

BIDIRECTIONAL STACK DECODING OF POLAR CODES

Aleksandar Đ. Minja^a, Dušan P. Dobromirov^b, Vojin I. Šenk^c

^a University of Novi Sad, Faculty of Technical Sciences,
IEM Department, Novi Sad, Republic of Serbia,
e-mail: aminja@uns.ac.rs,
ORCID iD: <https://orcid.org/0000-0001-6701-2258>

^b University of Novi Sad, Faculty of Technical Sciences,
IEM Department, Novi Sad, Republic of Serbia,
e-mail: ddbromirov@uns.ac.rs, **corresponding author**,
ORCID iD: <https://orcid.org/0000-0003-0553-7993>

^c University of Novi Sad, Faculty of Technical Sciences,
IEM Department, Novi Sad, Republic of Serbia,
e-mail: vojnin_senk@uns.ac.rs,
ORCID iD: <https://orcid.org/0000-0003-3029-0486>

DOI: 10.5937/vojtehg69-29858; <https://doi.org/10.5937/vojtehg69-29858>

FIELD: Telecommunications, Computer sciences

ARTICLE TYPE: Original scientific paper

Abstract:

Introduction/purpose: The paper introduces a reduced latency stack decoding algorithm of polar codes, inspired by the bidirectional stack decoding of convolutional codes and based on the folding technique.

Methods: The stack decoding algorithm (also known as stack search) that is useful for decoding tree codes, the list decoding technique introduced by Peter Elias and the folding technique for polar codes which is used to reduce the latency of the decoding algorithm. The simulation was done using the Monte Carlo procedure.

Results: A new polar code decoding algorithm, suitable for parallel implementation, is developed and the simulation results are presented.

Conclusions: Polar codes are a class of capacity achieving codes that have been adopted as the main coding scheme for control channels in 5G New Radio. The main decoding algorithm for polar codes is the successive cancellation decoder. This algorithm performs well at large blocklengths with a low complexity, but has very low reliability at short and medium blocklengths. Several decoding algorithms have been proposed in order to improve the error correcting performance of polar codes. The successive cancellation list decoder, in conjunction with a cyclic redundancy check, provides very

ACKNOWLEDGMENT: The research described in this paper has been supported by the Serbian Ministry of Education, Science and Technological Development through Project no. 451□03-68/2020-14/200156 titled "Innovative scientific and artistic research from the Faculty of Technical Sciences activity domain".

good error-correction performance, but at the cost of a high implementation complexity. The successive cancellation stack decoder provides similar error-correction performance at a lower complexity. Future machine-type and ultra reliable low latency communication applications require high-speed low latency decoding algorithms with good error correcting performance. In this paper, we propose a novel decoding algorithm, inspired by the bidirectional stack decoding of classical convolutional codes, with reduced latency that achieves similar performance as the classical successive cancellation list and successive cancellation stack decoding algorithms. The results are presented analytically and verified by simulation.

Key words: polar codes, successive cancellation, stack and list decoding, folding.

Introduction

Context and motivation. Polar codes, introduced in (Arikan, 2009), are the first class of codes with an explicit construction and a low complexity encoding and decoding that provably achieve the symmetric capacity of a binary input discrete memoryless channel (bi-DMC). The standard decoding algorithm for polar codes is the successive cancellation decoding (SCD). Although the SCD has good error correcting performance for large blocklengths and is an important procedure for proving the capacity achieving property of polar codes, it underperforms for short and medium blocklengths. Several new decoding algorithms are developed to correct this. The most important algorithms in this category are definitely the successive cancellation list decoding (SCLD), introduced in (Tal & Vardy, 2015) and improved in (Balatsoukas-Stimming et al., 2015) and (Hashemi et al., 2016), the successive cancellation stack decoding (SCSD) introduced in (Niu, K. & Chen, K., 2012a) and improved in (Xiang et al., 2019), (Aurora et al., 2018), (Xiang et al., 2020) and the belief propagation decoding (BPD) (Arikan, 2008), (Arikan, 2010). Cyclic redundancy check (CRC) aided polar codes, coupled with the SCLD or the SCSD (Niu & Chen, 2012b), (Li et al., 2012) have been proposed to further improve the error correcting performance of polar codes for practical applications. Other research direction deals with reducing the latency of the decoding algorithm. The simplified successive cancellation algorithm (SSCD) was proposed in (Alamdar-Yazdi & Kschischang, 2011) and further improved in (Sarkis et al., 2014), (Hanif & Ardakani, 2017). The main idea of the SSCD is to identify small constituent codes that can efficiently be decoded. The

four common constituent codes usually considered are the rate-0 code, the repetition code, the rate-1 code, and the single parity check code. The reduced latency SCLD was presented in (Li et al., 2013) and an efficient implementation of the SCSD was proposed in (Aurora et al., 2018). One way of reducing latency is to exploit the recursive nature of polar codes. This approach known as folding, was introduced in (Kahraman et al., 2013). Several decoding algorithms based on the folding technique were proposed in (Kahraman et al., 2014a), (Kahraman et al., 2014b), (Vangala et al., 2014a), (Vangala et al., 2014b), (Huang et al., 2018). A reduced latency implementation of the SCSD was given in (Xiang et al., 2020).

Polar codes have been adopted as the main coding scheme for control and physical broadcast channels in the enhanced mobile broadband (eMBB) and the ultra reliable low latency communications (URLLC) service categories, defined in the fifth generation (5G) wireless communications standard (-3rd Generation Partnership Project., 2016), (Won & Ahn, 2020), (Hashemi et al., 2020). It is well known that the CRC aided (2048, 1024) polar codes with list decoding can outperform LDPC and Turbo codes of the same length (Li et al., 2012). Although polar codes have good error correcting performance, the decoding latency still remains an important research topic (Hashemi et al., 2020). In this work, we focus on improving the decoding latency of the SCSD algorithm.

Contribution. In this paper, we present a novel low latency decoding algorithm that is constructed by applying the folding technique to the SCSD algorithm. The new decoding algorithm is in part inspired by the bidirectional decoding of convolutional codes (Senk & Radivojac, 1997) where the channel output is divided into two parts and each part is processed in parallel, while an inner component code is used to combine the results. We name this new decoding algorithm the bidirectional stack decoding (BSD). To the best of our knowledge, no such algorithms has ever been presented. We also show that the reduced latency decoding algorithm presented in (Xiang et al., 2020) can also be constructed by applying the folding technique to the classical SCSD algorithm. This reduced latency SCSD can be used as a component decoder of our algorithm, presented here.

Paper organization. The remainder of the paper is composed of four sections. The section [System model](#) introduces the polar codes and

gives a brief overview of the SCD, the SCLD and the SCSD algorithms. The section [Bidirectional stack decoding algorithm](#) describes the new decoding algorithm introduced in this paper. The next section presents the [Simulation results](#). The section [Conclusion](#) gives an overview of the future work and concludes the paper.

Notation. Throughout the paper, uppercase letters represent random variables, lowercase letters represent realizations of the corresponding random variables, uppercase bold letters represent random vectors, and lowercase bold letters represent their realization. The i -th component of a vector x is denoted x_i . $P[\cdot]$ represents the probability of an event, $E[\cdot]$ represents the mean of a random variable and $\text{Var}[\cdot]$ represents the variance of a random variable. Cursive uppercase letters represent sets, and \mathcal{A}^N represents the set of all N -tuples of a set \mathcal{A} . The cardinality of \mathcal{A} is denoted $|\mathcal{A}|$. Sometimes a set will be defined only by its elements, i.e. $\{a_1, a_2, \dots, a_N\}$. Given a set $\mathcal{A} = \{a_1, a_2, \dots, a_K\}$ and an N -dimensional vector x , such that $N \geq K$, we define a new vector $x_{\mathcal{A}} = [x_{a_1}, x_{a_2}, \dots, x_{a_K}]$. Other notation is introduced as it is used.

System model

In this paper, we consider only binary polar codes constructed by the Arikan kernel ([Arikan, 2009](#))

$$F_2 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}, \quad (1)$$

as they are most often used in practical applications. Given a vector $u = [u_0, u_1]$, we define $u \cdot F_2 = x = [x_0, x_1]$. A useful graphical representation of this multiplication is given in Fig. 1. Note that $u = xF_2^{-1} = xF_2$ can be represented by the same logical circuit in Fig. 1, by treating values x_0 and x_1 as the input, and u_0 and u_1 as the output. The Kronecker product of the matrix F_2 with itself is defined as

$$F_2 \otimes F_2 = \begin{bmatrix} 1 \cdot F_2 & 0 \cdot F_2 \\ 1 \cdot F_2 & 1 \cdot F_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}. \quad (2)$$

Given $N = 2^m$, we define an N -dimensional polar transform as $F_N = F_2^{\otimes m} = F_2^{\otimes m-1} \otimes F_2$. A logical circuit representation of a higher-order matrix

can be recursively constructed from smaller order circuits as shown in Fig. 2 for the case of the matrix in eq. (2).

The polar code $\mathcal{C} : (N, K, \mathcal{A})$ is a linear block code of length N and dimension K defined by a set of indices $\mathcal{A} \subset \{0, 1, \dots, N - 1\}$, such that $|\mathcal{A}| = K$. The generator matrix of the code \mathcal{C} is given by $G = F_{N, \mathcal{A}}$, which consists only of the rows of F_N specified by the values in \mathcal{A} . Encoding can be defined as

$$i \cdot G = x. \tag{3}$$

Alternatively, we can define an N -dimensional vector u , such that $u_{\mathcal{A}} = i$, and $u_{\mathcal{A}^c}$ bits are said to be frozen and set to some predefined value (usually all zeros) which is known both by the encoder and the decoder. This is useful as we can now use a logical circuit (as the one given in Fig. 1 and Fig. 2) to define the encoding and decoding. There are many methods for constructing the set \mathcal{A} (Vangala et al., 2015). In this paper, we used the method based on Bhattacharyya bound approximation proposed in (Arikan, 2009).

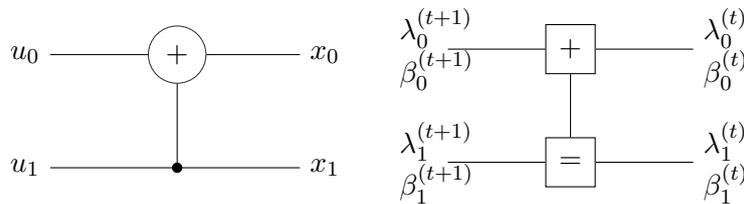


Figure 1 – Polar code unit circuit.
Encoding circuit (left) and decoding circuit (right).

Рис 1 – Схема блока полярного кода.
Схема кодирования (слева) и схема декодирования (справа).
Слика 1 – Основно коло полярног кода.
Коло за кодовање (лево) и коло за декодовање (десно).

Successive cancellation decoding

Let \mathbf{X} be a random codeword, and $\mathbf{C} = \phi(\mathbf{X})$, where $\phi(\cdot)$ represents the BPSK mapping, defined as $\phi(x) = 1 - 2x$. \mathbf{C} is transmitted over an additive white Gaussian channel (AWGN), and the channel output $\mathbf{Y} = \mathbf{C} + \mathbf{W}$ is received. \mathbf{W} represents the AWGN noise vector with mean 0 and variance σ^2 . Let \mathbf{y} be a specific channel output, then $\lambda = \frac{2\mathbf{y}}{\sigma^2}$ represents the log-likelihood ratio (LLR) vector.

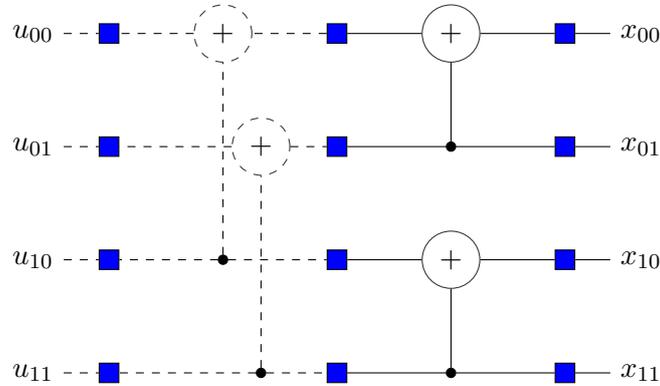


Figure 2 – Recursive construction of a 4-dimensional polar transform from two unit circuits

Рис 2 – Рекурсивное построение 4-мерного полярного преобразования из двух единичных схем

Слика 2 – Рекурзивна конструкција 4-димензионалне поларне трансформације спајањем два основна кола

The decoding factor graph can be constructed from the encoding circuit by replacing all summation blocks with check nodes and all junctions with variable nodes (Fig. 1) (Forney, 2001).

The SCD decoding algorithm works by passing the LLR values along such a defined factor graph from right to left, and hard decisions (β values) from left to right. When an updated LLR value reaches the end of the decoding circuit, a hard decision is made and propagated in the other direction. The SCD decoding procedure for the case of $N = 2$ is shown in Fig. 1. The update rules in Fig. 1 are given by:

$$\begin{aligned}
 \lambda_0^{(t+1)} &= f(\lambda_0^{(t)}, \lambda_1^{(t)}), \\
 \beta_0^{(t+1)} &= h(\lambda_0^{(t+1)}), \\
 \lambda_1^{(t+1)} &= g(\lambda_0^{(t)}, \lambda_1^{(t)}, \beta_0^{(t)}), \\
 \beta_1^{(t+1)} &= h(\lambda_1^{(t+1)}), \\
 \beta_0^{(t)} &= \beta_0^{(t+1)} + \beta_1^{(t+1)}, \\
 \beta_1^{(t)} &= \beta_1^{(t+1)},
 \end{aligned} \tag{4}$$

where the functions $f(\cdot)$, $g(\cdot)$ and $h(\cdot)$ are defined as

$$\begin{aligned} f(\lambda_0, \lambda_1) &\approx \text{sign}(\lambda_0) \text{sign}(\lambda_1) \min(|\lambda_0|, |\lambda_1|), \\ g(\lambda_0, \lambda_1, \beta) &= (-1)^\beta \cdot \lambda_0 + \lambda_1, \\ h(\lambda) &= \begin{cases} 0 & \lambda \geq 0 \\ 1 & \lambda < 0 \end{cases}. \end{aligned} \tag{5}$$

The high-level description of the SCD algorithm for $N = 2^m$, $m \geq 1$ is given in Alg. 1 (Vangala et al., 2015). Let $\Lambda \in \mathbb{R}^{N \times m+1}$ and $B \in \mathbb{F}_2^{N \times m+1}$, be two matrices for storing the calculated LLR and hard estimate values, respectively. Each column of the matrix Λ (B) corresponds to a given level (numerated from left to right) in the decoding factor graph. At the beginning of the algorithm, the first column of the matrix Λ is set to the channel LLRs (λ). Furthermore, let $\zeta(\cdot)$ represent the standard bit reversal function. For an efficient implementation of the SCD algorithm, see (Aurora & Gross, 2018).

Algorithm 1: SCD algorithm

Input : $\lambda, N, \mathcal{A}^c$
Output: β
Initialize Λ and B ;
 $\Lambda_{i,0} = \lambda_i \ i = 0, \dots, N - 1$;
for $i = 0, \dots, N - 1$ **do**
 $b = \zeta(i)$;
 recursively calculate $\Lambda_{b,m}$;
 if $b \in \mathcal{A}^c$ **then**
 $B_{b,m} = 0$;
 else
 $B_{b,m} = h(\Lambda_{b,m})$;
 end
 recursively propagate $B_{b,m}$;
end
 $\beta_i = B_{i,m} \ i = 0, \dots, N - 1$;

Successive cancellation list and stack decoding

The main drawback of the SCD algorithm is the fact that once the algorithm makes a hard decision, that decision is never again revisited, and an error can easily propagate. The list and stack algorithms were proposed to fix this problem. Let $s_l = (\Lambda, B, PM^i, i)$ represent the current state of the decoder. The matrices Λ and B are the same as before and i represents the index of the last decoded bit. PM^i represents the path metric at the time i and it is calculated as (Balatsoukas-Stimming et al., 2015)

$$PM^i = \begin{cases} PM^{i-1}, & B_{\zeta(i),m} = h(\Lambda_{\zeta(i),m}) \\ PM^{i-1} + |\Lambda_{\zeta(i),m}|, & B_{\zeta(i),m} \neq h(\Lambda_{\zeta(i),m}) \end{cases} \quad (6)$$

$$PM^{-1} = 0$$

The SCLD algorithm works by keeping a list of states. At each time step, all states in the list are expanded. Every time a decision is made, the other decision is also considered and a new element is added to the list. If the list size is greater than some predefined size L , the list is sorted according to the path metric and only the best L are kept. The SCSD algorithm works in a similar manner, but instead of a list of states, a stack of states is kept and at each step only the best (top) state is considered. As the SCSD is an important component of the BSD, the high level description of the SCSD algorithm is given in Alg. 2. For a description of the SCLD, see (Balatsoukas-Stimming et al., 2015). In Alg. 2 the function $\text{top}(\cdot)$ returns the top of the stack. The matrix \bar{B} is calculated by flipping a single bit in B , while the \bar{PM} is the path metric of this changed state. We assume the stack is sorted in the descending order of the path metric, so that the best path is at the top of the stack. If the stack size is greater than some predefined size L , then the worst path is discarded. For an efficient implementation of the SCSD, see (Aurora & Gross, 2018), (Xiang et al., 2020).

Bidirectional stack decoding algorithm

Let $\mathcal{I} = \{0, 1, \dots, N/2 - 1\}$ and $\mathcal{J} = \{N/2, \dots, N - 1\}$ be two non-overlapping index sets. Furthermore, let $\mathcal{E} \subset \{0, 1, \dots, N - 1\}$ be the set of all even numbers lower than N , and $\mathcal{O} \subset \{0, 1, \dots, N - 1\}$ be the set of all odd numbers lower than N . Using the recursive structure of polar codes, we can write (Kahraman et al., 2013)

Algorithm 2: SCSD algorithm

Input : $\lambda, N, \mathcal{A}^c$
Output: β
Initialize Λ and B and create an empty stack s ;
Add the state $(\Lambda, B, 0, -1)$ to the stack s ;
 $(\Lambda, B, i, PM) = s.top()$;
while $i < N$ **do**
 $b = \zeta(i + 1)$;
 recursively calculate $\Lambda_{b,m}$;
 if $b \in \mathcal{A}^c$ **then**
 $B_{b,m} = 0$;
 $i = i + 1$ and update PM using eq. (6);
 recursively propagate $B_{b,m}$;
 else
 $B_{b,m} = h(\Lambda_{b,m})$;
 $\overline{B}_{b,m} = B_{b,m} \oplus 1$;
 $i = i + 1$ and update PM and \overline{PM} using eq. (6);
 recursively propagate $B_{b,m}$ and $\overline{B}_{b,m}$;
 Add the state $(\Lambda, \overline{B}, i, \overline{PM})$ to the stack s ;
 end
 recursively propagate $B_{b,m}$;
 sort and prune s ;
 $(\Lambda, B, i, PM) = s.top()$;
end
 $\beta_i = B_{i,m} \quad i = 0, \dots, N - 1$;

$$\mathbf{x} = \mathbf{u} \cdot F_N = [\mathbf{u}_{\mathcal{I}}, \mathbf{u}_{\mathcal{J}}] \cdot \begin{bmatrix} F_{\frac{N}{2}} & \mathbf{0} \\ F_{\frac{N}{2}} & F_{\frac{N}{2}} \end{bmatrix}. \quad (7)$$

It follows that

$$\begin{aligned} \mathbf{x}_{\mathcal{I}} &= (\mathbf{u}_{\mathcal{I}} \oplus \mathbf{u}_{\mathcal{J}}) \cdot F_{\frac{N}{2}} \\ \mathbf{x}_{\mathcal{J}} &= \mathbf{u}_{\mathcal{J}} \cdot F_{\frac{N}{2}} \end{aligned} \quad (8)$$

This operation is called the basic folding and it splits the original code into two polar codes of the length $\frac{N}{2}$. A folded decoding factor graph corresponding to a polar code of the length $N = 8$ is shown in Fig. 3. This



decoder consists of two polar codes of the length $N = 4$ that can work in parallel, and a combination phase that is used to reconcile the upper and lower decoder. The upper and lower decoders are usually implemented as a SCD, while the choice of the combination decoder may vary. If the combination decoder is also a SCD, then the error performance would be equal to that of a classical SCD algorithm. Usually, the combination phase is implemented as a maximum likelihood (ML) decoder (Vangala et al., 2014b), (Li et al., 2014), a list decoder (Li et al., 2014) or a stack decoder (Xiang et al., 2020). The upper and lower decoders can be folded again in order to further reduce latency (Xiang et al., 2020). The multiple folding technique consists of applying the same operation several times (Kahraman et al., 2013), (Kahraman et al., 2014b).

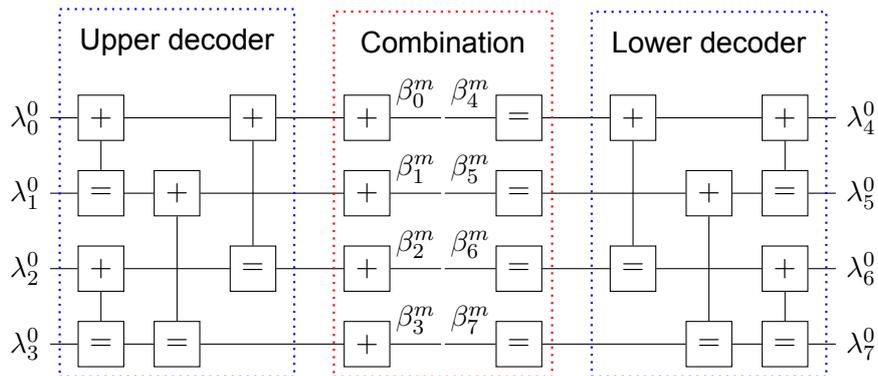


Figure 3 – Folded decoding factor graph for the polar codes of the length $N = 8$.
 Рис 3 – Граф сверточного кода для декодирования полярных кодов длиной $N = 8$.

Слика 3 – Граф савијеног фактора за декодовање поларних кодова дужине $N = 8$.

Note that a similar folding operation can be obtained by splitting the even and odd code bits into two parts and applying the polar transform in the opposite direction.

$$\begin{aligned} w_{\mathcal{E}} &= x_{\mathcal{E}} \oplus x_{\mathcal{O}} \\ w_{\mathcal{O}} &= x_{\mathcal{O}} \end{aligned} \tag{9}$$

This operation splits the code into two polar codes of half the length, similarly as before. An example of this operation is shown in Fig. 4 for

the case of a polar code of the length $N = 8$.

$$\begin{aligned} \mathbf{w}_E &= \mathbf{u}_E \cdot F_{\frac{N}{2}} \\ \mathbf{w}_O &= \mathbf{u}_O \cdot F_{\frac{N}{2}} \end{aligned} \tag{10}$$

By combining equations (9) and (10), we get

$$\begin{aligned} \mathbf{x}_E &= \mathbf{u}_E \cdot F_{\frac{N}{2}} \oplus \mathbf{u}_O \cdot F_{\frac{N}{2}} \\ \mathbf{x}_O &= \mathbf{u}_O \cdot F_{\frac{N}{2}} \end{aligned} \tag{11}$$

The same procedure can be applied to w_E and w_O to construct multiple folds.

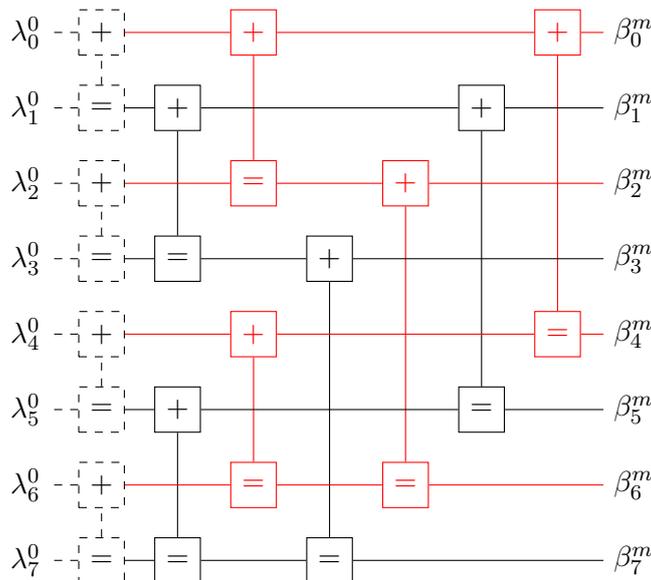


Figure 4 – Folded operation splits a polar code of the length $N = 8$ into two overlapping polar codes of the length $N = 4$.

Рис 4 – Операция свертки делит полярный код длиной $N = 8$ на два перекрывающихся полярных кода длиной $N = 4$.

Слика 4 – Савијање дели поларни код дужине $N = 8$ на два учешљана поларна кода дужине $N = 4$.

The new SCSD algorithm based on the odd-even folding is presented in Alg. 3. Let SCSD_E and SCSD_O represent the SCSD decoder of the even polar code and the odd polar code, respectively. Note that the reduced



latency SCSD decoder, presented in (Xiang et al., 2020), can be used as a component decoder in our algorithm. Let ENCO_ε represents the encoder of the even polar code. The described decoding algorithm performs slightly worse than the classical SCSD algorithm. The reason for this is the fact that we keep only the best surviving state of the SCSD_ε decoder and discard all the rest. In order to improve performance, we modify the component SCSD algorithm to return a list of D best candidate states. This can easily be implemented by popping the best state and rerunning the SCSD with a modified stack. The decoder output is selected amongst all D^2 paths based on the combined path metric or by checking the CRC. As D does not need to be very large, an ML decision can also be applied.

Algorithm 3: Bidirectional stack decoding algorithm

Input : λ, N
Output: β
 Initialize α as
for $i = 0, 2, 4, \dots, N - 1$ **do**
 | $\alpha_{i/2} = f(\lambda_i, \lambda_{i+1});$
end
 $\beta_\varepsilon = \text{SCSD}_\varepsilon(\alpha);$
 $w = \text{ENCO}_\varepsilon(\beta_\varepsilon);$
for $i = 0, 2, 4, \dots, N - 1$ **do**
 | $\alpha_{i/2} = g(\lambda_i, \lambda_{i+1}, w_{i/2});$
end
 $\beta_\mathcal{O} = \text{SCSD}_\mathcal{O}(\alpha);$

Simulation results

As we are primarily interested in URLLC applications, we consider the polar codes of rate 0.5 and block lengths 128 and 256. Although we present results only for short blocklengths, the proposed algorithm can be applied to polar codes of any length. The codes were constructed using the Bhattacharyya bound approximation method, and the simulation was done using the Monte Carlo method for different values of E_b/N_0 ranging from -1.6dB to 3.9dB with a step of 0.5dB . All simulations were run until a relative precision of $\delta = 0.05$ was reached.

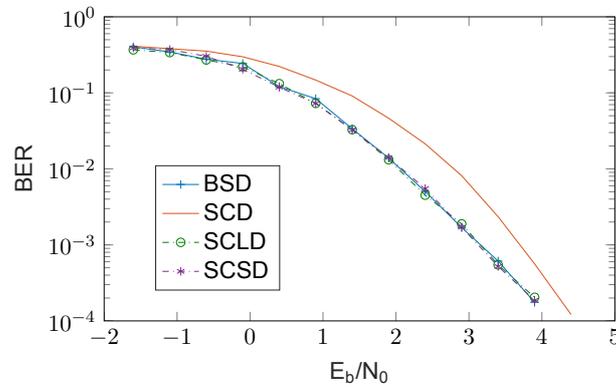


Figure 5 – BER of the $(128, 64)$ polar code using the BSD, SCD, SCLD and SCSD algorithms.

Рис 5 – BER кривая полярного кода $(128, 64)$ с использованием алгоритмов BSD, SCD, SCLD и SCSD.

Слика 5 – BER крива $(128, 64)$ полярног кода применом BSD, SCD, SCLD и SCSD алгоритма.

The SCLD algorithm was run with a list size of $L = 32$, while the SCSD algorithm was run with a stack of size $L = 100$. In the case of the bidirectional stack algorithm, the component SCSD algorithms were run with a stack of size $L = 32$, and they both returned a list of $D = 4$ candidate states. Out of $D^2 = 16$ possible candidates, we select the ML one. Fig. 5 shows the bit error rate (BER) as a function of E_b/N_0 for different decoders in the case of the $(128, 64)$ polar code.

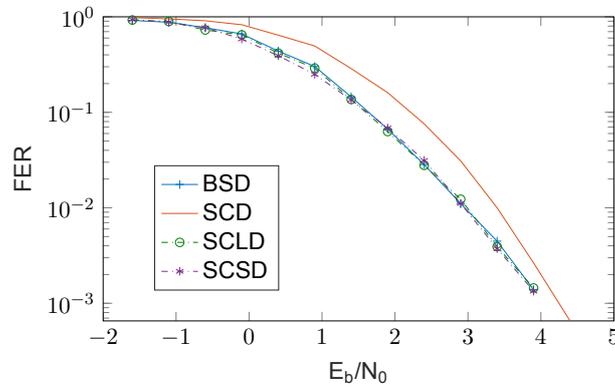


Figure 6 – FER of the (128, 64) polar code using the BSD, SCD, SCLD and SCSD algorithms.

Рис 6 – FER кривая полярного кода (128, 64) с использованием алгоритмов BSD, SCD, SCLD и SCSD.

Слика 6 – FER крива (128, 64) поларног кода применом BSD, SCD, SCLD и SCSD алгоритма.

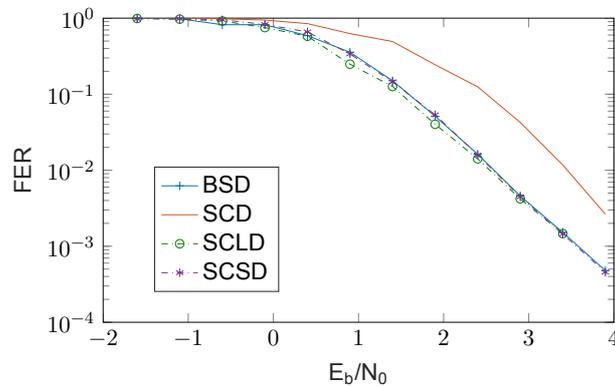


Figure 7 – FER of the (256, 128) polar code using the BSD, SCD, SCLD and SCSD algorithms.

Рис 7 – FER кривая полярного кода (256, 128) с использованием алгоритмов BSD, SCD, SCLD и SCSD.

Слика 7 – FER крива (256, 128) поларног кода применом BSD, SCD, SCLD и SCSD алгоритма.

Fig. 6 shows the frame error rate (FER) as a function of E_b/N_0 for different decoders in the case of the (128, 64) polar code. The FER of the (256, 128) polar code is shown in Fig. 7.

Based on these results, we see that the new BSD algorithm has the same error rate as the original SCSD algorithm. The speed of the proposed algorithm is higher than that of the original SCSD algorithm because of the smaller stack size and the fact that the algorithm is split into two parts, where each one is a SCSD of half the length of the original code.

Conclusion

In this paper, we presented a novel bidirectional stack decoding algorithm based on the folding technique. It was shown that the proposed algorithm has the same error performance as the existing algorithms. Future research will deal with further improving the bidirectional algorithm. The folding procedure is a powerful technique that can be used to construct a wide range of hybrid decoders. Different combinations of decoders could give better results - which is something that needs to be investigated. It is possible to further improve the proposed algorithm by applying the folding operation multiple times. The use of multiple CRC codes can also improve the performance of the decoder. An efficient hardware implementation of the proposed algorithm will also be developed.

References

- 3rd Generation Partnership Project. 2016. *Final report of 3GPP TSG RAN WG1 #87 v1.0.0*. [online]. Available at: <https://www.3gpp.org/DynaReport/TDocExMtg--R1-87--31665.htm> [Accessed: 1 February 2021].
- Alamdar-Yazdi, A. & Kschischang, F.R. 2011. A Simplified Successive-Cancellation Decoder for Polar Codes. *IEEE Communications Letters*, 15(12), pp. 1378–1380. Available at: <https://doi.org/10.1109/LCOMM.2011.101811.111480>.
- Arikan, E. 2008. A performance comparison of polar codes and Reed-Muller codes. *IEEE Communications Letters*, 12(6), pp. 447–449. Available at: <https://doi.org/10.1109/LCOMM.2008.080017>.
- Arikan, E. 2009. Channel Polarization: A Method for Constructing Capacity-Achieving Codes for Symmetric Binary-Input Memoryless Channels. *IEEE Transactions on Information Theory*, 55(7), pp. 3051–3073. Available at: <https://doi.org/10.1109/TIT.2009.2021379>.

Arikan, E. 2010. Polar codes: A pipelined implementation. In: *Proc. 4th Int. Symp. on Broad. Commun. ISBC 2010*. pp. 11–14.

Aurora, H., Condo, C. & Gross, W.J. 2018. Low-Complexity Software Stack Decoding of Polar Codes. In: *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*. pp. 1–5. Available at: <https://doi.org/10.1109/ISCAS.2018.8351832>.

Aurora, H. & Gross, W.J. 2018. *Towards Practical Software Stack Decoding of Polar Codes*. [online]. Available at: <https://arxiv.org/abs/1809.03606> [Accessed: 1 February 2021].

Balatsoukas-Stimming, A., Parizi, M.B. & Burg, A. 2015. LLR-Based Successive Cancellation List Decoding of Polar Codes. *IEEE Transactions on Signal Processing*, 63(19), pp. 5165–5179. Available at: <https://doi.org/10.1109/TSP.2015.2439211>.

Forney, G.D. 2001. Codes on graphs: normal realizations. *IEEE Transactions on Information Theory*, 47(2), pp. 520–548. Available at: <https://doi.org/10.1109/18.910573>.

Hanif, M. & Ardakani, M. 2017. Fast Successive-Cancellation Decoding of Polar Codes: Identification and Decoding of New Nodes. *IEEE Communications Letters*, 21(11), pp. 2360–2363. Available at: <https://doi.org/10.1109/LCOMM.2017.2740305>.

Hashemi, S.A., Condo, C. & Gross, W.J. 2016. Simplified Successive-Cancellation List decoding of polar codes. In: *2016 IEEE International Symposium on Information Theory (ISIT)*. pp. 815–819. Available at: <https://doi.org/10.1109/ISIT.2016.7541412>.

Hashemi, S.A., Mondelli, M., Fazeli, A., Vardy, A., Cioffi, J. & Goldsmith, A. 2020. *Parallelism versus Latency in Simplified Successive-Cancellation Decoding of Polar Codes*. [online]. Available at: <https://arxiv.org/abs/2012.13378> [Accessed: 1 February 2021].

Huang, Y., Zhang, Z., You, X. & Zhang, C. 2018. Efficient Folded SC Polar Line Decoder. In: *2018 IEEE 23rd International Conference on Digital Signal Processing (DSP)*. pp. 1–5. Available at: <https://doi.org/10.1109/ICDSP.2018.8631807>.

Kahraman, S., Viterbo, E. & Çelebi, M.E. 2013. Folded tree maximum-likelihood decoder for Kronecker product-based codes. In: *2013 51st Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. pp. 629–636. Available at: <https://doi.org/10.1109/Allerton.2013.6736584>.

Kahraman, S., Viterbo, E. & Çelebi, M.E. 2014a. Folded successive cancellation decoding of polar codes. In: *2014 Australian Communications Theory Workshop (AusCTW)*. pp. 57–61. Available at: <https://doi.org/10.1109/AusCTW.2014.6766428>.

Kahraman, S., Viterbo, E. & Çelebi, M.E. 2014b. Multiple Folding for Successive Cancellation Decoding of Polar Codes. *IEEE Wireless Communications Letters*, 3(5), pp. 545–548. Available at: <https://doi.org/10.1109/LWC.2014.2343970>.

Li, B., Shen, H. & Tse, D. 2012. An Adaptive Successive Cancellation List Decoder for Polar Codes with Cyclic Redundancy Check. *IEEE Communications Letters*, 16(12), pp. 2044–2047. Available at: <https://doi.org/10.1109/LCOMM.2012.111612.121898>.

Li, B., Shen, H. & Tse, D. 2013. *Parallel Decoders of Polar Codes*. [online]. Available at: <https://arxiv.org/abs/1309.1026v1> [Accessed: 1 February 2021].

Li, B., Shen, H., Tse, D. & Tong, W. 2014. Low-latency polar codes via hybrid decoding. In: *2014 8th International Symposium on Turbo Codes and Iterative Information Processing (ISTC)*. pp. 223–227. Available at: <https://doi.org/10.1109/ISTC.2014.6955118>.

Niu, K. & Chen, K. 2012b. CRC-Aided Decoding of Polar Codes. *IEEE Communications Letters*, 16(10), pp. 1668–1671. Available at: <https://doi.org/10.1109/LCOMM.2012.090312.121501>.

Niu, K. & Chen, K. 2012a. Stack decoding of polar codes. *Electronics Letters*, 48, pp. 695–697(2). Available at: <https://doi.org/10.1049/el.2012.1459>.

Sarkis, G., Giard, P., Vardy, A., Thibeault, C. & Gross, W.J. 2014. Fast Polar Decoders: Algorithm and Implementation. *IEEE Journal on Selected Areas in Communications*, 32(5), pp. 946–957. Available at: <https://doi.org/10.1109/JSAC.2014.140514>.

Senk, V. & Radivojac, P. 1997. The bidirectional stack algorithm. In: *Proceedings of IEEE International Symposium on Information Theory*. pp. 500–. Available at: <https://doi.org/10.1109/ISIT.1997.613437>.

Tal, I. & Vardy, A. 2015. List Decoding of Polar Codes. *IEEE Transactions on Information Theory*, 61(5), pp. 2213–2226. Available at: <https://doi.org/10.1109/TIT.2015.2410251>.

Vangala, H., Viterbo, E. & Hong, Y. 2014a. Improved multiple folded successive cancellation decoder for polar codes. In: *2014 XXXIth URSI General Assembly and Scientific Symposium (URSI GASS)*. pp. 1–4. Available at: <https://doi.org/10.1109/URSIGASS.2014.6929238>.

Vangala, H., Viterbo, E. & Hong, Y. 2014b. A new multiple folded successive cancellation decoder for polar codes. In: *2014 IEEE Information Theory Workshop (ITW 2014)*. pp. 381–385. Available at: <https://doi.org/10.1109/ITW.2014.6970858>.

Vangala, H., Viterbo, E. & Hong, Y. 2015. *A Comparative Study of Polar Code Constructions for the AWGN Channel*. [online]. Available at: <https://arxiv.org/abs/1501.02473> [Accessed: 1 February 2021].



Won, J.W. & Ahn, J.M. 2020. 3GPP URLLC patent analysis. *ICT Express*. Available at: <https://doi.org/10.1016/j.ict.2020.09.001>.

Xiang, L., Kaykac Egilmez, Z.B., Maunder, R.G. & Hanzo, L. 2019. CRC-Aided Logarithmic Stack Decoding of Polar Codes for Ultra Reliable Low Latency Communication in 3GPP New Radio. *IEEE Access*, 7, pp. 28559–28573. Available at: <https://doi.org/10.1109/ACCESS.2019.2901596>.

Xiang, L., Zhong, S., Maunder, R.G. & Hanzo, L. 2020. Reduced-Complexity Low-Latency Logarithmic Successive Cancellation Stack Polar Decoding for 5G New Radio and Its Software Implementation. *IEEE Transactions on Vehicular Technology*, 69(11), pp. 12449–12458. Available at: <https://doi.org/10.1109/TVT.2020.3026915>.

ДВУНАПРАВЛЕННОЕ СТЕКОВОЕ ДЕКОДИРОВАНИЕ ПОЛЯРНЫХ КОДОВ

*Александр Дж. Миня, Душан П. Добромиров, корреспондент,
Войин И. Шенк*

Нови-Садский университет, факультет технических наук,
кафедра телекоммуникаций и обработки сигналов,
г. Нови-Сад, Республика Сербия

РУБРИКА ГРНТИ: 28.00.00 КИБЕРНЕТИКА:

28.21.00 Теория информации;

28.21.19 Теория кодирования,

27.00.00 МАТЕМАТИКА:

27.03.00 Основания математики и
математическая логика;

27.03.17 Алгоритмы и вычислимые функции

ВИД СТАТЬИ: оригинальная научная статья

russian

Резюме:

Введение/цель: В данной статье представлен новый алгоритм декодирования полярных кодов с уменьшенной задержкой, вдохновленный двунаправленным стековым декодированием сверточных кодов, разработанный с помощью метода свертки.

Методы: Алгоритм стекового декодирования (также известный как стековый поиск), применяется при декодировании древовидных кодов, декодировании списка, разработанного Питером Элиасом, методе свертки, применяемой для минимизации задержки алгоритмов

декодирования. Моделирование проводилось методом Монте-Карло.

Результаты: Разработан новый алгоритм декодирования полярных кодов, подходящий для параллельного внедрения. В статье представлены сравнительный анализ нового алгоритма с существующими и результаты моделирования.

Выводы: Полярные коды – это класс кодов, который достигает пропускной способности каналов, используемый в качестве основной схемы кодирования для каналов управления в 5G New Radio. Основным алгоритмом декодирования полярных кодов является последовательное декодирование списка отмен (англ. Successive cancellation decoding - SCD). Данный алгоритм отличается простотой и отличными характеристиками относительно длинных кодов, но он ненадежен относительно коротких и средних кодов. В данной связи предлагаются несколько алгоритмов декодирования, с целью улучшения характеристик полярных кодов. Последовательное декодирование списка отмен в сочетании с циклическим избыточным кодом обеспечивает точное исправление ошибок, но за счет высокой сложности внедрения. Стековое декодирование отмен обеспечивает аналогичную точность при исправлении ошибок, но при меньшей сложности внедрения. Помимо высокой надежности, новые приложения в критических ситуациях требуют небольших задержек, поэтому необходимо минимизировать время работы декодера. В данной статье предлагается новый алгоритм двунаправленного декодирования, который обеспечивает значительно меньшую задержку и аналогичную надежность, как у наилучших существующих алгоритмов. В статье представлен сравнительный анализ нового алгоритма с существующими, а также результаты моделирования.

Ключевые слова: полярные коды, последовательное декодирование, стековое декодирование, декодирование списка отмен, свертка.



БИДИРЕКЦИОНО СТЕК ДЕКОДОВАЊЕ ПОЛАРНИХ КОДОВА

Александар Ђ. Миња, Душан П. Добромиров, аутор за преписку,
Војин И. Шенк

Универзитет у Новом Саду, Факултет техничких наука, Катедра за телекомуникације и обраду сигнала, Нови Сад, Република Србија

ОБЛАСТ: телекомуникације, рачунарске науке

ВРСТА ЧЛАНКА: оригинални научни рад

serbian

Сажетак:

Увод/циљ: Рад уводи нови алгоритам за декодовање поларних кодова са смањеним кашњењем који је инспирисан бидирекционим стек декодовањем класичних конволуционих кодова, а заснован је на техници савијања (енг. *foldng*).

Методe: Стек алгоритма (познат и као стек претрага) погодан је за декодовање кодова са структуром стабла, декодер са листом. Ова техника савијања, коју је увео Петар Елиас, користи се за смањивање кашњења алгоритама за декодовање. Симулације су рађене применом поступка Монте Карло.

Резултати: Развијен је нови алгоритам за декодовање поларних кодова који је погодан за паралелну имплементацију. Нови алгоритам упоређен је са постојећим, а резултати симулације су приказани.

Закључак: Поларни кодови су класа кодова која достиже капацитет канала, и усвојена је за кодовање контролних канала у 5G новом радију. Главни алгоритам декодовања за поларне кодове јесте секвенцијални алгоритам - тзв. декодер са сукцесивним поништавањем (енг. *Successive cancellation decoding - SCD*). Овај алгоритам има ниску комплексност и одличне перформансе у случају кодова са великом дужином, али и врло малу поузданост на кратким и средњим дужинама. Предложено је неколико алгоритама за декодовање како би се побољшале перформансе поларних кодова. Декодер са листом и сукцесивним поништавањем (енг. *Successive cancellation list decoding - SCLD*), заједно са цикличким кодом за проверу грешака

(енг. *Cyclic redundancy check - CRC*), пружа веома добре перформансе при исправљању грешака, али по цену велике комплексности имплементације. Стек декодер са сукцесивним поништавањем (енг. *Successive cancellation stack decoding - SCSD*) пружа сличне перформансе по питању вероватноће грешке при нижој комплексности. Поред ниске вероватноће грешке, нове апликације за рад у критичним ситуацијама захтевају и мало кашњење, па је потребно смањити време рада декодера. У раду је предложен нови бидирекциони алгоритам за декодовање, који постиже знатно мање кашњење уз приближно исту поузданост као и најбољи постојећи алгоритми. Нови алгоритам је упоређен са постојећим алгоритмима, а резултати симулације су приказани.

Кључне речи: поларни кодови, секвенцијални декодер, стак декодовање и декодер са листом, савијање.

Paper received on / Дата получения работы / Датум пријема чланка: 08.02.2021.
Manuscript corrections submitted on / Дата получения исправленной версии работы / Датум достављања исправки рукописа: 10.03.2021.

Paper accepted for publishing on / Дата окончательного согласования работы / Датум коначног прихватања чланка за објављивање: 12.03.2021.

© 2021 The Authors. Published by Vojnotehnički glasnik / Military Technical Courier (<http://vtg.mod.gov.rs>, <http://втр.мо.унр.спб>). This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/3.0/rs/>).

© 2021 Авторы. Опубликовано в "Военно-технический вестник / Vojnotehnički glasnik / Military Technical Courier" (<http://vtg.mod.gov.rs>, <http://втр.мо.унр.спб>). Данная статья в открытом доступе и распространяется в соответствии с лицензией "Creative Commons" (<http://creativecommons.org/licenses/by/3.0/rs/>).

© 2021 Аутори. Објавио Војнотехнички гласник / Vojnotehnički glasnik / Military Technical Courier (<http://vtg.mod.gov.rs>, <http://втр.мо.унр.спб>). Ово је чланак отвореног приступа и дистрибуира се у складу са Creative Commons лиценцом (<http://creativecommons.org/licenses/by/3.0/rs/>).

