

A New Polar Code Design Based on Reciprocal Channel Approximation

Hideki Ochiai¹, Senior Member, IEEE, Kosuke Ikeya¹, and Patrick Mitran², Senior Member, IEEE

Abstract—This paper revisits polar code design for a binary-input additive white Gaussian noise (BI-AWGN) channel when successive cancellation (SC) decoding is applied at the receiver. We focus on the so-called *reciprocal channel approximation (RCA)*, which is often adopted in the design of low-density parity-check (LDPC) codes. Implementation of RCA requires the computation of the mutual information of BPSK signaling as well as a corresponding function known as the *reciprocal channel mapping*, and thus we develop rigorous closed-form approximations of these that are easy to calculate numerically and also valid over a wide range of SNR. Through numerical evaluation we find that, compared to approaches based on the popular Gaussian approximation (GA) as well as the so-called improved GA (IGA), the proposed RCA approach offers better estimates of the bit error rate of polarized channels *with no additional computational cost*. As a result, polar codes designed by the proposed RCA can achieve further improvement in terms of block error rate (BLER) performance. The gain achieved by the new approach becomes significant as the codeword length increases.

Index Terms—Code construction, density evolution, Gaussian approximation, polar codes, reciprocal channel approximation.

I. INTRODUCTION

ONE of the most striking properties of polar codes [1] is their capacity approaching behavior that is achievable with low-complexity successive cancellation (SC) decoding. Specifically, for a codeword length of N bits, the decoding complexity is only $O(N \log N)$, which is significantly lower than other known capacity approaching codes that are available in practice. As a result of channel polarization, the design of polar codes is equivalent to the identification of good channels and bad channels, where the former channels are used for information transmission and the latter channels are left unused (i.e., *frozen*). The main focus of this paper is on the design of polar codes with long codeword lengths

Manuscript received 11 April 2022; revised 1 September 2022 and 16 November 2022; accepted 27 November 2022. Date of publication 7 December 2022; date of current version 16 February 2023. This work was supported in part by the Japan Society for the Promotion of Science (JSPS) through the Grants-in-Aid for Scientific Research (KAKENHI) under Grant 21H04873. The associate editor coordinating the review of this article and approving it for publication was P. Trifonov. (*Corresponding author: Hideki Ochiai.*)

Hideki Ochiai and Kosuke Ikeya are with the Department of Electrical and Computer Engineering, Yokohama National University, Yokohama 240-8501, Japan (e-mail: hideki@ynu.ac.jp; ikeya.kosuke.pbic@gmail.com).

Patrick Mitran is with the Department of Electrical and Computer Engineering, University of Waterloo, ON N2L 3G1, Canada (e-mail: pmitran@uwaterloo.ca).

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TCOMM.2022.3227280>.

Digital Object Identifier 10.1109/TCOMM.2022.3227280

over a binary-input additive white Gaussian noise (BI-AWGN) channel.

There have been various techniques proposed for polar code design over a BI-AWGN channel. The most accurate analytical approach is the use of density evolution [2], [3], originally developed for the design of low-density parity-check (LDPC) codes, and its applicability to polar codes has been identified in [4]. Density evolution tracks the probability distribution of the channel or its log-likelihood ratio (LLR), and in order to improve its accuracy, it should be computed with sufficient quantization and dynamic range. Therefore, density evolution is highly demanding in terms of space and computational complexity, especially when the codeword length increases. A more tractable approach with limited space complexity was proposed in [5]. Nevertheless, the approach involves quantization, and thus the overall complexity depends not only on the codeword length but also on the required precision. On the other hand, a significantly simpler approach that tracks density evolution by a single parameter is Gaussian approximation (GA) [6]. Also initially developed for the design of LDPC codes, its applicability to polar code design has been well investigated [7]. It has been pointed out in [8] that GA with the original approximation function developed in [6] may not necessarily work accurately when the equivalent signal-to-noise power ratio (SNR) values of the channels after polarization become low. Consequently, a modified version, which will be referred to as improved GA (IGA) in this work, has been proposed [8]. Furthermore, there have been various design approaches proposed in the recent literature, targeting specific polar decoding algorithms. For example, in the case of successive cancellation list (SCL) decoding [9] or belief propagation (BP) decoding [10], learning-based approaches such as genetic algorithms and reinforcement learning have been proposed in [11] and [12], respectively. The major limitations of these approaches are their lack of flexibility and scalability since computationally demanding training should be performed for each given combination of the code parameters such as codeword length and code rate. Other related works include multi-kernel polar codes [13] which generalize Arkan's kernel so as to offer code design flexibility, while GA would again be employed in the information bit selection process.

In this work, we consider the application of the so-called *reciprocal channel approximation (RCA)*, introduced by Chung in [14] as another single-parameter approximation of density evolution, and it is motivated by the duality property

of mutual information that holds between a repetition code and parity-check code over a binary erasure channel (BEC) [15]. This approach has been originally adopted in the design of LDPC and other related codes [14], [16], [17], [18], [19], and it was numerically demonstrated in [14, Chapter 7] that RCA may offer a better approximation even over BI-AWGN channels than the conventional GA. By numerical study and simulation, we will find that polar codes designed based on RCA exhibit better performance than those based on (improved) GA, and the gain becomes more significant as the codeword length increases. The superior performance improvement will be demonstrated for polar codes with very long lengths (up to $N = 2^{18}$ bits), as is considered in practical optical communications [20]. (See, also [21], where a polar code of length 2^{20} bits is considered.)

The design of polar codes based on RCA was initially proposed in [22] as well as [23]. RCA-based designs make use of the mutual information of BPSK signaling over an AWGN channel as well as its inverse function, which should be calculated numerically. There are several closed-form approximations available in the literature. In [23] the two approximations developed in [17] and [24] are compared, indicating the sensitivity of polar code performance to the choice of approximation. For this reason, in this work we introduce rigorous closed-form expressions to approximate the mutual information $C(\gamma)$ of BPSK as a function of SNR γ as well as its inverse function $C^{-1}(\cdot)$ that can be used to design polar codes with long codeword lengths.

The main contributions of this work are as follows:

- We derive a closed-form piece-wise continuous mutual information expression for BPSK signaling over AWGN channels with guaranteed convergence in the case of high and low SNR based on the asymptotic analysis of the mutual information function. This expression is exploited to design polar codes based on RCA supporting a wide range of SNR after polarization.
- We develop an explicit algorithm that identifies SNR after channel polarization using only closed-form equations, which can thus be calculated with low complexity or low latency.
- By simulation, we demonstrate that, with a moderate code rate (e.g., around 1/2), the RCA-based design can offer better bit error rate (BER) as well as block error rate (BLER) estimates compared to GA-based approaches, where the gap becomes more significant as the codeword length increases. As a result, the BLER performance of polar codes designed by the proposed RCA approach outperforms that of IGA when it is compared at the same design SNR.

In summary, with no additional computational cost, the proposed design approach based on RCA offers an improvement over conventional low-complexity design approaches in terms of the achievable BLER performance as well as their estimates, especially when the codeword length of interest is large.

The remainder of the paper is organized as follows. Section II reviews the principles of GA and RCA from the viewpoint of polar code design and describes a general algorithm for RCA. Upon development of RCA, the key

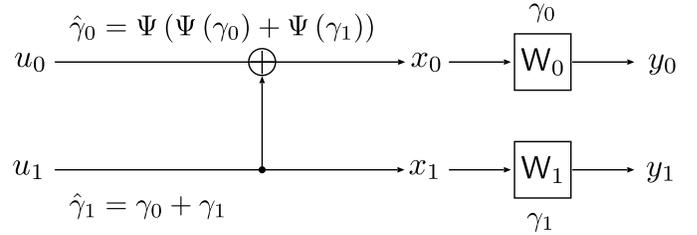


Fig. 1. Polar encoder and associated notations for $N = 2$.

function denoted by $\Lambda(\xi)$, which corresponds to the *reciprocal channel mapping* defined in [14] and with SNR given in the log domain, is introduced. Section III develops a new closed-form expression to approximate the mutual information for BPSK signaling as well as its inverse, and its accuracy is compared with conventional alternatives. Based on the developed mathematical tools, closed-form approximations for the associated function $\Lambda(\xi)$ are derived in Section IV, which completes the proposed algorithm. Simulation results and estimated BER/BLER values are compared in Section V, which reveals the effectiveness of the RCA-based polar code design using the proposed algorithm. Finally, concluding remarks are given in Section VI.

II. PRELIMINARIES

A. Polar Codes

We start with the simplest binary polar code of codeword (or block) length $N = 2$ (Arkan's kernel) shown in Fig. 1, where the information bits $u_0, u_1 \in \mathbb{F}_2$ and the coded bits $x_0, x_1 \in \mathbb{F}_2$ are related by

$$\begin{cases} x_0 = u_0 + u_1, \\ x_1 = u_1. \end{cases} \quad (1)$$

The coded bits x_0 and x_1 are modulated by BPSK and are transmitted over AWGN channels, denoted by W_0 and W_1 , where the SNRs of the channels are γ_0 and γ_1 , respectively, as illustrated in Fig. 1, with the received symbols given by y_0 and y_1 .

Assuming that u_0 is decoded first, since

$$u_0 = x_0 + x_1, \quad (2)$$

u_0 can be seen as a check node connected to x_0 and x_1 in the Tanner graph. Once the estimate of u_0 , denoted by $\hat{u}_0 \in \mathbb{F}_2$, is given, u_1 can be uniquely determined by either of x_0 or x_1 , i.e., u_1 is a variable node connected to both x_0 and x_1 .

Polar codes of length $N = 2^n$ for an integer $n > 1$ can be obtained by the recursive application of the above kernel [1].

B. Gaussian Approximation

We briefly summarize the Gaussian approximation (GA) approach, in the framework of polar codes based on the original work of [6]. (More details can be found in [8] and the references therein.) We note that GA is an approximation of density evolution through the single parameter corresponding to a mean value of the LLR, assuming that the LLR is

Gaussian distributed. Let L_0 and L_1 denote the LLR values corresponding to x_0 and x_1 , respectively. Without loss of generality, we assume that the all-zero input sequence and thus all-zero codeword is transmitted. For a BI-AWGN channel, it follows that $L_0 \sim \mathcal{N}(\gamma_0, 2\gamma_0)$ and $L_1 \sim \mathcal{N}(\gamma_1, 2\gamma_1)$, where $\mathcal{N}(\mu, \sigma^2)$ denotes a real-valued Gaussian distribution with mean μ and variance σ^2 . Due to the relationship in (2), the LLR corresponding to the check node u_0 , denoted by \hat{L}_0 , should satisfy [25]

$$\tanh\left(\frac{\hat{L}_0}{2}\right) = \tanh\left(\frac{L_0}{2}\right) \tanh\left(\frac{L_1}{2}\right). \quad (3)$$

The GA approach assumes that the LLR always follows a Gaussian distribution $\mathcal{N}(\gamma, 2\gamma)$ for some mean value γ . The mean value of \hat{L}_0 is then determined by taking expectations of both sides of (3). For $L \sim \mathcal{N}(\gamma, 2\gamma)$, by defining

$$\phi(\gamma) \triangleq 1 - E\left[\tanh\left(\frac{L}{2}\right)\right], \quad (4)$$

the mean value of \hat{L}_0 can be expressed as

$$\hat{\gamma}_0 = \phi^{-1}\left(1 - (1 - \phi(\gamma_0))(1 - \phi(\gamma_1))\right). \quad (5)$$

The LLR corresponding to the variable node u_1 , denoted by \hat{L}_1 , is then expressed as

$$\hat{L}_1 = L_0 + L_1. \quad (6)$$

Based on the assumption that L_0 and L_1 are independent Gaussian random variables, we have

$$\hat{\gamma}_1 = \gamma_0 + \gamma_1. \quad (7)$$

Since the key function $\phi(\gamma)$ defined in (4) cannot be expressed in closed form, the following approximation based on [6] has also been adopted for polar code design [26], [27]:

$$\phi(\gamma) \approx \begin{cases} e^{a\gamma^c + b}, & \gamma < 10, \\ \sqrt{\frac{\pi}{\gamma}} e^{-\frac{\gamma}{4}} \left(1 - \frac{10}{7\gamma}\right), & \gamma \geq 10, \end{cases} \quad (8a)$$

$$(8b)$$

where $(a, b, c) = (-0.4527, 0.0218, 0.86)$. We note that this approximation becomes inaccurate as γ decreases due to numerical error. Therefore, an improved version (IGA) that solves this numerical issue was proposed in [8], where several related approximations are also compared in detail.

Finally, it is worth noting that the inverse function of (8b), as required when calculating (5), should be numerically calculated using, e.g., the bisection method as discussed in [27]. This is also the case for most GA-based approaches including IGA of [8].

C. Reciprocal Channel Approximation

Following [19], we describe the principle of reciprocal channel approximation (RCA). Let us first consider a BEC where the erasure probability is e^{-s} and the non-erasure probability is e^{-r} , i.e., $e^{-s} + e^{-r} = 1$, for $s, r > 0$. Let $C(s)$ denote the capacity of this channel, i.e., $C(s) = 1 - e^{-s}$. Then, it follows that $C(s) + C(r) = 1$. Let s_i , $i \in \{0, 1\}$, denote the above parameter s for the bit x_i transmitted over the corresponding BEC and let us define r_i such that

$e^{-s_i} + e^{-r_i} = 1$. For a variable node u_1 connected to x_0 and x_1 , its erasure probability is given by $e^{-s_0}e^{-s_1} = e^{-(s_0+s_1)}$, i.e., it is characterized by the sum of s_0 and s_1 . For a check node u_0 connected to x_0 and x_1 , its corresponding *non-erasure* probability is given by $(1 - e^{-s_0})(1 - e^{-s_1}) = e^{-(r_0+r_1)}$, i.e., it is also characterized by the sum of r_0 and r_1 . In other words, for BEC, the parameter s is additive for variable nodes, whereas the parameter r is additive for check nodes [19]. In this sense, they have a reciprocity property.

We now apply the above reciprocity concept to the binary-input AWGN (BI-AWGN) channel. We define $C(\gamma)$ as the capacity of the BI-AWGN channel with the SNR given by γ . Similar to BEC, the additive property holds for the parameter γ in the case of the variable node as long as all the connecting nodes are associated with mutually independent AWGN channels. This stems from the fact that the LLR is additive at the variable node, and both mean and variance of the LLR corresponding to BI-AWGN channels are proportional to the channel SNR as discussed in the previous subsection. Let us now define the SNR parameter λ corresponding to the check node such that $C(\gamma) + C(\lambda) = 1$, even though this property does not strictly hold for general BI-AWGN channels.

For a given channel SNR γ , the parameter λ can be found by solving

$$C(\lambda) = 1 - C(\gamma), \quad (9)$$

for λ . Since $C(\gamma)$ is a strictly increasing function for $\gamma > 0$, we can define its inverse function $C^{-1}(\cdot)$ and write

$$\lambda = C^{-1}(1 - C(\gamma)) \triangleq \Psi(\gamma). \quad (10)$$

The above function corresponds to the *reciprocal channel mapping* introduced in [14, Definition 7.3], also referred to as *self-inverting reciprocal energy function* in [19], and applied to BI-AWGN channels. Since density evolution is approximated with a single parameter γ through the above reciprocal channel mapping principle, it is referred to as the *reciprocal channel approximation* (RCA) in [14, Section 7.4].

Our goal is to find a suitable expression for the function $\Psi(\gamma)$ that can support a wide range of SNR γ with high accuracy. Note that since (9) holds with λ and γ interchanged, one can easily verify that

$$\Psi(\Psi(\gamma)) = \gamma, \quad (11)$$

i.e., $\Psi(\gamma)$ is a self-inverse function satisfying $\Psi(\gamma) = \Psi^{-1}(\gamma)$ [14].

With γ_0 and γ_1 representing the SNRs of the channels corresponding to the bits x_0 and x_1 , respectively, the SNR corresponding to the variable node u_1 , denoted by $\hat{\gamma}_1$, is the same as GA, i.e., (7). On the other hand, the SNR corresponding to the check node u_0 , denoted by $\hat{\gamma}_0$, is similarly expressed based on the RCA principle [14, Section 7.4] as

$$\Psi(\hat{\gamma}_0) = \Psi(\gamma_0) + \Psi(\gamma_1), \quad (12)$$

from which we obtain

$$\begin{aligned} \hat{\gamma}_0 &= \Psi^{-1}(\Psi(\gamma_0) + \Psi(\gamma_1)) \\ &= \Psi(\Psi(\gamma_0) + \Psi(\gamma_1)). \end{aligned} \quad (13)$$

The above relationship between the pairs (γ_0, γ_1) and $(\hat{\gamma}_0, \hat{\gamma}_1)$ is illustrated in Fig. 1.

In what follows, we consider the case when γ is extremely large or small due to the polarization effect. From the viewpoint of numerical evaluation and similar to [8], it would be convenient to express γ in the log domain. Therefore, we define $\xi \triangleq \ln \gamma$ and also introduce the function

$$\Lambda(\xi) \triangleq \ln \Psi(e^\xi) = \ln \Psi(\gamma). \quad (14)$$

For the check node, by iteratively applying the relationship (14) to (13), the output $\hat{\xi}_0 \triangleq \ln \hat{\gamma}_0$ for a given pair of inputs $\xi_0 = \ln \gamma_0$ and $\xi_1 = \ln \gamma_1$ can be expressed as

$$\begin{aligned} \hat{\xi}_0 &= \ln \Psi(\Psi(\gamma_0) + \Psi(\gamma_1)) \\ &= \Lambda(\ln(\Psi(\gamma_0) + \Psi(\gamma_1))) \\ &= \Lambda\left(\ln\left(e^{\Lambda(\xi_0)} + e^{\Lambda(\xi_1)}\right)\right) \\ &= \Lambda\left(\max(\Lambda(\xi_0), \Lambda(\xi_1)) + \ln\left(1 + e^{-|\Lambda(\xi_1) - \Lambda(\xi_0)|}\right)\right). \end{aligned} \quad (15)$$

On the other hand, for the variable node, the output $\hat{\xi}_1$ is expressed as

$$\hat{\xi}_1 = \max(\xi_0, \xi_1) + \ln\left(1 + e^{-|\xi_0 - \xi_1|}\right). \quad (16)$$

Note that if the SNRs of the two channels are identical, i.e., $\gamma_0 = \gamma_1$, then we have $\xi_0 = \xi_1$ and $\Lambda(\xi_0) = \Lambda(\xi_1)$, and the above equations reduce to

$$\hat{\xi}_0 = \Lambda(\Lambda(\xi_0) + \ln 2), \quad (17)$$

$$\hat{\xi}_1 = \xi_0 + \ln 2. \quad (18)$$

Remark: While GA tracks the LLR distribution (as a simplification of density evolution), RCA tracks SNR using capacity formulas. Therefore, unlike GA, for check node evaluation RCA does not assume that LLRs are Gaussian distributed. Nevertheless, to estimate the bit error rate of each channel from SNR, in Section V we resort to the assumption that the LLR is Gaussian distributed.

D. General RCA Algorithm

In the case of a binary polar code of length $N = 2^n$, let $\gamma_0, \gamma_1, \dots, \gamma_{N-1}$ denote the channel SNRs of the coded bits x_0, x_1, \dots, x_{N-1} . Then the SNRs for the input bits u_0, u_1, \dots, u_{N-1} , denoted by $\hat{\gamma}_0, \hat{\gamma}_1, \dots, \hat{\gamma}_{N-1}$, can be obtained by the well-known recursive procedure in [1]. Based on (15) and (16), the corresponding RCA algorithm when each channel has a distinct SNR value is summarized in **Algorithm 1**. The algorithm can be significantly simplified when all the channels have the same SNR (or design SNR), i.e., $\gamma_0 = \gamma_1 = \dots = \gamma_{N-1}$. In this case, (17) and (18) are applicable and the algorithm reduces to **Algorithm 2**.

The remaining questions are: (i) how can one calculate the function $\Lambda(\xi)$ accurately with computational efficiency, i.e., without resorting to numerical integration, and (ii) how well does the algorithm operate compared to other known approaches of similar complexity? These will be addressed in the subsequent sections.

Algorithm 1 Channel Polarization With RCA (for Distinct Channel SNRs)

Input: $n = \log_2 N$, $\xi[0], \xi[1], \dots, \xi[N-1]$ as $\ln \gamma_0, \ln \gamma_1, \dots, \ln \gamma_{N-1}$.
Output: $\xi[0], \xi[1], \dots, \xi[N-1]$ as $\ln \hat{\gamma}_0, \ln \hat{\gamma}_1, \dots, \ln \hat{\gamma}_{N-1}$

- 1: **for** $i = 1 : n$ **do**
- 2: $J \leftarrow 2^i$
- 3: **for** $k = 0 : N/J - 1$ **do**
- 4: **for** $j = 0 : J/2 - 1$ **do**
- 5: $\xi_0 \leftarrow \xi[kJ + j]$
- 6: $\xi_1 \leftarrow \xi[kJ + j + J/2]$
- 7: $\Lambda_0 \leftarrow \Lambda(\xi_0)$
- 8: $\Lambda_1 \leftarrow \Lambda(\xi_1)$
- 9: $\xi[kJ + j] \leftarrow \Lambda(\max(\Lambda_0, \Lambda_1) + \ln(1 + e^{-|\Lambda_0 - \Lambda_1|}))$
- 10: $\xi[kJ + j + J/2] \leftarrow \max(\xi_0, \xi_1) + \ln(1 + e^{-|\xi_0 - \xi_1|})$
- 11: **end for**
- 12: **end for**
- 13: **end for**
- 14: **return** $\xi[0], \xi[1], \dots, \xi[N-1]$

Algorithm 2 Channel Polarization With RCA (for Uniform Channel SNR)

Input: $n = \log_2 N$, $\xi[0] = \ln \gamma_0$
Output: $\xi[0], \xi[1], \dots, \xi[N-1]$ as $\ln \hat{\gamma}_0, \ln \hat{\gamma}_1, \dots, \ln \hat{\gamma}_{N-1}$

- 1: **for** $i = 1 : n$ **do**
- 2: $J \leftarrow 2^i$
- 3: **for** $j = 0 : J/2 - 1$ **do**
- 4: $\xi_0 \leftarrow \xi[j]$
- 5: $\Lambda_0 \leftarrow \Lambda(\xi_0)$
- 6: $\xi[j] \leftarrow \Lambda(\Lambda_0 + \ln 2)$
- 7: $\xi[j + J/2] \leftarrow \xi_0 + \ln 2$
- 8: **end for**
- 9: **end for**
- 10: **return** $\xi[0], \xi[1], \dots, \xi[N-1]$

III. A NEW CAPACITY EXPRESSION

We first review popular BI-AWGN capacity approximation formulas available in the literature, based on which we derive a new expression.

A. Existing BI-AWGN Capacity Approximation Expressions

Let E_s denote the symbol energy of BPSK and the variance of real-valued Gaussian noise be $N_0/2$. We define the SNR parameter γ as $\gamma \triangleq E_s/N_0$ in what follows. To compute the mutual information $C(\gamma)$ between a BPSK input and its corresponding output over an AWGN channel, the following equivalent J -function (see [17]) is often adopted:

$$J(x) = 1 - \frac{1}{\sqrt{2\pi x^2}} \int_{-\infty}^{\infty} e^{-\frac{(t - \frac{x^2}{2})^2}{2x^2}} \log_2(1 + e^{-t}) dt, \quad (19)$$

where x corresponds to the standard deviation of the LLR and thus is related to γ by $x = \sqrt{8\gamma}$. Then $C(\gamma) = J(\sqrt{8\gamma})$.

According to [17], the function $J(x)$ is well approximated numerically by

$$J(x) \approx \begin{cases} a_1 x^3 + b_1 x^2 + c_1 x, & 0 \leq x \leq 1.6363, \\ 1 - e^{a_2 x^3 + b_2 x^2 + c_2 x + d}, & 1.6363 < x \leq 10, \\ 1, & x > 10, \end{cases} \quad (20)$$

with $a_1 = -0.0421061$, $b_1 = 0.209252$, $c_1 = -0.00640081$, $a_2 = 0.00181491$, $b_2 = -0.142675$, $c_2 = -0.0822054$, and $d = 0.0549608$. On the other hand, in [24], the approximation

$$J(x) \approx \left(1 - 2^{-H_1 x^{2H_2}}\right)^{H_3}, \quad (21)$$

was proposed with $H_1 = 0.3073$, $H_2 = 0.8935$, and $H_3 = 1.1064$.

B. A New Approximation Formula

From (19) and $C(\gamma) = J(\sqrt{8\gamma})$, we define

$$\begin{aligned} U(\gamma) &\triangleq 1 - C(\gamma) \\ &= \frac{1}{4\sqrt{\pi\gamma}} \int_{-\infty}^{\infty} e^{-\frac{1}{16\gamma}(t-4\gamma)^2} \log_2(1 + e^{-t}) dt. \end{aligned} \quad (22)$$

In what follows, we divide the range of γ , $\mathcal{R} \triangleq (0, \infty)$, into the four sub-regions \mathcal{R}_1 , \mathcal{R}_2 , \mathcal{R}_3 , and \mathcal{R}_4 , where $\mathcal{R}_1 = (0, \Gamma_1)$, $\mathcal{R}_2 = [\Gamma_1, \Gamma_2)$, $\mathcal{R}_3 = [\Gamma_2, \Gamma_3)$, and $\mathcal{R}_4 = [\Gamma_3, \infty)$, with Γ_1, Γ_2 , and Γ_3 representing appropriate boundaries to be determined numerically.

1) *Low SNR Case*: For the sub-region \mathcal{R}_1 , i.e., when $\gamma \approx 0$, by Maclaurin series expansion we obtain

$$C(\gamma) \approx \frac{1}{\ln 2} \left(\gamma - \gamma^2 + \frac{4}{3}\gamma^3 - \frac{10}{3}\gamma^4 + \frac{208}{15}\gamma^5 \right). \quad (23)$$

By truncating up to the third order, we may approximate $U(\gamma)$ as

$$U_1(\gamma) \triangleq 1 - \frac{1}{\ln 2} \left(\gamma - \gamma^2 + \frac{4}{3}\gamma^3 \right), \quad \gamma < \Gamma_1. \quad (24)$$

The upper boundary Γ_1 should be determined according to the target approximation error. In what follows, we set $\Gamma_1 = 0.04$ (i.e., -14 dB) resulting in a numerical error that is at most about 10^{-5} (as will be shown in Fig. 2 later). More precisely, the value of $U(\gamma)$ at $\gamma = \Gamma_1$ is calculated by numerical integration as

$$U(\Gamma_1) \approx 0.9444880, \quad (25)$$

whereas the corresponding approximation according to (24) is given by

$$U_1 \triangleq U_1(\Gamma_1) \approx 0.9444774. \quad (26)$$

Therefore, the approximation error of $U(\gamma)$ by $U_1(\gamma)$ in the region \mathcal{R}_1 is bounded by

$$|U(\gamma) - U_1(\gamma)| < 1.1 \times 10^{-5}, \quad \gamma < \Gamma_1. \quad (27)$$

2) *High SNR Case*: We next consider the sub-region \mathcal{R}_4 , i.e., when γ becomes larger than some boundary Γ_3 . By applying the series expansion of the exponential function $e^x = \sum_{k=0}^{\infty} \frac{x^k}{k!}$ to $U(\gamma)$ of (22), we obtain

$$\begin{aligned} U(\gamma) &= \frac{e^{-\gamma}}{4\sqrt{\pi\gamma}} \sum_{k=0}^{\infty} \frac{(-1)^k}{k! (16\gamma)^k} c_k \\ &= \frac{e^{-\gamma}}{4\sqrt{\pi\gamma}} \left\{ c_0 - \frac{c_1}{16\gamma} + \frac{c_2}{2 \cdot (16\gamma)^2} - \dots \right\}, \end{aligned} \quad (28)$$

where

$$c_k \triangleq \int_{-\infty}^{\infty} e^{\frac{t}{2}} t^{2k} \log_2(1 + e^{-t}) dt. \quad (29)$$

The above expression agrees with [3, Problem 4.12]. Note that c_k for specific values of k can be expressed in closed form, e.g.,

$$\begin{aligned} c_0 &= \frac{2\pi}{\ln 2}, \quad c_1 = \frac{2\pi(8 + \pi^2)}{\ln 2}, \\ c_2 &= \frac{2\pi(384 + 48\pi^2 + 5\pi^4)}{\ln 2}. \end{aligned} \quad (30)$$

Although the approximation becomes tighter as we incorporate more terms in (28), it becomes infeasible to find the corresponding inverse expression in closed form. Therefore, we adopt the expression

$$U_4(\gamma) \triangleq \alpha \frac{e^{-\gamma}}{\sqrt{\gamma}}, \quad \gamma > \Gamma_3, \quad (31)$$

where α and Γ_3 are appropriate constants to be determined. By setting the boundary as $\Gamma_3 = 10$, the value at the boundary becomes

$$U_3 \triangleq U(\Gamma_3) \approx 1.667 \times 10^{-5}, \quad (32)$$

from which we fix the constant α as

$$\alpha \approx 1.16125142, \quad (33)$$

such that $U(\Gamma_3) \approx U_4(\Gamma_3)$. Note that this is different from the exact coefficient of the first term in (28), which is given by

$$\alpha_0 \triangleq c_0 / (4\sqrt{\pi}) = \frac{\sqrt{\pi}}{2(\ln 2)} \approx 1.27856, \quad (34)$$

and thus the exact asymptotic form for large γ would be

$$U(\gamma) \rightarrow \frac{\sqrt{\pi}}{2(\ln 2)} \frac{e^{-\gamma}}{\sqrt{\gamma}}, \quad (35)$$

as discussed in [3] and [28]. As γ increases, the use of α_0 may become more accurate eventually, but for our purpose, the use of (33) may be more suitable as the piece-wise continuity of the approximate function can then be guaranteed.

3) *Moderate SNR Case:* We first note that the main advantage of the capacity approximation expression in (21) is that its inverse function can also be expressed in closed form. We thus adopt this form for the remaining sub-regions \mathcal{R}_2 and \mathcal{R}_3 and define the functions for the two adjacent regions as

$$U_2(\gamma) \triangleq 1 - \left(1 - e^{-H_{2,1}\gamma^{H_{2,2}}}\right)^{H_{2,3}}, \quad \Gamma_1 < \gamma < \Gamma_2, \quad (36)$$

$$U_3(\gamma) \triangleq 1 - \left(1 - e^{-H_{3,1}\gamma^{H_{3,2}}}\right)^{H_{3,3}}, \quad \Gamma_2 < \gamma < \Gamma_3, \quad (37)$$

where the constants $H_{i,j}$ for $i \in \{2,3\}$ and $j \in \{1,2,3\}$ should be determined appropriately depending on the boundary Γ_2 . In what follows, we fix the boundary as $\Gamma_2 = 1$ for simplicity of numerical evaluation. Then the precise value of $U(\gamma)$ at the boundary is obtained by numerical integration as

$$U_2 \triangleq U(\Gamma_2) \approx 0.2785484. \quad (38)$$

Furthermore, in order to guarantee the piece-wise continuity of the approximate function for $U(\gamma)$, it is required that

$$U_2(\Gamma_1) = U_1, \quad U_3(\Gamma_3) = U_3. \quad (39)$$

Let us first consider the function $U_2(\gamma)$. By the relationships at the boundaries, one may express

$$H_{2,1} = -\ln\left(1 - (1 - U_2)^{\frac{1}{H_{2,3}}}\right), \quad (40)$$

$$H_{2,2} = \left[\ln\left(\frac{\Gamma_2}{\Gamma_1}\right)\right]^{-1} \ln\left\{\frac{\ln\left(1 - (1 - U_2)^{\frac{1}{H_{2,3}}}\right)}{\ln\left(1 - (1 - U_1)^{\frac{1}{H_{2,3}}}\right)}\right\}. \quad (41)$$

As a consequence, $U_2(\gamma)$ can be expressed as a function of γ and $H_{2,3}$, which we explicitly write as $U_2(\gamma; H_{2,3})$. Based on the minimization of ∞ -norm, we may optimize the coefficient $H_{2,3}$ as

$$H_{2,3} = \arg \min_H \max_{\Gamma_1 < \gamma < \Gamma_2} |U_2(\gamma; H) - U(\gamma)|, \quad (42)$$

from which we numerically obtain

$$H_{2,1} = 1.396634, \quad H_{2,2} = 0.872764, \quad H_{2,3} = 1.148562.$$

Likewise, by denoting $U_3(\gamma)$ as a function of $H_{3,3}$, i.e., $U_3(\gamma; H_{3,3})$, since

$$H_{3,1} = -\ln\left(1 - (1 - U_2)^{\frac{1}{H_{3,3}}}\right), \quad (43)$$

$$H_{3,2} = \left[\ln\left(\frac{\Gamma_3}{\Gamma_2}\right)\right]^{-1} \ln\left\{\frac{\ln\left(1 - (1 - U_3)^{\frac{1}{H_{3,3}}}\right)}{\ln\left(1 - (1 - U_2)^{\frac{1}{H_{3,3}}}\right)}\right\}, \quad (44)$$

one may determine $H_{3,3}$ according to

$$H_{3,3} = \arg \min_H \max_{\Gamma_2 < \gamma < \Gamma_3} |U_3(\gamma; H) - U(\gamma)|, \quad (45)$$

and thus numerically obtain

$$H_{3,1} = 1.266967, \quad H_{3,2} = 0.938175, \quad H_{3,3} = 0.986830.$$

4) *Summary:* We summarize the closed-form piece-wise continuous approximate function $\hat{U}(\gamma)$ for $U(\gamma)$:

$$\hat{U}(\gamma) = \begin{cases} U_1(\gamma) = 1 - \frac{1}{\ln 2} \left(\gamma - \gamma^2 + \frac{4}{3}\gamma^3\right), & \gamma \in \mathcal{R}_1, & (46a) \\ U_2(\gamma) = 1 - \left(1 - e^{-H_{2,1}\gamma^{H_{2,2}}}\right)^{H_{2,3}}, & \gamma \in \mathcal{R}_2, & (46b) \\ U_3(\gamma) = 1 - \left(1 - e^{-H_{3,1}\gamma^{H_{3,2}}}\right)^{H_{3,3}}, & \gamma \in \mathcal{R}_3, & (46c) \\ U_4(\gamma) = \alpha \frac{e^{-\gamma}}{\sqrt{\gamma}}, & \gamma \in \mathcal{R}_4, & (46d) \end{cases}$$

where the boundaries of the four sub-regions are $\Gamma_1 = 0.04$, $\Gamma_2 = 1$, and $\Gamma_3 = 10$. As a consequence, the approximate formula of the capacity for the BI-AWGN channel, $\hat{C}(\gamma) \triangleq 1 - \hat{U}(\gamma)$, can be summarized as

$$\hat{C}(\gamma) = \begin{cases} \frac{1}{\ln 2} \left(\gamma - \gamma^2 + \frac{4}{3}\gamma^3\right), & \gamma \in \mathcal{R}_1, & (47a) \\ \left(1 - e^{-H_{2,1}\gamma^{H_{2,2}}}\right)^{H_{2,3}}, & \gamma \in \mathcal{R}_2, & (47b) \\ \left(1 - e^{-H_{3,1}\gamma^{H_{3,2}}}\right)^{H_{3,3}}, & \gamma \in \mathcal{R}_3, & (47c) \\ 1 - \alpha \frac{e^{-\gamma}}{\sqrt{\gamma}}, & \gamma \in \mathcal{R}_4. & (47d) \end{cases}$$

C. Numerical Results

To investigate the accuracy of the developed approximation, we define the error function as

$$\epsilon(\gamma) \triangleq U(\gamma) - \hat{U}(\gamma) = \hat{C}(\gamma) - C(\gamma). \quad (48)$$

We calculate $C(\gamma)$ based on numerical integration and compare with the developed expression as well as (20) of [17] and (21) of [24]. In Fig. 2, we plot the absolute value of the error function $\epsilon(\gamma)$ with respect to γ in dB. We observe that the absolute values of the approximation error for all the three expressions are less than 10^{-3} , and monotonically decrease for higher and lower SNR regions. Nevertheless, the most significant difference of the proposed expression is that the error becomes much less than 10^{-5} as SNR decreases since we selected the associated parameters such that the condition (27) holds. Also, since the proposed expression $\hat{C}(\gamma)$ in (47) is designed to be piece-wise continuous with respect to γ , we also observe that the error function is piece-wise continuous as well. By comparison, as observed in Fig. 2, the approximation based on (20) (introduced in [17]) exhibits discontinuities at the boundaries corresponding to $x = 1.6363$ and $x = 10$, i.e., $\gamma = -4.7536$ dB and $\gamma = 10.969$ dB, respectively. From a previous study on GA [8], if there is a discontinuity in the function, this may result in an inaccurate ordering of channel reliability. Therefore, we give more priority to continuity of the function rather than its accuracy.

IV. DERIVATION OF CLOSED-FORM EXPRESSION FOR $\Lambda(\xi)$

Our next step is to find a suitable approximate expression for $\Lambda(\xi)$ introduced in (14). We first consider an approximation for its constituent function $\Psi(\gamma)$. From (10), we have

$$\Psi(\gamma) = C^{-1}(1 - C(\gamma)) = C^{-1}(U(\gamma)). \quad (49)$$

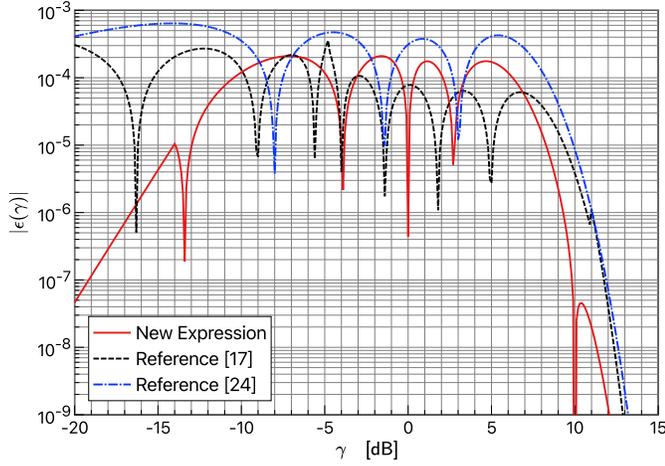


Fig. 2. Approximation error of the proposed expression. Those based on ten Brink et al. [17] and Brännström et al. [24] are also shown for comparison.

Therefore, it is necessary to find the inverse function for $\hat{C}(\gamma)$ defined in (47).

Let us first consider the case $\gamma \in \mathcal{R}_1$. From (47a), the inverse function for the equation $\hat{C}(\gamma) = c$ can be expressed as

$$\hat{C}^{-1}(c) = \frac{1}{4} \left(1 - \frac{3}{A(c)} + A(c) \right), \quad c < C_1, \quad (50)$$

where $C_1 \triangleq 1 - U_1$ and

$$A(c) \triangleq \left(-5 + 24 (\ln 2) c + 2\sqrt{13 + 12 (\ln 2) c (12 (\ln 2) c - 5)} \right)^{\frac{1}{3}}. \quad (51)$$

On the other hand, when $\gamma \in \mathcal{R}_2$, we have

$$\hat{C}^{-1}(c) = \left[-\frac{1}{H_{2,1}} \ln \left(1 - c^{\frac{1}{H_{2,2}}} \right) \right]^{\frac{1}{H_{2,2}}}, \quad C_1 < c < C_2, \quad (52)$$

where $C_2 \triangleq 1 - U_2$. The function $\hat{C}^{-1}(c)$ in the case of $\gamma \in \mathcal{R}_3$ can be derived in a similar manner. Finally, in the case of $\gamma \in \mathcal{R}_4$, we have

$$\hat{C}^{-1}(c) = \frac{1}{2} W_0 \left(2 \left(\frac{\alpha}{1-c} \right)^2 \right), \quad c > C_3, \quad (53)$$

where $C_3 \triangleq 1 - U_3$ and $W_0(x)$ is the corresponding Lambert W function [29], i.e., the value of w that satisfies $w e^w = x$ for a given $x > 0$.

In summary, we have

$$\hat{C}^{-1}(c) = \begin{cases} \frac{1}{4} \left(1 - \frac{3}{A(c)} + A(c) \right), & c < C_1, & (54a) \\ \left[-\frac{1}{H_{2,1}} \ln \left(1 - c^{\frac{1}{H_{2,2}}} \right) \right]^{\frac{1}{H_{2,2}}}, & C_1 \leq c < C_2, & (54b) \\ \left[-\frac{1}{H_{3,1}} \ln \left(1 - c^{\frac{1}{H_{3,2}}} \right) \right]^{\frac{1}{H_{3,2}}}, & C_2 \leq c < C_3, & (54c) \\ \frac{1}{2} W_0 \left(2 \left(\frac{\alpha}{1-c} \right)^2 \right), & c \geq C_3, & (54d) \end{cases}$$

where $C_1 \approx 0.055523$, $C_2 \approx 0.721452$, and $C_3 \approx 0.999983$.

By substituting $\hat{U}(\gamma)$ of (46) into c of (54), we obtain the approximate value of $\hat{\Psi}(\gamma) = \hat{C}^{-1}(\hat{U}(\gamma))$ for a given γ , which we denote by $\hat{\Psi}(\gamma) \triangleq \hat{C}^{-1}(\hat{U}(\gamma))$.

In what follows, we derive simple asymptotic closed-form expressions of $\hat{\Psi}(\gamma)$ and thus $\Lambda(\xi)$ when γ becomes extremely small or large in order to simplify the computation with negligible loss of accuracy. Also, as we will see, it is not actually necessary to implement the $W_0(\cdot)$ function for polar code construction.

A. For Small γ

We consider the case where γ is small and satisfies $\gamma \leq \Gamma_0$ for some boundary Γ_0 such that the condition of (54d) holds, i.e., Γ_0 is the value of γ that satisfies $\hat{U}(\gamma) = C_3 = 1 - U_3$. Among the four possible expressions of $\hat{U}(\gamma)$ in (46), if we apply (46a) with γ replaced by Γ_0 , then we have

$$\frac{1}{\ln 2} \left(\Gamma_0 - \Gamma_0^2 + \frac{4}{3} \Gamma_0^3 \right) = U_3. \quad (55)$$

By numerically solving the above equation with respect to Γ_0 , we obtain,

$$\Gamma_0 \approx 1.21974 \times 10^{-5}, \quad (56)$$

and thus we observe that $\Gamma_0 \ll \Gamma_1 = 0.04$, satisfying the condition of (46a) for γ . As a result, from (54d) with c replaced by $U_1(\gamma)$ of (46a), we have

$$\hat{\Psi}(\gamma) = \frac{1}{2} W_0 \left(2 \left(\frac{\alpha}{\frac{1}{\ln 2} (\gamma - \gamma^2 + \frac{4}{3} \gamma^3)} \right)^2 \right), \quad \gamma < \Gamma_0. \quad (57)$$

By defining $f(\gamma) \triangleq \gamma - \gamma^2 + \frac{4}{3} \gamma^3$, we may express (57) as

$$\hat{\Psi}(\gamma) = \frac{1}{2} W_0 \left(2 \alpha^2 (\ln 2)^2 f^{-2}(\gamma) \right), \quad \gamma < \Gamma_0. \quad (58)$$

Since the argument of the function $W_0(\cdot)$ in (58) becomes larger as γ decreases, we may invoke the following asymptotic form of $W_0(x)$ for large x [29]:

$$W_0(x) \approx \ln x - \ln \ln x + \frac{\ln \ln x}{\ln x} + \dots \quad (59)$$

On applying (59) to (58), we note that

$$\begin{aligned} & \ln \left(2 \alpha^2 (\ln 2)^2 f^{-2}(\gamma) \right) \\ &= \ln 2 + 2 \ln(\ln 2) + 2 \ln \alpha - 2 \ln \gamma - 2 \ln \left(1 - \gamma + \frac{4}{3} \gamma^2 \right), \end{aligned} \quad (60)$$

where the last term on the right hand side turns out to be negligible in the range of $\gamma < \Gamma_0$. Thus, by defining the dominant terms in (60) as

$$B(\xi) \triangleq \ln 2 + 2 (\ln \alpha + \ln(\ln 2)) - 2\xi, \quad (61)$$

where $\xi \triangleq \ln \gamma$, we may equivalently express $\ln \hat{\Psi}(\gamma)$ with respect to ξ , denoted by $\Lambda(\xi)$, as

$$\begin{aligned} \Lambda(\xi) &= \ln \hat{\Psi}(\gamma) \Big|_{\gamma=e^\xi} \\ &\approx \ln \left(B(\xi) + \left(\frac{1}{B(\xi)} - 1 \right) \ln B(\xi) \right) - \ln 2. \end{aligned} \quad (62)$$

The above approximation is valid in the range of $\gamma < \Gamma_0$, i.e.,

$$\xi < \Xi_0 \triangleq \ln \Gamma_0 \approx -11.3143. \quad (63)$$

B. For Large γ

When $\gamma \geq \Gamma_3$, i.e., $\gamma \in \mathcal{R}_4$, we observe from (46d) with the condition of (39) that $\hat{U}(\gamma) \leq U_3 \ll C_1$ and thus from (54a) we have

$$\hat{\Psi}(\gamma) = \frac{1}{4} \left(1 - \frac{3}{A(c)} + A(c) \right), \quad c = \alpha \frac{e^{-\gamma}}{\sqrt{\gamma}}. \quad (64)$$

For $A(c)$ in (51) with small c , applying the first-order approximation as

$$\sqrt{13 + 12(\ln 2)c(12(\ln 2)c - 5)} \approx \sqrt{13} - \frac{30}{\sqrt{13}}(\ln 2)c, \quad (65)$$

we have

$$A(c) \approx (-5 + 2\sqrt{13})^{\frac{1}{3}} \left(1 + \frac{4}{\sqrt{13}}(\ln 2)c \right), \quad (66)$$

and by substituting (66) into (64), we obtain

$$\hat{\Psi}(\gamma) \approx (\ln 2)c \left(1 - \frac{2(\sqrt{13}+1)(\ln 2)}{13+4\sqrt{13}(\ln 2)c} c \right). \quad (67)$$

Taking the logarithm of both sides, we have

$$\ln \hat{\Psi}(\gamma) \approx \ln c + \ln(\ln 2) + \ln \left(1 - \frac{2(\sqrt{13}+1)(\ln 2)}{13+4\sqrt{13}(\ln 2)c} c \right). \quad (68)$$

By substituting $c = \alpha \frac{e^{-\gamma}}{\sqrt{\gamma}}$ from (46d) and noticing that the third term on the right-hand side of (68) becomes negligible as γ increases, we have

$$\ln \hat{\Psi}(\gamma) \approx \ln \alpha + \ln(\ln 2) - \gamma - \frac{1}{2} \ln \gamma, \quad (69)$$

or in terms of ξ as

$$\Lambda(\xi) \approx \ln \alpha + \ln(\ln 2) - e^\xi - \frac{1}{2}\xi. \quad (70)$$

Note that (69) indicates that $\hat{\Psi}(\gamma)$ becomes infinitesimal as γ increases.

C. Complete RCA Algorithm

In Section II, the RCA algorithm was summarized as **Algorithm 1** and **Algorithm 2**, depending on the condition of the channel SNR. Based on the mathematical derivations in this section, the calculation process of the function $\Lambda(\xi)$ is summarized as **Algorithm 3**. As mentioned in Section II-B, GA-based approaches generally involve the use of an inverse function that cannot be described (or accurately approximated) using a closed form expression, and thus require some iterative algorithms (such as the bisection method) to guarantee the required precision. On the other hand, the proposed approach does not require such numerical algorithms and thus the computational effort is in general lower than those based on GA (or IGA).

Algorithm 3 Calculating $\Lambda(\xi)$

Input: $\xi = \ln \gamma$

Output: $\Lambda(\xi) = \ln \hat{\Psi}(\gamma)$

```

1:  $\alpha = 1.16125, \Gamma_1 = 0.04, \Gamma_2 = 1, \Gamma_3 = 10,$ 
    $\Xi_0 = -11.3143$ 
2:  $C_1 = 0.055523, C_2 = 0.721452$ 
3:  $H_{2,1} = 1.396634, H_{2,2} = 0.872764, H_{2,3} = 1.148562$ 
4:  $H_{3,1} = 1.266967, H_{3,2} = 0.938175, H_{3,3} = 0.986830$ 
5: if  $\xi < \Xi_0$  then
6:    $B = \ln 2 + 2 \ln(\ln 2) + 2 \ln \alpha - 2\xi$ 
7:   return  $\ln(B + (\frac{1}{B} - 1) \ln B) - \ln 2$ 
8: end if
9:  $\gamma \leftarrow \exp(\xi)$ 
10: if  $\gamma > \Gamma_3$  then
11:   return  $\ln(\ln 2) + \ln \alpha - \gamma - \xi/2$ 
12: else if  $\gamma < \Gamma_1$  then
13:    $U \leftarrow 1 - \frac{1}{\ln 2}(\gamma - \gamma^2 + \frac{4}{3}\gamma^3)$ 
14: else if  $\gamma < \Gamma_2$  then
15:    $U \leftarrow 1 - \left( 1 - e^{-H_{2,1}\gamma^{H_{2,2}}} \right)^{H_{2,3}}$ 
16: else
17:    $U \leftarrow 1 - \left( 1 - e^{-H_{3,1}\gamma^{H_{3,2}}} \right)^{H_{3,3}}$ 
18: end if
19: if  $U < C_1$  then
20:    $A = (-5 + 24(\ln 2)U$ 
       $+ 2\sqrt{13 + 12(\ln 2)U(12(\ln 2)U - 5)})^{\frac{1}{3}}$ 
21:   return  $\ln(1 - \frac{3}{A} + A) - 2 \ln 2$ 
22: else if  $U < C_2$  then
23:   return  $\frac{1}{H_{2,2}} \left[ \ln \left( -\ln \left( 1 - U^{\frac{1}{H_{2,3}}} \right) \right) - \ln H_{2,1} \right]$ 
24: else
25:   return  $\frac{1}{H_{3,2}} \left[ \ln \left( -\ln \left( 1 - U^{\frac{1}{H_{3,3}}} \right) \right) - \ln H_{3,1} \right]$ 
26: end if

```

V. SIMULATION RESULTS

In this section, we focus on AWGN channels with BPSK, where the SNRs of all the channels are identical. Let γ_0 denote the design SNR input to **Algorithm 2**, and $\hat{\gamma}_{0(\gamma_0)}, \hat{\gamma}_{1(\gamma_0)}, \dots, \hat{\gamma}_{N-1(\gamma_0)}$ denote all the output SNR values.

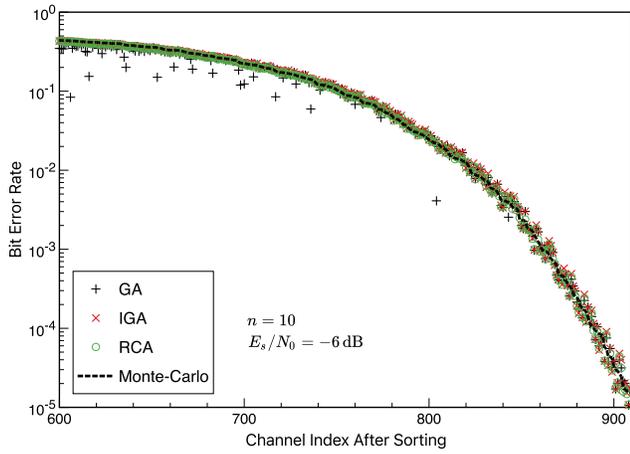
A. BER Estimation

Similar to the case of GA, for the purpose of BER estimation we assume that the distribution of the LLR corresponding to the k th bit channel is modeled as Gaussian with SNR $\hat{\gamma}_{k(\gamma_0)}$. Then, under the assumption that all the previous bits in the SC decoding are correctly decoded, the error rate of the k th bit channel can be estimated as

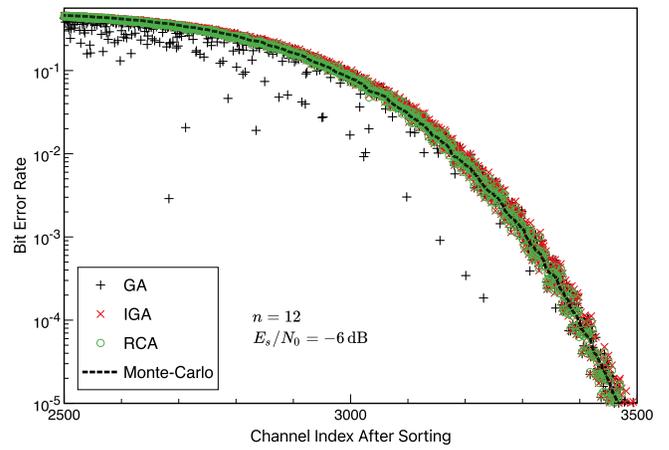
$$P_{b,k}(\gamma_0) = Q \left(\sqrt{\frac{\hat{\gamma}_{k(\gamma_0)}}{2}} \right), \quad (71)$$

with the Q -function defined as $Q(x) = \frac{1}{2} \operatorname{erfc} \left(\frac{x}{\sqrt{2}} \right) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-\frac{t^2}{2}} dt$.

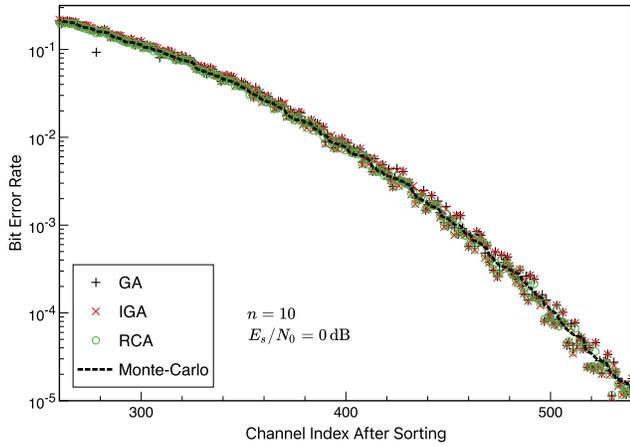
By Monte-Carlo (MC) simulations (with 10^8 trials) we obtained the BER for a given channel SNR γ_0 , and sorted the channels in descending order of BER. We then compared



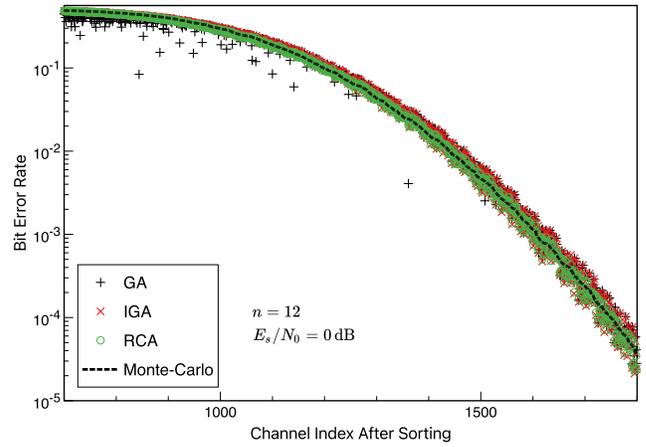
(a)



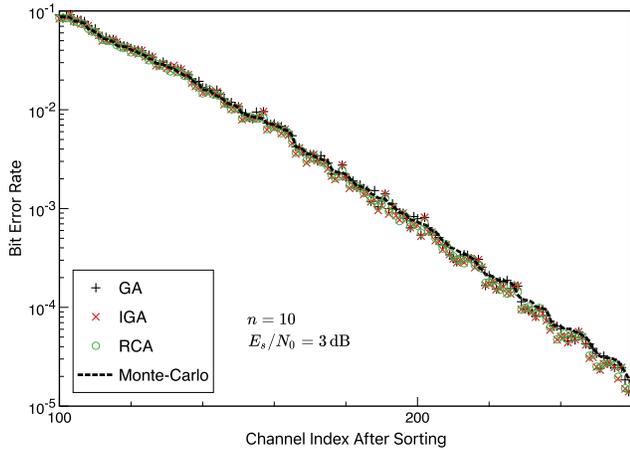
(a)



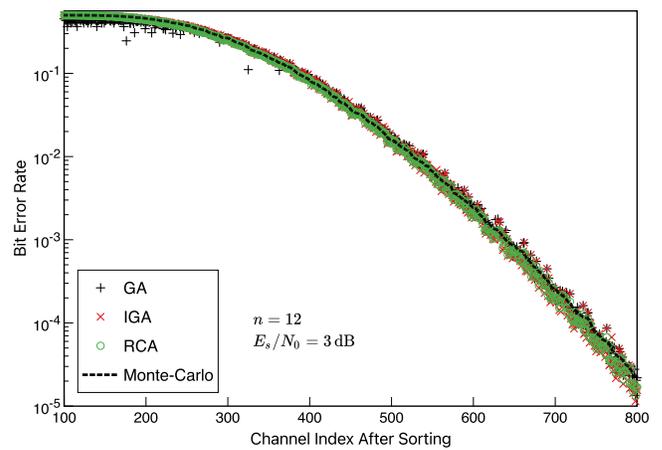
(b)



(b)



(c)



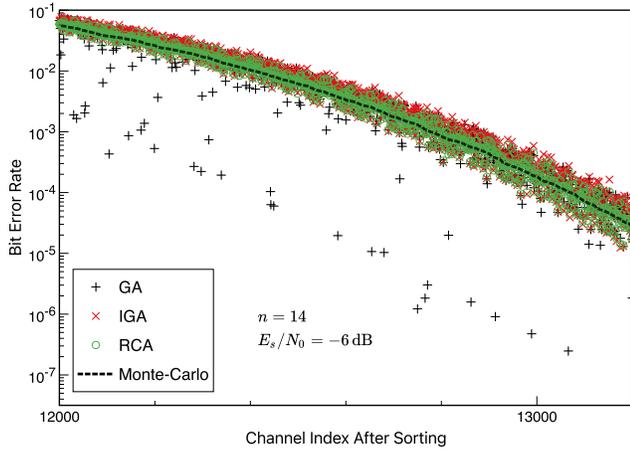
(c)

Fig. 3. Comparison of BER based on Monte-Carlo simulation as well as the corresponding BER estimates based on GA, IGA, and RCA. Polar code length is $N = 2^{10} = 1024$, and the channel indices are sorted according to the BER of Monte-Carlo simulation in descending order. Only a subset of BER above 10^{-5} is plotted. (a) $E_s/N_0 = -6$ dB. (b) $E_s/N_0 = 0$ dB. (c) $E_s/N_0 = 3$ dB.

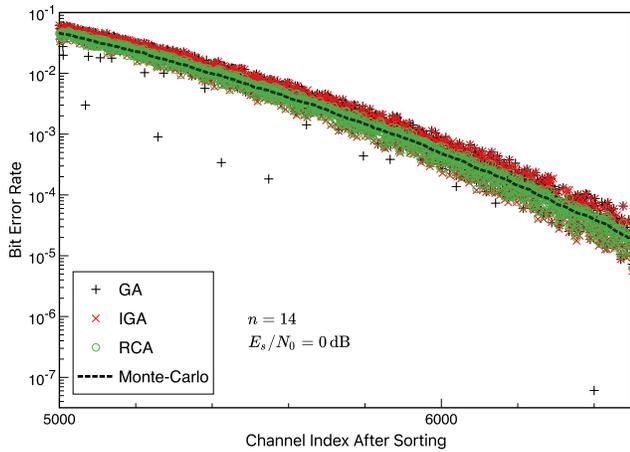
Fig. 4. Comparison of BER based on Monte-Carlo simulation as well as the corresponding BER estimates based on GA, IGA, and RCA. Polar code length is $N = 2^{12} = 4096$, and the channel indices are sorted according to the BER of Monte-Carlo simulation in descending order. Only a subset of BER above 10^{-5} is plotted. (a) $E_s/N_0 = -6$ dB. (b) $E_s/N_0 = 0$ dB. (c) $E_s/N_0 = 3$ dB.

the sorted BERs obtained by MC with the estimated BER (71) of the corresponding channels (i.e., sorted in the same order as those of the MC simulation) based on estimates obtained by

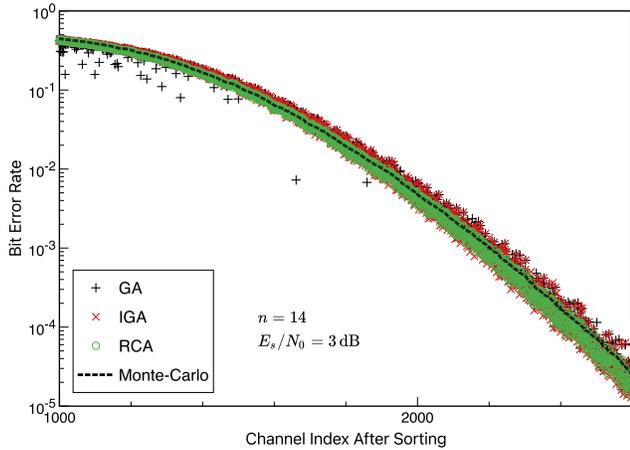
GA with (8), IGA of [8], and the proposed RCA. The results are shown in Figs. 3, 4, and 5, where the codeword lengths are chosen as $N = 2^{10} = 1024$, $2^{12} = 4096$, and $2^{14} = 16384$,



(a)



(b)

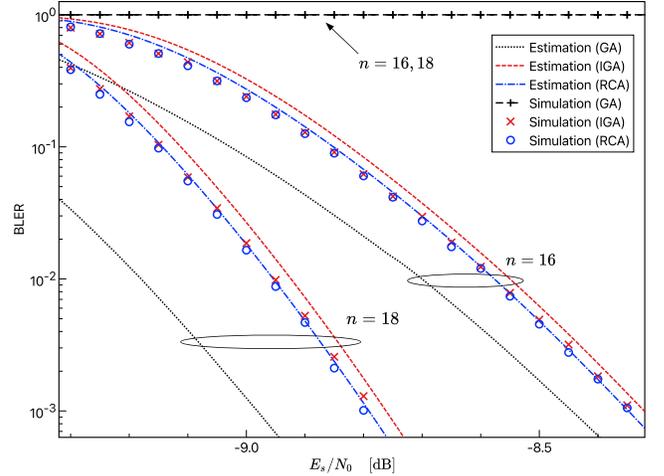


(c)

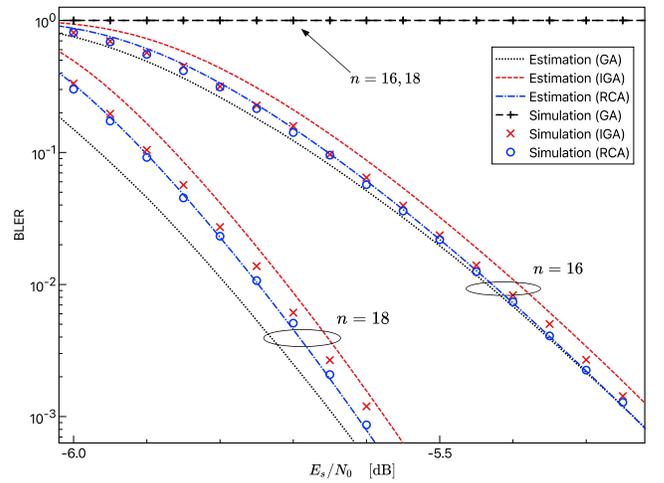
Fig. 5. Comparison of BER based on Monte-Carlo simulation as well as the corresponding BER estimates based on GA, IGA, and RCA. Polar code length is $N = 2^{14} = 16384$, and the channel indices are sorted according to the BER of Monte-Carlo simulation in descending order. Only a subset of BER above 10^{-5} is plotted (except for GA that significantly underestimates the BER). (a) $E_s/N_0 = -6$ dB. (b) $E_s/N_0 = 0$ dB. (c) $E_s/N_0 = 3$ dB.

respectively, and for each case the channel SNRs are selected as $\gamma_0 = -6$ dB, 0 dB, and 3 dB.

We observe that in the case of a relatively short codeword length ($N = 2^{10}$), all the three schemes show similar



(a)



(b)

Fig. 6. Simulated BLER of very low-rate polar codes with $N = 2^n$ ($n = 16$ and 18) designed at each channel SNR as well as the corresponding BLER estimates: (a) $R = 0.125$, (b) $R = 0.25$.

BER estimates as observed in Fig. 3. On the other hand, as N increases and γ_0 decreases, the BER estimates obtained by the GA based on (8) show significant gaps from those obtained by MC simulation. In particular, GA significantly underestimates the BER of many bit channels, which would lead to an inappropriate selection of unfrozen bits and poor BLER performance. In all the compared long codeword cases, we observe that RCA yields better BER estimates than IGA.

B. BLER Estimation and Comparison

We now estimate the BLER performance for a given code rate. We sort the estimated SNR such that

$$\hat{\gamma}_{I_0(\gamma_0)} \geq \hat{\gamma}_{I_1(\gamma_0)} \geq \dots \geq \hat{\gamma}_{I_{N-1}(\gamma_0)}, \quad (72)$$

where the subscript $I_k(\gamma_0)$ corresponds to the index of the bit channel having the k th highest SNR (with $k = 0$ representing the maximum), with emphasis on the fact that the design SNR is γ_0 .

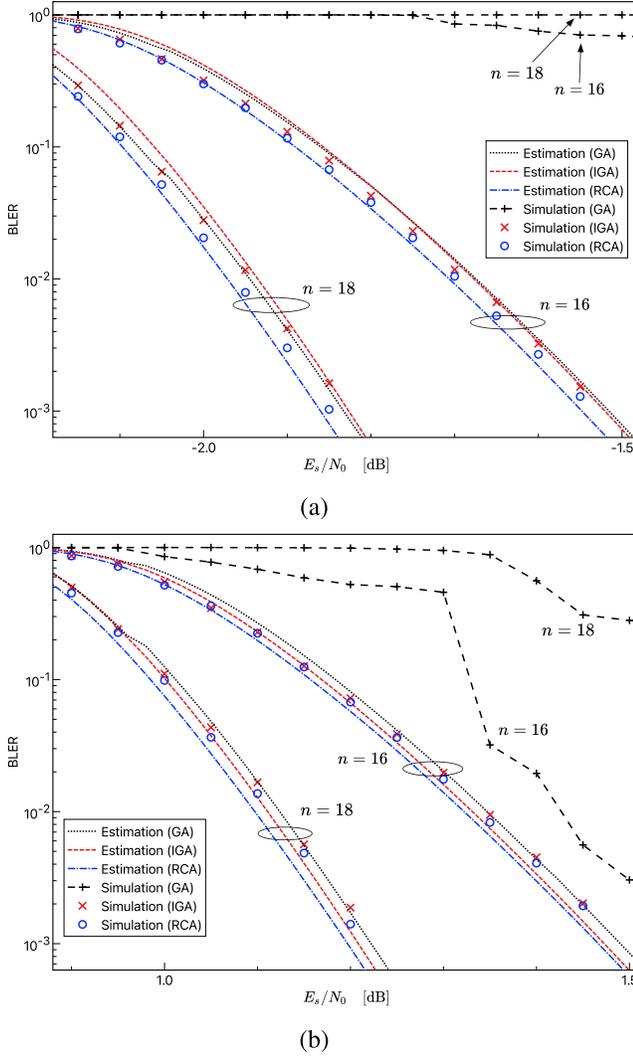


Fig. 7. Simulated BLER of moderate to high-rate polar codes with $N = 2^n$ ($n = 16$ and 18) designed at each channel SNR as well as the corresponding BLER estimates: (a) $R = 0.5$, (b) $R = 0.75$.

Let us define the index set $\mathcal{I}_K(\gamma_0)$ consisting of the K elements with highest SNRs as

$$\mathcal{I}_K(\gamma_0) \triangleq \{I_0(\gamma_0), I_1(\gamma_0), \dots, I_{K-1}(\gamma_0)\}, \quad (73)$$

where $R \triangleq K/N$ corresponds to the code rate. For each given channel SNR γ_0 , we may define the estimated BLER as

$$P_{BL}(K, \gamma_0) \triangleq 1 - \prod_{k \in \mathcal{I}_K(\gamma_0)} (1 - P_{b,k}(\gamma_0)). \quad (74)$$

The above estimate may serve as a good approximation if (i) the estimated SNR values in the selected information set $\mathcal{I}_K(\gamma_0)$ are accurate, and (ii) the LLRs of the corresponding bit channels are approximately Gaussian and each BER is then well characterized by (71).

Fig. 6 compares the BLER of polar codes designed based on GA, IGA, and RCA with codeword lengths $N = 2^{16}$ and 2^{18} in the case of very low code rates ($R = 1/8$ and $R = 1/4$). The moderate to higher code rate cases ($R = 1/2$ and $R = 3/4$) are shown in Fig. 7. The corresponding BLER estimates based on (74) are also plotted. Note that for all

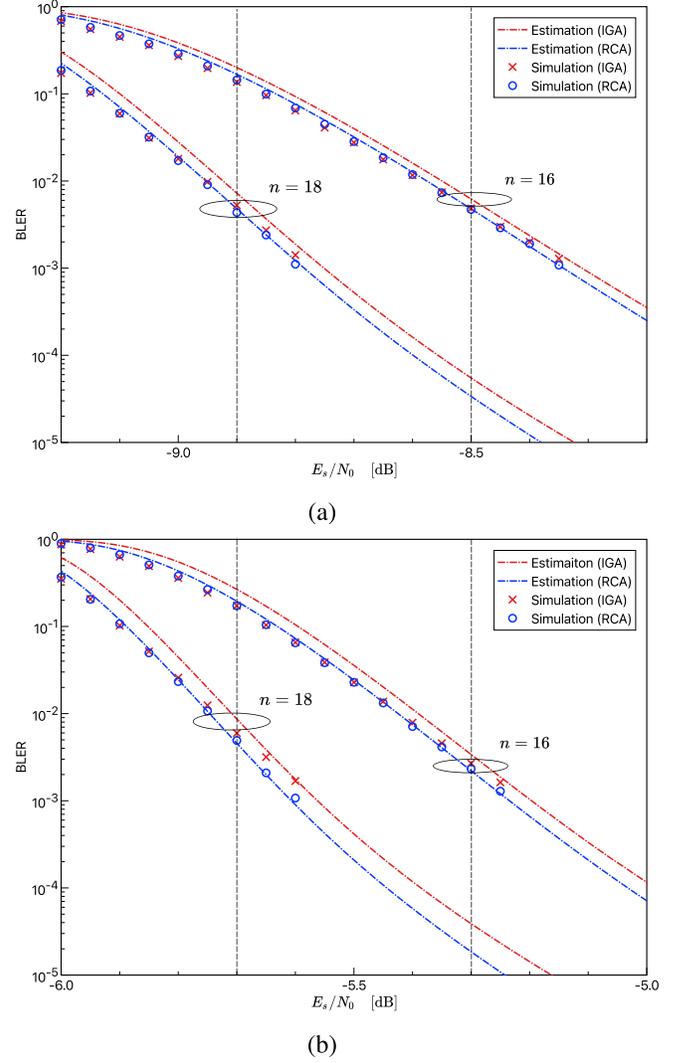


Fig. 8. Simulated BLER of very low-rate polar codes with $N = 2^n$ ($n = 16$ and 18) designed at specific SNR as well as the corresponding BLER estimates: (a) $R = 0.125$, (b) $R = 0.25$. The vertical lines indicate the corresponding design SNR.

simulation results presented here, the polar code is constructed for each given channel SNR. As one can observe, the conventional GA based on (8) fails to accurately estimate the BLER from (74) in all the cases compared here. Furthermore, the corresponding simulated BLER is almost 1 when the SNR region of interest is low, which can be inferred from the BER results shown in the previous subsection. Therefore, in what follows, we only focus on the performance achieved by RCA and IGA. From all the results compared, we observe that the polar codes designed based on the proposed RCA algorithms outperform those based on IGA, and the gap is especially noticeable when the code rate is moderate. We also notice that, as expected from the BER comparison results shown in Section V-A, the estimated BLER values computed from (74) and those obtained by simulation are in general closer when RCA is employed, indicating that the estimation accuracy of the equivalent SNR obtained by RCA is higher than that of IGA. Therefore, considering the fact that the computational complexity required for performing the proposed RCA is

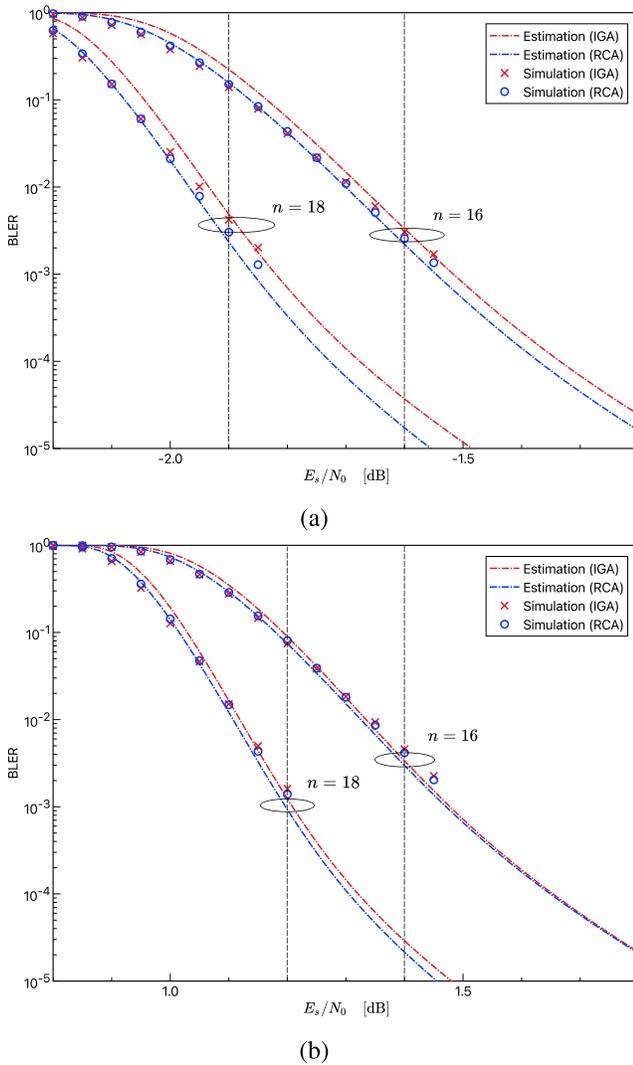


Fig. 9. Simulated BLER of moderate to high-rate polar codes with $N = 2^n$ ($n = 16$ and 18) designed at specific SNR as well as the corresponding BLER estimates: (a) $R = 0.5$, (b) $R = 0.75$. The vertical lines indicate the corresponding design SNR.

lower than those of GA and IGA, the proposed RCA may be preferable to GA-based approaches.

We also note that, as in Fig. 7(b), the BLER estimated by RCA is slightly optimistic for high rate code cases. This can also be inferred from the BER estimate results shown in Figs. 4(c) and 5(c), where RCA tends to offer slightly optimistic BER estimates when SNR increases.

C. BLER Comparison With Fixed Design SNR

Finally, we compare more practical cases where the code is designed for a certain design SNR. In other words, instead of (74), we evaluate

$$P_{BL}(K, \gamma_0, \gamma_c) \triangleq 1 - \prod_{k \in \mathcal{I}_K(\gamma_0)} (1 - P_{b,k}(\gamma_c)), \quad (75)$$

where γ_c is the actual channel SNR, whereas γ_0 remains the design SNR. In other words, (74) corresponds to the case with $\gamma_0 = \gamma_c$, i.e., $P_{BL}(K, \gamma_c, \gamma_c)$ of (75). The simulation results and the corresponding BLER estimates based on (75)

are compared in Fig. 8 for very low code rates ($R = 1/8$ and $R = 1/4$) and in Fig. 9 for moderate to higher code rate cases ($R = 1/2$ and $R = 3/4$), all of them with codeword lengths $N = 2^{16}$ and 2^{18} . (Note that the performance of the conventional GA based on (8) is not shown in the remaining figures since, as seen in the previous subsection, it is significantly worse than that of IGA.) For each result, the design SNR is indicated by the vertical line, and the same design SNR is applied to both IGA and RCA. Each design SNR is selected as an approximate SNR where their BLER estimate achieves below 10^{-2} . We observe that in these cases, if we compare the simulation results at the channel SNR equal to the design SNR or higher (i.e., $\gamma_c \geq \gamma_0$), RCA is better than IGA, while for lower channel SNR (i.e., $\gamma_c < \gamma_0$), IGA may be marginally better when the BLER is $\approx 10^{-1}$ or greater. This suggests that RCA has better estimation accuracy at the target channel SNR and thus outperforms IGA in terms of simulated BLER as well as its estimated value in the range of practical interest. Finally, we note that the BLER curves tend to decrease slowly as SNR increases in all the cases evaluated here. The reason is that the optimal code changes according to the channel SNR, and thus the code designed at some specific SNR will become sub-optimal when it is evaluated at different SNR due to the inaccurate polarization caused by SNR mismatch.

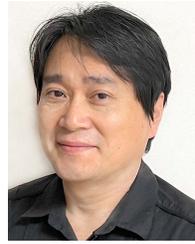
VI. CONCLUSION

In this work, we have proposed an explicit closed-form algorithm for polar code design based on RCA. Comparisons with MC simulation results have revealed that RCA can estimate BER and BLER more accurately than conventional low-complexity approaches including improved GA. As a result, polar code designed based on the proposed RCA outperforms those based on the popular GA-based approaches with no increase of computational complexity. The benefit of the proposed approach becomes significant for polar codes with large codeword lengths and moderate code rates.

REFERENCES

- [1] E. Arkan, "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] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 619–637, Feb. 2001.
- [3] T. J. Richardson and R. L. Urbanke, *Modern Coding Theory*. Cambridge, U.K.: Cambridge Univ. Press, 2008.
- [4] R. Mori and T. Tanaka, "Performance of polar codes with the construction using density evolution," *IEEE Commun. Lett.*, vol. 13, no. 7, pp. 519–521, Jul. 2009.
- [5] I. Tal and A. Vardy, "How to construct polar codes," *IEEE Trans. Inf. Theory*, vol. 59, no. 10, pp. 6562–6582, Oct. 2013.
- [6] S.-Y. Chung, T. J. Richardson, and R. L. Urbanke, "Analysis of sum-product decoding of low-density parity-check codes using a Gaussian approximation," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 657–670, Feb. 2001.
- [7] P. Trifonov, "Efficient design and decoding of polar codes," *IEEE Trans. Commun.*, vol. 60, no. 11, pp. 3221–3227, Nov. 2012.
- [8] H. Ochiai, P. Mitran, and H. V. Poor, "Capacity-approaching polar codes with long codewords and successive cancellation decoding based on improved Gaussian approximation," *IEEE Trans. Commun.*, vol. 69, no. 1, pp. 31–43, Jan. 2021.
- [9] I. Tal and A. Vardy, "List decoding of polar codes," *IEEE Trans. Inf. Theory*, vol. 61, no. 5, pp. 2213–2226, May 2015.

- [10] E. Arkan, "A performance comparison of polar codes and Reed–Müller codes," *IEEE Commun. Lett.*, vol. 12, no. 6, pp. 447–449, Jun. 2008.
- [11] A. Elkelesh, M. Ebada, S. Cammerer, and S. ten Brink, "Decoder-tailored polar code design using the genetic algorithm," *IEEE Trans. Commun.*, vol. 67, no. 7, pp. 4521–4534, Jul. 2019.
- [12] Y. Liao, S. A. Hashemi, J. M. Cioffi, and A. Goldsmith, "Construction of polar codes with reinforcement learning," *IEEE Trans. Commun.*, vol. 70, no. 1, pp. 185–198, Jan. 2022.
- [13] V. Bioglio, F. Gabry, I. Land, and J.-C. Belfiore, "Multi-kernel polar codes: Concept and design principles," *IEEE Trans. Commun.*, vol. 68, no. 9, pp. 5350–5362, Sep. 2020.
- [14] S.-Y. Chung, "On the construction of some capacity-approaching coding schemes," Ph.D. thesis, MIT, Cambridge, MA, USA, 2000.
- [15] A. Ashikhmin, G. Kramer, and S. ten Brink, "Extrinsic information transfer functions: Model and erasure channel properties," *IEEE Trans. Inf. Theory*, vol. 50, no. 11, pp. 2657–2673, Nov. 2004.
- [16] S. ten Brink and G. Kramer, "Design of repeat-accumulate codes for iterative detection and decoding," *IEEE Trans. Signal Process.*, vol. 51, no. 11, pp. 2764–2772, Nov. 2003.
- [17] S. ten Brink, G. Kramer, and A. Ashikhmin, "Design of low-density parity-check codes for modulation and detection," *IEEE Trans. Commun.*, vol. 52, no. 4, pp. 670–678, Apr. 2004.
- [18] E. Sharon, A. Ashikhmin, and S. Litsyn, "EXIT functions for binary input memoryless symmetric channels," *IEEE Trans. Commun.*, vol. 54, no. 7, pp. 1207–1214, Jul. 2006.
- [19] D. Divsalar, S. Dolinar, C. R. Jones, and K. Andrews, "Capacity-approaching protograph codes," *IEEE J. Sel. Areas Commun.*, vol. 27, no. 6, pp. 876–888, Aug. 2009.
- [20] B. P. Smith, A. Farhood, A. Hunt, F. R. Kschischang, and J. Lodge, "Staircase codes: FEC for 100 Gb/s OTN," *J. Lightw. Technol.*, vol. 30, no. 1, pp. 110–117, Jan. 1, 2012.
- [21] H. Mahdaviifar and A. Vardy, "Achieving the secrecy capacity of wiretap channels using polar codes," *IEEE Trans. Inf. Theory*, vol. 57, no. 10, pp. 6428–6443, Oct. 2011.
- [22] K. Vakilinia, D. Divsalar, and R. D. Wesel, "RCA analysis of the polar codes and the use of feedback to aid polarization at short blocklengths," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2015, pp. 1292–1296.
- [23] D. Dosio, "Polar codes for error correction: Analysis and decoding algorithms," M.S. thesis, Laurea Magistrale, Università di Bologna, Bologna, Italy, 2016.
- [24] F. Brännström, L. K. Rasmussen, and A. J. Grant, "Convergence analysis and optimal scheduling for multiple concatenated codes," *IEEE Trans. Inf. Theory*, vol. 51, no. 9, pp. 3354–3364, Sep. 2005.
- [25] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Trans. Inf. Theory*, vol. 42, no. 2, pp. 429–445, Mar. 1996.
- [26] 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.
- [27] H. Vangala, E. Viterbo, and Y. Hong, "A comparative study of polar code constructions for the AWGN channel," Jan. 2015, *arXiv:1501.02473*.
- [28] A. Lozano, A. M. Tulino, and S. Verdú, "Optimum power allocation for parallel Gaussian channels with arbitrary input distributions," *IEEE Trans. Inf. Theory*, vol. 52, no. 7, pp. 3033–3051, Jul. 2006.
- [29] R. M. Corless, G. H. Gonnet, D. E. G. Hare, D. J. Jeffrey, and D. E. Knuth, "On the Lambert W function," *Adv. Comput. Math.*, vol. 5, no. 1, pp. 329–359, Dec. 1996.



Hideki Ochiai (Senior Member, IEEE) received the B.E. degree in communication engineering from Osaka University, Osaka, Japan, in 1996, and the M.E. and Ph.D. degrees in information and communication engineering from The University of Tokyo, Tokyo, Japan, in 1998 and 2001, respectively.

From 2001 to 2003, he was a Research Associate at The University of Electro-Communications, Tokyo. From 2003 to 2004, he was a Visiting Scientist at Harvard University, Cambridge, MA, USA. From 2019 to 2020, he was a Visiting Professor at the University of Waterloo, ON, Canada, and a Visiting Fellow at Princeton University, Princeton, NJ, USA. Since April 2003, he has been with Yokohama National University, Yokohama, Japan, where he is currently a Professor. His research interests include wireless communications and networks. He has served as an Editor for the *IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS* from 2007 to 2011 and the *IEEE WIRELESS COMMUNICATIONS LETTERS* from 2011 to 2016. He has been elevated to IEEE Fellow for contributions to power and spectral efficient wireless communication in 2023.



Kosuke Ikeya received the B.E. degree from Yokohama National University, Yokohama, Japan, in 2021, where he is currently pursuing the M.E. degree in communication engineering with the Department of Electrical and Computer Engineering, Graduate School of Engineering.



Patrick Mitran (Senior Member, IEEE) received the bachelor's and master's degrees in electrical engineering from McGill University, Montreal, Canada, in 2001 and 2002, respectively, and the Ph.D. degree from the School of Engineering and Applied Sciences, Harvard University, Cambridge, MA, USA, in 2006.

In 2005, he was a Research Scientist with the Radio Communications Laboratory, Intel Corporation. From 2006 to 2007, he was a Lecturer in applied mathematics with the School of Engineering and Applied Sciences, Harvard University. He is currently a Professor with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. His current research interests include information theory and wireless communications, cross-layer design, and signal processing for 5G. He was a recipient of the Government of Ontario Early Researcher Award in 2011. He is a Licensed Professional Engineer in the Province of Ontario. From 2012 to 2016, he served as an Editor for the *IEEE TRANSACTIONS ON COMMUNICATIONS*. From 2016 to 2019, he served as an Associate Editor for the *IEEE TRANSACTIONS ON INFORMATION THEORY*.