Nested polarized codes: general design[1]

Ilya Dumer dumer@ee.ucr.edu
University of California at Riverside, USA

Abstract. We consider recursive constructions that employ the $(u, u+v)$ decomposition similarly to polar codes and Reed-Muller (RM) codes. The first difference of our design is that this partitioning ends at various short RM codes instead of the single information bits used as "end nodes" in polar codes. In addition, we use maximum-likelihood (ML) decoding for these end nodes. This combination of recursive cancellation technique with ML decoding can improve the output error rates of polarized constructions on moderate lengths.

1 Introduction

This paper concerns code constructions that repeatedly use shorter codes to recursively design and decode the longer ones. One classical example is the Plotkin $(u, u+v)$ construction that builds up RM codes $R(r,m)$ of length $n = 2^m$ and dimension $k(r,m) = \sum_{i=0}^{r} \binom{m}{i}$ for any parameters $0 \leq r \leq m$. Another recursive design introduces polar codes [1]. Here the same Plotkin construction first encodes an n-dimensional Hamming space into the full, non-redundant code $R(m,m)$. The second (non-constructive) step selects some high-fidelity information bits of code $R(m,m)$ that can be recovered on a given memoryless channel with high probability that approaches 1 for long codes.

Decoding of both RM codes and polar codes uses the same recursive successive algorithm [1]- [3] that repeatedly relegates processing of the original block of length n to the shorter blocks of length $n/2$, $n/4$ and so on, similarly to the Fast Fourier Transform. This design yields a one-by-one successive retrieval of information bits in the single-bit codes $R(0,0)$ and has a low complexity order of $n \log_2 n$. It also turns out that recursive recalculations lead to very different reliabilities of specific information bits. By freezing the most error-prone information bits as zeros, we obtain subcodes of RM codes that closely approach ML performance [3] on the relatively short lengths $n \leq 512$. The algorithm also benefits by using the lists of the most probable candidates [3]. However, these lists become prohibitively large even on the moderate lengths $n \geq 1000$.

Asymptotically, this recursive technique yields capacity-achieving codes [1] if bit-freezing is applied to the full spaces $R(m,m)$ instead of the smaller codes $R(r,m)$. Namely, the breakthrough result of [1] shows that on any memoryless channel successive cancellation decoding yields the maximum possible fraction

[1] This work is supported in part by NSF Grant 1018935

of information symbols that achieve a vanishing error probability as $m \to \infty$. However, it turns out that capacity-approaching performance is achieved only on the very long blocks. For this reason, *efficient polar design on the moderate lengths* has become one of the central problems in coding theory.

To design such codes, we wish to extend the existing polar constructions. In particular, we will simultaneously encode-decode small groups of bits instead of a bit-by-bit processing of polar codes. To this end, we choose some short RM codes $R(s,g)$ as the end nodes in our recursive design instead of individual information bits. We then employ ML decoding of these codes, while keeping the overall complexity order close to $n \log_2 n$. Our preliminary analysis implies that ML decoding of short end codes can improve recursive performance on the moderate lengths similarly to the results [3] that used ML decoding on the biorthogonal codes $R(1,g)$. For polar codes, we will use a similar justification. Indeed, some end codes $R(s,g)$ in polar design can include both low-and-high fidelity information bits on moderate lengths. Recursive decoding completely discards these unreliable information bits. By contrast, ML decoding will make all information bits of the end codes $R(s,g)$ equally reliable. This approach is mostly useful on the short and moderate lengths. Indeed, polarization technique of [1] shows that long polar codes achieve almost full polarization not only for individual bits $R(0,0)$ but also on the short-length end nodes $R(s,g)$ for $g = o(m)$, By contrast, the shorter constructions with moderate $m \leq 15$ still leave high- and low-fidelity information bits mixed in some end codes $R(s,g)$. For any such code, MLD will allow us to reliably decode former unreliable bits and add them to polar codes. Below we use a common recursive design of codes $R(r,m)$ and polar codes and introduce a family of nested codes. The subsequent paper proceeds with recursive decoding and optimized design.

2 Recursive design of RM and polar codes

Let $F(m,r)$ be the set of boolean polynomials of degree r or less in m binary variables $x_1, \ldots, x_m$, where $r \leq m$. We use vectors $x = (x_1, ..., x_m)$ to mark the positions of our code and define codewords of the RM code $R(r,m)$ using the map $\mathbb{F}_2^m \overset{f(x)}{\to} \mathbb{F}_2$ for each $f(x) \in F(m,r)$. We then consider the recursive form

$$f(x) = x_1 f_1(x_2, ..., x_m) + f_0(x_2, ..., x_m) \tag{1}$$

For any $0 < r < m$, polynomials f_1 and f_0 have degrees $\deg f_1 \leq r-1$ and $\deg f_0 \leq r$. Then any codeword $\mathbf{c}$ of code $R(r,m)$ has the form $\mathbf{c} = \mathbf{u}, \mathbf{u}+\mathbf{v}$ where codewords $\mathbf{u}$ and $\mathbf{v}$ of length $n/2$ are generated by the polynomials f_0 and f_1 and belong to the codes $R(r,m-1)$ and $R(r-1,m-1)$, respectively. Polar codes use a similar design and replace two bits u,v with the product $(u,v)\,G = (u,u+v)$, where

$$G = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

Below we first decompose vector $\mathbf{c} = \mathbf{u}, \mathbf{u} + \mathbf{v}$ onto the pair $\mathbf{v} \equiv \mathbf{v}_{(1)}$ and $\mathbf{u} \equiv \mathbf{u}_{(1)}$. We also mark vectors $\mathbf{v}$ and $\mathbf{u}$ with an edge $\xi_1 = 0$ and $\xi_1 = 1$, respectively. We continue this splitting process and double the number of "end" vectors $\mathbf{v}_{(i)}$ and $\mathbf{u}_{(i)}$ in each step i. For each vector $\mathbf{v}_{(i)}$, we complement its current path $(\xi_1, ..., \xi_{i-1})$ with an edge $\xi_i = 0$. We also add an edge $\xi_i = 1$ if we proceed to vector $\mathbf{u}_{(i)}$. Now let us define the end points of this splitting process. Here we consider three different designs for codes $R(r, m)$.

Design A [2,3] uses all k paths $\xi = (\xi_1, ..., \xi_m)$ of length m and Hamming weight $\mathrm{wt}(\xi) \geq m - r$. One information bit is assigned to each path. Note that at some intermediate step, any path ξ arrives at some repetition code $R(0, g)$ or full space $R(h, h)$. Code $R(0, g)$ then uses $(\mathbf{u}, \mathbf{u} + \mathbf{v})$ design with $\mathbf{v} = \mathbf{0}$ until we arrive at the one-bit code $R(0, 0)$. For code $R(h, h)$, a $(\mathbf{u}, \mathbf{u} + \mathbf{v})$ design has both parts $\mathbf{u}, \mathbf{v}$ taken from the full spaces $\mathbb{F}_2^{h-1}$. Then decomposition of $R(h, h)$ leads to 2^h codes $R(0, 0)$. The overall result is code $R(r, m)$. This bit-by-bit representation is described in Fig. 1 for code $\mathrm{RM}(4, 7)$. The solid part of design A is used below in design B.

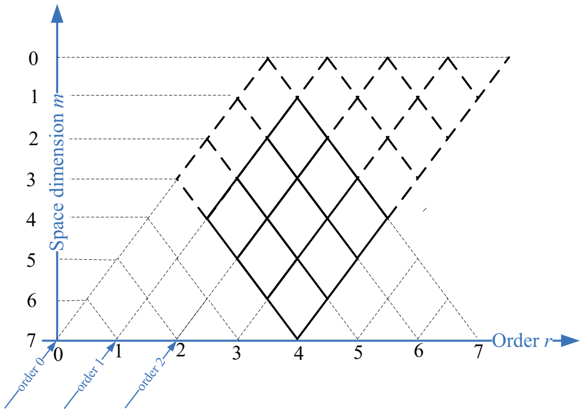

Figure 1: Decomposition of code $\mathrm{RM}(4, 7)$: bit-by-bit design A

Design B also follows [2,3] but takes the shorter paths $\xi = (\xi_1, ..., \xi_\ell)$ that end at some biorthogonal codes $R(1, g)$. Here some paths directly proceed from the origin (r, m) to the various end nodes $(1, g)$ with $g \geq 1$ and have lengths $\ell = m - g$. Other paths arrive at the full-space codes $R(h, h)$ and then split further until the extended paths $(\xi_1, ..., \xi_{m-1})$ end at the nodes $R(1, 1)$.

Finally, the design of Fig. 3 is similar to that of polar codes [1]. Here design A is applied to the full code $R(m, m)$ and removes the restriction on the Hamming weights $\mathrm{wt}(\xi) \geq m - r$ used in codes $R(r, m)$. An equivalent representation uses the generator matrix $G(m, m)$ of code $R(m, m)$, which is the Kronecker product $G^{\otimes m}$ of degree m. The rows in $G(m, m)$ are the maps of

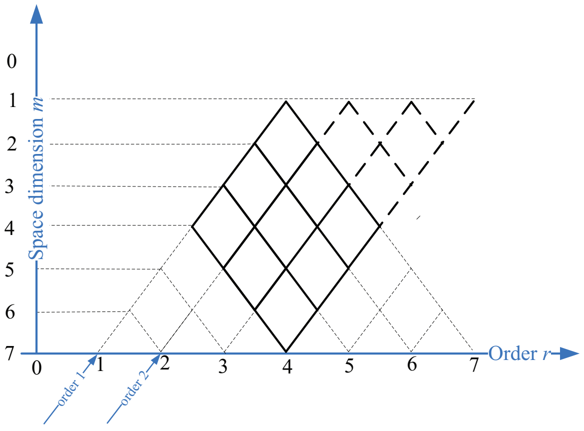

Figure 2: Decomposition of code RM$(4,7)$: design B with biorthogonal codes.

all m-variate monomials. Note also that every path $\xi=(\xi_1,...,\xi_m)$ represents the monomial $\prod_{i=1}^m x_i^{1-\xi_i}$ that gives a codeword of weight $2^{wt(\xi)}$, where $wt(\xi)$ is the Hamming weight of a path ξ. Similarly, the generator matrix for the code $R(r,m)$ consists of the rows that are the maps of monomials of degree r or less.

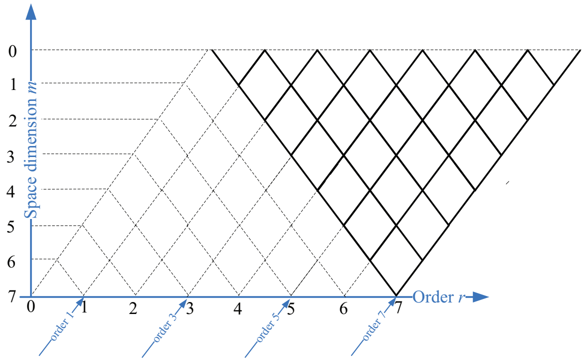

Figure 3: Decomposition of code RM$(7,7)$: bit-by-bit polar design C

We now proceed with the encoding of code $R(r,m)$. Let $\mathbf{a}_r^m$ be a block of k information bits used to give a code vector $\mathbf{u},\mathbf{u}+\mathbf{v}$. Then $\mathbf{a}_r^m$ can also be split into the sub-block $\mathbf{a}_r^{m-1}$ that encodes vector $\mathbf{u}$ and the sub-block $\mathbf{a}_{r-1}^{m-1}$ that encodes $\mathbf{v}$. Proceeding in the same way, we again split two information blocks. In the end, any given path $\xi=(\xi_1,...,\xi_\ell)$ arrives at some code $R(s,g)$. Here $s=g=0$ in designs A or C, while $s=1$ in design B. Let us assume that encoding gives some vector $\mathbf{c}_\ell=\mathbf{c}_\ell\,(\xi)$ of length 2^g on the end node $R(s,g)$.

Moving back to the origin $R(r,m)$ along the reverse path $(\xi_\ell,...,\xi_1)$, we obtain the longer vectors $\mathbf{c}_\ell,...,\mathbf{c}_0$ where

$$\mathbf{c}_{i-1} = (\xi_i\mathbf{c}_i,\mathbf{c}_i) = \begin{cases} (\mathbf{0},\mathbf{c}_i) & \text{if } \xi_i = 0 \\ (\mathbf{c}_i,\mathbf{c}_i) & \text{if } \xi_i = 1 \end{cases} \tag{2}$$

Then the entire path ξ gives some vector $\mathbf{c} = \mathbf{c}(\xi,\mathbf{c}_\ell)$ of length n that depends on the entry vector $\mathbf{c}_\ell$ and path ξ. In turn, a subset of prefix-free paths $T = \{\xi_{(i)},\, i = 1,...,N\}$ gives a codeword $\mathbf{c}(T)=\sum_{\xi\in T}\mathbf{c}(\xi,\mathbf{c}_\ell)$ obtained over all encoded paths. To generalize this design, we will now consider different paths $\xi = (\xi_1,...,\xi_\ell)$ of various lengths ℓ that will end at the non-trivial nodes $R(s,g)$ instead of $R(0,0)$. Here parameter $g = m-\ell$ is defined by the current length ℓ. Parameter $s = s(\xi)$ can also be fixed for each path ξ. Later s will be chosen to optimize decoding error probability. Let k_ξ denote the dimension of code $R(s,g)$ associated with path ξ and let $d_\xi = 2^{g-s+wt(\xi)}$.

Definition. Given some path ξ that ends at the node $R(s,g)$, define a code $C(\xi) = \{\mathbf{c}(\xi,\mathbf{c}_\ell) \mid \mathbf{c}_\ell \in R(s,g)\}$ that is obtained by encoding (2) of code vectors $\mathbf{c}_\ell \in R(s,g)$. Given a set of paths T of length m, let $C(m,T) = \cup_{\xi\in T}C(\xi)$.

Lemma 1. *Code $C(m,T)$ has length 2^m, dimension $k(m,T) = \sum_{\xi\in T}k_\xi$ and distance $d(m,T) = \min_{\xi\in T} d(\xi)$.*

Proof. Every code $C(\xi)$ is a subcode of $R(m,m)$ of dimension k_ξ. Its originating code $R(s,g)$ is an RM code generated by k_ξ monomials α of degree $\deg(\alpha) \le s$ in g variables $x_{\ell+1},...x_m$. Then code $C(\xi)$ is generated by monomials $\beta = \alpha\prod_{i=1}^{\ell} x_i^{1-\xi_i}$ that have $\deg\beta \le s+\ell-wt(\xi)$. Thus, code $C(\xi)$ is generated by different m-variate monomials of degree $\delta \le s+\ell-wt(\xi)$ and has distance $d \ge 2^{m-\delta} \ge d_\xi$. Note also that distance d_ξ is achieved on the monomials of maximal degree $s+\ell-wt(\xi)$. Next, note that different (prefix-free) paths ξ generate different sets of monomials β regardless of the chosen codes $R(s,g)$. Thus, different codes $C(\xi)$ are generated by different rows of the generator matrix $G(m,m)$ of the code $R(m,m)$. This gives the above dimension $k(m,T)$, since code $C(m,T)$ is a direct sum of codes $C(\xi)$. Its distance is defined by the fact that its generator matrix consists of different m-variate monomials whose highest degree is $\max_\xi(s+\ell-wt(\xi))$. □

At this point, code $C(m,T)$ is almost identical to polar design. Indeed, every node $R(s,g)$ itself is a collection of paths that connect $R(s,g)$ with some end nodes $R(0,0)$. Then all paths $\xi\in T$ can be extended to the length m and connect $R(m,m)$ to various nodes $R(0,0)$. This gives some subset T' of paths that is equivalent to design C. We note, however, that design $C(m,T)$ yields a more substantial difference in decoding procedures. Here we will end successive cancellation procedures at every point $R(s,g)$ instead of a bit recovery performed at the nodes $R(0,0)$. At every such node $R(s,g)$, we use some powerful decoding algorithm, such as MLD. Thus, design $C(m,T)$ becomes a polar design with the set of predefined end nodes $R(s,g)$. More generally, we can replace short RM codes $R(s,g)$ with end nodes, such as extended BCH codes or short polar codes. Further extensions can include some other recursive constructions, such as spherically restricted codes, whose length is different from 2^g.

Our main goal in this design is to keep all high-fidelity bits of polar codes, and also add the new multi-bit paths by using the end-node MLD instead of successive cancellation. To this end, we wish to extend the results of [3], which use MLD for biorthogonal nodes $R(1,g)$ instead of the full recursion. Fig. 4 depicts some of these improvements obtained for a short code $R(3,8)$ and its subcode of dimension 78. Here we exclude 15 low-fidelity information bits and use various lists of size $L \geq 1$. In the subsequent paper, we describe the simplest recursive algorithm with $L = 1$ and then optimize its complexity and output BER for the nested constructions.

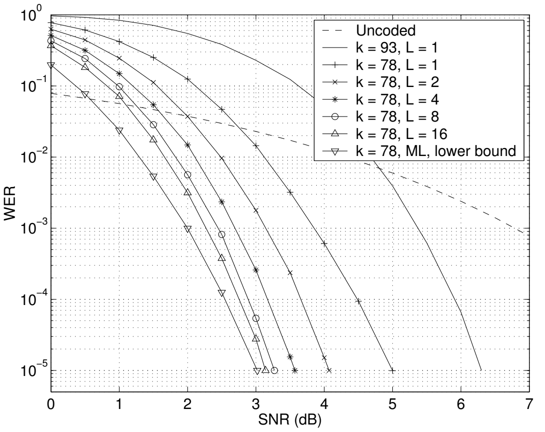

Figure 4: (256, 93) RM code $R(3,8)$ and its (256, 78)-subcode

References

[1] E. Arikan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," IEEE Trans. Info. Theory, vol. 55 , pp. 3051-3073, July 2009.

[2] I. Dumer, "Recursive decoding and its performance for low-rate Reed-Muller codes," IEEE Trans. Info. Theory, vol. 50, pp. 811-823, May 2004.

[3] I. Dumer and K. Shabunov, "Soft decision decoding of Reed-Muller codes: recursive lists," IEEE Trans. Info. Theory, vol. 52, pp. 1260-1266, March 2006.