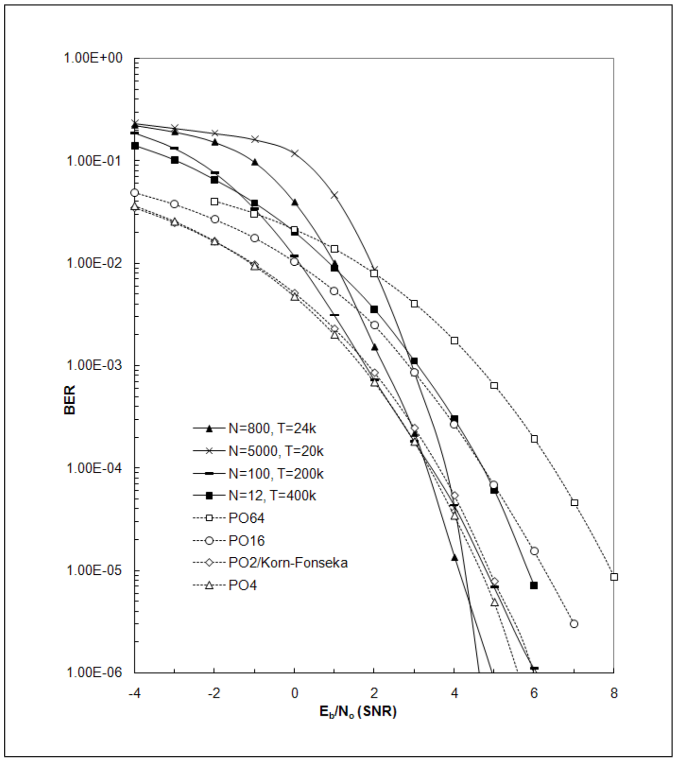

Figure 15: Performance of tandem source and channel coding scheme for various block lengths. Selected PO constellation performance shown for reference.

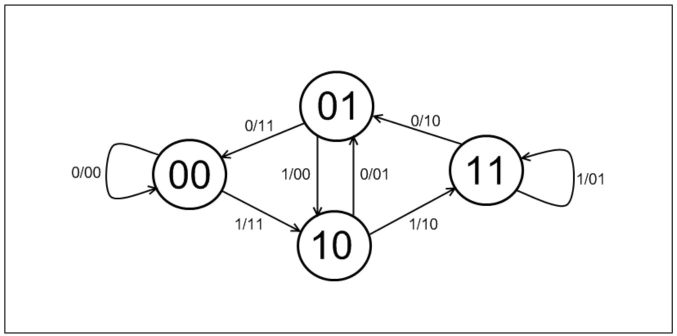

Figure 14: State machine representing the convolutional channel code with constraint length $k = 3$ and rate $r = 1/2$.