Improving Successive Cancellation Decoding of Polar Codes by Usage of Inner Block Codes

Mathis Seidl and Johannes B. Huber
Lehrstuhl für Informationsübertragung
Universität Erlangen-Nürnberg
Erlangen, Germany
Email: {seidl, huber}@LNT.de

***Abstract*—Polar coding is a recently introduced capacity-achieving code constructing method for binary-input discrete memoryless channels. We present a method to improve the finite-length performance of polar codes together with successive cancellation (SC) decoding by means of simple and short inner block codes. Example simulations show an improvement of about 0.3 dB.**

I. Introduction

Polar codes were recently introduced by E. Arıkan [1]. By a construction of recursively combining and splitting a number N of independent copies of a binary-input discrete memoryless channel (B-DMC), an alternative set of channels is generated which shows—with increasing block length—a polarization effect in the sense that the capacity of almost each channel is either near 0 or near 1. As the block length goes to infinity, the fraction of channels not being either completely noisy or completely noiseless tends to zero. However, for finite block lengths there exists a broad "grey zone" of channels whose capacities are far from being polarized.

For data transmission only the channels with highest capacity are used, referred to as information bits. The capacities can either be determined by simulation or by density evolution [2] [3]. The data transmitted over non-information channels (so-called frozen bits) are fixed values which are known to the decoder. We denote $\mathcal{A}$ the set of information bits and $\mathcal{A}_c$ that of frozen bits.

A. Decoding

The benefit of a binary length N polar code with dimension K is that decoding can be performed using a low-complexity $\mathcal{O}(N \log N)$ successive cancellation (SC) decoder. The algorithm successively generates estimates $\hat{u}_i$ $(i = 1 \dots K)$ on the bits u_i of the source vector by computing log-likelihood ratios (LLRs) of the form

$$L_i = \ln \left(\frac{\Pr\left(u_i = 0 | y_1^N, \hat{u}_1^{i-1}\right)}{\Pr\left(u_i = 1 | y_1^N, \hat{u}_1^{i-1}\right)} \right) \quad , \tag{1}$$

each depending on the decisions made before. Here, y_1^N denotes the received vector from index 1 to N.

If $i \in \mathcal{A}$, the bit $\hat{u}_i$ is set to the more probable value, otherwise the already known value of the frozen bit is used. We denote $p(\hat{u}_i)$ the (conditional) probability of the value $\hat{u}_i$ the decoder finally decides for at channel index i. Thus, given the estimated (or known) value $\hat{u}_i$, we define

$$p(\hat{u}_i) := \Pr\left(u_i = \hat{u}_i | y_1^N, \hat{u}_1^{i-1}\right) \quad . \tag{2}$$

Note that $p(\hat{u}_i)$ is not necessarily greater than $1/2$ if $i \in \mathcal{A}_c$.

The probability $P_e(i)$ of the SC decoder to fail at the ith channel provided that all previous decisions were correct is given by the error probability functional [4, p. 201]

$$\mathcal{P}_e(i) = \frac{1}{2} \cdot \int_{-\infty}^{\infty} l_i(x) \cdot \mathrm{e}^{-\frac{1}{2}(x+|x|)} \, \mathrm{d}x \tag{3}$$

with l_i denoting the L-density corresponding to the LLR L_i at the i-th channel which is obtained using density evolution. The frame error rate (FER) is upper bounded by the union bound

$$\mathcal{P}_B = \sum_{i \in \mathcal{A}} \mathcal{P}_e(i) \quad . \tag{4}$$

B. Partial polarization

We want to illustrate the problem of imperfect polarization for finite-length polar codes by introducing an example. Fig.

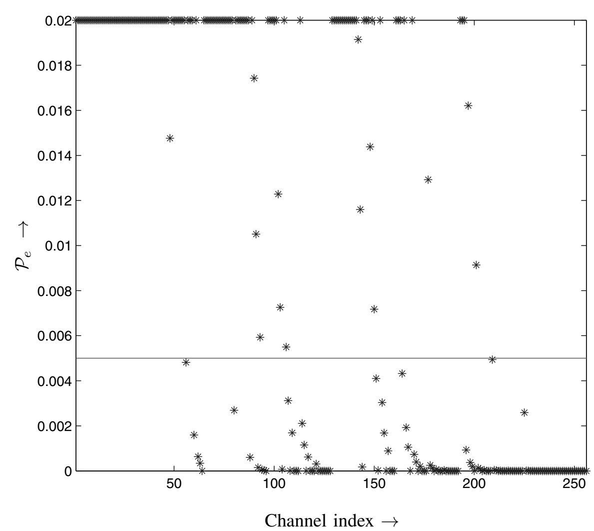

Fig. 1. Error probability $\mathcal{P}_e$ for a rate $1/2$ polar code of length $N = 256$ at $10 \log_{10}(E_\mathrm{b}/N_0) = 2.5$dB

1 depicts the distribution of the error probabilities $\mathcal{P}_e(i)$ with respect to the channel indices i. Here, a block length of $N = 256$ and rate $R = 1/2$ was used while transmitting over the binary-input AWGN channel (BIAWGNC) at $10\log_{10}(E_\mathrm{b}/N_0) = 2.5\mathrm{dB}$. Channels with $\mathcal{P}_e \geq 0.02$ are characterized by an asterisk at 0.02. The horizontal line at $\mathcal{P}_e = 0.005$ demarks the border between the sets $\mathcal{A}$ and $\mathcal{A}_c$. The FER bounded by $\mathcal{P}_B \approx 0.048$ is caused in a large part by only a very small fraction of the 128 information bits. Taking away only the 8 least reliable channels would more than halve the bound on FER resulting in $\mathcal{P}_B \approx 0.019$.

II. Inner block codes

In this section we demonstrate how the frame error probability can be reduced by combining the least reliable information bits with suitably chosen frozen bits. In contrast to usual concatenation schemes, our approach uses the polar code as an outer code where neither codeword length is increased nor code rate is changed. Fig. 1 shows that error probabilities do not decline monotonously with increasing channel index. The sets $\mathcal{A}$ and $\mathcal{A}_c$ are interlaced, i.e. information bits are followed by frozen bits and vice versa. Furthermore, there are (comparatively unreliable) information bits surrounded by frozen bits with error probabilities not being significantly higher. By using short block codes operating on a set consisting of both information and (now no longer) frozen bits we are able to achieve a significantly lower FER without changing the overall rate of the polar code.

A. Code construction

The construction of an inner code $\mathcal{C}_\mathrm{I}$ of length n and dimension k is carried out in 2 steps:

- First, we determine an ordered set $\mathcal{J}$ of length n including the channel indices to be used:

  $$\mathcal{J} = \{j_1, j_2, \ldots, j_n\} \quad , \quad 0 < j_1 < j_2 < \ldots < j_n < N$$

  with

  $$k = \#\{i \in \{1, \ldots, n\} : j_i \in \mathcal{A}\} \quad , \quad 0 < k < n$$

  being the number of information bits used and thus being the dimension of the inner code $\mathcal{C}_\mathrm{I}$. The set $\mathcal{J}$ does not necessarily have to consist of subsequent channel indices. Still, greater spacings between the elements of $\mathcal{J}$ increase the computational complexity as we will point out in the next section. Clearly, the choice of $\mathcal{J}$ depends on the channel properties as the sets $\mathcal{A}$ and $\mathcal{A}_c$ do.
- Now we choose a well suited short (n, k) block code (which should be ML-decodable by full search) to operate on the index set $\mathcal{J}$.

B. Decoding

For decoding, some modifications of the SC decoder are necessary. Plainly, the SC decoding task has to be performed either repeatedly or in parallel for each codeword of the inner code $\mathcal{C}_\mathrm{I}$, as described in the following:

k denotes the dimension of the inner code $\mathcal{C}_\mathrm{I}$. W.l.o.g. we suppose the inner coding to be systematic. Up to the decision index j_1, the SC decoding is performed as described in [1]. Now, a second instance of the SC decoder is started allowing for parallel decoding of both possibilities $u_{j_1} = 0$ and $u_{j_1} = 1$. The probabilities of the following decisions made in each instance $p(u_i|u_{j_1} = 0)$ and $p(u_i|u_{j_1} = 1)$ $(j_1 < i < j_2)$ are recorded separately regardless of whether i is an information bit or not. Reaching index j_2, two additional decoding instances are created in order to determine the likelihoods of each possible value combination of u_{j_1} and u_{j_2}.

In the same way, further splittings at the indices defined by $\mathcal{J}$ are performed until decision index j_k leading to a total of 2^k different paths corresponding to the codewords $c_l \in \mathcal{C}_\mathrm{I}$ $(l = 1, \ldots, 2^k)$. The parallel decoding of the 2^k paths then continues treating the following indices $j_{k+1}, \ldots, j_n$ from $\mathcal{J}$ as frozen bits with values set according to $\mathcal{C}_\mathrm{I}$ until reaching the last index j_n of the set $\mathcal{J}$.

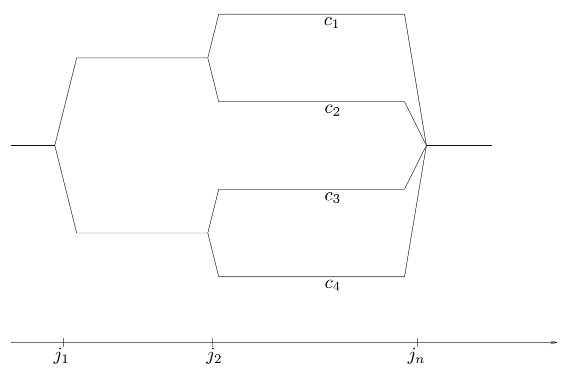

Fig. 2. Decoding scheme for an inner code of dimension $k = 2$

Now the decoder decides for the codeword c_{l^*} of the inner code with maximum probability:

$$l^* = \arg\max_l \left(\prod_{i=j_1}^{j_n} p_l(\hat{u}_i) \right)$$

with $p_l(\cdot)$ $(l = 1, \ldots, 2^k)$ denoting the particular probabilities according to eq. (2). Note that statistical dependencies of the $\hat{u}_i$ are included in the definition of the $p_l(\hat{u}_i)$. The SC algorithm finally continues in the usual way using the values of c_{l^*} and discarding the other instances. Fig. 2 shows an example scheme for $k = 2$.

Considering the computational complexity, obviously the length of the block $\mathcal{J}$ with respect to the outer code is an important parameter which we denote by

$$B_\mathcal{J} := j_n - j_1 + 1 \quad .$$

The additional computational complexity grows linearly with $B_\mathcal{J}$ and exponentially with the dimension k of the used inner block code $\mathcal{C}_\mathrm{I}$.

III. Simulation results

Finally, we present some exemplary realizations of inner block codes constructed by the just introduced method. For the following considerations a polar code of length $N = 256$ and rate $R = 1/2$ is used. Furthermore, for comparability reasons of the particular inner codes we concentrate on constructing inner block codes of length $n = 8$ and dimension $k = 4$. The error probability values given in the following subsections as well as the sets of information and frozen bits $\mathcal{A}$ and $\mathcal{A}_c$ refer to a signal-to-noise ratio of $10\log_{10}(E_\mathrm{b}/N_0) = 2.5\mathrm{dB}$. The construction for different SNRs is carried out in complete analogy.

Experimenting with simple length-2-repetition codes combining one bit from $\mathcal{A}$ with one from $\mathcal{A}_c$ we observed that the order plays a decisive role: With the information bit followed by the frozen bit we achieved a reduction of the FER while the converse constellation in most cases led to an actually higher error probability than without coding. Note in this context that the error probabilities $\mathcal{P}_e$ are conditional probabilities under the assumption that the previous bits have been decoded correctly.

A. Concatenated repetititon codes

The first inner coding scheme we consider is based on a simple repetition code of length 2. For each information bit that we wish to protect a redundancy bit from $\mathcal{A}_c$ is chosen. Because information bits and frozen bits in polar codes usually do not occur alternately but in clusters, we build coding blocks of four concatenated length-2-repetition codes by combining the elements $(1 \leftrightarrow 5)$, $(2 \leftrightarrow 6)$, $(3 \leftrightarrow 7)$ and $(4 \leftrightarrow 8)$ resulting in an usual (8,4) repetition code.

At first we specify the indices of the first information bits with expected error probabilities $\mathcal{P}_e > 0.0006$ (cf. Fig. 1):

$$56, 60, 80, 88, 107, 109, \ldots$$

As we want to use blocks of length $n = 8$ we now have to look for four frozen bits in the range $[57, 106]$ which are as reliable as possible. Table I lists the indices used for the first inner coding block together with their corresponding bit error probabilities $\mathcal{P}_e$. Information bits are highlighted to illustrate the previously mentioned cluster structure. We now proceed building blocks in the same way to protect further information bits. One possible result consisting of four coding blocks is shown below. The third line of (5) still improves the FER although for two of the length-2-repetition blocks $(1 \leftrightarrow 5)$ and $(2 \leftrightarrow 6)$ we had to place the frozen bit before the information bit due to the structure of the code.

TABLE I

i	**56**	**60**	**80**	**88**	91	93	103	106
$\mathcal{P}_e(i)$	0.0048	0.0016	0.0027	0.0006	0.0105	0.0059	0.0073	0.0055

$$\begin{bmatrix} \mathbf{56} & \mathbf{60} & \mathbf{80} & \mathbf{88} & 91 & 93 & 103 & 106 \\ \mathbf{107} & \mathbf{109} & \mathbf{114} & \mathbf{115} & 136 & 140 & 142 & 143 \\ 148 & 150 & \mathbf{151} & \mathbf{154} & \mathbf{155} & \mathbf{157} & 164 & 165 \\ \mathbf{166} & \mathbf{167} & \mathbf{168} & \mathbf{170} & 177 & 194 & 195 & 197 \end{bmatrix} \quad (5)$$

The additional complexity for the first line of (5) can be determined as follows: Between bit 56 and 91, the number of parallel branches increases at each element in (5) by a factor 2 up to a number of 16. From bit 91 on, each element causes a division of parallel branches by 2, ending with one resting branch at bit 106.
Considering the complete concatenation scheme, we obtain a complexity multiplied approximately by the factor 5 when compared to the case without concatenation.

B. Extended Hamming codes

Replacing the repetition code from the last subsection by an (8,4) extended Hamming code enlarges the number of possible constellations concerning the positioning of the elements in one block.

$$\begin{bmatrix} \mathbf{56} & \mathbf{60} & \mathbf{80} & \mathbf{88} & 91 & 93 & 103 & 106 \\ \mathbf{107} & \mathbf{109} & \mathbf{114} & \mathbf{115} & 142 & 143 & 148 & 150 \\ \mathbf{151} & \mathbf{154} & 163 & \mathbf{164} & 165 & \mathbf{166} & 169 & 177 \end{bmatrix} \quad (6)$$

The coding scheme (6) achieves a comparable performance while reducing the overall block lengths $B_\mathcal{J}$ of the concatenation scheme. Note that the last line of (6) is not a valid (8,4) repetition block as defined above.
On the other hand, despite of the shorter overall block lengths $B_\mathcal{J}$, the additional complexity is slightly higher than for the repetition case (factor ≈ 6 compared to the non-concatenated code), as here the 16 branches have to be computed in parallel from bit 4 to bit 8 of each line of (6).

C. Simulation

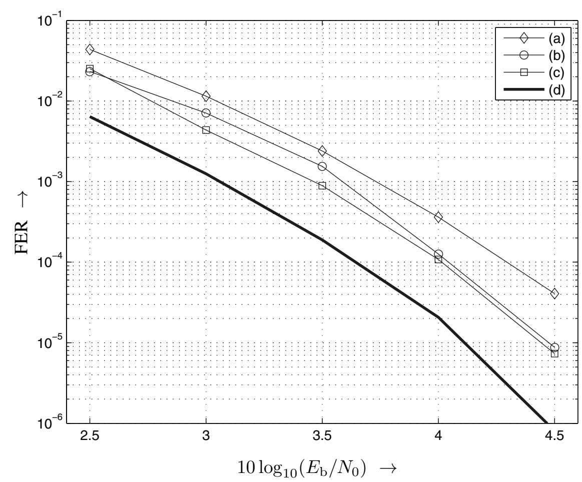

Fig. 3. Frame error rates on BIAWGNC for a (256,128) polar code using (a) no inner code, (b) a 4-block repetition code, (c) a 3-block (8,4) extended Hamming code as inner code

The bold line (d) in Fig. 3 shows the performance of a genie-aided decoder which automatically sets the 16 least reliable bits to the correct value, corresponding to a "perfect" inner code that never fails. Clearly, this demarks an upper bound on the performance gain that can be achieved using the concatenation approach presented in this paper.
As the sets $\mathcal{A}$ and $\mathcal{A}_c$ change with increasing SNR, slight adjustments of the coding schemes (5), (6) for each simulation

point had been applied. The simulations show a significant performance improvement of approx. 0.3 dB.

IV. Conclusion

We have pointed out a way to achieve better decoding results for finite-length polar codes by combining less reliable subchannels with suitable channels previously not being used for carrying information within short block codes.

References

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

[2] R. Mori and T. Tanaka, "Performance of polar codes with the construction using density evolution," *IEEE Comm. Letters.*, vol. 13, no. 7, pp. 519–521, 2009.

[3] S. H. Hassani, S. B. Korada, and R. L. Urbanke, "The compound capacity of polar codes," *CoRR*, vol. abs/0907.3291, 2009.

[4] T. Richardson and R. Urbanke, *Modern Coding Theory*. Cambridge University Press, March 2008.