Polar Code Constructions Based on LLR Evolution

Minghai Qin, Jing Guo, Aman Bhatia, Albert Guillén i Fàbregas, *Senior Member, IEEE*, and Paul H. Siegel, *Fellow, IEEE*

***Abstract*— Polar code constructions based on mutual information or Bhattacharyya parameters of bit-channels are intended for hard-output successive cancellation (SC) decoders, and thus might not be well designed for use with other decoders, such as soft-output belief propagation (BP) decoders or successive cancellation list (SCL) decoders. In this letter, we use the evolution of messages, i.e., log-likelihood ratios, of unfrozen bits during iterative BP decoding of polar codes to identify weak bit-channels, and then modify the conventional polar code construction by swapping these bit-channels with strong frozen bit-channels. The modified codes show improved performance not only under BP decoding, but also under SCL decoding. The code modification is shown to reduce the number of low-weight codewords, with and without CRC concatenation.**

***Index Terms*— Polar codes, belief propagation decoding, successive cancellation list decoding.**

I. Introduction

Polar codes were proposed in [2] as a coding technique that provably achieves the symmetric capacity of binary-input discrete memoryless channels (B-DMCs) with low encoding and decoding complexity. The analysis and construction of polar codes is based on a successive cancellation (SC) decoder. The effective channels seen by the SC decoder when making decisions are called bit-channels. As the code length tends to infinity, the bit-channels become either noiseless or completely noisy and the fraction of noiseless channels tends to the symmetric capacity. The symmetric capacity is achieved by transmitting information through the noiseless bit-channels. However, the performance of moderate-length polar codes suffers from the sub-optimality of SC decoding and imperfectly polarized bit-channels.

Several decoders with better finite-length performance than SC have been proposed. In [12], successive cancellation list (SCL) decoding was proposed, yielding performance comparable to maximum-likelihood (ML) decoding at high SNR. Belief propagation (BP) decoding over the polar code factor graph was also proposed, with parallel [1] and sequential [6] message scheduling.

In addition to the aforementioned improved decoders, modified constructions of polar codes have been considered. In particular, [4] reports a near-exponential rate of decay of the error probability using a concatenation with outer Reed-Solomon codes. A concatenated code employing an outer polar code and inner block codes is proposed in [11]. In [10], an interpolated construction that relates polar codes to Reed-Muller codes is proposed. In [7], BP decoding with a concatenation of LDPC codes to protect intermediate bit-channels is proposed. Interleaved concatenations with BCH and convolutional codes are proposed in [14].

In this letter, a simple modification to the conventional polar code construction method is proposed to improve code performance under BP decoding. By tracking the evolution of the LLR densities of unfrozen bits during iterative BP decoding, we identify weak unfrozen bit-channels. These are then replaced by strong frozen bit-channels for information transmission. This LLR-based bit-swapping construction yields performance improvements under BP decoding. Gains are also observed under SCL decoding and CRC-aided SCL decoding, due to a reduction in the number of low-weight codewords.

II. Preliminaries

We define $[b] \overset{\text{def}}{=} \{1, \ldots, b\}$ for $b \in \mathbb{Z}$. We use x_1^b to denote a length-b vector $(x_1, \ldots, x_b)$ and $\boldsymbol{A}^\top$ to denote the transpose of matrix $\boldsymbol{A}$. Row vectors are assumed.

Let $W : \mathcal{X} \to \mathcal{Y}$ denote a B-DMC, with input alphabet $\mathcal{X} = \{0, 1\}$, output alphabet $\mathcal{Y}$, and transition probability $W(y|x), x \in \mathcal{X}, y \in \mathcal{Y}$. The channel mutual information with equiprobable inputs, or symmetric capacity, is defined by

$$I(W) = \sum_{y \in \mathcal{Y}} \sum_{x \in \mathcal{X}} \frac{1}{2} W(y|x) \log \frac{W(y|x)}{\frac{1}{2}W(y|0) + \frac{1}{2}W(y|1)}, \quad (1)$$

and the corresponding Bhattacharyya parameter by

$$Z(W) = \sum_{y \in \mathcal{Y}} \sqrt{W(y|0)W(y|1)}. \quad (2)$$

Let N be the block length. Channel input and output sequences are denoted by x_1^N and y_1^N, respectively, with corresponding vector channel $W^N\left(y_1^N|x_1^N\right)$.

A. Channel Polarization

Consider the matrix $\boldsymbol{G}_2 = \left[\begin{smallmatrix} 1 & 0 \\ 1 & 1 \end{smallmatrix}\right]$, and let $\boldsymbol{G}_N = \boldsymbol{G}_2^{\otimes n}$ be the n-th Kronecker power of $\boldsymbol{G}_2$, where $n = \log_2 N$. Input bits are denoted by $u_1^N \in \{0, 1\}^N$. We define $W_N\left(y_1^N|u_1^N\right) = W^N\left(y_1^N|u_1^N \boldsymbol{G}_N\right)$ as the induced vector channel from the input bits. From $W^N\left(y_1^N|u_1^N\right)$, an SC decoder implicitly defines, for

Manuscript received December 3, 2016; accepted January 5, 2017. Date of publication January 20, 2017; date of current version June 8, 2017. This work was funded by National Science Foundation Grant CCF-1405119, European Research Council Grant 259663, and Spanish Ministry of Economy and Competitiveness Grant TEC2012-38800-C03-03. The associate editor coordinating the review of this letter and approving it for publication was C. Jego.

M. Qin is with WDC Research, Milpitas, CA 95035 USA (e-mail: qinminghai@gmail.com).

J. Guo is with the Department of Engineering, University of Cambridge, Cambridge CB2 1TN, U.K. (e-mail: jg582@cam.ac.uk).

A. Bhatia is with SK Hynix Memory Solutions, San Jose, CA 95134 USA (e-mail: aman.bhatia@outlook.com).

A. Guillén i Fàbregas is with the Department of Information and Communication Technologies, Universitat Pompeu Fabra, 08018 Barcelona, Spain, also with the Institució Catalana de Recerca i Estudis Avançats, 08010 Barcelona, Spain, and also with the Department of Engineering, University of Cambridge, Cambridge CB2 1PZ, U.K. (e-mail: guillen@ieee.org).

P. H. Siegel is with the Department of Electrical and Computer Engineering, University of California, San Diego, La Jolla, CA 92093 USA, and also with the Center for Memory and Recording Research, University of California, San Diego, La Jolla, CA 92093 USA (e-mail: psiegel@ucsd.edu).

Digital Object Identifier 10.1109/LCOMM.2017.2656126

$i \in [N]$, the bit-channel

$$W_N^{(i)}\left(y_1^N, u_1^{i-1}|u_i\right) = \sum_{u_{i+1}^N} \frac{1}{2^{N-i}} W_N\left(y_1^N|u_1^N\right). \qquad (3)$$

The channel polarization theorem [2] states that $I\left(W_N^{(i)}\right)$ converges to either 0 or 1 as N tends to infinity and the fraction of noiseless channel tends to $I(W)$.

Rate $R = K/N$ polar codes are constructed by selecting the $N-K$ indexes with the lowest $I\left(W_N^{(i)}\right)$ or the highest $Z\left(W_N^{(i)}\right)$, for $i \in [N]$. These are called the *frozen* set, $\mathcal{F}$, and the corresponding input bits are set to zero. The complementary *unfrozen* set of K indexes, $\mathcal{F}^c$, correspond to information bits.

It is impractical to precisely calculate $I\left(W_N^{(i)}\right)$ or $Z\left(W_N^{(i)}\right)$ since the output alphabet size of $W_N^{(i)}$ grows exponentially with N. However, a quantization can be used to closely approximate $W_N^{(i)}$ [13].

B. Successive Cancellation Decoding

In SC decoding, the information bits are estimated as

$$\widehat{u}_i = \arg\max_{u_i \in \{0,1\}} W_N^{(i)}\left(y_1^N, u_1^{i-1}|u_i\right), \quad i \in \mathcal{F}^c.$$

The time complexity of the SC decoder is $O(N \log N)$. A significant performance improvement is achieved by successive cancellation list (SCL) decoding [12], which conducts a breadth-first search of the polar code decoding tree [12] over L candidate paths with complexity $O(LN \log N)$.

C. Belief Propagation (BP) Decoding

BP is a message-passing algorithm that has been extensively studied for decoding graph-based codes. BP decoding of polar codes has been considered in [1], [5], [8] and it was shown that the decoding complexity is $O(I_{\text{ave}} N \log N)$, where I_{ave} is the average number of iterations.

Scheduling, i.e., choosing the order in which nodes compute their output messages, plays a key role in the performance and complexity of BP decoders [8]. The two main decoding schedules for polar decoders are called "flooding" [7] and soft-cancellation (SCAN) decoding [6]. SCAN has lower complexity than flooding; it uses a schedule similar to SC decoding, but yields better performance.

III. LLR-Based Constructions for BP Decoding

Recall that the bit-channels are defined as the virtual channels between the input sequence to the polar encoder and the channel output sequence seen by a genie-aided SC decoder. If we use another decoder, e.g., a BP decoder or SCL decoder, the virtual channels seen by the decoder differ from the bit-channels. For such a decoder, the conventional polar code construction might not be the best approach. In this section, we propose a modified polar code construction for use with BP and SCL decoding in the finite block length regime.

A. Evolution of LLRs During BP Decoding

As in [7], we denote the soft-output LLRs at bit u_i on the jth iteration of BP decoding by $L_{\text{out}}(i, j), i \in [N]$. The performance of BP decoding is determined by the distributions of $L_{\text{out}}(i, j)$, for $i \in \mathcal{F}^c$ and $j \in [I_{\max}]$, where $I_{\max}$ is the maximum number of iterations. We approximated these distributions by observing the LLR values during 5000 transmissions of the all-zero codeword, and recorded the evolution of the mean LLR values during BP decoding as a function of the iteration number. The sum-product algorithm was used for variable node (VN) and check node (CN) updates. The input LLR to the CN updating function was limited by a threshold value of 38 to avoid overflow of the hyperbolic tangent function [3].

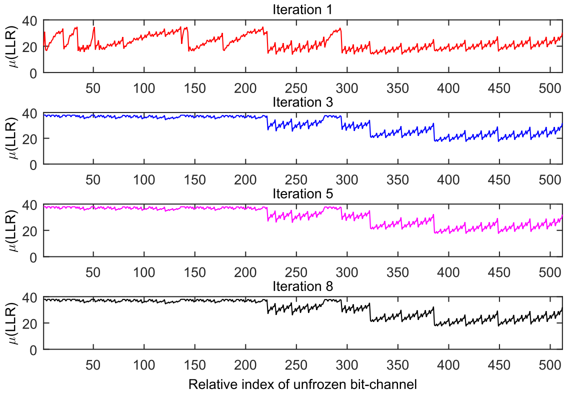

Fig. 1. Mean LLR values (μ(LLR)) under BP SCAN decoding of unfrozen bits of polar code with $N = 1024$, $R = \frac{1}{2}$, on AWGN channel, with $\frac{E_b}{N_0} = 2.25$ dB. Code optimized at $\frac{E_b}{N_0} = 2.25$ dB according to [13].

Fig. 1 shows mean LLR plots after iterations $j = 1, 3, 5, 8$ of BP SCAN decoding for a conventional polar code with length $N = 1024$ and rate $R = 0.5$, optimized at $\frac{E_b}{N_0} = 2.25$ dB. Note that each plot shows mean LLRs for the 512 unfrozen bits, displayed in increasing relative index order. The $L_{\text{out}}(i, j)$ distributions, and their mean values, were observed to stabilize after $j = 8$ iterations. The bottom subfigure of Fig. 1, corresponding to $j = 8$, shows that, at certain bit indexes, there is a substantial drop in the mean LLR value, followed by a period of small oscillations as the index increases.

B. LLR-Based Bit-Swapping Construction

In the simulations above, since the all-zero codeword was transmitted, correct decoding is likely to occur when $L_{\text{out}}(i, 8) > 0$. Thus, if the variances of LLR distributions $L_{\text{out}}(i, 8)$ are fixed for $i \in \mathcal{F}^c$, one might expect that a large drop in the mean LLR plot, followed by small oscillations, would correspond to a more probable unfrozen bit error that also propagates to subsequent bits.

With this interpretation, the patterns observed in Fig. 1 motivate a modified polar code construction in which a set of unfrozen bit-channels associated with the most severe mean LLR drops are replaced by an equal number of the most reliable frozen bit-channels. This procedure was implemented by swapping 12 bit-channels, the optimal number of swaps having been determined empirically.

Fig. 2 shows the mean LLR plots after iterations $j = 1, 3, 5, 8$ for the code obtained. The qualitative difference between these plots and those in Fig. 1 is evident, with fewer occurrences of large drops followed by small oscillations.

We refer to this method of code modification via bit-swapping as LLR-based code construction. In the next section, we show that it can improve the polar code performance under BP and SCL decoding.

Examination of the evolution of mean LLR values of unfrozen bits under BP decoding with a flooding schedule

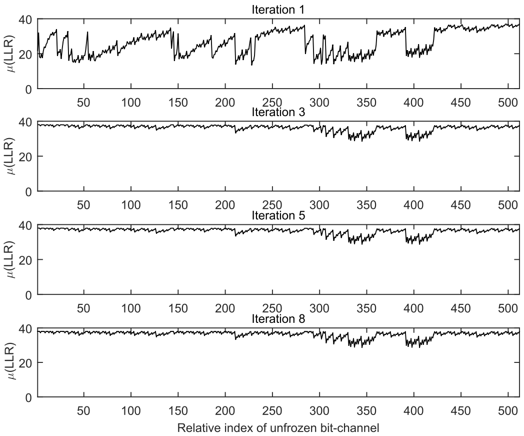

Fig. 2. Mean LLR values (μ(LLR)) under BP SCAN decoding of unfrozen bits of polar code in Fig. 1 after LLR-based bit-swapping of 12 bit-channels, on AWGN channel, $\frac{E_b}{N_0} = 2.25$ dB.

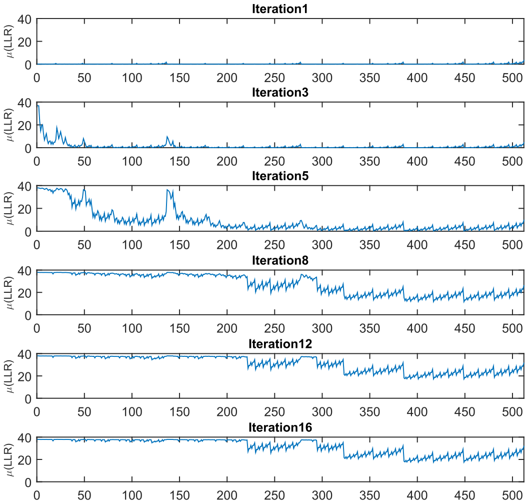

Fig. 3. Mean LLRs (μ(LLR)) under BP flooding decoding of unfrozen bits of polar code in Fig. 1.

reveals behavior similar to that found under SCAN decoding. Fig. 3 shows the evolution of mean LLRs using traces captured after $j = 1, 3, 5, 8, 12, 16$ decoder iterations. Comparison to Fig. 1 leads to two observations. First, the indexes where the major drops occur at iteration $j = 8$ and beyond are the same in both. This suggests that LLR-based bit-swapping will have a similar effect on LLR evolution with a flooding decoder, as was found to be the case. Second, the mean LLR plots for SCAN decoding appear to stabilize after fewer iterations than for flooding decoding, as can be seen by comparing the plots for iterations $j = 5$ and $j = 8$ in both figures. This suggests that SCAN decoding should converge faster that flooding decoding, requiring fewer iterations to decode. This is consistent with the measurements of the average number of iterations to reach convergence of SCAN and flooding decoders reported in [7]. The faster convergence of SCAN decoding translates to reduced time complexity to decode, and a lower error rate when the maximum allowed number of iterations is fixed [6].

C. Numerical Results

We simulated the performance of the conventional polar code and the modified code obtained by LLR-based bit-swapping, as described above, under BP SCAN-decoding on the AWGN channel. Decoding was terminated whenever the polar code parity-check equations were satisfied by the estimated codeword. The maximum allowable number of iterations was set to $I_{\max} = 200$. At high SNR values (e.g., 3 dB), we found that the average number of iterations required to decode was slightly greater than 1, which is consistent with results in [7]. Fig. 4 shows the frame error rate (FER) performance of the two codes in the [1.75, 3.25] dB SNR range. As can be seen, the LLR-based construction provides a gain of approximately 0.2 dB throughout almost the entire range.

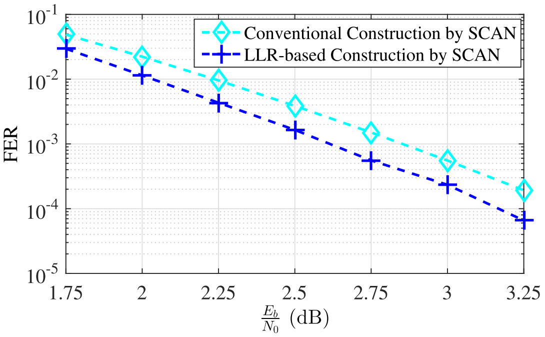

Fig. 4. FER of conventional and LLR-based polar codes under BP SCAN decoding on the AWGN channel.

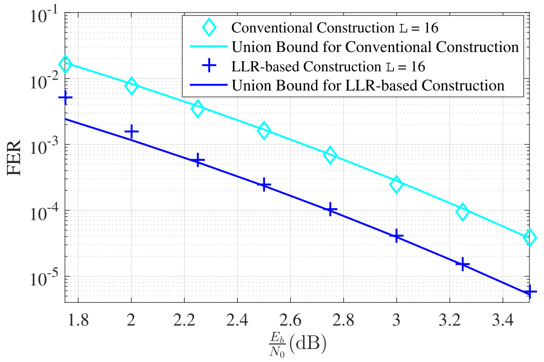

Fig. 5. Union bound approximation (5) and FER for SCL decoding.

IV. LLR-Based Construction With SCL Decoding

In the previous section, we showed that the LLR-based bit-swapping construction outperforms the conventional construction under iterative soft-output BP decoding. In this section, we demonstrate that the LLR-based construction technique can improve the performance of polar codes under hard-output SCL decoding, as well.

Fig. 5 compares the FER performance of the length $N = 1024$, rate $R = 0.5$ polar codes constructed using the conventional method and the LLR-based bit-swapping approach on the AWGN channel under SCL decoding with list size $L = 16$. Note that the improvement over BP SCAN decoding is approximately 0.12 dB for the conventional construction and 0.38 dB for the LLR-based design. Under SCL decoding, the LLR-based code provides a nearly 0.5 dB gain over the conventional code in almost the entire [1.75, 3.5] dB SNR range.

It has been observed that SCL decoder performance can approach that of ML decoding for sufficiently large list size. For a polar code of length $N = 2048$ and rate $R = 0.5$, constructed using the methods of [13] and optimized for an SNR value of 2 dB, a list size $L = 16$ was found to yield near-ML performance in the [1.5, 3] dB SNR range [12]. Due to complexity constraints, it is not possible to simulate ML decoding directly for the codes we have constructed, so instead we compare our results to an approximate bound on ML decoder performance.

For a linear block code transmitted over the AWGN channel, the union bound on FER with ML decoding is given by

$$P_e^{\mathrm{ML}} \leqslant \sum_d A_d Q(\sqrt{2d\mathrm{SNR}}), \tag{4}$$

where A_d is the number of codewords with Hamming weight d, and $Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{+\infty} \exp(-\frac{u^2}{2}) du$. In the high SNR regime, the upper bound is dominated by the term corresponding to the minimum non-zero codeword weight, i.e., the minimum distance of the code $d_{\min}$, and we have the familiar approximation:

$$P_e^{\mathrm{ML}} \approx A_{d_{\min}} Q(\sqrt{2d_{\min}\mathrm{SNR}}). \tag{5}$$

There is no closed-form expression for the weight enumerators A_d of polar codes. However, a method based upon adaptive SCL decoding, proposed in [9], can be used to identify low-weight codewords in polar codes. Evidence suggests that it finds all of the minimum weight codewords for sufficiently large list size. Applying the technique to the conventional and LLR-based polar codes, we found that both codes have minimum distance $d_{\min} = 16$, with corresponding weight enumerators $A_{16} = 34997$ and $A_{16} = 4896$, respectively. The approximate union bound in (5) was evaluated for both codes using these values of A_{16}, and the results are shown in Fig. 5. There is a very good match between the SCL decoder simulation results and the approximate ML bound. This suggests that most of the SCL decoding errors are caused by misdecoding to the nearest codewords in Hamming distance. The comparison also provides an explanation for the improved performance of the LLR-based code: the bit-swapping construction reduced the number of minimum-weight codewords substantially, by a factor of more than 7.

In [12], it was shown that the concatenation of a polar code with a simple CRC could significantly improve the performance under SCL decoding at the price of a small rate loss. We concatenated the conventional and LLR-based polar codes with an 8-bit CRC, and evaluated their FER performance on the AWGN channel using SCL decoding with list size $L = 16$ in the [1.75, 2.75] dB SNR range. The results are shown in Fig. 6. The LLR-based code provides better performance for SNRs greater than 2 dB. Using the same technique as above, we also determined the minimum distance of both CRC-aided codes to be 16, with corresponding weight enumerators $A_{16} = 2237$ and $A_{16} = 56$, respectively. Once again, the LLR-based code has significantly fewer minimum-weight codewords. The approximate ML decoding bounds given by (5) are also plotted in Fig. 6. The agreement between the bounds and simulation results is not as good as in Fig. 5, possibly because the operating SNR is too low. However, the results suggest that the improved performance of the LLR-based polar code is again attributable to the significant reduction in the number of minimum-weight codewords.

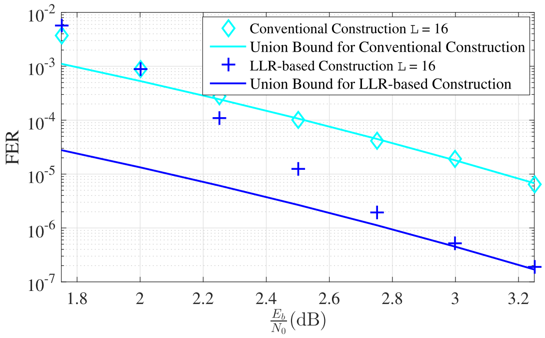

Fig. 6. Union bound approximation (5) and FER for CRC-aided SCL decoding.

V. Conclusions

In this letter, we proposed a modification of the conventional polar code construction for use with BP and SCL decoders. The evolution of unfrozen bit LLR distributions during iterative BP decoding was used to identify the most vulnerable unfrozen bit-channels. These were replaced with the same number of most reliable frozen bit-channels. This LLR-based bit swapping construction improves code performance under BP decoding, as well as under SCL decoding, with and without CRC concatenation. The gains can be attributed to a significant reduction in the number of low-weight codewords.

References

[1] E. Arıkan, "A performance comparison of polar codes and Reed-Muller codes," *IEEE Commun. Lett.*, vol. 12, no. 6, pp. 447–449, Jun. 2008.

[2] E. Arıkan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Trans. Inf. Theory*, vol. 55, no. 7, pp. 3051–3073, Jul. 2009.

[3] B. K. Butler and P. H. Siegel, "Numerical issues affecting LDPC error floors," in *Proc. GLOBECOM*, Anaheim, CA, USA, Dec. 2012, pp. 3201–3207.

[4] M. Bakshi, S. Jaggi, and M. Effros, "Concatenated polar codes," in *Proc. IEEE Int. Symp. Inf. Theory*, Jun. 2010, pp. 918–922.

[5] A. Eslami and H. Pishro-Nik, "On bit error rate performance of polar codes in finite regime," in *Proc. 48th Annu. Allerton Conf. Commun., Control Comput.*, Monticello, IL, USA, Sep. 2010, pp. 188–194.

[6] U. U. Fayyaz and J. R. Barry, "Polar codes for partial response channels," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2013, pp. 4337–4341.

[7] J. Guo, M. Qin, A. Guillén i Fàbregas, and P. H. Siegel, "Enhanced belief propagation decoding of polar codes through concatenation," in *Proc. IEEE Int. Symp. Inf. Theory*, Jun. 2007, pp. 1166–1170.

[8] N. Hussami, S. B. Korada, and R. Urbanke, "Performance of polar codes for channel and source coding," in *Proc. IEEE Int. Symp. Inf. Theory*, Seoul, South Korea, Jun. 2009, pp. 1488–1492.

[9] B. Li, H. Shen, and D. Tse, "An adaptive successive cancellation list decoder for polar codes with cyclic redundancy check," *IEEE Commun. Lett.*, vol. 16, no. 12, pp. 2044–2047, Dec. 2012.

[10] M. Mondelli, S. H. Hassani, and R. L. Urbanke, "From polar to Reed-Muller codes: A technique to improve the finite-length performance," in *Proc. IEEE Int. Symp. Inf. Theory*, Honolulu, HI, USA, Jun./Jul. 2014, pp. 131–135.

[11] M. Seidl and J. B. Huber, "Improving successive cancellation decoding of polar codes by usage of inner block codes," in *Proc. 6th Int. Symp. Turbo Codes Iterative Inf. Process.*, Brest, France, Sep. 2010, pp. 103–106.

[12] I. Tal and A. Vardy, "List decoding of polar codes," in *Proc. IEEE Int. Symp. Inf. Theory*, St. Petersburg, Russia, Jul./Aug. 2011, pp. 1–5.

[13] I. Tal and A. Vardy, "How to construct polar codes," *IEEE Trans. Inf. Theory*, vol. 59, no. 10, pp. 6562–6582, Oct. 2013.

[14] Y. Wang, K. R. Narayanan, and Y.-C. Huang, "Interleaved concatenations of polar codes with BCH and convolutional codes," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 2, pp. 267–277, Feb. 2016.