Improved Multiple Folded Successive Cancellation Decoder for Polar Codes *

Harish Vangala [1]*, Emanuele Viterbo** [1]*, and Yi Hong* [1]

[1]Dept. of Electrical and Computer Systems Eng., Monash University, Wellington Rd, Clayton, VIC 3800, Australia
Email: {harish.vangala, emanuele.viterbo, yi.hong}@monash.edu

Abstract

Folding is a technique to modify the decoder graph of polar codes based on successive cancellation decoding (SCD). Folding was proposed in [1] and extended to multiple folding (κ times) in [2] as a technique to significantly reduce the latency of the SCD of polar codes up to a factor of 2^κ, without performance loss. The extension in [2] shows folding only for $\kappa \leq 3$ due to the rapidly increasing complexity. In this paper, we propose an alternative implementation of the multiple-folding SC decoder, to significantly reduce its complexity. The complexity of the new algorithm is only slightly higher than that of the original SCD for polar codes. As a result, the new algorithm enables to decode longer codes with larger κs, which exhibit significant performance gains, in addition to the latency gain.

1. Introduction

Polar codes proposed by Arikan [3] are a recent breakthrough in coding theory, which stand out as the first provable class of capacity achieving codes. From a practical point of view, Polar codes have an explicit construction and very low complexity encoding and decoding.

The successive cancelation decoder (SCD) for polar codes plays an important role in proving the capacity theorems. Additionally, the decoder has many attractive features for use in practical applications such as a highly regular structure and a constant complexity of $\mathcal{O}(N \log N)$, where N is the code-length. Although the decoder achieves the channel capacity asymptotically in the code length, the SCD is observed to have worse performance when compared to the best available LDPC codes at finite code lengths of practical interest. Additionally, the decoding suffers from large decoding latency equal to $(2N-1)$ units, in its most parallel implementation [3].

A significant number of papers have attempted to improve the SCD both in latency and decoder performance [4–11]. Several algorithmic solutions [8–11] have been reported to improve latency up to 50% when compared to the standard SCD. Folding is a technique proposed in [12] in the context of maximum-likelihood (ML) sphere decoding of polar codes. For the first time, this technique has allowed polar codes to be optimally ML decoded up to code length of 256. Later in [1] the folding technique was extended for use with SCD, saving 50% of decoding latency, but requiring a higher complexity. This work has been generalized to *multiple folding* in [2], where the latency advantage is further reduced by a factor of 2^κ, when folded κ times. However, the increasing complexity by an exponential factor did not exhibit any performance gain, since only short codes were analyzed. The current paper precisely addresses this issue of high complexity and demonstrates the potential performance gains for longer codes.

A brief summary of the contributions of this paper is provided below:

1. We propose a low-complexity version of the multiple folding SCD in [2], with latency reduced by a factor of 2^κ.
2. We show that the joint density processing proposed in $[1,2]$ can be replaced by (possibly parallel) processing of the binary marginal densities.
3. We show that the gains with multiple folding can be significant for longer codes and higher rates.

2. System Model

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

*This work was supported by NPRP grant #NPRP5-597-2-241 from the Qatar National Research Fund (a member of Qatar Foundation). Yi Hong's contribution was supported by the Australian Research Council under Discovery grants ARC DP 130100103.

Consider the following index notation: Given any subset of element indices (column indices) $\mathcal{I}$ from a vector $\mathbf{x}$ (or a matrix $\mathbf{A}$), we denote the corresponding sub-vector of $\mathbf{x}$ by $\mathbf{x}_\mathcal{I}$ (and the corresponding sub-matrix of $\mathbf{A}$ by $\mathbf{A}_\mathcal{I}$). Using this notation, we describe the encoding operation of a $(N, K, \mathcal{F})$ polar code, given a vector of information bits $\mathbf{u}$ of length K. Let $n \triangleq \log_2(N)$ and $\mathbf{G} \triangleq \mathbf{F}^{\otimes n} = \mathbf{F} \otimes \cdots \otimes \mathbf{F}$ be the n-fold Kronecker product of

$$\mathbf{F} \triangleq \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

Then, a codeword is generated as

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

where $\mathcal{F}^c \triangleq \{0, 1, \ldots, N-1\} \backslash \mathcal{F}$ 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 $\mathbf{d}_\mathcal{F} = \mathbf{0}$ and $\mathbf{d}_{\mathcal{F}^c} = \mathbf{u}$. An efficient implementation of (2) proposed by Arikan, is shown in Fig. 1.

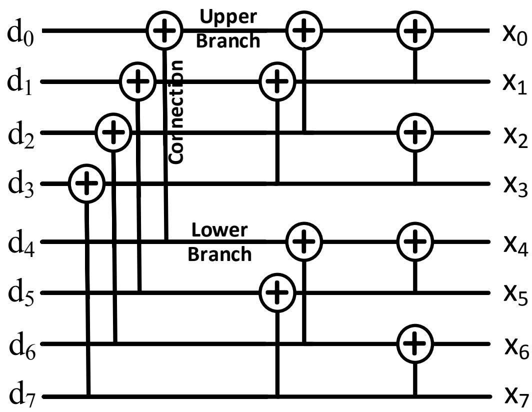

Figure 1: Arikan's $O(N \log_2 N)$ complexity encoder graph implementing (2) for $N = 8$

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, 13–17]. For simplicity, we use the original polar code construction algorithm from [13] with a good seed as in for e.g. [16].

Successive Cancellation Decoder (SCD) – The SCD algorithm [3] essentially follows the same encoder diagram as in Fig.1. The likelihoods evolve in the reverse direction from right-to-left. A detailed description and a full implementation may be found in [3, 18].

3. The Improved Multiple Folded Successive Cancellation Decoder

Single Folding — From Fig. 2 we observe that after the first stage of encoding of the input $\mathbf{d}$, the first half and second half of $\mathbf{d}$ see the same encoding circuit (dotted boxes) for the following stages. This suggests that we may process in parallel the operations in two dotted boxes at both encoder and decoder. This is the basic idea of the *folding* operation, which can be mathematically described as follows:

$$\mathbf{x} = \mathbf{F}^{\otimes n}\mathbf{d} = \begin{bmatrix} \mathbf{F}^{\otimes n-1} & \mathbf{F}^{\otimes n-1} \\ 0 & \mathbf{F}^{\otimes n-1} \end{bmatrix} \mathbf{d} \tag{3}$$

$$\Longrightarrow \begin{bmatrix} x_0\ x_{N/2} \\ \vdots \\ x_{(N/2)-1}\ x_{N-1} \end{bmatrix} = \mathbf{F}^{n-1} \begin{bmatrix} d'_0\ d'_{N/2} \\ \vdots \\ d'_{(N/2)-1}\ d'_{N-1} \end{bmatrix}, \text{ where } d'_j = \begin{cases} d_j \oplus d_{N/2+j}, & \text{if } 0 \le j < N/2 \\ d_j, & \text{if } N/2 \le j < N \end{cases} \tag{4}$$

$$\Longrightarrow \mathbf{x}' = \mathbf{F}^{\otimes n-1}\mathbf{d}' \tag{5}$$

We can interpret (5) as an encoding operation for a q-ary polar code as follows. Consider the pairs of bits quaternary symbols from $GF(4)$ ($q = 4$), taking values $\{00, 01, 10, 11\}$. The pairs of bits in the rows of the matrix $\mathbf{d}'$ are formed from $\mathbf{d}$ using (4). Alternatively, $\mathbf{d}'$ can be viewed as vector of q-ary symbols, which is encoded by using a matrix of half the dimension i.e. $\mathbf{F}^{\otimes n-1}$. Similarly the coded bit vector $\mathbf{x}$ is transformed to the quaternary vector of half length $\mathbf{x}'$. Note that the matrix vector multiplication in (5) should be interpreted over $GF(4)$.

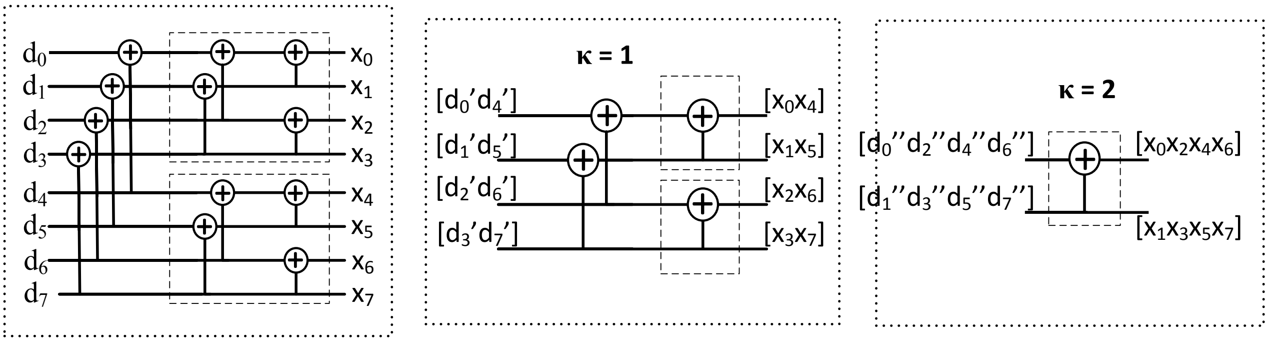

Figure 2: Folding technique illustrated at $N = 8$, folding length $\kappa = 1, 2$

During the decoding of a received vector $\mathbf{y}$ we similarly rearrange the elements to obtain $\mathbf{y}'$, to start decoding on the reduced encoding graph of q-ary symbols. We start doing so by forming the joint density functions of the q-ary alphabet instead of the likelihoods of bits in $\mathbf{x}$, and then following the density transformations in [1]. When we reach the left end of the graph, the decision element performs an ML decoding of size equal to two bits, according to the transformation $\mathbf{d}$ to $\mathbf{d}'$. We can easily verify from the transformation equations in [1] that, the density transformations cause a complexity increase by a factor of 8 but reduces the decoding graph size by a factor of 2, thus reducing the decoding latency by a factor of 2 i.e., from $(2N-1)$ to $(N-1)$.

Multiple Folding — The above concept of single folding can be extended κ times as proposed in [2], to produce a graph of reduced dimension $N/2^\kappa$, where each symbol contains 2^κ bits and hence the alphabet size increases to $q = 2^{2^\kappa}$ (see Fig. 2). This enables parallel processing of 2^κ bits at a time, which reduces latency by a factor of 2^κ. However, the overall complexity increases by a factor of $2^{2^\kappa}2^\kappa$, even when an efficient implementation of the q-ary density transformations is based on Hadamard transform. Note that the ML-decisions in the final decision stage introduce additional complexity and latency, though marginal when κ is small.

In this paper note that this q-ary interpretation of the vectors is not necessary. Instead, we may decode the bits in each q-ary symbol separately and parallely, which avoids the large complexity of joint density evolution with the q-ary alphabets of size $q = 2^{2^\kappa}$ and makes it equal to that in a standard SCD. This is seen from the (4), where each column of bits in q-ary vectors $\mathbf{x}'$ and $\mathbf{d}'$ can be processed separately. Since encoder has this clear separation, the decoding may exploit it to implement parallel decoding. However, note that, both columns in (4) are indeed correlated at the last stage ML-decision elements.

When repeated κ times, the overall likelihood evolution becomes equivalent to the evolution of likelihoods of 2^κ bits in parallel on the encoding graphs of shorter length $N/2^\kappa$. This finally results in the latency gain by a factor 2^κ. However, the additional latency & complexity of the final stage ML decision elements still needs to be accounted for. The ML decisions are also highly parallelizable and since they operate on very short blocks of 2^κ bits they have a low complexity for $\kappa \leq 3$.

The overall complexity with the our new proposal becomes almost the same as that in standard SCD, and the latency is reduced by almost a factor of 2^κ compared to that in standard SCD, if we neglect the ML decision elements of the last stage for small κs.

Note that the complexity of a short 2^κ length ML-decision element is comparable to that of a 2^κ length SCD which was embedded in the last stage and is now replaced by an MLD. Hence we may expect a gain in latency, up to a factor of 2^κ and only a negligible increase in the overall complexity. We show our simulation results in the next section with this latest proposed, improved multiple folding decoder model. The reduced complexity of our new proposed model enables usage of the decoder for higher blocklengths, where we saw significant performance gains with our improved multiple folded SCD.

4. Simulations

We consider polar encoding of code length $N = 8192$ and rate $R = 0.8$. We consider Additive White Gaussian Noise (AWGN) channel and BPSK modulation. We consider the polar code constructions from [13, 16].

Fig. 3 shows the gain in Frame Error Rate (FER), and the gain in Bit Error Rate (BER). As expected, the new multiple folded decoder performs at least as good as the standard SCD. Further we see that the potential gain can be significant, in addition to the gain in latency.

5. References

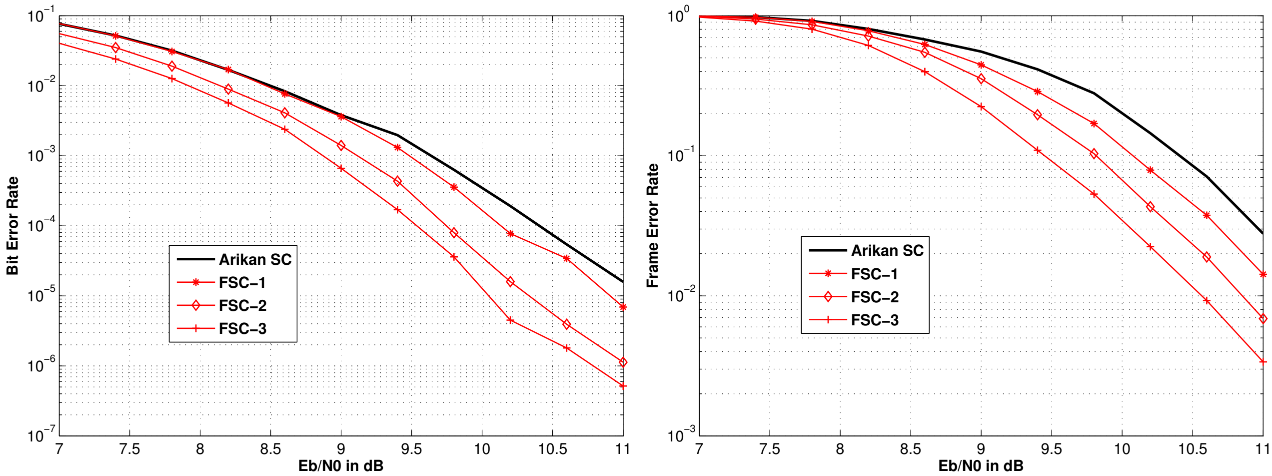

Figure 3: Performance comparison in Frame Error Rate and Bit Error Rate, with various κ at $N = 8192, R = 0.8$

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

[2] ——, "Multiple folding for successive cancelation decoding of polar codes," 2014, in preparation.

[3] E. Arikan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Transactions on Information Theory*, vol. 55, no. 7, pp. 3051–3073, July 2009.

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

[5] 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, December 2012.

[6] Z. Huang, C. Diao, and M. Chen, "Multiple candidates successive cancellation decoding of polar codes," in *International Conference on Wireless Communications and Signal Processing (WCSP)*, Huangshan, 2012, pp. 1–4.

[7] K. Niu and K. Chen, "Stack decoding of polar codes," *Electronics Letters*, vol. 48, no. 12, pp. 695–697, June 2012.

[8] 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, November 2012.

[9] 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.

[10] 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.

[11] 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.

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

[14] 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.

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

[16] 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),*, Wuhan, September 2011, pp. 1–4.

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

[18] V. Harish, E. Viterbo, and Y. Hong, "Permuted successive cancellation decoder for polar codes," in *International Symposium on Information Theory an Applications*, 2014.