

# A Survey of Reed-Muller Codes from Polar Coding Perspective

Erdal Arıkan  
Bilkent University, Ankara, Turkey

**Abstract**—A survey of Reed-Muller (RM) coding is given with the goal of establishing a continuity between RM codes and polar codes. The focus is mainly on recursive decoding methods for RM codes and other ideas that are most relevant to polar coding.

**Index Terms**—Reed-Muller codes, polar codes, channel polarization.

## I. RM CODES

RM codes were discovered by Muller [1] and Reed [2]. For every pair of integers  $0 \leq r \leq m$ , there is an RM code, denoted  $RM(r, m)$ , with block-length  $n = 2^m$ , dimension  $k = \sum_{i=0}^r \binom{m}{i}$ , and minimum distance  $d = 2^{m-r}$ . For each  $m \geq 0$ , the code  $RM(0, m)$  is the repetition code and  $RM(m, m)$  is the trivial code consisting of all binary  $n$ -tuples. All other RM codes can be obtained by the Plotkin construction [3], also known as the  $|u|u + v|$  construction, by the recursion

$$RM(r, m) = \{(u, u + v) : u \in RM(r, m - 1), v \in RM(r - 1, m - 1)\}. \quad (1)$$

A more explicit algebraic description of RM codes is given by first specifying a generator matrix for  $RM(m, m)$  as

$$G_{RM}(m, m) = F^{\otimes m}, \quad F = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}, \quad (2)$$

where  $F^{\otimes m}$  denotes the  $m$ th Kronecker power of  $F$ . Then, for  $0 \leq r \leq m$ ,  $G_{RM}(r, m)$ , the generator matrix of  $RM(r, m)$ , is taken as the submatrix of  $G_{RM}(m, m)$  consisting of rows with Hamming weights  $\geq 2^{m-r}$ . The equivalence of these two constructions is well-known (see, e.g., [4, p. 114]).

For example, for  $m = 3$ , we have

$$G_{RM}(3, 3) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \quad (3)$$

and  $RM(1, 3)$  is the code with generator matrix

$$G_{RM}(1, 3) = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \quad (4)$$

The encoding operation for all RM codes of a given length  $n = 2^m$  can be represented universally as a transform

$$\mathbf{x} = \mathbf{d}F^{\otimes m} \quad (5)$$

where  $\mathbf{d} = (d_0, \dots, d_{n-1})$  and  $\mathbf{x} = (x_0, \dots, x_{n-1})$  represent data and code words, respectively. Depending on the order of the RM code to be encoded, some of the elements of  $\mathbf{d}$  are fixed to zero. For example, for the  $RM(1, 3)$  code, the fixed coordinates are  $d_0, d_1, d_2$ , and  $d_4$ . The coordinates that are not fixed are free to take on any 0-1 value.

The universal encoder can be implemented by a circuit of complexity  $O(n \log n)$ , as shown in Fig. 1 for  $n = 8$ . The figure also shows how a codeword  $\mathbf{x}$  splits into two parts  $\mathbf{u}$  and  $\mathbf{v}$ , each of which in turn splits again into two parts, and so on, until one reaches single data elements  $d_i$ . For example,  $\mathbf{u}$  has the form  $|r|r + s|$ , and  $\mathbf{s}$  has the form  $|d_5|d_4 + d_5|$ . Thus, this code has a three-level  $|u|u + v|$  structure.

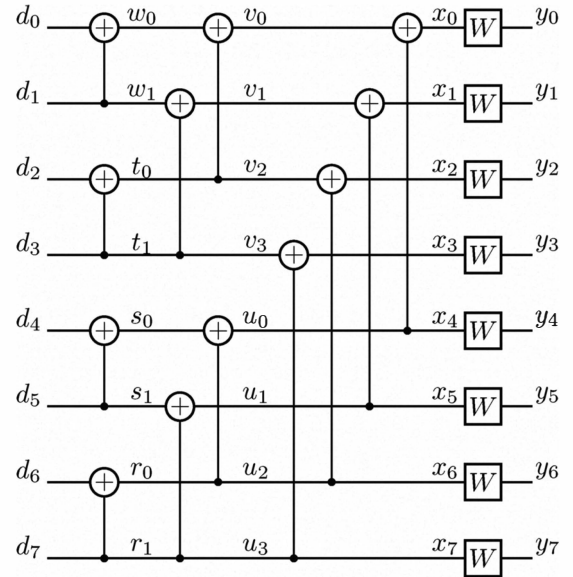


Fig. 1. Encoding and transmission of data. A codeword  $\mathbf{x}$  is obtained by a circuit that multiplies a data word  $\mathbf{d}$  by  $F^{\otimes 3}$ . The codeword is sent over a memoryless channel  $W$  and a channel output  $\mathbf{y}$  is received.

The transform (5) can be used as an encoder for a broader class of codes than just RM codes. For a given pair of code parameters  $(n, k)$ , there are  $\binom{n}{k}$  codes that can be encoded, each corresponding to a particular way of choosing the  $(n - k)$

frozen coordinates (coordinates of  $\mathbf{d}$  that are fixed as zero). Each code in this class admits a multi-level  $|\mathbf{u}| + |\mathbf{v}|$  decomposition, beginning at  $\mathbf{x}$  and progressing recursively down to individual elements of  $\mathbf{d}$ .

Polar codes are members of this class. They choose the frozen coordinates using a channel-specific rule, as described in [5]. By adapting the code to the channel, polar coding is able to achieve the capacity of any symmetric binary-input channel.

Since RM codes and polar codes share a common universal encoder, it is natural to expect that they can be decoded by similar algorithms. The following survey of decoding algorithms for RM codes shows that this is indeed the case. Many of the elements of polar coding can be found in the vast literature on RM coding.

## II. DECODING OF RM CODES

We will examine four categories of RM decoders: (i) maximum-likelihood (ML), (ii) majority-logic, (iii) recursive, and (iv) belief-propagation (BP) decoders. Our emphasis will be mainly on recursive decoders with soft-decisions since these are the most relevant ones for the purposes of this study. We have made no attempt to compile a complete list of references on RM decoders since this would be impossible within the scope of this brief survey.

### A. ML Decoders

All known methods of ML decoding for general RM codes have exponential complexity. However, for first-order RM codes,  $\text{RM}(1, m)$ , there exist ML decoders of complexity  $O(n \log n)$ . Such a decoder was developed by Green [6] and applied to the  $\text{RM}(1, 5)$  code used in NASA's 1972 Mariner 9 mission to send photographs of Mars. Green's design was based on a fast Hadamard transform (FHT) approach and its description may be found in [7, p. 419]. For first order RM codes, a symbol-by-symbol maximum a-posteriori probability (MAP) decoder of complexity  $O(n \log n)$  has also been found [8].

Although ML decoding is not a practical alternative for RM codes in general, it can be used if the code block-length is sufficiently small. The trellis-based approach may be the most efficient implementation of ML decoding for RM codes. Trellis representations of RM codes have been studied in [9] and RM trellis complexity is known [10]. A comparison of RM codes and polar codes under trellis-based ML decoding has been carried out in [11] using BPSK modulation over an AWGN channel. These experiments show that for a given code size  $(n, k)$ , polar codes tend to have smaller trellis complexity; the two codes have similar error-rates at low SNR, but RM codes perform visibly better at high SNR. This is explained by the better minimum distance properties of RM codes compared to polar codes [12].

A fundamental open problem on this subject is whether RM codes are capacity-achieving under ML decoding, at least for some channels with sufficient amount of symmetry. Some evidence in support of an affirmative answer to this open

problem is presented in [13, Sect. III-B] based on different considerations.

### B. Majority-Logic Decoders

The first low-complexity decoder for RM codes was the one proposed by Reed [2]. Descriptions of Reed's decoder can be found in [7, p. 385] and [4, p. 107]. This decoder had complexity  $O(nk)$  and could correct all error patterns up to half the minimum distance,  $d/2$ . Later work on Reed's decoder [14] showed that, asymptotically as the code length increased, it could in fact correct most error patterns of weight less than  $(d \ln d)/4$  for any fixed rate  $0 < R < 1$ . A soft-decision version of Reed's algorithm was introduced in [15], which increased the number of correctable errors by a factor of  $4/\pi$ .

Reed's decoder is mainly of historical interest since it has been superseded by more powerful and less complex decoding algorithms. Historically, it was the first example of a "successive cancellation" decoder. It made decisions on information bits one at a time and applied cancellation (decision feedback) before moving on to the next bit in the decoding sequence. Reed's decoder was also the first majority-logic decoder, a term that came to be used for a more general class of decoders [16].

### C. Recursive Decoders

By recursive decoders for RM codes, we mean decoders that take advantage of the multi-level  $|\mathbf{u}| + |\mathbf{v}|$  structure of RM codes. These decoders incorporate multiple decision stages and each stage feeds its decision to the next stage. There exist such decoders with computational complexity  $O(n \log n)$  and performance better than the majority-logic decoders discussed above.

There appears to be a gap of more than two decades between the discovery of RM codes and the introduction of recursive decoders for RM codes. The 1977 book [7] by MacWilliams and Sloane discusses RM codes in encyclopedic detail but does not mention any recursive decoder based on their  $|\mathbf{u}| + |\mathbf{v}|$  structure. The slow pace of advance in recursive decoding techniques for RM codes may have been due to a lack of interest in these codes apparently because they were perceived inferior to other alternatives.<sup>1</sup>

Renewed interest in  $|\mathbf{u}| + |\mathbf{v}|$  constructions, and RM codes in particular, seems to have been awakened as part of a wave of interrelated ideas known as "multi-level coding" [17], "generalized concatenated coding" [18], [19], and "trellis-based representations and decoding of block codes" [20]. Examples of papers that treat RM codes from these broader perspectives are [9], [21], and [22]. An exposition of this line of work on recursive decoders designed for multi-level  $|\mathbf{u}| + |\mathbf{v}|$  codes, and RM codes in particular, is given in [4, Chapter 15].

We have not been able to pin-point the first paper where a recursive decoder for RM codes was given. For the purposes of this survey, the paper [23] by Schnabl and Bossert is a

<sup>1</sup>A recent review paper by Costello and Forney [13, Sections III.B and IV.E] gives a more positive assessment of RM codes.

good starting point to discuss recursive RM decoders since it describes such a decoder explicitly and uses a simple setting and notation.<sup>2</sup> We will describe the main ideas of the decoding algorithm in [23] using a small example. Our presentation differs from that in [23] and is designed to show the connections to polar coding.

Consider using the code  $\text{RM}(1, 3)$  over a discrete memoryless channel (DMC)  $W$ . The system model is that depicted in Fig. 1 with the data bits  $u_0, u_1, u_2$ , and  $u_4$  set to zero. Given a channel output  $\mathbf{y}$ , the recursive decoding algorithm splits the decoding task for  $\mathbf{x} \in \text{RM}(1, 3)$  into a first decoding task for  $\mathbf{v} \in \text{RM}(0, 2)$  and a second one for  $\mathbf{u} \in \text{RM}(1, 2)$ . In decoding for  $\mathbf{v}$ , the part  $\mathbf{u}$  is treated as pure binary noise with independent  $\text{Bernoulli}(\frac{1}{2})$  components. This is a simplifying assumption, which is actually false in this example. Effectively, the decoding problem for  $\mathbf{v}$  is now reduced to the case shown in Fig. 2.

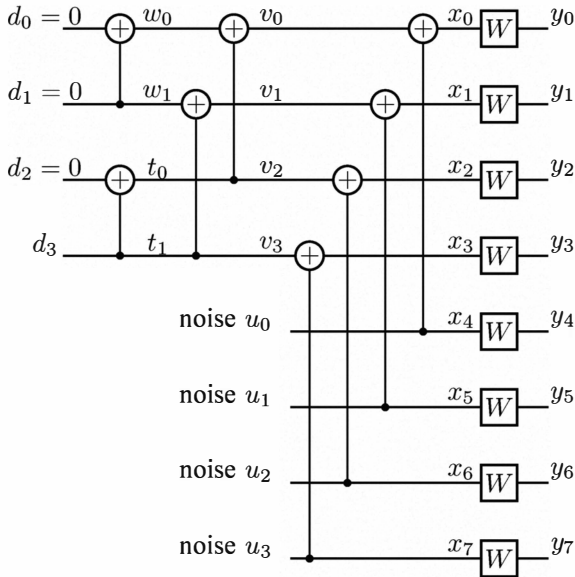


Fig. 2. System model for decoding  $\mathbf{v} \in \text{RM}(0, 2)$ . The variables  $\mathbf{u}$  are treated as pure noise.

By straightforward likelihood-ratio combining algebra, the decoding problem is simplified further to the form in Fig. 3 where  $W^-$  is a DMC with input  $v_i$ , output  $y_i^- = (y_i, y_{i+4})$ , and transition probabilities

$$W^-(y_i^- | v_i) = \frac{1}{2} W(y_i | v_i) W(y_{i+4} | 0) + \frac{1}{2} W(y_i | v_i + 1) W(y_{i+4} | 1). \quad (6)$$

The decoding problem for  $\mathbf{v} \in \text{RM}(0, 2)$  is now in the form of a general decoding problem for RM codes, as in the beginning, only the size of the problem is smaller. So, the problem size can be reduced still further by using the

<sup>2</sup>We should mention that [24] and [25] are two earlier papers often cited on this subject but we have not been able to obtain copies of these two papers.

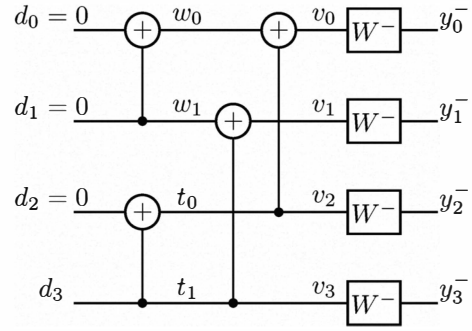


Fig. 3. Reduced decoding task for  $\mathbf{v}$  with an effective channel  $W^-$ .

same technique again. However, the particular algorithm in [23] stops and applies ML decoding as soon as the code at hand is a repetition code  $\text{RM}(0, m')$  or a single parity-check (SPC) code  $\text{RM}(m' - 1, m')$ , for some  $m' \leq m$ . Since  $\mathbf{v}$  in this example is already a repetition codeword, the decoder proceeds with ML decoding, using the model in Fig. 3; it generates an estimate  $\hat{d}_3$  of the unknown data bit  $d_3$  and an estimate  $\hat{\mathbf{v}}$  of  $\mathbf{v}$ .

The decoding algorithm now returns to the first branching point and starts decoding for  $\mathbf{u}$  armed with the estimate  $\hat{\mathbf{v}}$ . The algorithm uses the system model in Fig. 4 for carrying out this task. This model is not a precise model of the underlying system since it assumes that the estimate  $\hat{\mathbf{v}}$  is always correct, without allowance for chance of error.

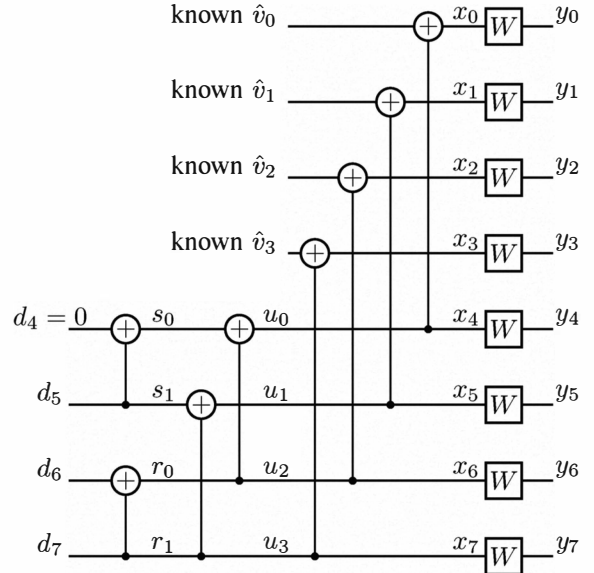


Fig. 4. System model for decoding  $\mathbf{u} \in \text{RM}(1, 2)$ . The estimates  $\hat{\mathbf{v}}$  are assumed correct.

The algorithm simplifies the problem by defining an equivalent channel  $W^+$  with input  $u_i$ , output  $y_i^+ = (y_i, y_{i+4}, \hat{v}_i)$ ,

and transition probabilities

$$W^+(y_i^+|u_i) = \frac{1}{2}W(y_i|\hat{v}_i + u_i)W(y_{i+4}|u_i) \quad (7)$$

This reduces the present decoding problem to the case shown in Fig. 5. The problem is now in the same form as the original RM decoding problem, but the size of the problem is smaller. Again, the problem size can be reduced still further by recursion. However, in this example, the algorithm proceeds with ML decoding at this point since  $\mathbf{u}$  is already a SPC codeword. Estimates  $\hat{d}_5, \hat{d}_6, \hat{d}_7$  of the unknown data bits are generated, along with an estimate  $\hat{\mathbf{u}}$  of  $\mathbf{u}$ . The decoding task is thus completed.

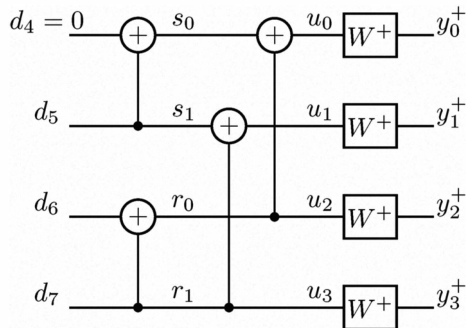


Fig. 5. Reduced decoding task for  $\mathbf{u}$  with an effective channel  $W^+$ .

The general form of the recursive decoder should be apparent from the above example. The actual algorithm in [23] makes some simplifications in likelihood-ratio calculations so as to reduce computational complexity. However, these simplifications do not change the asymptotic complexity of the recursive decoder; with or without such approximations, the asymptotic complexity remains as  $O(n \log n)$ . Further refinements and a list-decoding version of this algorithm are given in [26], [27].

Clearly, the above recursive decoder can be applied to any code in the broader class of codes that can be encoded using the transform (5), in particular, to polar codes. It is interesting to compare the above recursive decoder with the one in [5], which was used to prove the capacity-achieving nature of polar codes. Although the two decoders initially look different, on closer inspection, it is seen that they are essentially the same [28]. A minor difference between the two decoders is that the decoder in [5] processes the codeword  $\mathbf{x}$  and the channel output  $\mathbf{y}$  in bit-reversed index order. Also, for the purposes of [5], it was necessary that the decoder there avoid making any approximations in calculating likelihood-ratios and continue the recursion until reaching individual data bits  $d_i$ ,  $1 \leq i \leq n$ .

Significant advances in the analysis and refinement of the recursive RM decoder have been made in the last decade by Dumer and his co-authors. In [29], the algorithm in [23] was modified by applying ML decoding as soon as a first-order RM code is encountered, rather than waiting for a repetition code. In the above example, this modified algorithm would apply

ML decoding to the RM(1, 3) code at the outset, instead of splitting the decoding task. Since there exist ML decoders of complexity  $O(n \log n)$  for first-order RM codes, as already mentioned above, this modification preserves the  $O(n \log n)$  complexity of the overall algorithm but may improve performance significantly, as shown in [29] by examples.

An idea proposed in [30] and [31] is to combine recursive decoding of RM codes with “permutation decoding.” The recursive RM decoder, as any successive cancellation decoder, is sensitive to the order in which the decisions about the data bits are made. Since RM codes have a rich group of symmetries (automorphism group) under permutation of codeword coordinates [7, p. 398], a recursive decoder for RM codes can make independent decoding trials on various permutations of the received word. Thus, a list of decisions can be compiled and the most reliable one in the list can be selected as the final decoder decision. As expected, this improves the decoder performance.

An important observation in [29] was that the error probability of the recursive RM decoder is dominated by the probability of false decisions at a small number of weak codes in the decoding hierarchy. These weak codes are typically the first codes where final decoding decisions are made (such as the RM(1,  $m - r + 1$ ) code in the decoding of RM( $r, m$ )). As a remedy, [29] suggests identifying such weak codes and fixing their information bits to zero, effectively replacing the original RM code with one of its subcodes. It turns out these subcodes, if chosen properly, can be far more reliable under recursive decoding than the original code, but this comes at the expense of a loss in code rate. This idea is studied in [32] and [31].

In more recent work [33], [34], [35], Burnashev and Dumer refine the earlier analyses on the probability of error for recursive RM decoders. The primary goal in these papers is to find analytical bounds on the probability of decision error at each terminal code (first-order RM and SPC codes) in the decoding hierarchy. The papers succeed in obtaining some partial results; however, the general problem remains unsolved.

In some sense, the problem of estimating the error probability for a recursive decoder is also the central problem of polar coding. This problem has been addressed in [5] by using martingale techniques, which were sufficient to prove that polar codes are capacity-achieving. In [36], the same problem has been approached from a computational perspective using density evolution techniques that were originally developed in [37].

#### D. BP Decoders

We close this review by briefly mentioning BP decoders. RM codes can be represented by sparse factor graphs as shown by Forney [38]. Thus, they can be decoded using iterative graph-based decoding algorithms, which we refer to collectively as BP decoders. The complexity of BP decoding for RM codes is  $O(n \log n)$  per iteration. The flow of soft-information in a BP decoder for RM codes is similar to that in a recursive RM decoder; however, BP decoders avoid making

hard decisions prematurely, thus alleviating the problem of propagation of decision errors. Simulations reported in [39] show that polar codes perform significantly better than RM codes under BP decoding. A more extensive study of polar coding under variants of BP decoding as well as ML decoding is reported in [12].

### III. CONCLUSIONS

This survey had the goal of tracing the main ideas of polar coding in the RM coding literature. The survey shows that polar coding rests on a collection of encoding and decoding methods that existed for many years in a vast body of literature spanning not only RM codes but more general coding techniques. However, these methods had not been synthesized into a capacity-achieving coding technique prior to polar coding. Polar coding achieved this synthesis through the “channel polarization” method introduced in [5].

### ACKNOWLEDGMENT

The author wishes to thank Satish B. Korada for helpful comments on the paper. This work was supported in part by The Scientific and Technological Research Council of Turkey (TÜBİTAK) under contract no. 107E216, and in part by the European Commission FP7 Network of Excellence NEWCOM++ (contract no. 216715).

### REFERENCES

- [1] D. E. Muller, “Application of Boolean algebra to switching circuit design and to error correction,” *IRE Trans. Electronic Computers*, vol. EC-3, pp. 6–12, Sept. 1954.
- [2] I. Reed, “A class of multiple-error-correcting codes and the decoding scheme,” *IRE Trans. Inform. Theory*, vol. 4, pp. 39–44, Sept. 1954.
- [3] M. Plotkin, “Binary codes with specified minimum distance,” *IRE Trans. Inform. Theory*, vol. 6, pp. 445–450, Sept. 1960.
- [4] S. Lin and D. J. Costello, Jr., *Error Control Coding, (2nd ed)*. Pearson: N.J., 2004.
- [5] E. Arıkan, “Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels,” *IEEE Trans. Inform. Theory*, vol. 55, pp. 3051–3073, July 2009.
- [6] R. R. Green, “US Patent 3537103: Serial Digital Decoder,” Oct. 1970.
- [7] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*. North Holland: Amsterdam, 1977.
- [8] A. Ashikhmin and S. Litsyn, “Simple MAP decoding of first-order Reed-Muller and Hamming codes,” *IEEE Trans. Inform. Theory*, vol. 50, pp. 1812–1818, Aug. 2004.
- [9] G. D. Forney Jr., “Coset codes—Part II: Binary lattices and related codes,” *IEEE Trans. Inform. Theory*, vol. IT-34, pp. 1152–1187, Sept. 1988.
- [10] C.-C. Lu and S.-H. Huang, “On bit-level trellis complexity of Reed-Muller codes,” *IEEE Trans. Inform. Theory*, vol. 41, pp. 2061–2064, Nov 1995.
- [11] E. Arıkan, H. Kim, G. Markarian, Ü. Özgür, and E. Poyraz, “Performance of short polar codes under ML decoding,” in *Proc. ICT Mobile Summit 2009*, (Santander, Spain), 10-12 June 2009.
- [12] N. Hussami, S. B. Korada, and R. Urbanke, “Performance of polar codes for channel and source coding,” in *Proc. 2009 IEEE Int. Symp. Inform. Theory*, (Seoul, South Korea), pp. 1488–1492, July 2009.
- [13] D. J. Costello and G. D. Forney Jr., “Channel coding: The road to channel capacity,” *Proc. IEEE*, vol. 95, pp. 1150–1177, June 2007.
- [14] R. Krichevskiy, “On the number of Reed-Muller code correctable errors,” *Dokl. Soviet Acad. Sci.*, no. 191, pp. 541–547, 1970.
- [15] I. Dumer and R. Krichevskiy, “Soft-decision majority decoding of Reed-Muller codes,” *IEEE Trans. Inform. Theory*, vol. 46, pp. 258–264, Jan. 2000.
- [16] J. L. Massey, “Threshold Decoding,” RLE 410, Mass. Inst. Tech., RLE, April 1963.
- [17] H. Imai and S. Hirakawa, “A new multilevel coding method using error correcting codes,” *IEEE Trans. Inform. Theory*, vol. IT-23, pp. 371–377, May 1977.
- [18] E. L. Blokh and V. V. Zyablov, “Coding of generalized concatenated codes,” *Probl. Peredachi Inf.*, vol. 10, no. 3, pp. 45–50, 1974.
- [19] V. A. Zinov’ev, “Generalized cascade codes,” *Probl. Peredachi Inf.*, vol. 12, no. 1, pp. 5–15, 1976.
- [20] J. Wolf, “Efficient maximum likelihood decoding of linear block codes using a trellis,” *IEEE Trans. Inform. Theory*, vol. 24, pp. 76–80, Jan. 1978.
- [21] A. Calderbank, “Multilevel codes and multistage decoding,” *IEEE Trans. Comm.*, vol. 37, pp. 222–229, Mar. 1989.
- [22] F. Hemmati, “Closest coset decoding of  $|u|u + v|$  codes,” *IEEE J. Sel. Areas Comm.*, vol. 7, pp. 982–988, Aug. 1989.
- [23] G. Schnabl and M. Bossert, “Soft-decision decoding of Reed-Muller codes as generalized multiple concatenated codes,” *IEEE Trans. Inform. Theory*, vol. IT-41, pp. 304–308, Jan. 1995.
- [24] S. N. Litsyn, “On decoding complexity of low-rate Reed-Muller codes,” in *All-Union Conference on Coding Theory and Information Transmission*, pp. 202–204, 1988.
- [25] G. A. Kabatyanskii, “On decoding Reed-Muller codes in semicontinuous channel,” in *Proc. 2nd Int. Workshop on Algebraic and Combinatorial Coding Theory*, (Leningrad, U.S.S.R.), pp. 87–91, 1990.
- [26] R. Lucas, M. Bossert, and A. Dammann, “Improved soft-decision decoding of Reed-Muller codes as generalized multiple concatenated codes,” in *2nd. ITG Conf. Sour. Chan. Coding*, (Aachen, Germany), pp. 137–142, 1998.
- [27] R. Lucas, M. Bossert, and A. Dammann, “Improved list-decoding for Reed-Muller codes as generalized multiple concatenated (GMC) codes,” in *Proc. 1998 IEEE Int. Symp. Inform. Theory*, (Cambridge, MA), p. 337, Aug. 1998.
- [28] F. Didier. Private communication, June 2009.
- [29] I. Dumer and K. Shabunov, “Recursive decoding of Reed-Muller codes,” in *Proc. 2000 IEEE Int. Symp. Inform. Theory*, (Sorrento, Italy), p. 63, 2000.
- [30] I. Dumer and K. Shabunov, “Recursive and permutation decoding for Reed-Muller codes,” in *Proc. 2002 IEEE Int. Symp. Inform. Theory*, (Lausanne, Switzerland), p. 146, 2002.
- [31] I. Dumer and K. Shabunov, “Soft-decision decoding of Reed-Muller codes: recursive lists,” *IEEE Trans. Inform. Theory*, vol. 52, pp. 1260–1266, March 2006.
- [32] I. Dumer and K. Shabunov, “Near-optimum decoding for subcodes of Reed-Muller codes,” in *Proc. 2001 IEEE Int. Symp. Inform. Theory*, (Washington, D.C., USA), p. 329, 2001.
- [33] M. Burnashev and I. Dumer, “Error exponents for recursive decoding of Reed-Muller codes on a binary-symmetric channel,” *IEEE Trans. Inform. Theory*, vol. 52, pp. 4880–4891, Nov. 2006.
- [34] M. Burnashev and I. Dumer, “Error exponents for two soft decision decoding algorithms of Reed-Muller codes,” in *Proc. 2008 IEEE Int. Symp. Inform. Theory*, (Toronto, Canada), pp. 71–75, July 2008.
- [35] M. Burnashev and I. Dumer, “Error exponents for two soft-decision decoding algorithms of Reed-Muller codes,” *IEEE Trans. Inform. Theory*, vol. IT-55, pp. 4108–4118, Sept. 2009.
- [36] R. Mori and T. Tanaka, “Performance and construction of polar codes on symmetric binary-input memoryless channels,” 2009.
- [37] T. Richardson and R. Urbanke, *Modern Coding Theory*. Cambridge University Press: Cambridge, 2008.
- [38] G. D. Forney Jr., “Codes on graphs: Normal realizations,” *IEEE Trans. Inform. Theory*, vol. IT-47, pp. 520–548, Feb. 2001.
- [39] E. Arıkan, “A performance comparison of polar codes and Reed-Muller codes,” *IEEE Comm. Letters*, vol. 12, pp. 447–449, June 2008.