A Simplified Decoding Method of Polar Codes Based on Hypothesis Testing

He Sun, Rongke Liu, *Senior Member, IEEE*, and Chenyu Gao

***Abstract*— In this letter, a simplified successive cancellation decoding algorithm of polar codes is proposed, where a hard decision rule is adopted by exploiting the prior information in frozen bits for better decoding performance. Considering the difference of the node reliability, a hypothesis-testing-based strategy is designed to select reliable unstructured nodes for hard decision. Then the reliable nodes are decoded by the proposed hard decision method without serial recursion, which reduces the decoding latency. Simulation results show the advantages of the proposed method in terms of the decoding latency and error-correction ability compared with the existing methods.**

***Index Terms*— Polar codes, hypothesis testing, hard decision.**

I. Introduction

POLAR codes are capacity-achieving channel codes with a simple encoder and the successive cancellation (SC) decoder [1]. However, the SC decoder suffers from high decoding latency due to the serial nature of decoding. To address this problem, the simplified SC (SSC) decoder is proposed, in which the Rate-0 and Rate-1 nodes are defined to be decoded instantly by hard decision. Based on particular structures of rate-r nodes, fast-SSC [2] defines the single-parity-check (SPC) node, the repetition (REP) node and some merged nodes [3], which include the REP-SPC, 0-SPC and 01 nodes. In [4], five new constituent nodes are proposed to increase the decoding speed. These methods leverage specific node structures to reduce latency. However, the nodes without specific structures, called unstructured nodes [5], cannot be simplified by these structured SC decoding methods or their list decoding versions [6], [7]. Independent of specific node structures, the SC decoders with look-ahead increase the parallelism of decoding operations from the perspective of decoding schedules [8] or improve the decoding performance by exploiting multiple codewords [9]. To simplify the decoding of unstructured nodes, the NEP-SSC [5] adopts a threshold based on the Monte Carlo simulation to identify reliable nodes. Then hard decision is implemented directly on reliable unstructured nodes. However, there is at least one frozen bit in each unstructured node, which provides more prior information. The direct implementation of hard decision on unstructured nodes ignores the prior information. Therefore, there is still potential for performance improvement.

Manuscript received October 27, 2019; revised December 2, 2019; accepted December 29, 2019. Date of publication January 1, 2020; date of current version March 10, 2020. This work was supported by the National Natural Science Foundation of China (No. 61871009). The associate editor coordinating the review of this letter and approving it for publication was M.-R. Sadeghi. *(Corresponding author: Rongke Liu.)*

The authors are with the School of Electronic and Information Engineering, Beihang University, Beijing 100191, China (e-mail: sunhe1710@buaa.edu.cn; rongke_liu@buaa.edu.cn; gaochenyu@buaa.edu.cn).

Digital Object Identifier 10.1109/LCOMM.2019.2963645

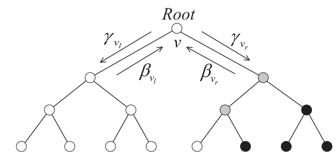


Fig. 1. The SC decoding tree of polar codes with $N = 8$.

This work is aimed to simplify the decoding of unstructured nodes and improve decoding performance, where a hypothesis-testing-based hard decision (HTHD) algorithm is proposed. In the HTHD method, a novel hard decision rule exploits the prior information of frozen bits to improve decoding performance. In view of the difference in the node reliability, a hypothesis testing strategy is devised to select reliable unstructured nodes without Monte Carlo simulation, which has better robustness and universal applicability. Then reliable unstructured nodes are decoded by the proposed hard decision method without complex recursion, thereby reducing the decoding latency. Moreover, the HTHD method can be integrated with the structured methods for lower latency. Simulation results show that the HTHD method significantly reduces the decoding latency with better error-correction performance.

II. Preliminaries

A. Polar Codes

A polar code of length $N = 2^n$ is generated by $x_1^N = u_1^N G_N$, where $u_1^N = (u_1, u_2, \ldots, u_N)$ is the input sequence and $x_1^N = (x_1, x_2, \ldots, x_N)$ is the codeword sequence. The generator matrix is G_N. After encoding, the codeword x_1^N is modulated by binary phase shift keying (BPSK) and sent to the channel, and then the channel outputs y_1^N. To construct a polar code (N, K) with code length N and K information bits, the K information bits are assigned to the reliable sub-channels. The remaining $N-K$ bits, called frozen bits, are set to values known by both the encoder and decoder. In this letter, we assume that the values of frozen bits are all set to 0, which is called the freezing constraint. Let $\mathcal{F}$ and $\mathcal{F}^c$ denote the index sets of the frozen bits and the information bits, respectively.

B. SC and SSC Decoder

Fig. 1 shows the decoding procedure in a binary tree. The black and white leaf nodes are information and frozen bits, respectively. The white, gray and black nodes represent the Rate-0, Rate-r and Rate-1 nodes, respectively. The local decoder begins at the root node, which is initialized by the received messages $\gamma_{v_{root}} = [\lambda_v^{(1)}, \ldots, \lambda_v^{(N)}]$, where

$\lambda^{(i)} = \log\left[\frac{\Pr(y_i|x_i=0)}{\Pr(y_i|x_i=1)}\right]$ is the log-likelihood ratio (LLR). The decoder at node v calculates γ_{v_l} and starts the calculation of its left node v_l by

$$\lambda_{v_l}^{(i)} = 2\tanh^{-1}\left\{\tanh\left[\frac{\lambda_v^{(2i-1)}}{2}\right]\tanh\left[\frac{\lambda_v^{(2i)}}{2}\right]\right\}, \quad (1)$$

where $i = 1, \ldots, 2^{n-d_v-1}$ and d_v represents the depth of the node v. The vector γ_{v_l} can be calculated in parallel with sufficient processing elements in one time step [5], where time steps denote the sum of clock cycles required by decoding operations [2], [5], [6]. After receiving decision results $\hat{\beta}_{v_l} = \left[\hat{\beta}_{v_l}^{(1)}, \ldots, \hat{\beta}_{v_l}^{\left(2^{n-d_v-1}\right)}\right]$ from v_l, node v starts the calculation of γ_{v_r} on its right descendant node v_r by

$$\lambda_{v_r}^{(i)} = \lambda_v^{(2i-1)}\left(1 - 2\hat{\beta}_{v_l}^{(i)}\right) + \lambda_v^{(2i)}. \quad (2)$$

When the node v receives $\hat{\beta}_{v_r}$ from v_r, it obtains $\hat{\beta}_v$ by

$$\hat{\beta}_v^{(2i)} = \hat{\beta}_{v_r}^{(i)}, \quad \hat{\beta}_v^{(2i-1)} = \hat{\beta}_{v_l}^{(i)} \oplus \hat{\beta}_{v_r}^{(i)}, \quad (3)$$

where $\oplus$ is the exclusive-or operation. After obtaining $\hat{\beta}_v$, the decoding procedure of v is finished, and the same operations are carried out at the next node. Especially, the decoder performs hard decision at the information leaf nodes by

$$\hat{\beta}_{v_{leaf}} = \begin{cases} 0, & \text{if } \lambda_{v_{leaf}} \geq 0, \\ 1, & \text{otherwise.} \end{cases} \quad (4)$$

To reduce the decoding latency, the SSC decoder is proposed, where hard decision is performed at the Rate-0 and Rate-1 nodes immediately, which omits the subsequent decoding recursion in the descendant nodes.

III. Hypothesis-Testing-Based Hard Decision Method for Unstructured Nodes

The freezing constraint limits the valid codeword space. However, the NEP-SSC [5] performs hard decision directly in reliable nodes, which might obtain invalid codewords that do not meet freezing constraints. To deal with this problem, we design a novel hard decision method, which exploits the prior information of frozen bits and ensures that the decision results are compliant with the freezing constraint.

A. A Novel Hard Decision Method

The original hard-decision in node $v_1^c = (v_1, \ldots, v_c)$ is

$$\hat{v}_1^c = \arg\left[\max_{\hat{v}_1^c \in \mathbb{B}^c}\left(p(\hat{v}_1)\, p(\hat{v}_2) \ldots p(\hat{v}_c)\right)\right], \quad (5)$$

where $\mathbb{B}$ denotes the set of binary numbers. For $i \in \{1, \ldots, c\}$, $p(\hat{v}_i) = W\left(y_1^N, \hat{u}|\hat{v}_i\right)$. $W\left(y_1^N, \hat{u}|\hat{v}_i\right)$ is the transition probability [1]. The relative order in (5) remains unchanged after dividing the products by a same number. Therefore, we have

$$\hat{v}_1^c = \arg\left[\max_{\hat{v}_1^c \in \mathbb{B}^c}\left(\frac{p(\hat{v}_1)\, p(\hat{v}_2) \ldots p(\hat{v}_c)}{p(v_1 = 1) \ldots p(v_c = 1)}\right)\right]. \quad (6)$$

Another expression of (6) in the logarithmic domain is

$$\hat{v}_1^c = \arg\left[\max_{\hat{v}_1^c \in \mathbb{B}^c}\sum_{i=1}^{c}(1 - \hat{v}_i)\,\lambda_i\right], \quad (7)$$

where $\lambda_i = \ln\left[p(\hat{v}_i = 0)/p(\hat{v}_i = 1)\right]$ denotes the LLR of v_i. Let $\mathcal{D}$ express the logarithmic-domain metric $\sum_{i=1}^{c}(1 - \hat{v}_i)\,\lambda_i$. Due to the freezing constraints, the correct decoding result must meet $u_i = 0$ for $i \in \mathcal{F}$, where $u_1^c = v_1^c G_c$. Owing to the short node length c, a look-up table is used to convert the decision results $\hat{v}_1^c$ into original sequence $\hat{u}_1^c$ for the freezing check in one time step. The table stores the mapping relationships between the $\hat{u}_1^c$ and the codeword $\hat{v}_1^c$. For a reliable node with length c, there are 2^c groups of one-to-one mapping relationships. All nodes with the same length share the same look-up table. The proposed hard-decision rules are

$$\hat{u}_1^c = \arg\left[\max_{u_1^c \in \mathbb{B}^c}(\mathcal{D})\right] \quad \text{s.t.}\, \hat{u}_i = 0, \quad i \in \mathcal{F}. \quad (8)$$

In (8), the decoding path with the largest metric among the candidate paths which satisfy the freezing constraint is selected as the result of the hard decision. However, the number of candidate paths is 2^c. Excessive complexity is involved in decoding all the candidate paths. To solve this problem, we propose a heuristic method to construct candidate paths with large metrics. Obviously, the path with the largest $\mathcal{D}$ is

$$\hat{v}_i = \begin{cases} 0 & \text{if } \lambda_i \geq 0, \\ 1 & \text{if } \lambda_i < 0. \end{cases} \quad (9)$$

Let $\hat{v}_1^c[1]$ denote the decision results with the largest $\mathcal{D}$. We propose the following method to obtain candidate paths.

Proposition 1: For $2 \leq k \leq 3$ and $j \in \{1, 2, \ldots, c\}$, the path $\hat{v}_1^c[k]$ with the k-th largest metric can be represented as

$$\hat{v}_j[k] = \begin{cases} \hat{v}_j[1] \oplus 1 & \text{when } j = \operatorname{argmin}_{k-1,\, i \in \{1,\ldots,c\}}\{|\lambda_i|\}, \\ \hat{v}_j[1] & \text{otherwise,} \end{cases} \quad (10)$$

where $\operatorname{argmin}_{k,\, i \in \{1,\ldots,c\}}\{|\lambda_i|\}$ is the index of the k-th smallest $|\lambda|$.

Proof: Based on the definition of $\mathcal{D}$, the maximum is

$$\mathcal{D}_1 = \frac{1}{2}\sum_{i=1}^{c}[1 + \operatorname{sign}(\lambda_i)]\lambda_i. \quad (11)$$

It can also be observed that the metric $\mathcal{D}$ is determined by the estimated results $\hat{v}_1^c$ and LLRs. Since the LLRs are fixed, other candidates can be obtained by flipping the estimated bit $\hat{v}_t$. For each $\hat{v}_t$, when it flips into $\hat{v}_t \oplus 1$, we have

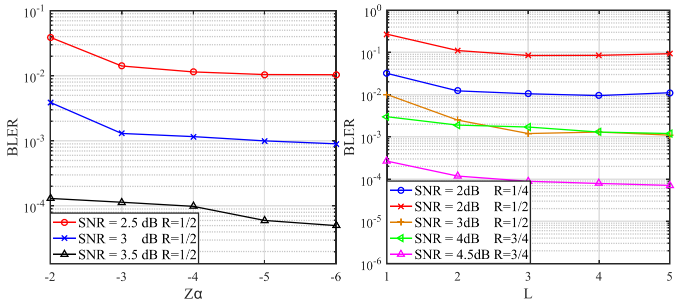
$$\begin{aligned} \mathcal{D}(v_t) &= \sum_{i=1, i \neq t}^{c}(1 - \hat{v}_i)\lambda_i + (1 - \hat{v}_t \oplus 1)\,\lambda_t, \\ &= \mathcal{D}_1 - |\lambda_t|. \end{aligned} \quad (12)$$

Therefore, the second largest metric is

$$\mathcal{D}_2 = \mathcal{D}_1 - \min_{t \in Z_1^c}\{|\lambda_t|\}, \quad (13)$$

where $Z_1^c = \{1, 2, \ldots, c\}$. The second largest path can be obtained by flipping $\hat{u}_j$ of $\hat{v}_1^c[1]$, where $j = \arg\min_{t \in Z_1^c}\{|\lambda_t|\}$. Then, the third largest metric is

$$\mathcal{D}_3 = \mathcal{D}_1 - \min_{t \in Z_1^c} 2\{|\lambda_t|\}, \quad (14)$$



(a) (b)

Fig. 2. BLER performance with different z_α (a) and L (b). ($N = 1024$ bit).

where $\min\ 2\{\cdot\}$ denotes the second smallest value. The third largest path can be obtained by flipping $\hat{v}_j$ in $\hat{v}_1^c[1]$, where $j = \arg\min_{t \in Z_1^c} 2\{|\lambda_t|\}$. Then we obtain (10). ■

Eq. (10) indicates that the path with the k-th largest $\mathcal{D}$ can be obtained by flipping the bit with the (k-1)-th smallest $|\lambda|$ in the largest path $\hat{v}_1^c[1]$. The Chase decoding [10] and the fast list decoder [7] construct limited candidate codewords by considering several unreliable bits. We adopt this method to limit the complexity of hard decision decoding. Since the paths with small metrics are unreliable, we consider the first L paths with the largest metrics. Through simulation in Fig. 2(b), we found that a good error-correction ability can be obtained with $L = 3$. Owing to building multiple paths and exploiting the freezing constraints, the performance of hard decision decoding is further improved.

B. Evaluation of Node Reliability With Hypothesis Testing

Due to the channel polarization and the noise, the node reliability varies among unstructured nodes. In [5], a threshold based on the bit error probability and a multiplicative factor is adopted to select reliable nodes. However, the factor is determined by Monte Carlo simulation, which might cause instability. To solve this problem with a more robust method, we propose a hypothesis-testing-based method to select reliable nodes for hard decision, where a closed-form expression of a dynamic threshold is devised without Monte Carlo simulation.

To measure the reliability of node v_1^c, we define the null hypothesis and the alternative hypothesis as follows

$$\mathcal{H}_0 : v_1^c \in \Theta, \quad \mathcal{H}_1 : v_1^c \in \Theta^c, \tag{15}$$

where Θ is the set of reliable nodes. When $\mathcal{H}_0$ is accepted, the hard decision is performed on node v_1^c. Let v_i denote the element in node v_1^c, where $i \in \{1, \ldots, c\}$. Because the absolute value of LLR monotonically decreases with the increase of node error probability, we adopt the LLR as the test statistic. Let T_i denote the threshold for v_i. If $\forall\ i \in \{1, \ldots, c\}$, $|\lambda_{v_i}| > T_i$, the $\mathcal{H}_0$ will be accepted and hard decision is performed. Let $\hat{u}_1^t$ be the estimated bits. The bit error probability of hard decision is

$$\begin{aligned} &P_{ed}\left(\hat{\beta}_{v_i} \neq \beta_{v_i} | y_1^N, \hat{u}_1^t = u_1^t\right) \\ &= P\left(\lambda_{v_i} < -T_i\right) P\left(\beta_{v_i} = 0\right) + P\left(\lambda_{v_i} > T_i\right) P\left(\beta_{v_i} = 1\right). \end{aligned} \tag{16}$$

Assuming that an all-zero codeword $u_1^N = \mathbf{0}$ is encoded and then modulated by BPSK, the distribution of LLR arising in the Arikan polarizing transformation can be approximated by Gaussian, i.e., $\lambda_i \sim \mathcal{N}(\mu_i, 2\mu_i)$ in binary-input additive white Gaussian noise (BAWGN) channel [11], [12]. Given code length N and rate R, the mean value μ can be calculated by Gaussian Approximation (GA) offline at given signal-to-noise ratio (SNR). Then the probability in (16) is represented as

$$P_{ed} = P\left(\lambda_{v_i} < -T_i\right). \tag{17}$$

To restrict decoding performance loss, we set

$$P\left(\lambda_{v_i} < -T_i\right) \leq \alpha, \tag{18}$$

where α is the significance level of hypothesis testing. To maximize the number of nodes simplified by hard decision, it is recommended that $P_{ed} = \alpha$, which is equivalent to

$$P\left(\lambda_{v_i} < -T_i\right) = P\left(z_i < z_\alpha\right) = \alpha, \tag{19}$$

where $z_i = \frac{\lambda_{v_i} - \mu_i}{\sqrt{2\mu_i}} \sim \mathcal{N}(0, 1)$, z_α is the quantile of standard normal distribution at level α. Therefore, we have

$$\lambda_{v_i} < z_\alpha \sqrt{2\mu_i} + \mu_i. \tag{20}$$

The threshold T_i can be calculated by

$$T_i = -z_\alpha \sqrt{2\mu_i} - \mu_i. \tag{21}$$

The proposed threshold is a function of the quantile and the mean value, where the influences of code length, code rate and SNR on node reliability are comprehensively considered. Consequently, the threshold has better robustness and universal applicability. There are $N(\log N\text{-}2)$ threshold values.

If the LLRs on node $v_1^c = (v_1, \ldots, v_c)$ satisfy $|\lambda_i| > T_i$ for $i = 1, 2, \ldots, c$, the $\mathcal{H}_0$ is accepted and node v will be decoded by the proposed hard decision method, where the computations for the descendant nodes are avoided and the decoding latency is reduced. Otherwise, this node is treated as unreliable node and the decoding in its descendant nodes will be performed.

C. Decoding Latency Analysis

To evaluate node reliability, comparison operation is performed simultaneously with the calculation of γ [5]. Besides, the proposed hard decision method contains flipping bits to obtain candidate paths, checking the freezing constrains and selecting the correct path, which takes three time steps.

Moreover, for a reliable node with length N_v, the calculation in its $2N_v - 2$ descendant nodes is omitted, which reduces the decoding latency and complexity. Because the dependence on node structures is overcame, the N_v is expanded in high SNR region, thereby further reducing the decoding latency.

IV. Simulation Results

Simulation is performed to evaluate the block error rate (BLER) and the decoding latency. Codewords are modulated by BPSK in the BAWGN channel. Polar codes (1024, 512), (4096, 2048) are designed by GA. Since the length of the node affects the node partition, we set the minimum node length to 4 or 8 in simulation for our proposed HTHD method.

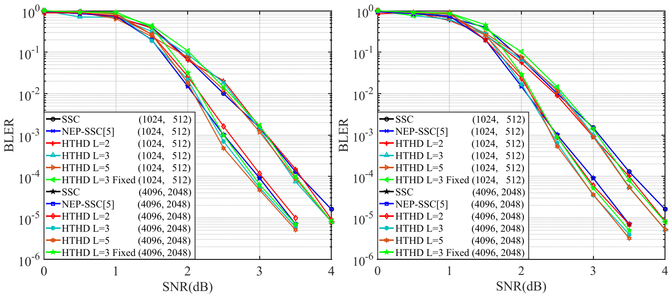


Fig. 3. Comparisons of the BLER with $z_\alpha = -3$ (a) and $z_\alpha = -5$ (b).

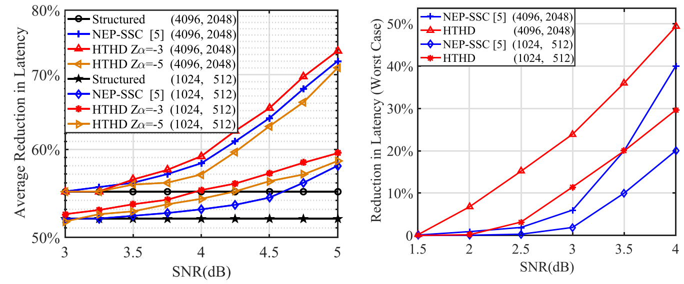


Fig. 4. (a) Depicts the average latency reduction proportion and (b) shows the worst case of reduction proportion in decoding latency with $z_\alpha = -3$.

A. Decoding performance

Fig. 2 depicts the BLER with different z_α and L. The BLER tends to be stable with the decrease of z_α and the increase of L. A small z_α corresponds to a strict constraint. According to the results in Fig. 2 (a), we set $z_\alpha \in \{-3, -5\}$ in simulation. Confirming to Section III-A, when $L > 3$, there is no significant BLER gain as L increases.

Fig. 3 shows that the BLER of the HTHD method is nearly the same with that of the SSC decoder. For $L = 3$, the HTHD achieves approximately 0.2 dB SNR gain in high SNR region because the proposed hard decision rules exploit the freezing constraints and utilize multiple paths synthetically. Besides, the BLER with $z_\alpha{=}-5$ is slightly lower than that with $z_\alpha{=}-3$ because a smaller z_α corresponds to a tighter constraint, which indicates that the hypothesis testing can improve the decoding performance by setting strict test conditions. Considering that the accurate SNR often cannot be obtained in actual systems, we also give the BLER performance of the HTHD method with a fixed set of thresholds and a fixed reliability sequence constructed at SNR $=$ 3.5 dB. In Fig. 3, the curves labeled by "fixed" show that the proposed HTHD method works well with the fixed construction of thresholds and polar codes.

B. Decoding Latency

Assuming that the processing elements are capable of calculating the message vectors simultaneously, the calculation of messages γ_v needs one time step. In the structured methods, a rate-r node v takes two time steps to calculate γ_{v_l} and γ_{v_r}, while waiting for t_l and t_r time steps during the information transfer process in its left and right descendant nodes, respectively [5]. According to Section III-C, the proposed hard decision on an unstructured node takes three time steps, which reduces $t_l + t_r - 1$ time steps relative to the structured decoder.

Since the proposed HTHD method simplifies the decoding of unstructured nodes, it can be combined with the structured methods for lower latency. Comparisons are made among the structured decoder, the NEP-SSC and the HTHD methods. Fig. 4 (a) depicts the reduction in terms of the decoding latency of the three methods relative to the SSC decoder, respectively. For $N{=}1024$ and $z_\alpha{=}-5$, the HTHD method yields lower latency with SNR gain of 0.2 dB than the NEP-SSC at BLER$=10^{-5}$. For $N{=}4096$, the quantile realizes the tradeoff between the BLER and latency. The average latency reduction by HTHD with $z_\alpha{=}-3$ is larger than that with $z_\alpha{=}-5$ because a small z_α leads to a small set of reliable nodes. For $z_\alpha{=}-3$, the HTHD has the lowest latency and nearly the same BLER with the SSC method. Fig. 4 (b) is given in the worst cases, where all of the rate-r nodes are treated as unstructured nodes and the latency is only reduced by simplifying the decoding of reliable nodes. Fig. 4 (b) indicates that the proposed HTHD method achieves lower latency than the NEP-SSC decoder.

V. Conclusion

A HTHD decoder is proposed to reduce the decoding latency of unstructured nodes, where the node reliability is evaluated by the hypothesis testing. The reliable nodes are then decoded by a novel hard decision method. Simulation verifies that the HTHD significantly reduces the decoding latency.

References

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

[2] G. Sarkis, P. Giard, A. Vardy, C. Thibeault, and W. J. Gross, "Fast polar decoders: Algorithm and implementation," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 5, pp. 946–957, May 2014.

[3] P. Giard, A. Balatsoukas-Stimming, G. Sarkis, C. Thibeault, and W. J. Gross, "Fast low–complexity decoders for low–rate polar codes," *IEEE J. Signal Process. Syst.*, vol. 90, no. 5, pp. 675–685, May 2018.

[4] M. Hanif and M. Ardakani, "Fast successive–cancellation decoding of polar codes: Identification and decoding of new nodes," *IEEE Commun. Lett.*, vol. 21, no. 11, pp. 2360–2363, Nov. 2017.

[5] S. Li, Y. Deng, L. Lu, J. Liu, and T. Huang, "A low–latency simplified successive cancellation decoder for polar codes based on node error probability," *IEEE Commun. Lett.*, vol. 22, no. 12, pp. 2439–2442, Dec. 2018.

[6] S. A. Hashemi, C. Condo, and W. J. Gross, "Fast and flexible successive–cancellation list decoders for polar codes," *IEEE Trans. Signal Process.*, vol. 65, no. 21, pp. 5756–5769, Nov. 2017.

[7] G. Sarkis, P. Giard, A. Vardy, C. Thibeault, and W. J. Gross, "Fast list decoders for polar codes," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 2, pp. 318–328, Feb. 2016.

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

[9] M. Vajha, V. S. C. Mukka, and P. V. Kumar, "Backtracking and look-ahead decoding algorithms for improved successive cancellation decoding performance of polar codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2019, pp. 31–35.

[10] D. Chase, "Class of algorithms for decoding block codes with channel measurement information," *IEEE Trans. Inf. Theory*, vol. IT-18, no. 1, pp. 170–182, Jan. 1972.

[11] D. Wu, Y. Li, and Y. Sun, "Construction and block error rate analysis of polar codes over AWGN channel based on Gaussian approximation," *IEEE Commun. Lett.*, vol. 18, no. 7, pp. 1099–1102, Jul. 2014.

[12] P. Trifonov, "Efficient design and decoding of polar codes," *IEEE Trans. Commun.*, vol. 60, no. 11, pp. 3221–3227, Nov. 2012.