ICT-MobileSummit 2009 Conference Proceedings
Paul Cunningham and Miriam Cunningham (Eds)
IIMC International Information Management Corporation, 2009
ISBN: 978-1-905824-12-0

Performance of short polar codes under ML decoding

Erdal Arıkan[1], Haesik Kim[2], Garik Markarian[3], Üstün Özgür[4], and Efecan Poyraz[5]
[1 4 5]*Electrical-Electronics Eng. Dept., Bilkent University, Ankara, 06800, Turkey*
Tel: +90-312-2901347, Fax: +90-312-2664192, Emails: arikan@ee.bilkent.edu.tr, ustun@ee.bilkent.edu.tr, e_poyraz@ug.bilkent.edu.tr
[2 3]*Dept. of Communication Systems, Lancaster University, Lancaster, LA1 4WA, UK*
Tel: +44 1524 510394, Fax: +44 1524 510493, Emails: h.kim2@lancaster.ac.uk, g.markarian@lancaster.ac.uk

Abstract: Polar codes are a recently introduced class of codes that achieve the capacity of arbitrary symmetric binary-input channels. This capacity-achieving performance is obtained by encoders and decoders of complexity $O(N \log N)$ where N is the code block-length. The performance of polar coding under belief propahation (BP) decoding has been studied before, using Reed-Muller (RM) codes as a benchmark. This work studies the performance of polar coding under trellis-based maximum-likelihood (ML) decoding, again using RM codes as a benchmark. One finding is that RM codes perform better than polar codes under ML decoding for certain short codes. On the other hand, polar codes have a lower trellis complexity. A second finding is that BP decoding offers performance comparable to ML decoding as the block-length is increased.

Keywords: Error correcting codes, capacity-achieving codes, polar codes, Reed-Muller codes, trellis decoding.

1. Introduction

Polar codes are a class of capacity-achieving codes introduced recently in [1]. The main motivation for the introduction of polar codes was theoretical, namely, to show the existence of a family of codes that are *provably* capacity-achieving and have low-complexity encoding and decoding algorithms. The well-established LDPC and turbo codes fall short in this regard because we still do not have a fully rigorous proof that they achieve channel capacity. Polar codes are the first familiy of codes to close this theoretical gap.

Polar coding owes its analytical tractability to its recursive structure. This recursive structure also leads to low-complexity encoding and decoding algorithms for polar coding. It is shown in [1], [2] that asymptotically, as a function of code block-length N, polar codes can be encoded in complexity $O(N \log N)$, decoded using a successive-cancellation decoder in complexity $O(N \log N)$, while achieving an overall block-decoding error probability that is bounded as $O(2^{-N^{\beta}})$ for any fixed $\beta < \frac{1}{2}$ and fixed code rate below the channel capacity.

Encouraged by this asymptotic result, an initial study of practical merits of polar coding was conducted in [3], where polar codes were compared with Reed-Muller (RM) codes under belief-propagation (BP) decoding. The conclusion of that study was that polar codes performed better than RM codes under BP decoding. The goal of the present paper is to study the performance of polar codes under ML decoding. Specifically, we consider trellis-based representations of polar codes and consider their bit error rate (BER) and frame error rate (FER) performance under BCJR and Viterbi algorithms. RM codes will serve as a benchmark for performance comparisons, as in the earlier study [3].

Unlike BP and SC decoders, the complexity of ML decoding increases exponentially as the code block-length is increased for any fixed non-zero coding rate. So, BP decoding of polar codes becomes infeasible after a certain block length. It is of interest to determine the range of block-lengths where ML decoding is feasible for polar codes and also compare ML and BP decoder performance to find out how much loss is incurred by using the sub-optimal BP decoder. These questions are addressed in this paper.

The study of polar coding at small to moderate block-lengths (64 to 512) may be of interest in applications where there is little tolerance for delay but the BER requirements are not very stringent. In such cases, polar codes may be applied as stand-alone codes. Polar codes under ML decoding may also be employed as component codes in iterative coding schemes.

The paper is organized as follows. In Section 2, we define polar and RM code constructions in a common framework. In Section 3, we consider the trellis representation of polar codes. Trellis complexities are given for comparable polar and RM codes. In Section 4, we offer some simulation results on polar codes that demonstrate the relative performance of polar and RM codes under ML decoding. Simulations are also given for comparing the performance of polar codes under ML and BP decoding. In Section 5, we summarize the results.

2. Code construction

We write $\mathbf{RM}(N, K)$ and $\mathbf{P}(N, K)$ to denote, respectively, polar and RM codes with block-length N and dimension K. These codes have similar constructions. First, we recall the well-known Plotkin construction for RM codes. Let $F = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$ and $F^{\otimes n}$ denote the nth Kronecker power of F. The generator matrix $G_{RM}(N, K)$ of an $\mathbf{RM}(N, K)$ code can be taken as any submatrix of $F^{\otimes \log_2 N}$ consisting of K distinct rows whose Hamming weights are as large as possible.

For example, to construct the $\mathbf{RM}(8, 5)$ code, we first compute

$$F^{\otimes 3} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \quad (1)$$

and select 5 of its heaviest rows to obtain

$$G_{RM}(8, 5) = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \quad (2)$$

This is one of three possible choices for $G_{RM}(8,5)$.

Polar code construction is similar to RM code construction in that the generator matrix $G_P(N,K)$ of the $\mathbf{P}(N,K)$ code, with $N=2^n$, is also selected as a submatrix of $F^{\otimes n}$, only by a different rule. The exact polar code construction rule is too complicated to be presented here. Instead, we shall use a heuristic rule, which was also used in [3]. First, a *ranking* vector $z_N = (z_{N,1}, \ldots, z_{N,N})$ is computed through the recursion

$$z_{2k,j} = \begin{cases} 2\,z_{k,j} - z_{k,j}^2, & \text{for } 1 \le j \le k \\ z_{k,j-k}^2, & \text{for } k+1 \le j \le 2k \end{cases} \tag{3}$$

for $k = 2^\ell$, $0 \le \ell \le n-1$, starting with $z_{1,1} = 1/2$. Next, one orders the elements of z_N to obtain a permutation $\pi_N = (i_1, \ldots, i_N)$ of the indices $(1, \ldots, N)$ so that $z_{N,i_j} \le z_{N,i_k}$ holds for all $1 \le j < k \le N$. The generator matrix $G_P(N,K)$ is selected as the submatrix of $F^{\otimes n}$ consisting of rows with indices $i_1, \ldots, i_K$.

For $(N,K) = (8,5)$, the ranking vector is computed as z_8 =(0.996, 0.684, 0.809, 0.121, 0.879, 0.191, 0.316, 0.004), which gives $\pi_8 = (8,4,6,7,2,3,5,1)$, and the generator matrix is uniquely determined as

$$G_P(8,5) = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{4}$$

Thus, there is no essential difference between $\mathbf{RM}(8,5)$ and $\mathbf{P}(8,5)$ codes, since one possible choice for $G_{RM}(8,5)$ coincides with $G_P(8,5)$. It turns out $(N,K) = (32,16)$ is the smallest dimension when the two codes differ in an essential manner. The generator matrix $G_P(32,16)$ employs a weight-4 row of $F^{\otimes 5}$ and leaves out a weight-8 row, unlike $G_{RM}(32,16)$ that employs no weight-4 row. RM codes and polar codes begin to differ significantly as the block-length N increases.

3. Trellis representation of polar codes

In this section, we study the performance and complexity of polar coding under trellis-based ML decoding. For terminology and details of trellis-based representation of block-codes, we refer to [4] and [5].

We list in Table 1, the trellis complexity of the two families of codes for various code dimensions (N,K). Listed in the table are the code minimum distance d, the state complexity V, and the branch complexity E. The parameters V and E equal, respectively, the total number of vertices and the total number of edges in the trellis representation. There are two types of trellis representations listed in the table: *bit-level* and *parallelized*. Both representations consist of N trellis sections; however, in the parallellized representation, the trellis is split into a number of identical and disjoint sub-trellises using the method described in [4, Sect. 9.7]. Although parallelization does not reduce the total state complexity, it simplifies the decoder implementation both in hardware and software. The complexity numbers in the table for the case of parallelized trellises refer to the complexity of one component sub-trellis.

We note that the bit-level trellis complexity numbers in the table for the RM codes agree with those in Table II of [6]. The table shows that for a given (N,K), polar codes

Table 1: Trellis complexity of polar and RM codes.

		Polar					RM				
			bit-level		parallelized			bit-level		parallelized	
N	K	d	V	E	V	E	d	V	E	V	E
32	16	4	1222	1628	232	292	8	4798	6396	434	540
32	26	4	638	1180	390	700	4	638	1180	390	700
64	57	2	818	1556	818	1556	4	2638	5084	1630	3100
128	120	2	3338	6516	3338	6516	4	10734	21084	6670	13020

tend to have significantly lower bit-level trellis complexity than RM codes when the two codes differ. (For the parameter $(32, 26)$, the two codes coincide.) The table also shows that RM codes benefit from parallelization more than polar codes do; however, polar codes still remain less complex.

To gain some insight into the trellis complexity issue, let us look at the $\mathbf{RM}(64, 57)$ and $\mathbf{P}(64, 57)$ codes. The matrices $G_P(64, 57)$ and $G_{RM}(64, 57)$ happen to differ only in one row. The RM generator matrix uses the row $10^{15}10^{15}10^{15}10^{15}$, where 0^{15} denotes 15 consecutive 0s; instead, the polar generator matrix uses the row $1^2 0^{62}$. This choice reduces the minimum Hamming distance of the code from 4 to 2; however, it also reduces the trellis complexity since the second row has a shorter span of 1s. The preference of low-span rows by polar coding is also in evidence in the (8,5) code example whose generator matrix was given above.

4. Simulations

We now give simulation results for the BER and FER performance of polar codes and RM codes over a BPSK channel with additive Gaussian noise. Due to space limitations, we will present only a limited number of simulation results. The conclusions that will be drawn on the given examples are consistent with experimental results performed for other code lengths and rates.

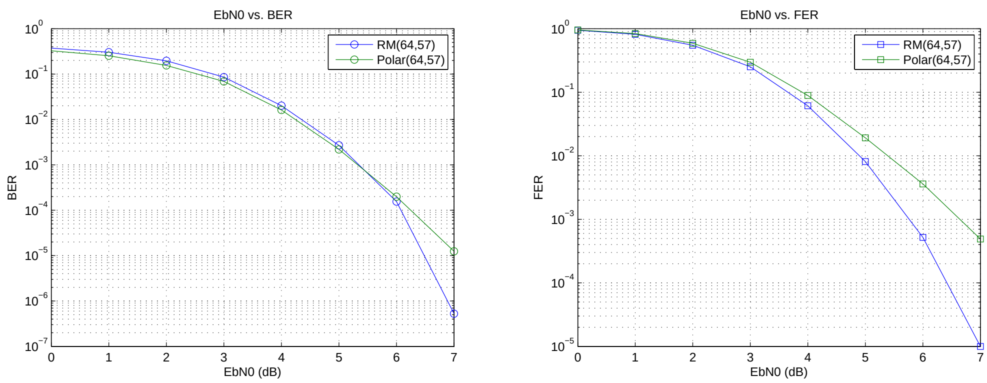

Figure 1: Error rates for (64,57) polar and RM codes on a BSPK channel.

Figure 1 shows the performance for $(64, 57)$ codes under Viterbi decoding. We observe that polar codes have a slightly better BER performance at low EbN0, while at high SNR, RM codes outperform polar codes. The FER performance of RM codes appears better than that of polar codes at all EbN0 values. The significantly better performance of RM codes at high EbN0 is explained by their better minimum distance. In return for their relatively poor performance, polar codes have a significantly lower trellis complexity. In applications where complexity is important and a BER of 1E-4 may be tolerated, the P(64,57) code appears to be a viable alternative to the RM(64,57) code.

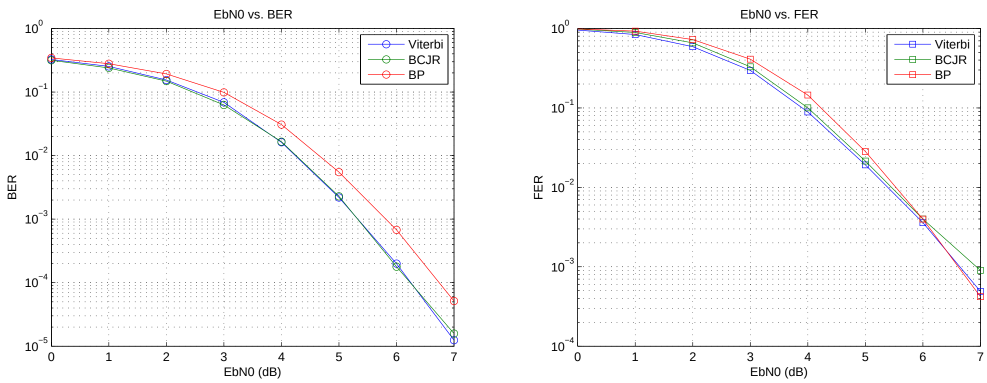

Figure 2: Error rates for the (64,57) polar code under various decoding algorithms.

Figure 2 shows the performance of the P(64,57) code under Viterbi, BCJR, and BP decoders. We observe that the Viterbi and BCJR algorithms give roughly the same performance throughout the EbN0 range. The BP algorithm on the other hand has a slightly worse BER performance than the other two, while its FER performance is not markedly different. Due to its $O(N \log N)$ complexity, the BP algorithm can be used at much higher block-lengths N compared to the ML algorithms (Viterbi and BCJR) for which the complexity is exponential in N.

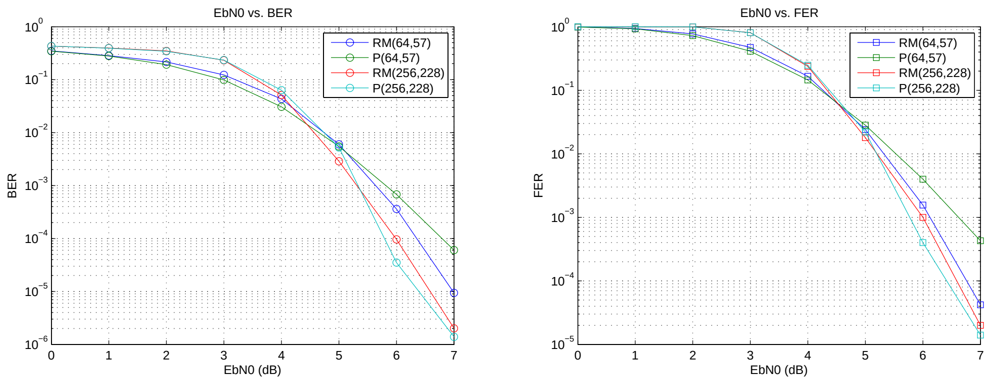

Figure 3: Error rates for polar and RM codes of sizes (64,57) and (256,228) under BP decoding on a BSPK channel.

Figure 3 illustrates that, under BP decoding, a performance advantage emerges in favor of polar codes as one increases the block-length. The figure shows results under BP decoding for the codes RM(64,57), P(64,57), RM(256,228), and P(256,228), all of which have the same rate. It is seen that while the high-EbN0 performance of the RM code is better than that of the polar code for a code size of (64,57), the situation is reversed when the code size is increased. This result confirms earlier findings of [3].

5. Summary and conclusions

We have compared polar codes and RM codes under trellis-based ML decoding at short block-lengths. One observation has been that, when the two codes differ for a given code size (N, K), the polar code tends to have a significantly lower trellis complexity, while the RM code has a larger minimum distance. The minimum-distance advantage of RM codes translates into better performance at high SNR, although polar codes seem to perform slightly better at low SNR.

A second issue addressed in the paper has been the performance comparison between BP and ML decoding algoritms. BP decoders are suboptimal, however, they have complexity $O(N \log N)$. On the other hand, ML decoders have exponential complexity in N. So, for a given level of complexity, one may try to make up for the deficiency of BP decoding relative to ML decoding by using a code with a larger block-length. The experimental results suggest that the performance disadvantage of BP decoding relative to ML decoding is not too large even at short block-lengths (the comparisons for P(64,57) code). By increasing the block-length somewhat, BP decoding starts to outperform the ML decoder (the comparison of codes of sizes (256,228) vs. (64,57)). At large block-lengths, where ML decoding is no longer feasible, polar codes appear to be better than RM codes when both are decoded by a BP decoder. It appears that the minimum-distance advantage of RM codes over polar codes becomes a moot point as the block-length is increased.

Acknowledgement

This work was supported in part by The Scientific and Technological Research Council of Turkey (TÜBİTAK) under project no 107E216 and in part by the European Community under FP7 NoE NEWCOM++ (grant no 216715), FP7 STREP WiMAGIC (grant no 215167), and FP7 Intra-European Fellowship Wi-CODING (grant no 219612).

References

[1] E. Arıkan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," 2008. [Online]. Available: http://arxiv.org/abs/0807.3917. (To appear in *IEEE Trans. Inform. Theory.*)

[2] E. Arıkan and E. Telatar, "On the rate of channel polarization," 2008. [Online]. Available: http://arxiv.org/abs/0807.3806.

[3] E. Arıkan, "A performance comparison of polar codes and Reed-Muller codes," *IEEE Comm. Letters*, vol. 12, pp. 447–449, June 2008.

[4] S. Lin and D. J. Costello, Jr., *Error Control Coding, (2nd ed)*. Pearson: N.J., 2004.

[5] B. Honary and G. Markarian, *Trellis Decoding of Block Codes*. Kluwer, 1997.

[6] A. B. Kiely, S. J. Dolinar, R. J. McEliece, L. L. Ekroot, and W. Lin, "Trellis decoding complexity of linear block codes," *IEEE Trans. Inform. Theory*, vol. IT-42, pp. 1687–1697, Nov. 1996.

