A New Multiple Folded Successive Cancellation Decoder for Polar Codes

Harish Vangala, Emanuele Viterbo, and Yi Hong,
Dept. of Electrical and Computer Systems Eng.,
Monash University, Melbourne, VIC 3800, Australia.
Email: {harish.vangala, emanuele.viterbo, yi.hong}@monash.edu

***Abstract*—We consider a new variant of successive cancellation decoder (SCD) for polar codes based on the concept of *folding*, which was proposed in [1], [2] as technique to reduce the decoding latency at the cost of a higher computational complexity. In this paper, we first formally define the multiple folding operation (iterated κ times), which decomposes the original encoding graph into a number of smaller polar encoding graphs. More specifically, we show that the multiple folding gives rise to a two stage interpretation of the graph representing the polar encoder and the SCD. Based on this, we propose the improved multiple folded successive cancellation decoder (IMFSCD), which combines SCD in one stage and maximum-likelihood decoding in the other. This decoder exhibits a latency gain by a factor of 2^{κ}, still retaining a complexity close to the classic SCD. The small increase in complexity is due to a short maximum likelihood decoder (MLD) used in place of a SCD in the last decoding stage within the IMFSCD. Moreover, we observe by simulation that the decoder exhibits a significant performance gain at high rates and longer codes.**

***Keywords*—Successive cancellation decoder, multiple folded successive cancellation decoder, partial ML decoding of polar codes, low latency decoder, two stage polar encoder, two stage polar decoder, polar code concatenation.**

I. Introduction

Polar codes proposed by Erdal Arikan [3] have attracted significant research interest as the first provably capacity achieving family of codes with low complexity encoding and decoding of the order of $O(N \log N)$, where N is the code-length. The original successive cancellation decoder (SCD) for polar codes proposed by Arikan is an important element in proving the capacity theorems for polar codes. Even in practice, the SCD exhibits several useful properties such as fixed, deterministic complexity and good error performance. Hence, polar codes are becoming attractive for practical implementation. Indeed, several implementations are reported, with large code-lengths up to 2^{17} [4]–[7].

Two well-known problems with SCD are the decoding delay of the order of $(2N-1)$ units [3], [5] and worse performance when compared with the best available LDPC codes of the same length [5], [8], [9].

Many alternative decoders are available [10]–[12] for polar codes improving performance relative to SCD, but such gains are achieved at the cost of higher complexity and/or higher delay. Driven by this fact, considerable research has focused on improving the error performance and latency of the SCD, without much increase in complexity [1], [4]–[9], [13]–[18].

This work was supported by NPRP grant #NPRP5-597-2-241 from the Qatar National Research Fund (a member of Qatar Foundation).

Folding is a technique introduced in [12] to provide for the first time, a Maximum Likelihood (ML) decoding algorithm for polar codes with code-lengths up to 256. In [1], [2], the *folding* is used to reduce the latency of the SCD algorithm at the cost of a much higher complexity. The complexity increases by a factor $2^{2^{\kappa}}$ in their best implementation, where κ represents the number of foldings (see Sec. III). Also, the performance is reported to be the same as that of an SCD, while we identify a possible gain in performance. A first attempt to reduce complexity, while retaining all the advantages of *multiple folding*, was made in [19].

In the later stages of our work on folding, we found interesting similarities with the concatenated polar codes in [20], when using the new interpretation of folding, provided in this paper. However, our work significantly differs in both our main objective and approach. Specifically, we aim to achieve highly desired gains in delay (upto 87%, at $\kappa = 3$) using very short length ML decoding (length 2^{κ}, for small κ) of a part of the polar code. In contrast, [20] attempts to achieve gains in performance, using suboptimal, high complexity and high latency decoding algorithms for different outer codes with longer length. It is worth noting that such a performance gain is easily attainable using much smaller complexity decoding algorithms such as CRC-aided list decoding [8], without making significant changes to the original polar code.

In the current paper, we revisit the folding technique for application to SCD and propose an improved formulation. We show that by folding the graph κ times ($1 \leq \kappa \leq \log_2(N)-1$), the polar encoder can be visualized as two concatenated shorter length polar encoding stages. Following this new interpretation, we use a maximum-likelihood decoder (MLD) in one stage, while the other stage uses the standard SCD. This results in a new decoder with a much smaller complexity of the one given in [2]. We refer to this as the *improved multiple folded successive cancellation decoder* (IMFSCD). Similar to [2], we observe that the IMFSCD exhibits latency reduction by a factor of 2^{κ} over the conventional SCD. Using our low complexity decoder implementation, we are able to extend the analysis to longer codes and we find that the performance gain can be significant for rates above half.

Below is a summary of our contributions in this paper:

- We provide a new analytical formulation of the multiple folding technique, allowing a range of new decoders.
- We show that the classic polar encoder can be implemented with same complexity in two independent stages.

978-1-4799-5999-0/14/$31.00 ©2014 IEEE

- We show that a range of new decoders can be proposed by choosing different pairs of decoders for the two stages.
- We show that the processing of q-ary symbol's reliability proposed in [1], [2] is not necessary in the current framework. This is the key to a significant reduction in complexity.
- We show that, in addition to the significant latency gain up to a factor of 2^κ, the IMFSCD has a significant performance gain at longer code-lengths and high rates.

The rest of the paper is organized as follows. Sec. II introduces the notation and briefly describes the polar codes. In Sec. III, we discuss the main IMFSCD algorithm in detail along with our simulations and we finally conclude in Sec. IV.

II. Polar codes

Notation — An index notation used to define the encoding operation of polar codes is given as follows. Given any subset of indices $\mathcal{I}$ from a vector $\mathbf{x}$, we denote the corresponding sub-vector as $\mathbf{x}_\mathcal{I}$. Given any subset of column indices $\mathcal{I}$ from a matrix $\mathbf{A}$, we denote the corresponding sub-matrix by $\mathbf{A}_\mathcal{I}$.

The following notation will used in defining *multiple folding* with *folding-length* κ in Sec. III. Given a vector $\mathbf{v} = [v_1 v_2 \dots v_L]^T$ and integers l and d such that $l = L/d$, we define the *interleaved sub-vectors* with *separation* d and *length* l, as the set of sub-vectors of $\mathbf{v}$, of the form $\mathbf{v}_j \triangleq [v_j, v_{j+d}, \dots, v_{j+(l-1)d}]^T,\ j = 1, 2, \dots d$. Furthermore, if we form a new vector by stacking all the sub-vectors as $\mathbf{v}' \triangleq [\mathbf{v}_1^T \mathbf{v}_2^T \dots \mathbf{v}_d^T]^T$, the result is a *block-interleaver* (or *rectangular interleaver*) permutation of the vector $\mathbf{v}$ with an interleaver depth d. Denoting by $\mathbf{P}$, such permutation matrix, we have $\mathbf{v}' = \mathbf{P}\mathbf{v}$.

Polar Coding — A *polar code* is completely specified by the three-tuple $(N, K, \mathcal{F})$, where N is the *code length* in bits, K is the number of information bits encoded per codeword (or *code dimension*), and $\mathcal{F}$ is a subset of $N - K$ indices from $\{0, 1, \dots, N-1\}$ (*frozen bit locations*).

For a $(N, K, \mathcal{F})$ polar code we describe below the encoding operation for a vector of information bits $\mathbf{u}$ of length K. Let $n \triangleq \log_2(N)$ and $\mathbf{G} \triangleq \mathbf{F}^{\otimes n} = \mathbf{F} \otimes \dots \otimes \mathbf{F}$ be the n-fold Kronecker product of $\mathbf{F} \triangleq \left[\begin{smallmatrix} 1 & 1 \\ 0 & 1 \end{smallmatrix}\right]$.

Then, a codeword is generated as

$$\mathbf{x} = \mathbf{G}_{\mathcal{F}^c}\ \mathbf{u}, \tag{1}$$

where $\mathcal{F}^c \triangleq \{0, 1, \dots, N-1\} \backslash \mathcal{F}$ corresponds to the non-frozen bit indices. Alternatively,

$$\mathbf{x} = \mathbf{G}\ \mathbf{d} = \mathbf{F}^{\otimes n}\mathbf{d} \tag{2}$$

where $\mathbf{d} \in \{0, 1\}^N$ such that $\mathbf{d}_\mathcal{F} = 0$ and $\mathbf{d}_{\mathcal{F}^c} = \mathbf{u}$.

Note that the *frozen bits* $\mathbf{d}_\mathcal{F}$ from Arikan's original formulation [3] are set to zeros. Arikan proposed the efficient implementation of the encoding equation (2) shown in Fig. 1.

Construction of Polar Codes — The choice of the set $\mathcal{F}$ is an important step in polar coding often referred to as *polar code construction*. A significant amount of literature is devoted to this operation [3], [20]–[24]. The original algorithm, proposed in [21] and improved in [24], is based on the Bhattacharya bound approximation. Later proposed algorithms improve on this approximation at the cost of higher complexity. For simplicity, we use the original polar code construction algorithm.

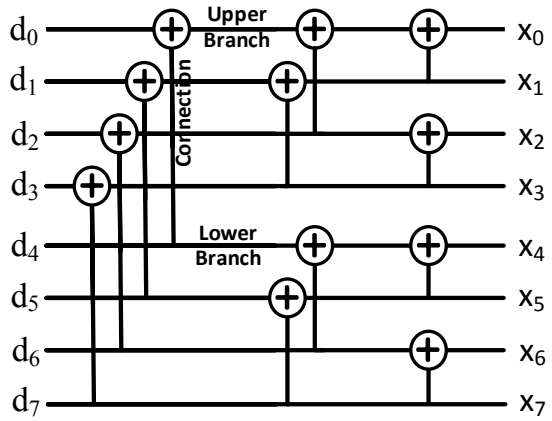

Fig. 1. Arikan's $O(N \log_2 N)$ complexity encoder to implement (2) at $N = 8$

Successive Cancellation Decoder (SCD) — The SCD algorithm [3] essentially follows the same encoder diagram in Fig.1. The likelihoods evolve in the reverse direction from right-to-left, as explained in [3]. A complete and detailed implementation of SCD is available in [14].

III. Improved Multiple Folded Successive Cancellation Decoding

The folded successive cancellation decoding is based on the ideas introduced in [1], [2], [12] exploiting the recursive structure in the encoding equation and its implementation graph. A formal definition of the technique is provided in the following.

Folding Operation at the Encoder – Consider below the encoding equation rewritten using the well-known Kronecker product property $\mathbf{AB} \otimes \mathbf{CD} = (\mathbf{A} \otimes \mathbf{C}) \cdot (\mathbf{B} \otimes \mathbf{D})$,

$$\mathbf{x} = \mathbf{F}^{\otimes n}\mathbf{d} \tag{3}$$

$$= \left(\mathbf{F}^{\otimes \kappa} \otimes \mathbf{F}^{\otimes n-\kappa}\right)\ \mathbf{d} \tag{4}$$

$$= \left(\mathbf{I}_{2^\kappa} \cdot \mathbf{F}^{\otimes \kappa} \otimes \mathbf{F}^{\otimes n-\kappa} \cdot \mathbf{I}_{2^{n-\kappa}}\right)\ \mathbf{d} \tag{5}$$

$$= (\mathbf{I}_{2^\kappa} \otimes \mathbf{F}^{\otimes n-\kappa}) \cdot \underbrace{(\mathbf{F}^{\otimes \kappa} \otimes \mathbf{I}_{2^{n-\kappa}})\ \mathbf{d}}_{\mathbf{v}} \tag{6}$$

$$= \begin{bmatrix} \mathbf{F}^{\otimes n-\kappa} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \mathbf{F}^{\otimes n-\kappa} \end{bmatrix} \begin{bmatrix} \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_{2^\kappa} \end{bmatrix} = \begin{bmatrix} \mathbf{F}^{\otimes n-\kappa} \cdot \mathbf{v}_1 \\ \vdots \\ \mathbf{F}^{\otimes n-\kappa} \cdot \mathbf{v}_{2^\kappa} \end{bmatrix} \tag{7}$$

$$\text{where, } \mathbf{v} = \begin{bmatrix} \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_{2^\kappa} \end{bmatrix} \triangleq (\mathbf{F}^{\otimes \kappa} \otimes \mathbf{I}_{2^{n-\kappa}})\, \mathbf{d}, \tag{8}$$

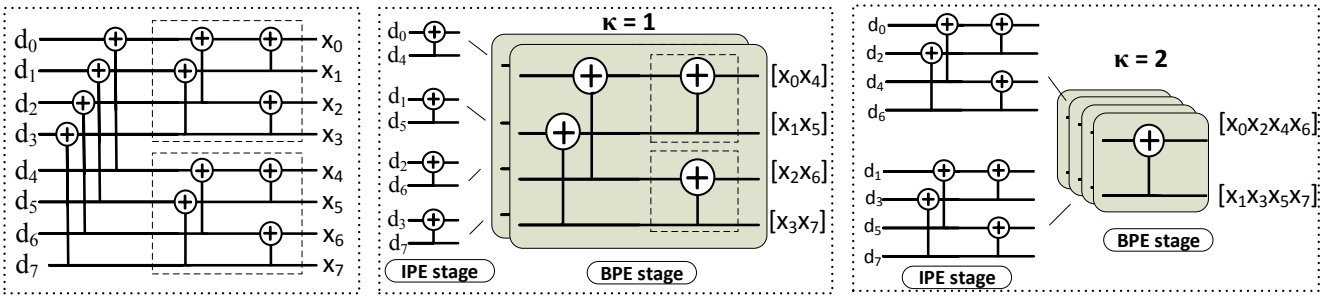

Fig. 2. Illustrating folding for $N = 8$

$\mathbf{v}_1, \cdots, \mathbf{v}_{2^\kappa}$, are the sub-vectors of $\mathbf{v}$ of length $2^{n-\kappa}$ or $N/2^\kappa$.

We may infer from (7) that given any $\kappa \leq n-1$, the encoding equation for code-length N can be visualized as a collection of several independent polar encodings of shorter code-length $N/2^\kappa$ on the partitions of vector $\mathbf{v}$. Therefore we denote this encoding stage as *block-wise polar encoding* (BPE) stage. However, we need a pre-processing step (8) on the information vector $\mathbf{d}$ to obtain $\mathbf{v}$. Interestingly, the pre-processing operation (8) can also be implemented by another set of independent polar encoders of code-length 2^κ, as shown below. We call this pre-processing stage as *interleaved polar encoding* (IPE) stage. Clearly, the two stages give rise to the original polar encoding of code-length N.

Recall the following property of the Kronecker product of square matrices $\mathbf{A}$ and $\mathbf{B}$, with possibly different dimensions:

$$\mathbf{P}^T(\mathbf{A} \otimes \mathbf{B})\mathbf{P} = (\mathbf{B} \otimes \mathbf{A}) \tag{9}$$

where $\mathbf{P}$ is a block-interleaver permutation matrix of rows, with depth equal to the dimension of square matrix $\mathbf{A}$. This can be easily verified by writing the expansions of Kronecker products $(\mathbf{A} \otimes \mathbf{B})$ and $(\mathbf{B} \otimes \mathbf{A})$ and then identifying the permutation involved to match them.

Using (9), we can rewrite (8) as follows.

$$\mathbf{v} = (\mathbf{F}^{\otimes\kappa} \otimes \mathbf{I}_{2^{n-\kappa}})\,\mathbf{d} \tag{10}$$

$$= \mathbf{P}^T(\mathbf{I}_{2^{n-\kappa}} \otimes \mathbf{F}^{\otimes\kappa})\mathbf{P}\,\mathbf{d} \tag{11}$$

$$= \mathbf{P}^T \begin{bmatrix} \mathbf{F}^{\otimes\kappa} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \mathbf{F}^{\otimes\kappa} \end{bmatrix} \begin{bmatrix} \mathbf{d}_1 \\ \vdots \\ \mathbf{d}_{2^{n-\kappa}} \end{bmatrix} \tag{12}$$

where $\mathbf{P}$ is the permutation matrix corresponding to a block-interleaver of depth $N/2^\kappa$. From (12), it is evident that we can implement the IPE stage by applying $2^{n-\kappa}$ parallel polar encoders to the interleaved sub-vectors of $\mathbf{d}$ of length 2^κ and then de-interleaving their output before feeding the BPE stage. Fig. 2 illustrates the folding process for $N = 8, \kappa = 1, 2$. Note that all the three implementations in Fig. 2 are equivalent polar encoders. A general implementation of the new encoder is elaborated below. Note also that, the frozen bits will be scattered to the component codes in a way that each one of them is not necessarily a polar code.

By observing Fig. 2, the BPE stage may be vectorized by grouping every 2^κ bits output from an IPE stage encoder. The BPE stage encoding can also be viewed as operating over $GF(q)$, with $q = 2^{2^\kappa}$, where the addition operator is the element-wise XOR on bit components. As a result, each layer in the BPE stage processes the sub-vectors from $\mathbf{v}$, $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{2^\kappa}$, and hence implements (7).

A simple interpretation of the multiple-folded polar encoder with folding-length κ is given as follows. Consider a rectangular interleaver with 2^κ columns holding N bits. First, we fill the interleaver memory column-wise with the elements of $\mathbf{d}$. Then, in IPE stage we encode in-place, all rows in parallel, using polar encoders of code-length 2^κ. Then, the BPE stage encodes in-place, all columns in parallel, using polar encoders of code-length $N/2^\kappa$. Finally, the memory is read column-wise to obtain the polar codeword $\mathbf{x}$. A similar interpretation will be used for the decoder, based on a rectangular memory of N real values representing the bit likelihoods.

Folding operation at the decoder — A key element in decoding a polar code is the use of the encoder graph in reverse direction. After folding, we split the encoder graph in *two* stages, which in turn are formed by different polar encoders of shorter code-lengths 2^κ and $N/2^\kappa$. With this new interpretation, we may decode *independently* each of the component codes in each stage, for example, BPE stage can be decoded with a standard successive cancellation decoder and the IPE stage can be decoded by using a MLD or a list decoder.

Following the q-ary interpretation of the BPE stage encoding, an extension of SCD was proposed for BPE stage decoding in [1], [2], on the q-ary symbols from $GF(q)$ with $q = 2^{2^\kappa}$, formed by the groups of 2^κ bits in Fig. 2. As a result, the *probability mass function* (pmf) of each q-ary symbol gets transformed (replacing likelihood transformation of binary symbols) at different component 2×2 blocks of the decoder circuit. Such a pmf transformation takes $O(2^{2^\kappa} \cdot 2^\kappa)$ complexity, in their best implementation. In this paper, we observe that this q-ary interpretation is not necessary using our new interpretation because the bit likelihoods within each q-ary symbol, corresponding to different layers in the BPE stage decoding, are independent of each other. In other words, layers of the BPE stage are independent, and hence the bits within each q-ary symbol can be processed independently and

parallelly at a complexity of $O(2^\kappa)$ only. This is a significant complexity reduction from the earlier work [1], [2] which has $O(2^{2^\kappa} \cdot 2^\kappa)$ complexity.

Choice of decoders — We propose to use a standard SCD algorithm in BPE stage parallelly at each layer, and MLD at the IPE stage. Note that, for small values of κ the IPE stage MLD deals with codes of length 2^κ and hence has negligible complexity. We name this new decoder an *improved multiple folded successive cancellation decoder* (IMFSCD) with folding length κ. Note that had we used SCD at both the stages, the result would have been exactly equal to the standard SCD algorithm of code-length N. By replacing an SCD with an MLD, we expect IMFSCD to perform at least as well as the or better than the standard SCD of code-length N. This is confirmed by our simulations for various test cases.

Similar to the encoder, a simple interpretation of the IMFSCD can be given. Consider a rectangular interleaver with 2^κ columns and $2^{n-\kappa}$ rows, holding N *real* values. The rectangular interleaver memory is filled column-wise in *natural order*, with the likelihoods computed from the channel observations. Each column is then decoded in parallel SCDs for codes of length $N/2^\kappa$. This results in a row-wise evolution of the likelihoods stored separately, in the well-known *bit-reversed* order of rows. Once we have a row of bit-likelihoods, we pass it on to an MLD for the code of length 2^κ, which gives a row of 2^κ decisions including frozen bits. The appropriate polar encoded decisions are then broadcasted parallelly just as in a standard SCD. The information bit decisions are stored separately in a similar rectangular interleaver of bits, in the same locations and finally read column-wise in the *natural order* to output the decisions.

Fig. 3 shows our simulations of polar codes performance under IMFSCD, assuming BPSK modulation and Additive White Gaussian Noise (AWGN) channel for a code-length $N = 8,192$ and rate $R = 0.8$. The code construction is taken from [21]. The plots show the comparison of an IMFSCD with $\kappa = 1, 2, 3$ versus a standard SCD. We will see in Fig. 3 that the IMFSCD can give a significant performance gain over the standard SCD for long code-lengths and high rates. However, we also found by further simulations that the gain reduces for shorter code-lengths or lower rates.

Latency gain and complexity of IMFSCD — The proposed MLD replacing the SCD for the IPE-stage, will have higher complexity compared to a standard SCD, but this increase is not significant so far as the value of κ is small e.g., $\kappa \leq 3$. Note that the previously proposed decoders [1], [2] have an much higher increase in complexity by a factor of 2^{2^κ} in their best implementation, due to the q-ary symbol processing.

The latency of the IMFSCD of length N is equal to a standard SCD of length $N/2^\kappa$, plus the latency of the MLD in the IPE stage. However, unlike SCD, MLD can be highly parallelized. Therefore, the difference in latency can be minimized, which is the key in reducing the latency $(2N - 1)$ of Arikan's SCD (see [3]) under maximum parallelized implementation. Then the latency of IMFSCD can be reduced to a value up to $(N/2^{\kappa-1} - 1)$, which is a significant reduction by a factor of 2^κ, when compared to standard SCD.

A Comparison with concatenated codes — As we noted earlier, a range of new decoders can be proposed using the above interpretation of *multiple folding*. This model closely resembles the architecture of a concatenated code with an inner code and an outer code. Interestingly, we may note that neither the outer codes nor the inner codes in multiple folding architecture are polar codes (i.e., the frozen bits are not selected according to the standard construction). In the IPE stage, this is due to frozen bits being spread randomly across the bits seen by IPE stage encoders. In the BPE stage, the locations of frozen bits are not compatible with a polar code construction.

In the light of this observation, an interesting comparison is to verify how a concatenated code performs compared to the IMFSCD polar code with the same rate and code-length. As a simple example, we use a single parity check (SPC) code at the IPE-stage and the same polar code for all component codes of the BPE-stage. This choice is motivated by the fact that SPC codes are a classic example of low-complexity ML-decodable codes, matching the choice of an IPE stage decoding in an IMFSCD. The rate of the inner polar code is adjusted so as to match the overall rate of the concatenated code.

The simulation of a SPC code of length 8, concatenated with polar code of length 1024 at overall rate $R = 0.5$ is considered in Fig. 4. We compare the performance of this concatenated code with polar codes at $N = 1024, R = 0.5$ and $N = 8096, R = 0.5$, being decoded using standard SCDs. We see that the concatenated code performs worse when compared to a classic polar code of an equal length and rate. This implies that the joint optimization performed by polar code construction is more powerful than a simple SPC code concatenation. Further comparisons are relegated to our future work.

IV. Conclusion

We have proposed a new formulation of the folding technique, by using which we have introduced a two stage concatenated interpretation of the standard polar encoder. Following this interpretation of the encoder, we have proposed a new low complexity decoder IMFSCD, with much lower latency and improved performance over the standard SCD. The latency gain can be up to a factor of 2^κ depending upon the ML stage implementation, with significantly lower complexity than the earlier decoders based on folding. The performance gains with IMFSCD can be significant for longer codes and high rates.

References

[1] S. Kahraman, E. Viterbo, and M. E. Celebi, "Folded successive cancellation decoding of polar codes," in *Australian Communications Theory Workshop (AusCTW), 15th Annual*, Sydney, Feb 2014, pp. 57–61.

[2] ——, "Multiple folding for successive cancelation decoding of polar codes," *IEEE Wireless Communication Letters*, 2014, (accepted).

[3] E. Arikan, "Channel polarization: A method for constructing capacity-

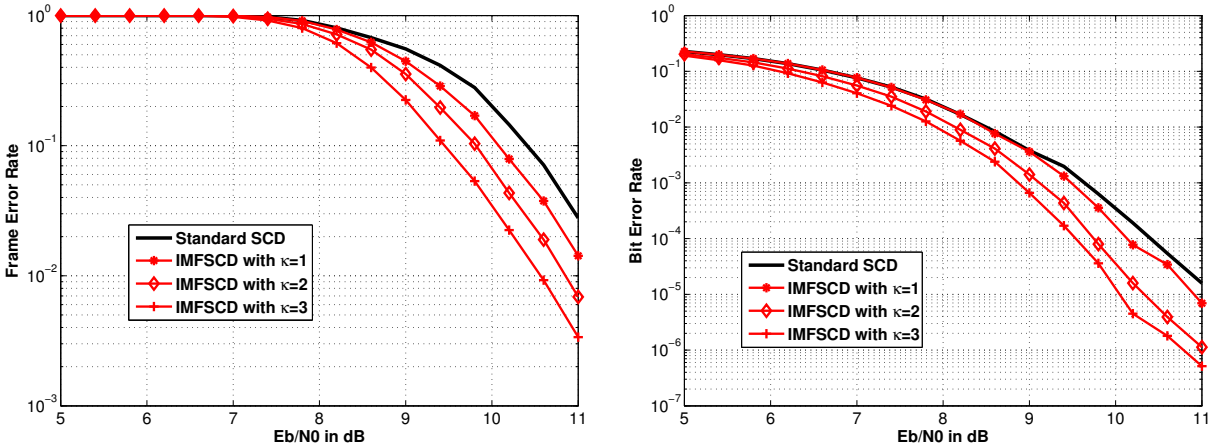

Fig. 3. Performance comparison of IMFSCD($\kappa = 1, 2, 3$) vs. SCD, in both FER, BER at $N = 8192, R = 0.8$ and a construction using [21]

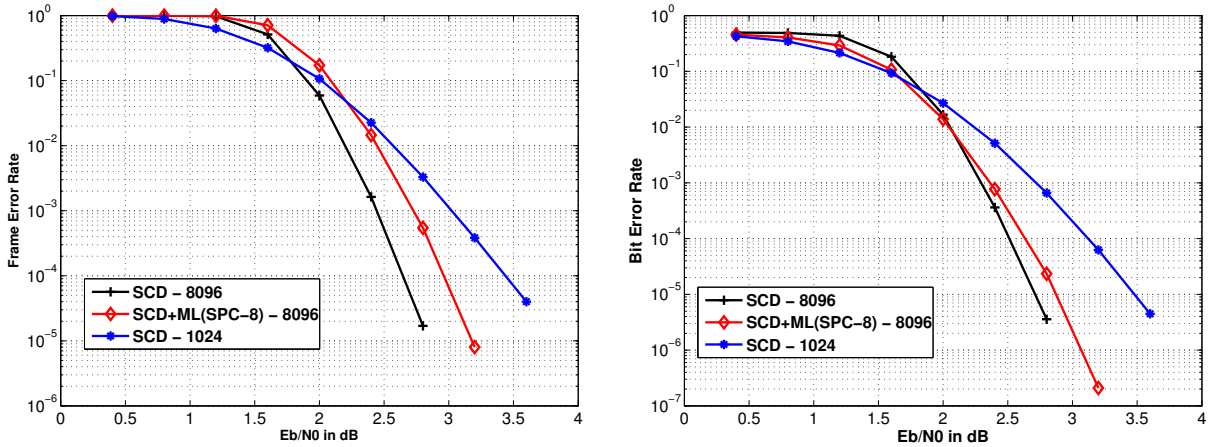

Fig. 4. FER Performance of a single parity check code of length 8 concatenated with a polar code of length 1024, at effective rate 0.5 compared with a polar codes with $N = 8096, R = 0.5$ and $N = 1024, R = 0.5$

achieving codes for symmetric binary-input memoryless channels," *IEEE Trans. on Inf. Theory*, vol. 55, no. 7, pp. 3051–3073, July 2009.

[4] C. Zhang, B. Yuan, and K. K. Parhi, "Reduced-latency sc polar decoder architectures," in *International Conference on Communications (ICC)*, Ottawa, ON, June 2012, pp. 3471–3475.

[5] C. Zhang and K. K. Parhi, "Low-latency sequential and overlapped architectures for successive cancellation polar decoder," *IEEE Transactions on Signal Processing*, vol. 61, no. 10, pp. 2429–2441, May 2013.

[6] A. Pamuk and E. Arikan, "A two phase successive cancellation decoder architecture for polar codes," in *International Symposium on Information Theory Proceedings (ISIT)*, Istanbul, July 2013, pp. 957–961.

[7] C. Leroux, A. J. Raymond, G. Sarkis, and W. J. Gross, "A semi-parallel successive-cancellation decoder for polar codes," *IEEE Transactions on Signal Processing*, vol. 61, no. 2, pp. 289–299, January 2013.

[8] I. Tal and A. Vardy, "List decoding of polar codes," in *International Symposium on Information Theory*, August 2011, pp. 1–5.

[9] B. Yuan and K. K. Parhi, "Low-latency successive-cancellation polar decoder architectures using 2-bit decoding," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 99, pp. 1–14, 2014.

[10] N. Goela, S. B. Korada, and M. Gastpar, "On LP decoding of polar codes," in *IEEE Information Theory Workshop (ITW)*, September 2010.

[11] A. Eslami and H. Pishro-Nik, "On finite-length performance of polar codes: Stopping sets, error floor and concatenated design," *IEEE Transactions on Communications*, vol. 61, no. 3, pp. 919–929, March 2013.

[12] S. Kahraman, E. Viterbo, and M. E. Celebi, "Folded tree maximum-likelihood decoder for Kronecker product-based codes," in *Allerton Conference on Communication, Control and Computing, 51st Annual*, Monticello, IL, October 2013, pp. 629–636.

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

[14] H. Vangala, E. Viterbo, and Y. Hong, "Permuted successive cancellation decoder for polar codes," in *International Symposium on Information Theory and Applications*, Melbourne, October 2014, (accepted).

[15] K. Niu and K. Chen, "Crc-aided decoding of polarcodes," *IEEE Communications Letters*, vol. 16, no. 10, pp. 1668–1671, October 2012.

[16] ——, "Stack decoding of polar codes," *Electronics Letters*, vol. 48, no. 12, pp. 695–697, June 2012.

[17] Z. L. Huang, C. J. Diao, and M. Chen, "Latency reduced method for modified successive cancellation decoding of polar codes," *Electronics Letters*, vol. 48, no. 23, pp. 1505–1506, Nov. 2012.

[18] G. Sarkis and W. J. Gross, "Increasing the throughput of polar decoders," *IEEE Communications Letters*, vol. 17, no. 4, pp. 725–728, April 2013.

[19] H. Vangala, E. Viterbo, and Y. Hong, "Improved multiple folded successive cancellation decoder for polar codes," in *31st International Union of Radio Science - General Assembly and Scientific Symposium (URSI-GASS)*, Beijing, China, August 2014, (invited paper).

[20] P. Trifonov, "Efficient design and decoding of polar codes," *IEEE Transactions on Communications*, vol. 60, no. 11, pp. 1–7, Nov. 2012.

[21] E. Arikan, "Performance comparison of polar codes and Reed-Muller codes," *IEEE Communications Letters*, vol. 12, no. 6, pp. 447–449, June 2008.

[22] R. Mori and T. Tanaka, "Performance and construction of polar codes on symmetric binary-input memoryless channels," in *International Symposium on Information Theory (ISIT)*, 2009, pp. 1496–1500.

[23] I. Tal and A. Vardy, "How to construct polar codes," *IEEE Transactions on Information Theory*, vol. 59, no. 10, pp. 6562–6582, October 2013.

[24] S. Zhao, P. Shi, and B. Wang, "Designs of bhattacharya parameter in the construction of polar codes," in *Wireless Communications, Networking and Mobile Computing (WiCOM), 7th international conference on*, Wuhan, September 2011, pp. 1–4.