CRC-Aided List Decoding of Convolutional Codes in the Short Blocklength Regime

Hengjie Yang, *Student Member, IEEE*, Ethan Liang, *Student Member, IEEE*, Minghao Pan, and Richard D. Wesel, *Fellow, IEEE*

***Abstract*—We consider the concatenation of a convolutional code (CC) with an optimized cyclic redundancy check (CRC) code as a promising paradigm for good short blocklength codes. The resulting CRC-aided convolutional code naturally permits the use of serial list Viterbi decoding (SLVD) to achieve maximum-likelihood decoding. The convolutional encoder of interest is of rate-$1/\omega$ and the convolutional code is either zero-terminated (ZT) or tail-biting (TB). The resulting CRC-aided convolutional code is called a CRC-ZTCC or a CRC-TBCC. To design a good CRC-aided convolutional code, we propose the *distance-spectrum optimal (DSO)* CRC polynomial. A DSO CRC search algorithm for the TBCC is provided. Our analysis reveals that the complexity of SLVD is governed by the expected list rank which converges to 1 at high SNR. This allows a good performance to be achieved with a small increase in complexity. In this paper, we focus on transmitting 64 information bits with a rate-$1/2$ convolutional encoder. For a target error probability 10^{-4}, simulations show that the best CRC-ZTCC approaches the random-coding union (RCU) bound within 0.4 dB. Several CRC-TBCCs outperform the RCU bound at moderate SNR values.**

***Index Terms*—Convolutional code, cyclic redundancy check code, list Viterbi decoding, negative acknowledgement, undetected errors.**

I. Introduction

Recently, the coding theory community has witnessed a growing interest in designing powerful short blocklength codes (e.g., codes with a thousand or fewer information bits). This renewed interest is mainly driven by the stringent requirement of new ultra-reliable low-latency communication in 5G [4], and advances in the finite-blocklength information theory developed by Polyanskiy, Poor and Verdú [5]. The basic question of finite-blocklength information theory asks: what is the maximal channel coding rate achievable at a given blocklength n and error probability ϵ? To answer this question, Polyanskiy *et al.* developed the *random-coding union (RCU) bound* $\mathrm{rcu}(n, M)$ [5, Theorem 16] and the *meta-converse (MC) bound* $\mathrm{mc}(n, M)$) [5, Theorem 27] that provide, respectively, tight upper and lower bounds on the error probability $P_e^*(n, M)$ of the best (n, M) code of length n and M codewords. Namely,

$$\mathrm{mc}(n, M) \leq P_e^*(n, M) \leq \mathrm{rcu}(n, M). \tag{1}$$

They also provide the *normal approximation* [5, Eq. 223] that tightly approximates the performance of the best (n, M) code. Thereafter, these bounds serve as benchmarks to assess the performance of a given finite-blocklength code over a broad class of channels, including the discrete memoryless channel (DMC) and the additive white Gaussian noise (AWGN) channel. Due to the prohibitive complexity of an exact computation of the RCU and MC bounds, saddlepoint approximations of these two bounds were developed that are shown to be numerically accurate [6].

For coding theorists, a central task is to construct *structured* short-blocklength codes for the binary-input AWGN channel such that the probability of error falls into the region delimited by the RCU bound and the MC bound at a reasonable decoding complexity. There are numerous approaches to achieve this goal. As a comprehensive overview, Coşkun *et al.* surveyed in detail the contemporary short-blocklength code designs developed in recent decades [7]. Important examples include extended BCH codes under ordered statistics decoding (OSD) [8], [9], tail-biting convolutional codes under wrap-around Viterbi algorithm (WAVA) [10], non-binary low-density parity-check codes [11], [12], non-binary turbo codes [13], [14] and polar codes [15], [16]. Recent advances also include the polarization-adjusted convolutional codes proposed by Arıkan [17], [18]. It is worth noting that if no restrictions are imposed on what kind of codes should be used for the AWGN channel, Shannon [19] has ingeniously shown that the optimal (n, M) code should be placed on a sphere in the n-dimensional Euclidean space such that the total solid angle is evenly split between the M Voronoi regions and every Voronoi region is a perfect circular cone in order to achieve the minimum probability of error.

While there are many possible structures for short-blocklength coding, this paper focuses on the concatenation of a convolutional code with a cyclic redundancy check (CRC) code. The resulting concatenated code is called the *CRC-aided convolutional code*. Convolutional codes were first introduced by Elias [20]. Viterbi decoding of convolutional codes was developed by Viterbi [21] and its maximum-likelihood (ML) nature was recognized by Forney [22], [23]. Advantages of convolutional codes include low decoding latency [24], [25]

This paper was presented in part at IEEE GLOBECOM 2018, 2019 [1], [2] and IEEE ISIT 2020 [3].

This research is supported in part by National Science Foundation (NSF) grant CCF-2008918 and a Qualcomm Faculty Award. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the NSF or of Qualcomm.

H. Yang is with the Department of Electrical and Computer Engineering, University of California, Los Angeles, Los Angeles, CA, 90095 USA (e-mail: hengjie.yang@ucla.edu).

E. Liang is with the Department of Electrical Engineering, Stanford University, Stanford, CA, 94305 USA (e-mail: emliang@stanford.edu)

M. Pan is with the Department of Mathematics, University of California, Los Angeles, Los Angeles, CA, 90095 USA (e-mail: minghaopan@ucla.edu).

R. D. Wesel is with the Department of Electrical and Computer Engineering, University of California, Los Angeles, Los Angeles, CA, 90095 USA (e-mail: wesel@ucla.edu).

and good error correction performance at short blocklength. The term “CRC” stems from the use of cyclic codes for error detection [26], where the cyclic codeword can be put into systematic form with the parity bits easily generated by a linear sequential circuit. As explained in [27], CRC codes are possibly shortened cyclic codes generated by a polynomial whose leading and zero coefficients are nonzero. The order of the generator polynomial defines the blocklength of the associated cyclic code. However, in practice, the CRC code is a subcode of this cyclic code whose blocklength is less than the polynomial order.

The structure of concatenating a convolutional code with a CRC code was first proposed in the context of hybrid automatic repeat request (ARQ) [28] and is used in numerous practical systems where the convolutional code serves as an inner error-correcting code to combat channel errors and the CRC code serves as an error-detecting code to verify if a codeword has been correctly received. Examples include the 3GPP cellular communication standards of both 3G [29] and 4G LTE [30].

The classical decoding approach for a CRC-aided convolutional code in a hybrid ARQ setting is Viterbi decoding with CRC verification. The input sequence identified by Viterbi decoding is checked to determine whether it is divisible by the CRC polynomial. This indicates whether a valid message has been decoded. If the decoded sequence is divisible by the CRC polynomial, the message segment of the decoded sequence is declared as the most likely message. Otherwise, a negative acknowledgement (NACK) is declared and perhaps a retransmission request is sent to the transmitter.

Unfortunately, the classical approach of Viterbi decoding with CRC verification conceals the true potential of the CRC-aided convolutional code. Performing a single Viterbi decoding step causes the decoder to give up too early, often before encountering a convolutional codeword whose input sequence passes the CRC verification. To unleash the power of the CRC-aided convolutional code, we consider the serial list Viterbi decoding (SLVD) pioneered by Seshadri and Sundberg [31]. SLVD sequentially produces a rank ordered list of codewords according to their likelihoods. Hence, CRC verification can naturally be used as a termination criterion for this list decoding.

Practical implementation of the SLVD typically assumes a *constrained maximum list size* Ψ to limit the peak decoding complexity. The SLVD terminates either when an input sequence passes the CRC verification or when the list rank reaches Ψ. The list rank at which the decoder stops is called the *terminating list rank* L. However, it is not always possible to have $L = \Psi$. This is because Ψ can be set arbitrarily large, yet only finitely many codewords exist. This implies that L has an intrinsic maximum achievable value independent of Ψ which is referred to as the *supremum list rank* λ. Consequently, L is a bounded random variable between 1 and $\min\{\lambda, \Psi\}$. Since the decoding complexity is a function of L, the average decoding complexity is a function of the average list rank $\mathsf{E}[L]$.

Assume that $\Psi < \lambda$. In this case, there are three possible outcomes associated with the SLVD: 1) a correct decoding if SLVD identifies the transmitted message within Ψ trials; 2) an undetected error (UE) if an erroneous input sequence found by SLVD passes the CRC verification within Ψ trials; and 3) a NACK and the forced termination of the decoder if the SLVD fails to find an input sequence that passes CRC verification within Ψ trials. In contrast, any value of Ψ with $\Psi \geq \lambda$ gives the same decoder behavior where no NACK is produced. In this case, the SLVD is an implementation of ML decoding of the CRC-aided convolutional code. In the extreme case where $\Psi = 1$, the SLVD reduces to the classical Viterbi decoding with CRC verification.

A classical list decoder [32] assumes a fixed list size and declares decoding success as long as the transmitted codeword is in the list. In contrast, the SLVD has a more stringent requirement for success that can lead to a higher error probability than for the classical list decoder. Several upper bounds on error probability were developed for the classical list decoder, e.g., [33], [34]. However, these results are not directly applicable to the SLVD.

This paper focuses on the concatenation of a rate-$1/\omega$ convolutional code with an optimized CRC code. We explore both zero-terminated convolutional code (ZTCC) and tail-biting convolutional code (TBCC) [35]. The resulting concatenated code is called a *CRC-ZTCC* in the first case and a *CRC-TBCC* in the second case. For CRC-ZTCCs, Lou *et al.* [36] realized that previous designs of CRC polynomials typically ignore the structure of the inner error-correcting code, which leads to suboptimal performance. Lou *et al.* designed optimal CRC polynomials for a given ZTCC such that the probability of UE is minimized for a single Viterbi decoding attempt followed by CRC verification. A key point in their analysis is that when the target probability of UE is low enough, the design principle is equivalent to maximizing the minimum distance of the CRC-ZTCC. However, Lou *et al.* did not address the optimal CRC design for a TBCC and did not consider SLVD.

Compared to the ZTCC, the TBCC has the advantage of avoiding the rate loss incurred by the overhead associated with the zero tail that follows the information sequence, but this overhead reduction comes with an increase in decoding complexity. A TB codeword requires that the initial and terminating states be the same, which can be achieved, for example, by setting the initial encoder memory to be the final bits of the information sequence. However, this requirement increases the difficulty of efficiently identifying the ML path on the trellis because the common value of the initial and terminating states is unknown at the decoder.

One approach to ML decoding of a TBCC is to perform Viterbi decoding from every possible initial state [35]. Various *approximate* algorithms are proposed for decoding the TBCC based on either ML or maximum *a posteriori* probability criterion, e.g., [37]–[40]. Among these algorithms, the WAVA [40] proves to be both efficient and near-ML. Shankar *et al.* [41] introduced an efficient, iterative, two-phase algorithm for *exact* ML decoding of TBCC, where an A* algorithm is applied in the second phase, using information from the first phase to compute the heuristic function. To make the exact SLVD of TBCC possible and efficient, this paper extends Shankar *et al.*’s algorithm to accommodate the CRC polynomial. Specifically, if a traceback identifies a TB path, the CRC of the corresponding input sequence is checked. If the input sequence passes the CRC verification, the algorithm terminates. Otherwise, the algorithm locates the next rank ordered path.

A. Contributions

This paper provides a design paradigm for both CRC-ZTCCs and CRC-TBCCs, a suite of tools for performance analysis of these codes, and a complexity analysis showing that SLVD allows low-complexity decoding at low probability of UE for $\Psi \geq \lambda$, i.e., an average decoding complexity similar to standard Viterbi decoding of the convolutional code alone. These contributions combine to yield, for example, CRC-aided convolutional codes that closely approach the RCU bound while requiring decoding complexity similar to Viterbi decoding on a convolutional code trellis with 2^8 states.

The main contributions of this paper are summarized below.

1) CRC-Aided Convolutional Code Design: This paper introduces the concept of the *distance-spectrum optimal (DSO) CRC polynomial*, which minimizes the theoretical union bound of the probability of UE for $\Psi \geq \lambda$. Theorem 1 shows that for high SNR, the DSO CRC polynomial reduces to the one that obtains the best minimum distance $d^l_{\min}$. Theorem 2 provides a sharp upper bound on the achievable $d^l_{\min}$ based on the distance spectrum of the convolutional code. For low target probability of UE, we present an efficient algorithm for finding DSO CRC polynomials for TBCCs of arbitrary rate, and provide these polynomials for ZTCCs and TBCCs for optimum rate-$1/2$ convolutional encoders in [42] at 64 information bits.

2) CRC-Aided Convolutional Code Performance Analysis: The performance of a CRC-aided convolutional code with the constrained maximum list size Ψ is measured by three probabilities: probability of correct decoding $P_{c,\Psi}$, probability of UE $P_{e,\Psi}$ and probability of NACK $P_{\mathit{NACK},\Psi}$, where $P_{c,\Psi} + P_{e,\Psi} + P_{\mathit{NACK},\Psi} = 1$. This paper provides bounds, approximations, and simulation results characterizing how these probabilities vary with Ψ and with SNR. Theorems 4 – 6 describe how performance evolves as Ψ increases, the existence and behavior of the supremum list rank λ, and performance (in terms of $P_{c,\Psi}$, $P_{e,\Psi}$, and $P_{\mathit{NACK},\Psi}$) as a function of SNR for extreme values of $\Psi = 1$ and $\Psi = \lambda$.

3) CRC-Aided Convolutional Code Decoding Complexity: This paper provides expressions for the complexity of SLVD for CRC-ZTCCs and CRC-TBCCs. These expressions reveal that complexity is a function of the expected list rank $\mathsf{E}[L]$. This paper characterizes $\mathsf{E}[L]$ including a new approach to computing $\mathsf{E}[L]$ in the limit of low SNR, a new analysis of conditional expected list rank given the noise magnitude, and two new approaches for approximating the conditional expected list rank. Our parametric approximation on the conditional expected list rank naturally leads to an accurate approximation of $\mathsf{E}[L]$ as a function of $P_{e,\lambda}$ which shows that as $P_{e,\lambda}$ converges to 0, $\mathsf{E}[L]$ converges to 1 (see Approximation 3 to follow). We see that for practically interesting operating points of $P_{e,\lambda}$ such as 10^{-6}, $\mathsf{E}[L] \approx 1$ for typical CRC lengths. This implies that for an interesting range of CRC lengths, the CRC length can be increased with negligible impact on complexity. Moreover, for these CRC lengths, the complexity of SLVD for the CRC-aided convolutional code is very similar to that of standard Viterbi decoding of the convolutional code alone.

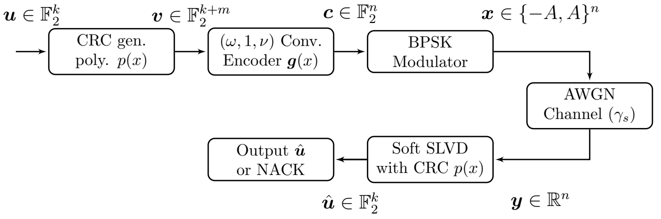
$\boldsymbol{u} \in \mathbb{F}_2^k$
$\boldsymbol{v} \in \mathbb{F}_2^{k+m}$
$\boldsymbol{c} \in \mathbb{F}_2^n$
$\boldsymbol{x} \in \{-A, A\}^n$
CRC gen. poly. $p(x)$
$(\omega, 1, \nu)$ Conv. Encoder $\boldsymbol{g}(x)$
BPSK Modulator
AWGN Channel (γ_s)
Soft SLVD with CRC $p(x)$
Output $\hat{\boldsymbol{u}}$ or NACK
$\hat{\boldsymbol{u}} \in \mathbb{F}_2^k$
$\boldsymbol{y} \in \mathbb{R}^n$

Fig. 1. Block diagram of the CRC-aided list decoding of convolutional codes.

4) Achieving the RCU Bound with Practical Complexity: This paper focuses on designing good CRC-aided convolutional codes for transmitting 64 information bits. Simulation results show that the CRC-ZTCC with 8 memory elements can approach the RCU bound within 0.4 dB with decoding complexity similar to standard Viterbi decoding of the ZTCC. The best CRC-TBCC with 8 memory elements essentially achieves the RCU bound, but requires increased decoding complexity.

B. Organization

This paper is organized as follows: Section II introduces notation, the system architecture, TB trellises, Polyanskiy *et al.*'s finite-blocklength bounds, and the related saddlepoint approximations. Section III introduces the concept of the DSO CRC polynomial, shows that at high SNR the DSO CRC can be obtained by maximizing $d^l_{\min}$, provides an upper bound on $d^l_{\min}$, and gives a DSO CRC design algorithm for TBCCs of arbitrary rate at high SNR. Section IV presents the performance and complexity analyses of SLVD of a given CRC-aided convolutional code. Section V presents simulation results of our designed CRC-aided convolutional codes and a comparison of $(128, 64)$ linear block codes. Section VI concludes the paper.

II. Preliminaries

A. Notation

Let $\mathbb{F}_2 = \{0, 1\}$ denote the binary field. $\mathbb{F}_2^n$ denotes the set of n-dimensional binary sequences. $\mathbb{F}_2[x]$ denotes the set of binary polynomials. The indicator function $\mathbf{1}_E$ takes the value 1 if the event E occurs, and 0 otherwise. The polynomial $u(x) = \sum_{i=0}^{n-1} u_i x^i \in \mathbb{F}_2[x]$ and its row vector form $\boldsymbol{u} = [u_0, u_1, \ldots, u_{n-1}] \in \mathbb{F}_2^n$ are used interchangeably. The CRC polynomial is represented in hexadecimal when its binary coefficients are written from the highest to lowest order. For instance, 0xD represents $x^3 + x^2 + 1$. The convolutional generator polynomial is represented in octal when the binary coefficients of each generator polynomial are written from the lowest to highest order. For instance, $(13, 17)$ represents $(1+x^2+x^3, 1+x+x^2+x^3)$. Let $w_H(\cdot)$, $d_H(\cdot, \cdot)$ and $\|\cdot\|$ denote the Hamming weight, Hamming distance, and Euclidean norm respectively. Finally, $\mathrm{cl}(S)$ and $\partial(S)$ denote the closure and the boundary of a subset $S \subseteq \mathbb{R}^n$, respectively.

B. Architecture

This paper considers CRC-aided list decoding of convolutional codes, as depicted in Fig. 1. Let $u(x) = \sum_{i=0}^{k-1} u_i x^i \in \mathbb{F}_2[x]$ denote the k-bit binary information sequence, where u_{k-1} is the first bit entering the CRC encoder. The information sequence $u(x)$ is first encoded with a degree-m CRC generator polynomial $p(x) = 1+p_1x+\cdots+p_{m-1}x^{m-1}+x^m \in \mathbb{F}_2[x]$ to obtain m parity check bits $r(x) = x^m u(x) \mod p(x)$. Thus, we obtain $v^*(x) = x^m u(x) + r(x)$ which is divisible by the CRC polynomial $p(x)$. The final CRC-coded sequence $v(x)$ is produced by reversing $v^*(x)$, i.e., $v(x) = x^{k+m-1}v^*(x^{-1})$. This guarantees that the first bit entering the encoder, namely, u_{k-1} in $u(x)$, is always the lowest degree term of $v(x)$, consistent with common representation. The concatenated codeword $\boldsymbol{c} \in \mathbb{F}_2^n$ of blocklength n is obtained by convolutionally encoding $\boldsymbol{v}$ with a minimal, feedforward, $(\omega, 1, \nu)$ encoder $\boldsymbol{g}(x) = [g_1(x), g_2(x), \ldots, g_\omega(x)]$, $g_i(x) = \sum_{j=0}^{\nu} g_{i,j}x^j$, with ν memory elements. To terminate a convolutional code into a linear block code, we consider either the ZT or TB method.

This paper focuses on CRC-aided convolutional codes, but our analysis also involves the higher-rate convolutional code for which the CRC codeword $\boldsymbol{v}$ is the input message. To describe the two codes of interest as concisely as possible, define the higher-rate code $\mathcal{C}_h$ and the lower-rate code $\mathcal{C}_l$, where the latter is the CRC-aided convolutional code, as follows:

$$\mathcal{C}_h \triangleq \left\{\boldsymbol{c} \in \mathbb{F}_2^n : \boldsymbol{c} = \boldsymbol{v}\boldsymbol{G}, \forall \boldsymbol{v} \in \mathbb{F}_2^{k+m}\right\}, \tag{2}$$
$$\mathcal{C}_l \triangleq \left\{\boldsymbol{c} \in \mathbb{F}_2^n : \boldsymbol{c} = \boldsymbol{v}\boldsymbol{G}, \forall \boldsymbol{v} \in \mathbb{F}_2^{k+m} \text{ s.t. } p(x)|v^*(x)\right\}, \tag{3}$$

where $\boldsymbol{G} \in \mathbb{F}_2^{(k+m)\times n}$ is the matrix representation of the convolutional encoder. Intuitively, the effect of $p(x)$ is to obtain a subcode $\mathcal{C}_l$ from the given higher-rate code $\mathcal{C}_h$. The exact definition of $\mathcal{C}_h$ and $\mathcal{C}_l$ require the specification of the ZTCC or TBCC. For a ZTCC, $n = \omega(k + m + \nu)$ and

$$\boldsymbol{G} = \begin{bmatrix} G_0 & G_1 & \cdots & G_\nu & & & \\ & G_0 & G_1 & \cdots & G_\nu & & \\ & & \ddots & \ddots & \ddots & \ddots & \\ & & & G_0 & G_1 & \cdots & G_\nu \end{bmatrix},$$

where

$$G_i = \begin{bmatrix} g_{1,i} & g_{2,i} & \cdots & g_{\omega,i} \end{bmatrix}, \quad i = 1, 2, \ldots, \nu.$$

Similarly, for a TBCC, $n = \omega(k + m)$ and

$$\boldsymbol{G} = \begin{bmatrix} G_0 & G_1 & \cdots & \cdots & G_\nu & & & \\ & G_0 & G_1 & \cdots & \cdots & G_\nu & & \\ & & \ddots & \ddots & \ddots & & \ddots & \\ & & & G_0 & G_1 & \cdots & \cdots & G_\nu \\ G_\nu & & & & G_0 & G_1 & \cdots & G_{\nu-1} \\ G_{\nu-1} & G_\nu & & & & \ddots & \ddots & \vdots \\ \vdots & & \ddots & & & & \ddots & G_1 \\ G_1 & G_2 & \cdots & G_\nu & & & & G_0 \end{bmatrix}.$$

Clearly, $\mathcal{C}_l \subseteq \mathcal{C}_h$, $|\mathcal{C}_h| = 2^{k+m}$ and $|\mathcal{C}_l| = 2^k$. The rate of the CRC-aided convolutional code (i.e., the lower-rate code) $R = k/n$. A fundamental quantity associated with a linear block code is its minimum distance. To aid our discussion, we define

$$d^h_{\min} \triangleq \min\{w_H(\boldsymbol{c}) : \boldsymbol{c} \in \mathcal{C}_h \setminus \{\boldsymbol{0}\}\}, \tag{4}$$
$$d^l_{\min} \triangleq \min\{w_H(\boldsymbol{c}) : \boldsymbol{c} \in \mathcal{C}_l \setminus \{\boldsymbol{0}\}\}. \tag{5}$$

As a corollary, $0 < d^h_{\min} \leq d^l_{\min}$. Note that for a ZTCC, $d^h_{\min}$ is in fact an order-$(k + m - 1)$ row distance and is thus no less than the free distance of the convolutional code [43].

The binary phase shift keying (BPSK) modulated sequence $\boldsymbol{x} = [x_0, x_1, \ldots, x_{n-1}]$ for codeword $\boldsymbol{c}$ is obtained via $x_i = (1 - 2c_i)A$, where A is the BPSK amplitude, and is then transmitted over the AWGN channel with channel SNR γ_s. Therefore, the channel model is

$$y_i = x_i + z_i, \quad i = 0, 1, \ldots, n - 1, \tag{6}$$

where z_i's are independent and identically distributed (i.i.d.) according to the standard normal distribution. Thus, $\gamma_s = A^2$ or $A = \sqrt{\gamma_s}$.

Upon receiving the channel observations $\boldsymbol{y}$, the (soft) SLVD with a constrained maximum list size Ψ using CRC polynomial $p(x)$ is employed to determine the most likely information sequences $\hat{u}(x)$ from the trellis of the higher-rate code $\mathcal{C}_h$ based on $\boldsymbol{y}$ in a sequential manner using a maximum of Ψ trials. We assume that the SLVD sequentially produces rank ordered codewords[1] that are also higher-rate codewords in $\mathcal{C}_h$. This is true when $\mathcal{C}_h$ is a ZTCC and may not be true when it is a TBCC in practice. If an input sequence $\hat{v}^*(x)$ associated with a higher-rate codeword passes the CRC verification, decoding terminates and the list stops growing. The corresponding list rank is marked as the terminating list rank L and the most likely information sequence $\hat{u}(x)$ is recovered from the last k bits of $\hat{v}^*(x)$. If an input sequence divisible by $p(x)$ is not found after Ψ attempts, the decoder terminates at list rank Ψ with a NACK as the output. As mentioned earlier, there exists a supremum list rank λ (whose formal definition will be given in (44)) which is independent of Ψ. If $\Psi \geq \lambda$, no NACK will occur. Consequently, L is always bounded between 1 and $\min\{\lambda, \Psi\}$.

A UE occurs if the SLVD erroneously identifies an input sequence $\hat{v}^*(x)$ that is divisible by $p(x)$ and $\hat{v}^*(x) \neq v^*(x)$. This is equivalent to the case where the UE polynomial $\hat{v}^*(x) - v^*(x) \in \mathbb{F}_2[x]$ is nonzero and is divisible by $p(x)$. Hence, an *error event* is given by the input-output pair $(\hat{v}(x) - v(x), \hat{c}(x) - c(x))$, where $\hat{v}(x) \neq v(x)$ and $\hat{c}(x)$ is a higher-rate codeword associated with $\hat{v}(x)$. By linearity, each error event corresponds to a pair of a nonzero input sequence $v(x)$ and its corresponding codeword $c(x)$. When restricted to convolutional codes, we can also use a trellis path to represent an error event.

The performance of the CRC-aided convolutional code is measured by three probabilities: probability of correct decoding $P_{c,\Psi}$, probability of UE $P_{e,\Psi}$, and probability of NACK $P_{\mathit{NACK},\Psi}$, where $P_{c,\Psi} + P_{e,\Psi} + P_{\mathit{NACK},\Psi} = 1$. In the special case where $\Psi \geq \lambda$, $P_{c,\Psi} + P_{e,\Psi} = 1$. For ease of reference, we use $P_{e,\lambda}$ to represent $P_{e,\Psi}$ for which $\Psi \geq \lambda$.

[1] The input sequence that generates this higher-rate codeword is also known simultaneously.

C. Tail-Biting Trellises

We follow [44] in describing a TB trellis. Let V be a set of vertices (or states). The set $\mathcal{A}$ is the output alphabet, and E is the set of edges described as ordered triples (v, a, v') with $v, v' \in V$, and $a \in \mathcal{A}$. In words, $(v, a, v') \in E$ denotes an edge that starts at v, ends at v' and has output a.

Definition 1 (Tail-biting trellises)**.** *A tail-biting trellis $T = (V, E, \mathcal{A})$ of depth N is an edge-labeled directed graph with the following property: the vertex set V can be partitioned as*

$$V = V_0 \cup V_1 \cup \cdots \cup V_{N-1} \tag{7}$$

such that every edge in T either begins at a vertex of V_i and ends at a vertex of V_{i+1} for some $i = 0, 1, \ldots, N-2$, or begins at a vertex of V_{N-1} and ends at a vertex of V_0.

Geometrically, a TB trellis can be viewed as a cylinder of N sections defined on some circular time axis. Alternatively, we can also define a TB trellis on a sequential time axis $\mathcal{I} = \{0, 1, \ldots, N\}$ with the restriction that $V_0 = V_N$ so that we obtain a conventional trellis.

For a trellis T of depth N, a trellis section connecting time i and $i+1$ is a subset $T_i \subseteq V_i \times \mathcal{A} \times V_{i+1} \subseteq E$ that specifies the allowed combination (s_i, a_i, s_{i+1}) of state $s_i \in V_i$, output symbol $a_i \in \mathcal{A}$, and state $s_{i+1} \in V_{i+1}$, $i = 0, 1, \ldots, N-1$. Such allowed combinations are called trellis branches. A trellis path $(\boldsymbol{s}, \boldsymbol{a}) \in T$ is a state/output sequence pair, where $\boldsymbol{s} \in V_0 \times V_1 \times \cdots \times V_N$, $\boldsymbol{a} \in \mathcal{A}^N$. Since $\boldsymbol{s}$ equivalently specifies the input sequence, an error event can also be described by its corresponding trellis path $(\boldsymbol{s}, \boldsymbol{a})$.

For a TB trellis T of depth N, a TB path $(\boldsymbol{s}, \boldsymbol{a})$ of length N on T is a *closed* path through N vertices. If T is defined on a sequential time axis $\mathcal{I} = \{0, 1, \ldots, N\}$, then any TB path $(\boldsymbol{s}, \boldsymbol{a})$ of length N satisfies $s_0 = s_N$.

D. Finite-Blocklength Bounds and Approximations

In [5], Polyanskiy *et al.* derived the RCU bound and the MC bound that upper and lower bound the probability of error of the best (n, M) code. These two bounds serve as benchmarks to assess the performance of a given finite-blocklength code.

We follow the notation in [6] to introduce the RCU bound and the MC bound. Let $W^n(\cdot|\cdot)$ denote a length-n channel. Let $\alpha_\beta(P, Q)$ denote the smallest type-I error probability among all tests discriminating between distributions P and Q, with a type-II error probability at most β [45, Chapter 11.7]. For a random-coding ensemble defined over distribution P^n, the RCU bound is given by

$$\mathrm{rcu}(n, M) \triangleq \mathsf{E}[\min\{1, (M-1)\,\mathrm{pep}(X^n, Y^n)\}], \tag{8}$$

where $(X^n, Y^n) \sim P^n \times W^n$ and the pairwise error probability $\mathrm{pep}(x^n, y^n)$ is defined as

$$\mathrm{pep}(x^n, y^n) \triangleq \mathsf{P}\big(W^n(y^n|\bar{X}^n) \geq W^n(y^n|x^n)\big),$$

with $\bar{X}^n \sim P^n$. The MC bound is a minimax of a particular smallest type-I error probability

$$\mathrm{mc}(n, M) \triangleq \min_{P^n} \max_{Q^n} \left\{ \alpha_{\frac{1}{M}}(P^n \times W^n, P^n \times Q^n) \right\}, \tag{9}$$

where the minimization is over all input distributions P^n, and the maximization is over a set of auxiliary, independent of the input, output distributions Q^n.

An exact evaluation of the RCU bound and the MC bound involves integrating tail probabilities of n-dimensional random variables, which is computationally difficult even for simple channels and moderate values of n. In [6], the authors provided saddlepoint approximations of these two bounds for memoryless symmetric channels, including the binary-input AWGN channel. These approximations are shown to be tight for a wide range of rates and blocklengths. Section V uses saddlepoint approximations to evaluate the RCU bound and the MC bound for the binary-input AWGN channel.

Approximation 1 (MC bound, [6])**.** *For memoryless symmetric channels for which $Y \sim W(\cdot|x)$ is independent of x,*

$$\mathrm{mc}(n, M) \approx \max_{\rho \geq 0} \Big\{ e^{-n(E_0(\rho) - \rho E_0'(\rho))} \cdot \Big(\psi\big(\sqrt{nU(\rho)}\big) + \psi\big(\rho\sqrt{nU(\rho)}\big) - e^{-n(R - E_0'(\rho))} \Big) \Big\}, \tag{10}$$

where

$$E_0(\rho, P) = -\log \int_{\mathcal{Y}} \Big(\sum_{x \in \mathcal{X}} P(x) W(y|x)^{\frac{1}{1+\rho}} \Big)^{1+\rho} \mathrm{d}y, \tag{11}$$

$$E_0(\rho) = \max_P E_0(\rho, P), \tag{12}$$

$$\psi(x) = \frac{1}{2} \operatorname{erfc}\left(\frac{|x|}{\sqrt{2}} \right) e^{\frac{x^2}{2}} \operatorname{sign}(x), \tag{13}$$

$$U(\rho) = -(1+\rho) E_0''(\rho), \tag{14}$$

where $\mathcal{X}$ and $\mathcal{Y}$ denote the input and output alphabets of the channel, and the maximization in (12) *is over all possible probability distributions on $\mathcal{X}$.*

Approximation 2 (RCU bound, [6])**.** *For memoryless symmetric channels for which $Y \sim W(\cdot|x)$ is independent of x,*

$$\mathrm{rcu}(n, M) \approx \tilde{\xi}_n(\hat{\rho}) + \varphi_n(\hat{\rho}) e^{-n(E_0(\hat{\rho}, P) - \hat{\rho} R)}, \tag{15}$$

where $\hat{\rho}$ is the value for which $E_0'(\rho, P) = R$, and

$$Q_\rho(y) = \frac{1}{e^{-E_0(\rho, P)}} \Big(\sum_{x \in \mathcal{X}} P(x) W(y|x)^{\frac{1}{1+\rho}} \Big)^{1+\rho}, \tag{16}$$

$$\bar{\omega}''(\hat{\rho}) = \int_{\mathcal{Y}} Q_{\hat{\rho}}(y) \left[\frac{\partial^2}{\partial \tau^2} \Big(\log \sum_{x \in \mathcal{X}} P(x) W(y|x)^\tau \Big) \Big|_{\tau = \hat{\tau}} \right] \mathrm{d}y, \tag{17}$$

$$\theta_n(\hat{\rho}) = \frac{1}{\sqrt{1+\hat{\rho}}} \left(\frac{1+\hat{\rho}}{\sqrt{2\pi n \bar{\omega}''(\hat{\rho})}} \right)^{\hat{\rho}}, \tag{18}$$

$$\tilde{\xi}_n(\hat{\rho}) = \begin{cases} 1, & \hat{\rho} < 0 \\ 0, & 0 \leq \hat{\rho} \leq 1 \\ e^{-n(E_0(1,P) - R)} \theta_n(1), & \hat{\rho} > 1, \end{cases} \tag{19}$$

$$V(\hat{\rho}) = -E_0''(\hat{\rho}, P), \tag{20}$$

$$\varphi_n(\hat{\rho}) = \theta_n(\hat{\rho}) \Big(\psi\big(\hat{\rho}\sqrt{nV(\hat{\rho})}\big) + \psi\big((1-\hat{\rho})\sqrt{nV(\hat{\rho})}\big) \Big). \tag{21}$$

III. The search for the DSO CRC polynomial

In this section, we seek to design good CRC-aided convolutional codes that provide the lowest possible probability of UE $P_{e,\lambda}$. To this end, for a given convolutional code, we design CRC polynomials that minimize the union bound on the probability of undetected error $P_{e,\lambda}$. The resulting CRC polynomial is known as the DSO CRC polynomial.

A. General Theory

For a given convolutional code and a desired CRC degree m, we wish to identify the degree-m CRC polynomial

$$p(x) = 1 + p_1 x + \cdots + p_{m-1}x^{m-1} + x^m \in \mathbb{F}_2[x] \qquad (22)$$

that minimizes the probability of UE $P_{e,\lambda}$. Since the exact probability $P_{e,\lambda}$ has no closed-form expression that can facilitate a design procedure, we use the union bound as an objective function that only involves the *distance spectrum*, $C_{d^l_{\min}}, \ldots, C_n$, of the lower-rate code $\mathcal{C}_l$, where C_d denotes the number of codewords in $\mathcal{C}_l$ of Hamming weight d, $d^l_{\min} \leq d \leq n$. The distance spectrum of the lower-rate code $\mathcal{C}_l$ is a function of both the CRC polynomial $p(x)$ and the higher-rate code $\mathcal{C}_h$. For any candidate polynomial $p(x)$, the union bound on $P_{e,\lambda}$ is given by

$$\begin{aligned} P_{e,\lambda} &\leq \sum_{\boldsymbol{c}\in\mathcal{C}_l\backslash\{\bar{\boldsymbol{c}}\}} \mathsf{P}\Big(Z > \frac{1}{2}\|\boldsymbol{x}(\boldsymbol{c}) - \boldsymbol{x}(\bar{\boldsymbol{c}})\| \Big| \boldsymbol{X} = \boldsymbol{x}(\bar{\boldsymbol{c}})\Big) \\ &= \sum_{d=d^l_{\min}}^{n} C_d Q\big(A\sqrt{d}\big), \end{aligned} \qquad (23)$$

where $\bar{\boldsymbol{c}} \in \mathcal{C}_l$ is the transmitted codeword, $\boldsymbol{x}(\boldsymbol{c}) \in \{-A, A\}^n$ is the BPSK-modulated point for codeword $\boldsymbol{c}$, $Z \sim \mathcal{N}(0,1)$, and

$$Q(x) \triangleq \int_x^\infty \frac{1}{\sqrt{2\pi}} e^{-u^2/2}\,\mathrm{d}u \qquad (24)$$

is the complementary Gaussian cumulative distribution function. $Q\big(A\sqrt{d}\big)$ computes the pairwise error probability of two codewords at distance d. For a given higher-rate code $\mathcal{C}_h$, a given SNR γ_s (i.e., $A = \sqrt{\gamma_s}$), and a CRC degree m, we define the degree-m *DSO CRC polynomial* as the one that minimizes the union bound on $P_{e,\lambda}$. Namely, the degree-m DSO CRC polynomial is the solution to the following optimization problem:

$$\min_{p(x)} \quad \sum_{d=d^l_{\min}}^{n} C_d Q\big(A\sqrt{d}\big). \qquad (25)$$

Theoretically, the distance spectrum $C_{d^l_{\min}}, \ldots, C_n$ of $\mathcal{C}_l$ can be found through Viterbi search of the trellis of the higher-rate code $\mathcal{C}_h$, retaining only codewords whose input sequences are divisible by the candidate CRC polynomial $p(x)$. However, this approach requires the calculation of distance spectra for 2^{m-1} candidate CRC polynomials and quickly becomes computationally expensive as the information length k gets large. The degree-m DSO CRC polynomial depends on the specific higher-rate code and the SNR at which $P_{e,\lambda}$ is being minimized. Note that the DSO CRC polynomial can be different for different values of k. In [36], Lou *et al.* investigated how DSO CRC polynomials vary with information length k. Their essential finding is that a DSO CRC polynomial for a large k is usually "good" for shorter k. If the SNR is not sufficiently high, the CRC polynomial that minimizes the union bound in (23) may not minimize the actual $P_{e,\lambda}$.

Nevertheless, when SNR is sufficiently high or equivalently if the target probability of UE $P_{e,\lambda}$ is sufficiently low (typically less than 10^{-6}), the union bound (23) will be dominated by its first term $C_{d^l_{\min}} Q\Big(A\sqrt{d^l_{\min}}\Big)$ which becomes asymptotically tight to $P_{e,\lambda}$. Furthermore, in most cases at high SNR where the operating A is large enough, the first term in (23) is only dominated by $d^l_{\min}$. The following theorem justifies this statement.

Theorem 1. *For a given higher-rate code $\mathcal{C}_h$, let $C_{d^l_{\min,1}}, \ldots, C_n$ and $C'_{d^l_{\min,2}}, \ldots, C'_n$ be two distance spectra associated with lower-rate codes generated by CRC polynomials $p_1(x)$ and $p_2(x)$, respectively. If $d^l_{\min,1} < d^l_{\min,2}$, there exists a positive threshold A^* such that if $A > A^*$,*

$$\sum_{d=d^l_{\min,1}}^{n} C_d Q\big(A\sqrt{d}\big) > \sum_{d=d^l_{\min,2}}^{n} C'_d Q\big(A\sqrt{d}\big). \qquad (26)$$

In the special case where $d^l_{\min,1} = d^l_{\min,2}$ and $C_{d^l_{\min,1}} > C'_{d^l_{\min,2}}$, the above conclusion still holds.

Proof: Assume that $d^l_{\min,1} < d^l_{\min,2}$. Since coefficients $C_{d^l_{\min,1}}, C'_{d^l_{\min,2}}$ are positive and bounded,

$$\lim_{A\to\infty} \frac{\sum_{d=d^l_{\min,1}}^{n} C_d Q(A\sqrt{d})}{\sum_{d=d^l_{\min,2}}^{n} C'_d Q(A\sqrt{d})} \qquad (27)$$

$$= \lim_{A\to\infty} \frac{C_{d^l_{\min,1}} \exp\Big(-\frac{A^2 d^l_{\min,1}}{2}\Big)}{C'_{d^l_{\min,2}} \exp\Big(-\frac{A^2 d^l_{\min,2}}{2}\Big)} \cdot \frac{\Big[1 + \sum_{d=d^l_{\min,1}+1}^{n} \frac{C_d}{C_{d^l_{\min,1}}} \exp\Big(-\frac{A^2(d-d^l_{\min,1})}{2}\Big)\Big]}{\Big[1 + \sum_{d=d^l_{\min,2}+1}^{n} \frac{C'_d}{C'_{d^l_{\min,2}}} \exp\Big(-\frac{A^2(d-d^l_{\min,2})}{2}\Big)\Big]} \qquad (28)$$

$$= \lim_{A\to\infty} \frac{C_{d^l_{\min,1}}}{C'_{d^l_{\min,2}}} \exp\Big(\frac{A^2}{2}(d^l_{\min,2} - d^l_{\min,1})\Big) \qquad (29)$$

$$=\infty.$$

Hence, there exists a threshold A^* such that when $A > A^*$, $\sum_{d=d^l_{\min,1}}^{n} C_d Q(A\sqrt{d}) > \sum_{d=d^l_{\min,2}}^{n} C'_d Q(A\sqrt{d})$. In the special case where $d^l_{\min,1} = d^l_{\min,2}$ and $C_{d^l_{\min,1}} > C'_{d^l_{\min,2}}$, the limit in (29) is still greater than 1. Thus, the same conclusion follows. ∎

For sufficiently low target $P_{e,\lambda}$, the operating amplitude A is typically large enough such that $A > A^*$ is easily met in practice. In these common situations, the DSO CRC design principle reduces to maximizing the minimum distance $d^l_{\min}$ of the lower-rate code.

As an illustrative example, Fig. 2 shows the union bounds (23) for three degree-5 CRC polynomials among the 16

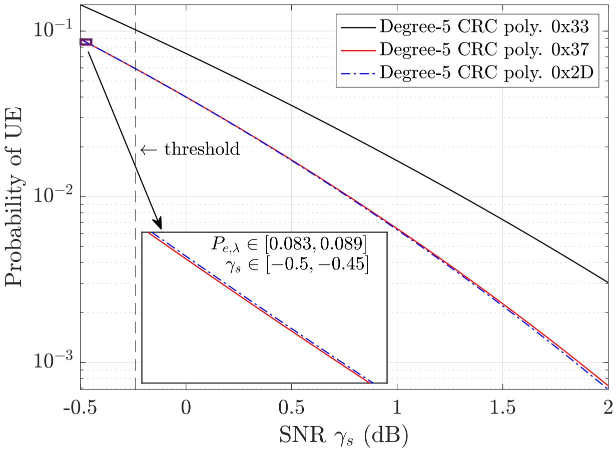

Fig. 2. Comparison of the DSO CRC polynomials for $k = 10$, $m = 5$ and ZTCC $(13, 17)$. The blocklength of the CRC-ZTCC $n = 36$. The threshold value is -0.2398 dB.

candidates for $k = 10$ and ZTCC $(13, 17)$. The CRC 0x37 minimizes the union bound at low SNR, whereas the CRC 0x2D minimizes the union bound at high SNR. On the contrary, the CRC 0x33 yields the worst possible union bound among all candidates. A detailed computation reveals that $d^l_{\text{min}} = 11$, $C_{d^l_{\text{min}}} = 17$ for 0x37, $d^l_{\text{min}} = 12$, $C_{d^l_{\text{min}}} = 76$ for 0x2D. Thus, the DSO CRC may not necessarily have the best minimum distance. The worst CRC polynomial 0x33 has $d^l_{\text{min}} = 8$, $C_{d^l_{\text{min}}} = 10$. In this example, the threshold at which the DSO CRC polynomial switches from 0x37 to 0x2D is -0.2398 dB. However, the gap between the performance of the two CRC polynomials is minimal, especially at low SNR. Nevertheless, both 0x37 and 0x2D achieve a gain of 0.5 dB compared to 0x33 at 10^{-2}, showing that the optimal CRC polynomial is crucial to achieving good performance.

For a given convolutional code and a specified CRC degree m, one may ask: how large can d^l_{min} be? The next theorem gives a tight upper bound on d^l_{min} in terms of the distance spectrum of the higher-rate code $\mathcal{C}_h$.

Theorem 2. *Given a specified CRC degree m and a higher-rate code $\mathcal{C}_h$ with distance spectrum $B_{d^h_{\text{min}}}, \ldots, B_n$, define w^* as the minimum w for which $\sum_{d=d^h_{\text{min}}}^{w} B_d \geq 2^m$. For any degree-$m$ CRC polynomial, we have $d^l_{\text{min}} \leq 2w^*$.*

Proof: Define the set $V(\boldsymbol{c})$ to be the set of codewords from the higher-rate code $\mathcal{C}_h$ that unambiguously decode to codeword $\boldsymbol{c}$ of the lower-rate code $\mathcal{C}_l$. Specifically, for each $\boldsymbol{c} \in \mathcal{C}_l$, define

$$V(\boldsymbol{c}) \triangleq \{\boldsymbol{r} \in \mathcal{C}_h : d_H(\boldsymbol{r}, \boldsymbol{c}) < d_H(\boldsymbol{r}, \boldsymbol{c}'),\ \forall \boldsymbol{c}' \in \mathcal{C}_l\}. \tag{30}$$

Hence, by linearity of the higher-rate code, the cardinality of $V(\boldsymbol{c})$ for every $\boldsymbol{c} \in \mathcal{C}_l$ is exactly the same. Hence,

$$|V(\boldsymbol{c})| \leq \frac{|\mathcal{C}_h|}{|\mathcal{C}_l|} = 2^m, \tag{31}$$

where (31) is an inequality because some codewords $\boldsymbol{r} \in \mathcal{C}_h$ may be equidistant from two or more lower-rate codewords.

Next, we show that for a given $\boldsymbol{c} \in \mathcal{C}_l$, $d_H(\boldsymbol{r}, \boldsymbol{c}) < \frac{1}{2} d^l_{\text{min}}$ implies that $\boldsymbol{r} \in V(\boldsymbol{c})$. By definition of the minimum distance, for two arbitrary distinct codewords $\boldsymbol{c}, \boldsymbol{c}' \in \mathcal{C}_l$, $d_H(\boldsymbol{c}, \boldsymbol{c}') \geq d^l_{\text{min}}$. Hence, for any $\boldsymbol{r} \in \mathcal{C}_h$, by triangle inequality,

$$d_H(\boldsymbol{r}, \boldsymbol{c}) + d_H(\boldsymbol{r}, \boldsymbol{c}') \geq d_H(\boldsymbol{c}, \boldsymbol{c}') \geq d^l_{\text{min}}. \tag{32}$$

Thus, if $d_H(\boldsymbol{r}, \boldsymbol{c}) < \frac{1}{2} d^l_{\text{min}}$, this implies that $d_H(\boldsymbol{r}, \boldsymbol{c}') > \frac{1}{2} d^l_{\text{min}}$ for any other $\boldsymbol{c}' \in \mathcal{C}_l$, i.e., $d_H(\boldsymbol{r}, \boldsymbol{c}) < d_H(\boldsymbol{r}, \boldsymbol{c}')$ for all $\boldsymbol{c}' \in \mathcal{C}_l$. By definition of $V(\boldsymbol{c})$, we conclude that $\boldsymbol{r} \in V(\boldsymbol{c})$.

TABLE I
Comparison Between d^l_{min} Associated With the DSO CRC Polynomial and $2w^*$ Computed From Theorem 2 for $k = 64$

m	ZTCC $(13, 17)$			TBCC $(13, 17)$		
	$p(x)$	d^l_{min}	$2w^*$	$p(x)$	d^l_{min}	$2w^*$
0	0x1	6	12	0x1	6	12
3	0x9	10	12	0xF	8	12
4	0x1B	10	12	0x1F	9	12
5	0x2D	12	12	0x2D	10	12
6	0x43	12	12	0x63	12	12
7	0xB5	13	14	0xED	12	14
8	0x107	14	14	0x107	12	14
9	0x313	14	16	0x349	14	16
10	0x50B	15	18	0x49D	14	18

By law of contraposition, if $\boldsymbol{r} \notin V(\boldsymbol{c})$, then $d_H(\boldsymbol{r}, \boldsymbol{c}) \geq \frac{1}{2} d^l_{\text{min}}$. Indeed, when $\sum_{d=d^h_{\text{min}}}^{w} B_d \geq 2^m$ (i.e., $\sum_{d=0}^{w} B_d \geq 2^m + 1$), by pigeonhole principle, there exists a codeword $\boldsymbol{r} \in \mathcal{C}_h$ that is outside of $V(\boldsymbol{c})$ and whose distance from $\boldsymbol{c}$ satisfies $d_H(\boldsymbol{r}, \boldsymbol{c}) \leq w$. Therefore, for this codeword $\boldsymbol{r}$, $w \geq d_H(\boldsymbol{r}, \boldsymbol{c}) \geq \frac{1}{2} d^l_{\text{min}}$ or equivalently, $d^l_{\text{min}} \leq 2 d_H(\boldsymbol{r}, \boldsymbol{c}) \leq 2w$. Since this holds for any w satisfying $\sum_{d=d^h_{\text{min}}}^{w} B_d \geq 2^m$, the minimum such value w^* yields the tightest upper bound. ■

Table I shows the comparison between d^l_{min} and the upper bound $2w^*$ in Theorem 2 for both ZTCC and TBCC generated with the rate-$1/2$ convolutional encoder $(13, 17)$ at $k = 64$. We see that the upper bound is sharp as there exist DSO CRC polynomials that achieve this bound.

B. A Two-Phase DSO CRC Design Algorithm for TBCCs

We focus on finding the DSO CRC polynomial for low target $P_{e,\lambda}$. As discussed earlier, the design principle under this circumstance conveniently reduces to maximizing the d^l_{min} of the lower-rate code. Thus, the optimal CRC polynomial depends on the convolutional code but not the SNR.

In principle, the DSO CRC design algorithm for low target $P_{e,\lambda}$ comprises a *collection phase* that gathers error events of the higher-rate code $\mathcal{C}_h$ up to a certain distance $\tilde{d}$, and a *search phase* that identifies the degree-m DSO CRC polynomial using the error events gathered in the collection phase. In this section, we propose a two-phase DSO CRC design algorithm particularized to TBCCs of arbitrary rate (including rate $1/\omega$). Later, we point out that our algorithm is also applicable to ZTCCs of arbitrary rate with a few distinctions.

The difficulty of designing DSO CRC polynomials for a TB trellis lies in the fact that a TB trellis is a union of 2^ν subtrellises that share trellis branches in the middle. Thus, to collect error events that meet the TB condition, a straightforward collection method is to perform Viterbi search separately at each possible start state to identify the *irreducible error event* (IEE) that leaves the start state once and rejoins it once, and then use them to reconstruct length-N TB paths with distance less than $\tilde{d}$. These IEEs constitute the error events of interest. However, this scheme will be *inefficient* in that for each nonzero start state, there exists a catastrophic IEE that spends a majority of time in the self-loop of the zero state. Such an IEE has the catastrophic property that its length grows unbounded with a finite weight. As a consequence, they are rarely used during reconstruction yet occupy a significant portion of total IEEs.

The algorithm we are about to propose follows the straightforward algorithm with the distinction in collecting IEEs. To circumvent the aforementioned catastrophic IEEs, we wish to identify IEEs whose weight is proportional to its length. To this end, we first partition the TB trellis into several sets that are closed under cyclic shifts. Next, all elements in each set are reconstructed via the concatenation of the corresponding IEEs and circular shifts of the resulting path.

For a given length-N TB trellis associated with a minimal convolutional encoder $\boldsymbol{g}(x)$, let $V_0 = \{0, 1, \ldots, 2^{\nu} - 1\}$ be the set of possible encoder states. We seek a partition of the TB trellis, i.e., mutually exclusive sets that, together, contain all length-N TB paths. To do this, we define $\mathrm{TBP}(0)$ as the set that contains all TB paths that traverse state 0; $\mathrm{TBP}(1)$ contains the TB paths that traverse state 1 but not state 0; and so on. In general, the set $\mathrm{TBP}(\sigma)$ for $\sigma \in V_0$ is defined as follows:

$$\begin{aligned}\mathrm{TBP}(\sigma) \triangleq \big\{&(\boldsymbol{s}, \boldsymbol{a}) \in V_0^{N+1} \times \mathcal{A}^N : s_0 = s_N;\\ &\exists i \in \mathcal{I} \text{ s.t. } s_i = \sigma;\ \forall i \in \mathcal{I},\ s_i \notin \{0, 1, \ldots, \sigma - 1\}\big\}. \quad (33)\end{aligned}$$

An important property of the above decomposition is that each set $\mathrm{TBP}(\sigma)$ is closed under cyclic shifts, as circularly shifting a TB path preserves the sequence of states that it traverses. Furthermore, such a partition of the TB trellis motivates the following IEE.

Definition 2 (Irreducible error events). *For a TB trellis T on sequential time axis $\mathcal{I} = \{0, 1, \ldots, N\}$, the set of irreducible error events $(\boldsymbol{s}, \boldsymbol{a})$ at state $\sigma \in V_0$ is defined as*

$$\mathrm{IEE}(\sigma) \triangleq \bigcup_{i=1,2,\ldots,N} \overline{\mathrm{IEE}}(\sigma, i), \quad (34)$$

where

$$\begin{aligned}\overline{\mathrm{IEE}}(\sigma, i) \triangleq \{&(\boldsymbol{s}, \boldsymbol{a}) \in V_0^{i+1} \times \mathcal{A}^i : s_0 = s_i = \sigma;\\ &\forall j, 0 < j < i,\ s_j \notin \{0, 1, \ldots, \sigma\}\}. \quad (35)\end{aligned}$$

For ZTCCs, Lou *et al.* [36] considered finding IEEs that start and end at the zero state and counting the allowed combinations. Hence, The IEE defined above generalizes Lou *et al.*'s IEEs. Since for a nonzero start state, no IEE can traverse the zero state, this guarantees that the weight of the IEE grows proportionally with its length, thus avoiding the catastrophic IEEs incurred in the straightforward algorithm.

With the sets $\mathrm{TBP}(\sigma)$ defined as above, the following theorem describes how to efficiently find all elements in each $\mathrm{TBP}(\sigma)$ via the corresponding IEEs.

Theorem 3. *Every TB path $(\boldsymbol{s}, \boldsymbol{a}) \in \mathrm{TBP}(\sigma)$ can be constructed from the IEEs in $\mathrm{IEE}(\sigma)$ via concatenation and subsequent cyclic shifts.*

Algorithm 1 The Collection Procedure

```
Input: The TB trellis $T$, threshold $\tilde{d}$
Output: The list of IEEs $\mathcal{L}_{\mathrm{IEE}}(\tilde{d}) = \{(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{v})\}$
Initialize lists $\mathcal{L}_\sigma$ to be empty for all $\sigma \in V_0$;
for $\sigma \leftarrow 0, 1, \ldots, |V_0| - 1$ do
    Perform Viterbi search at $\sigma$ on $T$ to collect list $\mathcal{L}_\sigma(\tilde{d})$ of all IEEs of distances less than $\tilde{d}$;
end for
return $\mathcal{L}_{\mathrm{IEE}}(\tilde{d}) \leftarrow \bigcup_{\sigma \in V_0} \mathcal{L}_\sigma(\tilde{d})$;
```

Algorithm 2 The Search Procedure

```
Input: The trellis length $N$, degree $m$, list of IEEs $\mathcal{L}_{\mathrm{IEE}}(\tilde{d})$
Output: The degree-$m$ DSO CRC polynomial $p(x)$
Initialize the list $\mathcal{L}_{\mathrm{CRC}}$ of $2^{m-1}$ CRC candidates and empty lists $\mathcal{L}_{\mathrm{TBP}}(d)$ of TBPs, $d = 1, \ldots, \tilde{d} - 1$;
for $d \leftarrow 1, 2 \ldots, \tilde{d} - 1$ do
    Construct all TBPs $(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{v})$ from $\mathcal{L}_{\mathrm{IEE}}(\tilde{d})$ s.t. $w_H(\boldsymbol{a}) = d$, $|\boldsymbol{v}| = N$, via concatenation and cyclic shifts;
    For each TBP, $\mathcal{L}_{\mathrm{TBP}}(d) \leftarrow \mathcal{L}_{\mathrm{TBP}}(d) \cup \{(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{v})\}$;
end for
$\mathrm{Candi}(1) \leftarrow \mathcal{L}_{\mathrm{CRC}}$;
for $d \leftarrow 1, \ldots, \tilde{d} - 1$ do
    for $p_i(x) \in \mathrm{Candi}(d)$ do
        Pass all $\boldsymbol{v}(x) \in \mathcal{L}_{\mathrm{TBP}}(d)$ to $p_i(x)$;
        $C^{(i)} \leftarrow$ the number of divisible $\boldsymbol{v}(x)$ of dist. $d$;
    end for
    $C^* \leftarrow \min_{i \in \mathrm{Candi}(d)} C^{(i)}$
    $\mathrm{Candi}(d+1) \leftarrow \{p_i(x) \in \mathrm{Candi}(d) : C^{(i)} = C^*\}$;
    if $|\mathrm{Candi}(d+1)| = 1$ then
        return $\mathrm{Candi}(d+1)$;
    end if
end for
```

Proof: Let us consider T as a TB trellis defined on a sequential time axis $\mathcal{I} = \{0, 1, \ldots, N\}$. For any TB path $(\boldsymbol{s}, \boldsymbol{a}) \in \mathrm{TBP}(\sigma)$ of length N on T, we can first circularly shift it to some other TB path $(\boldsymbol{s}^{(0)}, \boldsymbol{a}^{(0)}) \in \mathrm{TBP}(\sigma)$ on T such that $s_0^{(0)} = s_N^{(0)} = \sigma$.

Now, we examine $\boldsymbol{s}^{(0)}$ over $\mathcal{I}$. If $\boldsymbol{s}^{(0)}$ is already an element of $\mathrm{IEE}(\sigma)$, then there is nothing to prove. Otherwise, there exists a time index j, $0 < j < N$, such that $s_j = \sigma$. In this case, we break the TB path $(\boldsymbol{s}^{(0)}, \boldsymbol{a}^{(0)})$ at time j into two sub-paths $(\boldsymbol{s}^{(1)}, \boldsymbol{a}^{(1)})$ and $(\boldsymbol{s}^{(2)}, \boldsymbol{a}^{(2)})$, where

$$\begin{aligned}\boldsymbol{s}^{(1)} &= (s_0, s_1, \ldots, s_j),\ \boldsymbol{a}^{(1)} = (a_0, a_1, \ldots, a_{j-1}),\\ \boldsymbol{s}^{(2)} &= (s_j, s_{j+1}, \ldots, s_N),\ \boldsymbol{a}^{(2)} = (a_j, a_{j+1}, \ldots, a_{N-1}).\end{aligned}$$

Note that after segmentation of $(\boldsymbol{s}^{(0)}, \boldsymbol{a}^{(0)})$, the resultant two sub-paths, $(\boldsymbol{s}^{(1)}, \boldsymbol{a}^{(1)})$ and $(\boldsymbol{s}^{(2)}, \boldsymbol{a}^{(2)})$, still meet the TB condition. Repeat the above procedure on $(\boldsymbol{s}^{(1)}, \boldsymbol{a}^{(1)})$ and $(\boldsymbol{s}^{(2)}, \boldsymbol{a}^{(2)})$. Since the length of a new sub-path is strictly decreasing after each segmentation, the boundary case is the atomic sub-path $(\boldsymbol{s}, \boldsymbol{a})$ of some length j^* satisfying $s_0 = s_{j*} = \sigma$, $s_{j'} \neq \sigma$, $\forall j' \in (0, j^*)$. Clearly, this atomic path is an element of $\mathrm{IEE}(\sigma)$. Thus, we successfully decompose a length-N TB path into elements of $\mathrm{IEE}(\sigma)$. Hence, reversing

TABLE II
OPTIMUM RATE-$1/2$ ZTCCS AND THEIR DSO CRC POLYNOMIALS FOR $k = 64$ AT SUFFICIENTLY LOW PROBABILITY OF UE $P_{e,\lambda}$

ν	ZTCC $\boldsymbol{g}(x)$	DSO CRC Polynomials							
		$m = 3$	4	5	6	7	8	9	10
3	$(13, 17)$	9	1B	2D	43	B5	107	313	50B
4	$(27, 31)$	F	15	33	4F	D3	13F	2AD	709
5	$(53, 75)$	9	11	25	49	EF	131	23F	73D
6	$(133, 171)$	F	1B	23	41	8F	113	2EF	629
7	$(247, 371)$	9	13	3F	5B	E9	17F	2A5	61D
8	$(561, 753)$	F	11	33	49	8B	19D	27B	4CF
9	$(1131, 1537)$	D	15	21	51	B7	1D5	20F	50D
10	$(2473, 3217)$	F	13	3D	5B	BB	105	20D	6BB

TABLE III
OPTIMUM RATE-$1/2$ TBCCS AND THEIR DSO CRC POLYNOMIALS FOR $k = 64$ AT SUFFICIENTLY LOW PROBABILITY OF UE $P_{e,\lambda}$

ν	TBCC $\boldsymbol{g}(x)$	DSO CRC Polynomials							
		$m = 3$	4	5	6	7	8	9	10
3	$(13, 17)$	F	1F	2D	63	ED	107	349	49D
4	$(27, 31)$	F	11	33	4F	B5	1AB	265	4D1
5	$(53, 75)$	9	11	3F	63	BD	16D	349	41B
6	$(133, 171)$	F	1B	3D	7F	FF	145	2BD	571
7	$(247, 371)$	F	11	33	63	EF	145	3A1	5D7
8	$(561, 753)$	F	11	33	7F	FF	1AB	301	4F5
9	$(1131, 1537)$	D	15	33	51	C5	1FF	349	583
10	$(2473, 3217)$	F	1B	33	79	BB	199	217	4DD

the above procedure will turn elements of $\text{IEE}(\sigma)$ into a length-N TB path. ∎

We now present our two-phase DSO CRC polynomial design algorithm for TBCCs of arbitrary rate (including rate $1/\omega$) at low target $P_{e,\lambda}$ that consists of the collection procedure as described in Algorithm 1 and the search procedure as described in Algorithm 2. In the collection procedure, $(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{v})$ denotes the triple of states $\boldsymbol{s}$, outputs $\boldsymbol{a}$ and inputs $\boldsymbol{v}$, where the inputs $\boldsymbol{v}$ are uniquely determined by state transitions $s_i \to s_{i+1}$, $i = 0, 1, \ldots, N-1$. The TB trellis considered in the collection procedure should set a sufficiently large trellis length so that IEEs with bounded distance less than $\tilde{d}$ are fully collected. Once the collection procedure is done, one can reuse the collected IEEs in the search procedure for various trellis lengths. For a given higher-rate code $\mathcal{C}_h$ and a specified CRC degree m, according to Theorem 2, it suffices to consider distance threshold $\tilde{d} \leq 2w^* + 1$, where w^* is the minimum weight determined in the theorem, to identify the degree-m DSO CRC polynomial.

In the search procedure, let $|\boldsymbol{v}|$ denote the length of $\boldsymbol{v}$. Steps from lines 2 to 5 use the IEEs to build all length-N trellis paths with distance less than $\tilde{d}$. In practice, this can be accomplished using dynamic programming. Specifically, for a given state $\sigma \in V_0$, let $\mathcal{L}_\sigma(w, l)$ denote the list of TB paths of weight w, of length l, and with initial state σ, $0 \leq w < \tilde{d}$, $1 \leq l \leq N$. Then, the update rule of $\mathcal{L}_\sigma(w, l)$ is as follows: given an IEE $(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{v}) \in \text{IEE}(\sigma)$ with $w_H(\boldsymbol{a}) \leq w$ and $|\boldsymbol{v}| < l$,

$$\mathcal{L}_\sigma(w, l) \leftarrow \mathcal{L}_\sigma(w, l) \cup \{\mathcal{L}_\sigma(w - w_H(\boldsymbol{a}), l - |\boldsymbol{v}|) \oplus (\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{v})\},$$

where $\mathcal{L}_\sigma(w, l) \oplus (\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{v})$ denotes appending $(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{v})$ to the rear of each element in $\mathcal{L}_\sigma(w, l)$. The update rule inherently requires that w, l be enumerated in ascending order and $w_H(\boldsymbol{a}), |\boldsymbol{v}|$ in descending order. Finally, the set of length-N TB paths of distance less than $\tilde{d}$ via direct concatenation are given by $\bigcup_{\sigma \in V_0} \mathcal{L}_\sigma(\tilde{d}-1, N)$. The rest of the TB paths are obtained by circularly shifting elements in $\bigcup_{\sigma \in V_0} \mathcal{L}_\sigma(\tilde{d}-1, N)$.

We remark that our algorithm can be generalized to ZTCCs of arbitrary rate yet comes with the following distinctions: the collection procedure only collects IEEs that start and terminate at the zero state; the search procedure only performs dynamic programming to reconstruct all ZT paths with the target trellis length N and distances less than $\tilde{d}$; termination tails of each ZT path should be removed before CRC verification. For interested readers, the MATLAB routines are available for ZTCCs [46] and for TBCCs [47].

Table II presents the DSO CRC polynomials of degree m from 3 to 10 that maximize d^l_{min} of CRC-ZTCCs based on a family of optimum rate-$1/2$ convolutional encoders in [42, Table 12.1(c)] with constraint length v from 3 to 10 for $k = 64$. These DSO CRC polynomials are for a sufficiently low $P_{e,\lambda}$. Table III presents the TBCC counterpart in the same setting. The code generated by the DSO CRC polynomial and convolutional encoder in the above tables is our designed CRC-aided convolutional code. In Section V, we will present the performance and complexity trade-off of these codes.

IV. PERFORMANCE AND COMPLEXITY OF SLVD

This section explores the performance and complexity of SLVD. For a specified CRC-aided convolutional code, performance under SLVD is characterized by three probabilities: $P_{c,\Psi}$, $P_{e,\Psi}$ and $P_{\mathit{NACK},\Psi}$. The average decoding complexity of SLVD is a function of expected list rank $\mathsf{E}[L]$. In order to understand the performance-complexity trade-off, we investigate how these quantities vary with system parameters including the SNR γ_s and the constrained maximum list size Ψ.

Geometrically speaking, the process of SLVD is to draw a list decoding sphere around the received sequence $\boldsymbol{y}$ with an increasing radius until the sphere touches the closest lower-rate codeword. To formalize this procedure, let us consider the set of received sequences $\boldsymbol{y} \in \mathbb{R}^n \setminus \mathcal{N}$ where $\mathcal{N}$ is the probability-zero set defined by $\mathcal{N} \triangleq \{\boldsymbol{y} \in \mathbb{R}^n : \exists \boldsymbol{c}_1, \boldsymbol{c}_2 \in \mathcal{C}_h \text{ s.t. } \|\boldsymbol{y} - \boldsymbol{x}(\boldsymbol{c}_1)\| = \|\boldsymbol{y} - \boldsymbol{x}(\boldsymbol{c}_2)\|\}$. For every $\boldsymbol{y} \in \mathbb{R}^n \setminus \mathcal{N}$, let

$$\boldsymbol{c}_1(\boldsymbol{y}), \boldsymbol{c}_2(\boldsymbol{y}), \ldots, \boldsymbol{c}_{|\mathcal{C}_h|}(\boldsymbol{y}) \tag{36}$$

be an enumeration of $\mathcal{C}_h$ such that

$$\|\boldsymbol{y} - \boldsymbol{x}(\boldsymbol{c}_1(\boldsymbol{y}))\| < \|\boldsymbol{y} - \boldsymbol{x}(\boldsymbol{c}_2(\boldsymbol{y}))\| < \cdots < \|\boldsymbol{y} - \boldsymbol{x}(\boldsymbol{c}_{|\mathcal{C}_h|}(\boldsymbol{y}))\|.$$

Using the above enumeration, we formally define the terminating list rank $L(\boldsymbol{y})$ and the terminating Euclidean distance $d_t(\boldsymbol{y})$ for $\boldsymbol{y}$ as follows:

$$L(\boldsymbol{y}) \triangleq \min\{s \in \{1, 2, \ldots, |\mathcal{C}_h|\} : \boldsymbol{c}_s(\boldsymbol{y}) \in \mathcal{C}_l\}, \tag{37}$$

$$d_t(\boldsymbol{y}) \triangleq \min_{\boldsymbol{c} \in \mathcal{C}_l} \|\boldsymbol{y} - \boldsymbol{x}(\boldsymbol{c})\|. \tag{38}$$

Thus, the list decoding sphere of $\boldsymbol{y}$ can be expressed as

$$\mathcal{B}_{\text{SLVD}}(\boldsymbol{y}) = \{\boldsymbol{c} \in \mathcal{C}_h : \|\boldsymbol{y} - \boldsymbol{x}(\boldsymbol{c})\| \leq d_t(\boldsymbol{y})\}. \tag{39}$$

Clearly, $L(\boldsymbol{y}) = |\mathcal{B}_{\text{SLVD}}(\boldsymbol{y})|$.

The concepts above are defined for each individual received point $\boldsymbol{y} \in \mathbb{R}^n \setminus \mathcal{N}$. Alternatively, we can also consider the decoding region $\mathcal{Y}(\boldsymbol{c})$ (i.e., the Voronoi region) of each lower-rate codeword $\boldsymbol{c} \in \mathcal{C}_l$:

$$\begin{aligned}\mathcal{Y}(\boldsymbol{c}) \triangleq \{&\boldsymbol{y} \in \mathbb{R}^n \setminus \mathcal{N} : \|\boldsymbol{y} - \boldsymbol{x}(\boldsymbol{c})\| < \|\boldsymbol{y} - \boldsymbol{x}(\boldsymbol{c}')\|,\\ &\forall \boldsymbol{c}' \in \mathcal{C}_l \setminus \{\boldsymbol{c}\}\}.\end{aligned} \tag{40}$$

For SLVD, the decoding region $\mathcal{Y}(\boldsymbol{c})$ can be further decomposed into finer subsets according to the list rank. Namely, for each $\boldsymbol{c} \in \mathcal{C}_l$ and a particular list rank $s \in \{1, 2, \ldots, |\mathcal{C}_h| - |\mathcal{C}_l| + 1\}$,

$$\begin{aligned}\mathcal{Z}_s(\boldsymbol{c}) \triangleq \Big\{&\boldsymbol{y} \in \mathbb{R}^n \setminus \mathcal{N} : \exists \boldsymbol{c}_1, \ldots, \boldsymbol{c}_{s-1} \in \mathcal{C}_h \setminus \mathcal{C}_l \text{ s.t.}\\ &\|\boldsymbol{y} - \boldsymbol{x}(\boldsymbol{c})\| > \max_{j=1,2,\ldots,s-1} \|\boldsymbol{y} - \boldsymbol{x}(\boldsymbol{c}_j)\| \text{ and}\\ &\|\boldsymbol{y} - \boldsymbol{x}(\boldsymbol{c})\| < \min_{\boldsymbol{c}' \notin \mathcal{C}_h \setminus \{\boldsymbol{c}, \boldsymbol{c}_1, \ldots, \boldsymbol{c}_{s-1}\}} \|\boldsymbol{y} - \boldsymbol{x}(\boldsymbol{c}')\|\Big\}.\end{aligned} \tag{41}$$

Here, each $\mathcal{Z}_s(\boldsymbol{c})$ is referred to as the *order-s decoding region of $\boldsymbol{c}$*. Obviously, for each $\boldsymbol{c} \in \mathcal{C}_l$, we have

$$\mathcal{Z}_{s_1}(\boldsymbol{c}) \cap \mathcal{Z}_{s_2}(\boldsymbol{c}) = \varnothing, \quad \text{if } s_1 \neq s_2 \tag{42}$$

$$\mathcal{Y}(\boldsymbol{c}) = \bigcup_{s=1,2,\ldots,|\mathcal{C}_h|-|\mathcal{C}_l|+1} \mathcal{Z}_s(\boldsymbol{c}). \tag{43}$$

By linearity of the code, the order-s decoding regions of all lower-rate codewords are isomorphic. With BPSK modulation, the bisection hyperplane of any two codewords passes through the origin of $\mathbb{R}^n$, making each order-s decoding region a polyhedron. Note that there exists a *supremum list rank* λ

$$\lambda \triangleq \max\{s : \mathcal{Z}_s(\boldsymbol{c}) \neq \varnothing, \forall \boldsymbol{c} \in \mathcal{C}_l\}. \tag{44}$$

Here, the supremum list rank λ only depends on $\mathcal{C}_l$ and $\mathcal{C}_h$ and is independent of Ψ. Hence, if $\Psi \geq \lambda$, the possible outcomes of SLVD include only correct decoding or UE. Namely, NACKs are not possible.

A. Performance Analysis

We first give our results on how $P_{c,\Psi}, P_{e,\Psi}$ and $P_{\mathit{NACK},\Psi}$ vary with Ψ for a fixed SNR. Each of these probabilities may be understood as the probability of an event defined as a set of received sequences $\boldsymbol{y}$. For example, with $\bar{\boldsymbol{c}} \in \mathcal{C}_l$ as the transmitted codeword, by linearity, we have

$$\begin{aligned}P_{c,\Psi} &= \mathsf{P}\left(\bigcup_{s=1,2\ldots,\lambda\wedge\Psi} \mathcal{Z}_s(\bar{\boldsymbol{c}}) \middle| \boldsymbol{X} = \boldsymbol{x}(\bar{\boldsymbol{c}})\right)\\ &= \sum_{s=1}^{\lambda\wedge\Psi} \mathsf{P}\big(\mathcal{Z}_s(\bar{\boldsymbol{c}}) | \boldsymbol{X} = \boldsymbol{x}(\bar{\boldsymbol{c}})\big),\end{aligned} \tag{45}$$

$$\begin{aligned}P_{e,\Psi} &= \sum_{\boldsymbol{c} \in \mathcal{C}_l \setminus \{\bar{\boldsymbol{c}}\}} \mathsf{P}\left(\bigcup_{s=1,2,\ldots\lambda\wedge\Psi} \mathcal{Z}_s(\boldsymbol{c}) \middle| \boldsymbol{X} = \boldsymbol{x}(\bar{\boldsymbol{c}})\right)\\ &= \sum_{s=1}^{\lambda\wedge\Psi} \sum_{\boldsymbol{c} \in \mathcal{C}_l \setminus \{\bar{\boldsymbol{c}}\}} \mathsf{P}\big(\mathcal{Z}_s(\boldsymbol{c}) | \boldsymbol{X} = \boldsymbol{x}(\bar{\boldsymbol{c}})\big),\end{aligned} \tag{46}$$

where $\lambda \wedge \Psi \triangleq \min\{\lambda, \Psi\}$.

Theorem 4. *For a given CRC-aided convolutional code decoded with SLVD at a fixed SNR, $P_{c,\Psi}$ and $P_{e,\Psi}$ are both strictly increasing in Ψ and will converge to $P_{c,\lambda}$ and $P_{e,\lambda}$ respectively, where $P_{c,\lambda} + P_{e,\lambda} = 1$.*

Proof: According to (45) and (46), $P_{c,\Psi}$ and $P_{e,\Psi}$ are summations of the order-s decoding regions $\mathsf{P}(\mathcal{Z}_s(\boldsymbol{c}) | \boldsymbol{X} = \boldsymbol{x}(\bar{\boldsymbol{c}}))$, thus are non-decreasing in Ψ. For each $\boldsymbol{c} \in \mathcal{C}_l$ and $s = 1, 2, \ldots, \lambda$, $\mathsf{P}(\mathcal{Z}_s(\boldsymbol{c}) | \boldsymbol{X} = \boldsymbol{x}(\bar{\boldsymbol{c}}))$ is solely determined by the SNR value and is independent of Ψ. Since every order-s decoding region $\mathcal{Z}_s(\boldsymbol{c})$ is the intersection of halfplanes, it follows that each $\mathcal{Z}_s(\boldsymbol{c})$ is an open set. Hence, it suffices to show that each $\mathcal{Z}_s(\boldsymbol{c})$ is nonempty. To this end, we use induction to show that all $\mathcal{Z}_s(\boldsymbol{c})$, $s = 1, 2, \ldots, \lambda$, are open and nonempty.

By definition, $\mathcal{Z}_\lambda(\boldsymbol{c})$ is open and nonempty. Assume $\mathcal{Z}_s(\boldsymbol{c})$ is open and nonempty for some fixed $s \leq \lambda$. Hence, there exists $\boldsymbol{y} \in \mathcal{Z}_s(\boldsymbol{c})$ with $\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_s \in \mathcal{B}_{\text{SLVD}}(\boldsymbol{y})$, where $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_{s-1} \in \mathcal{C}_h \setminus \mathcal{C}_l$ and $\boldsymbol{c}_s \in \mathcal{C}_l$. Next, we show that with probability 1, a point $\boldsymbol{y}'$ can be constructed from $\boldsymbol{y}$ such that $\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_{j-1}, \boldsymbol{c}_{j+1}, \ldots, \boldsymbol{c}_{s-1}, \boldsymbol{c}_s \in \mathcal{B}_{\text{SLVD}}(\boldsymbol{y}')$ for some $j \in \{2, 3, \ldots, s-2\}$.

The new point $\boldsymbol{y}'$ is constructed as $\boldsymbol{y}' = \boldsymbol{y} + t(\boldsymbol{x}(\boldsymbol{c}_s) - \boldsymbol{y})$, where $t \in [0, 1]$. Hence,

$$\|\boldsymbol{x}(\boldsymbol{c}_s) - \boldsymbol{y}'\| = (1-t)\|\boldsymbol{y} - \boldsymbol{x}(\boldsymbol{c}_s)\|. \tag{47}$$

Therefore, it is equivalent to showing that there exists $t \in (0, 1)$ such that for some $j \in \{1, 2, \ldots, s-1\}$,

$$\|\boldsymbol{y}' - \boldsymbol{x}(\boldsymbol{c}_j)\| > (1-t)\|\boldsymbol{y} - \boldsymbol{x}(\boldsymbol{c}_s)\| \tag{48}$$

$$\max_{i \in \{1,\ldots,s-1\}\setminus\{j\}} \|\boldsymbol{y}' - \boldsymbol{x}(\boldsymbol{c}_i)\| < (1-t)\|\boldsymbol{y} - \boldsymbol{x}(\boldsymbol{c}_s)\|. \tag{49}$$

To this end, we show that the set of $\boldsymbol{y}$ for which no such t exists has a probability of zero. First, consider function

$$F(t) \triangleq \max_{i=1,2,\ldots,s-1} \|\boldsymbol{y}' - \boldsymbol{x}(\boldsymbol{c}_i)\| - (1-t)\|\boldsymbol{y} - \boldsymbol{x}(\boldsymbol{c}_s)\|.$$

Since each $\|\boldsymbol{y}' - \boldsymbol{x}(\boldsymbol{c}_i)\|$, $i = 1, 2, \ldots, s-1$, is a continuous function in t, $F(t)$ is also a continuous function in $t \in [0, 1]$. Note that

$$F(0) = \max_{i=1,2,\ldots,s-1} \|\boldsymbol{y} - \boldsymbol{x}(\boldsymbol{c}_i)\| - \|\boldsymbol{y} - \boldsymbol{x}(\boldsymbol{c}_s)\| < 0 \tag{50}$$

$$F(1) = \max_{i=1,2,\ldots,s-1} \|\boldsymbol{x}(\boldsymbol{c}_s) - \boldsymbol{x}(\boldsymbol{c}_i)\| > 0. \tag{51}$$

By the intermediate value theorem, there exists a $t^* \in (0, 1)$ such that

$$\max_{i=1,2,\ldots,s-1} \|\boldsymbol{y}' - \boldsymbol{x}(\boldsymbol{c}_i)\| = (1-t^*)\|\boldsymbol{y} - \boldsymbol{x}(\boldsymbol{c}_s)\|. \tag{52}$$

Hence, the converse case can only occur if there exist two codewords $\boldsymbol{c}_{j_1}$ and $\boldsymbol{c}_{j_2}$, $j_1 \neq j_2$, such that

$$\|\boldsymbol{y}' - \boldsymbol{x}(\boldsymbol{c}_{j_1})\| = \|\boldsymbol{y}' - \boldsymbol{x}(\boldsymbol{c}_{j_2})\| = (1-t^*)\|\boldsymbol{y} - \boldsymbol{x}(\boldsymbol{c}_s)\|. \tag{53}$$

If (53) holds, this implies that $\boldsymbol{y}'$ lies on the intersection of two hyperplanes: one that bisects $\boldsymbol{x}(\boldsymbol{c}_{j_1})\boldsymbol{x}(\boldsymbol{c}_s)$ and the other that bisects $\boldsymbol{x}(\boldsymbol{c}_{j_2})\boldsymbol{x}(\boldsymbol{c}_s)$. Namely, $\boldsymbol{y}'$ lies on an $(n-2)$-dimensional hyperplane that crosses the origin. Hence, such $\boldsymbol{y}'$ only occurs if line segment $\boldsymbol{y}\boldsymbol{x}(\boldsymbol{c}_s)$ intersects with any of these

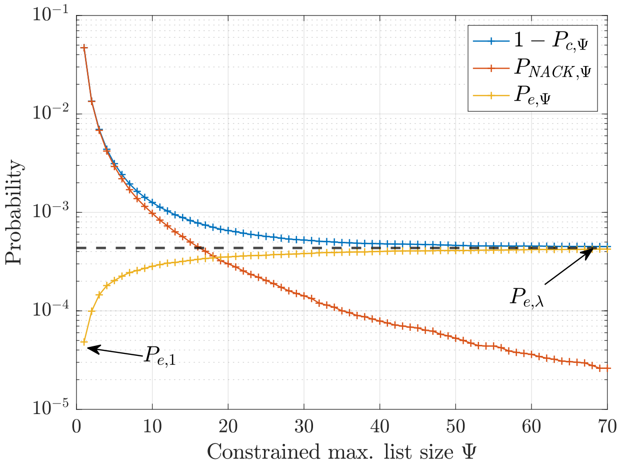

Fig. 3. $1 - P_{c,\Psi}, P_{NACK,\Psi}, P_{e,\Psi}$ vs. the constraint maximum list size Ψ at SNR $\gamma_s = 3$ dB for ZTCC $(13, 17)$, degree-6 DSO CRC polynomial 0x43 and $k = 64$ in Table II. The black, dashed line represents $P_{e,\lambda}$.

$(n-2)$-dimensional hyperplanes. Therefore, the set of $\boldsymbol{y}$ for which the converse case occurs is the union of finitely many $(n-1)$-dimensional hyperplanes, and thus has probability of zero. Hence, we can construct a $\boldsymbol{y}'$ from $\boldsymbol{y} \in \mathcal{Z}_s(\boldsymbol{c})$ such that $L(\boldsymbol{y}') = s - 1$ with probability 1. Namely, $\mathcal{Z}_{s-1}(\boldsymbol{c})$ is open and nonempty.

By induction, every order-s decoding region $\mathcal{Z}_s(\boldsymbol{c})$, $s = 1, 2, \ldots, \lambda$, is open and nonempty. Thus, $P_{c,\Psi}$ and $P_{e,\Psi}$ are both strictly increasing in Ψ and will converge to $P_{c,\lambda}$ and $P_{e,\lambda}$ respectively provided that $\Psi \geq \lambda$. ■

As an example, Fig. 3 shows the probability of UE $P_{e,\Psi}$ and probability of NACK $P_{NACK,\Psi}$ vs. the constrained maximum list size Ψ for $k = 64$, degree-6 DSO CRC polynomial 0x43 and ZTCC $(13, 17)$. It can be seen that $P_{e,\Psi}$ quickly increases and converges to $P_{e,\lambda}$ when Ψ is relatively small.

The monotone property of $P_{e,\Psi}$ with Ψ in Theorem 4 indicates that for a fixed SNR value,

$$P_{e,1} \leq P_{e,\Psi} \leq P_{e,\lambda}, \quad \forall \Psi \in \mathbb{N}^+. \tag{54}$$

The proof of Theorem 4 also implies that the closure of the order-λ decoding region must intersect with the boundary of $\mathcal{Y}(\boldsymbol{c})$, $\boldsymbol{c} \in \mathcal{C}_l$. We formalize this notion in Theorem 5.

Theorem 5. *For any lower-rate codeword $\boldsymbol{c} \in \mathcal{C}_l$, $\mathrm{cl}(\mathcal{Z}_\lambda(\boldsymbol{c})) \cap \partial\mathcal{Y}(\boldsymbol{c}) \neq \varnothing$.*

Proof: Fix a lower-rate codeword $\boldsymbol{c} \in \mathcal{C}_l$. Let $\boldsymbol{y} \in \mathcal{Z}_\lambda(\boldsymbol{c})$. Consider $\boldsymbol{y}' = \boldsymbol{y} + t(\boldsymbol{y} - \boldsymbol{x}(\boldsymbol{c}))$, $t \geq 0$. By the proof in Theorem 4, if $\boldsymbol{y}' \in \mathcal{Y}(\boldsymbol{c})$, $L(\boldsymbol{y}') \geq L(\boldsymbol{y}) = \lambda$. Since λ is the maximum list rank, $L(\boldsymbol{y}') = \lambda$ for all $0 \leq t < t^*$, where t^* is the threshold at which $\boldsymbol{y}' \in \partial\mathcal{Y}(\boldsymbol{c})$. This implies that $\mathrm{cl}(\mathcal{Z}_\lambda(\boldsymbol{c})) \cap \partial\mathcal{Y}(\boldsymbol{c}) \neq \varnothing$. ■

Theorem 5 indicates that one can find λ by following along the boundary of $\mathcal{Y}(\boldsymbol{c})$ and making a slight deviation towards the decoding region $\mathcal{Y}(\boldsymbol{c})$. This approach is computationally challenging in $\mathbb{R}^n$ for interesting values of n. While $\lambda \leq |\mathcal{C}_h| - |\mathcal{C}_l| + 1$ provides an initial upper bound on λ, it remains an open problem to identify a tighter bound on λ and to develop an efficient algorithm to compute λ.

We next direct our attention to quantifying $P_{e,1}$, $P_{e,\lambda}$ in terms of the SNR (or equivalently in terms of amplitude A) and the distance spectra of both the lower-rate code $\mathcal{C}_l$ and the higher-rate code $\mathcal{C}_h$.

Theorem 6. *Under SLVD of a CRC-aided convolutional code with higher-rate distance spectrum $B_{d^h_{\min}}, \ldots, B_n$ and lower-rate distance spectrum $C_{d^l_{\min}}, \ldots, C_n$,*

$$P_{e,1} \leq \min\left\{2^{-m},\ \sum_{d=d^l_{\min}}^{n} C_d Q\big(A\sqrt{d}\big)\right\} \tag{55}$$

$$\approx \min\left\{2^{-m},\ C_{d^l_{\min}} Q\Big(A\sqrt{d^l_{\min}}\Big)\right\}, \tag{56}$$

$$P_{e,\lambda} \leq \min\left\{1,\ \sum_{d=d^l_{\min}}^{n} C_d Q\big(A\sqrt{d}\big)\right\} \tag{57}$$

$$\approx \min\left\{1,\ \sum_{d=d^l_{\min}}^{\tilde{d}} C_d Q\big(A\sqrt{d}\big)\right\}, \tag{58}$$

$$P_{NACK,1} \approx \min\Bigg\{1 - 2^{-m},\ \sum_{d=d^h_{\min}}^{\tilde{d}} B_d Q\big(A\sqrt{d}\big) - C_{d^l_{\min}} Q\Big(A\sqrt{d^l_{\min}}\Big)\Bigg\}, \tag{59}$$

where the second approximation in braces in (56) *is called the nearest neighbor approximation, and the second approximation in* (58) *is called the truncated union bound (TUB) at distance $\tilde{d}$.*

Proof: First, note that $P_{e,\Psi}$ is a monotonically decreasing function of A for any Ψ. This can be seen from (46) where as A increases, the center of the Gaussian density is moving away from every $\boldsymbol{x}(\boldsymbol{c})$ for $\boldsymbol{c} \in \mathcal{C}_l \setminus \{\bar{\boldsymbol{c}}\}$. Hence, the corresponding probability $\mathsf{P}(\mathcal{Z}_s(\boldsymbol{c})|\boldsymbol{X} = \boldsymbol{x}(\bar{\boldsymbol{c}}))$ decreases with A, causing $P_{e,\Psi}$ to decrease with A.

Now we focus on the $\Psi = 1$ case. The previous paragraph reveals that $P_{e,1}$ has its maximum value at $A = 0$. As $A \to 0$, the transmitted point converges to the origin $\boldsymbol{O}$ in $\mathbb{R}^n$. At the limit where $\boldsymbol{x}(\bar{\boldsymbol{c}}) = \boldsymbol{O}$, the symmetry of the Gaussian density and linearity of the code ensures that each order-1 decoding region has a probability of $2^{-(k+m)}$. Hence,

$$P_{e,1} = \sum_{\boldsymbol{c} \in \mathcal{C}_l \setminus \{\bar{\boldsymbol{c}}\}} \mathsf{P}(\mathcal{Z}_1(\boldsymbol{c})|\boldsymbol{X} = \boldsymbol{x}(\bar{\boldsymbol{c}})) \tag{60}$$

$$\leq \lim_{A \to 0} \sum_{\boldsymbol{c} \in \mathcal{C}_l \setminus \{\bar{\boldsymbol{c}}\}} \mathsf{P}(\mathcal{Z}_1(\boldsymbol{c})|\boldsymbol{X} = \boldsymbol{x}(\bar{\boldsymbol{c}})) \tag{61}$$

$$= \sum_{\boldsymbol{c} \in \mathcal{C}_l \setminus \{\bar{\boldsymbol{c}}\}} \mathsf{P}(\mathcal{Z}_1(\boldsymbol{c})|\boldsymbol{X} = \boldsymbol{O})) \tag{62}$$

$$= (2^k - 1)2^{-(k+m)} \leq 2^{-m}. \tag{63}$$

For any SNR value, $P_{e,1} < P_{e,\lambda}$ so that the union bound (23) is also an upper bound for $P_{e,1}$. Hence, the minimum between the two is an upper bound on $P_{e,1}$. As SNR increases, the majority of probability will concentrate on the nearest

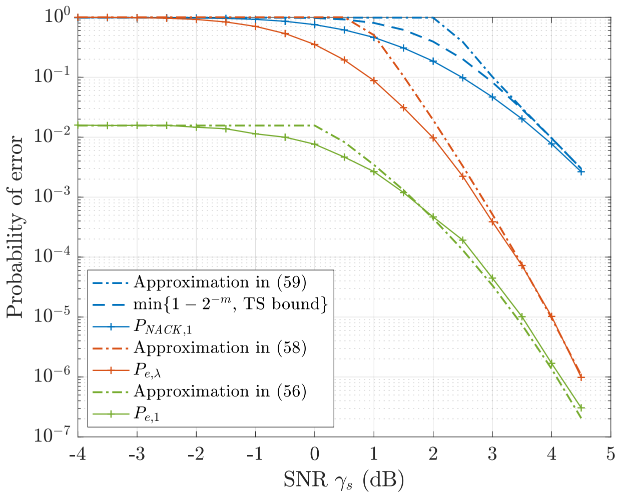

Fig. 4. $P_{\mathit{NACK},1}$, $P_{e,\lambda}$ and $P_{e,1}$ vs. SNR γ_s for ZTCC $(13,17)$, degree-6 DSO CRC polynomial 0x43 and $k=64$ in Table II. The TUBs in (58) and (59) are obtained at $\tilde{d}=24$. The TS bound on $P_{\mathit{NACK},1}$ is plotted using [48, Eq. (14)].

neighbors of $\bar{\boldsymbol{c}}$, hence, we can approximate $P_{e,1}$ only using the nearest neighbors.

For $P_{e,\lambda}$, we upper bound it by the union bound (23). For ease of computation, we can consider the TUB up to a sufficient distance $\tilde{d}$ to approximate the original union bound.

For $P_{\mathit{NACK},1}$, in the extremely low SNR regime (i.e., when A is close to 0), $P_{c,1}\approx 2^{-(k+m)}$ and $P_{e,1}\approx 2^{-m}(1-2^{-k})$. It follows that

$$P_{\mathit{NACK},1}=1-P_{e,1}-P_{c,1}\approx 1-2^{-m}. \tag{64}$$

For an arbitrary SNR, invoking the union bound on $P_{\mathit{NACK},1}+P_{e,1}$ yields

$$P_{\mathit{NACK},1}+P_{e,1}\le\sum_{d=d^h_{\min}}^{n}B_dQ\big(A\sqrt{d}\big)\approx\sum_{d=d^h_{\min}}^{\tilde{d}}B_dQ\big(A\sqrt{d}\big).$$

Hence,

$$P_{\mathit{NACK},1}\approx\sum_{d=d^h_{\min}}^{\tilde{d}}B_dQ\big(A\sqrt{d}\big)-C_{d^l_{\min}}Q\left(A\sqrt{d^l_{\min}}\right). \tag{65}$$

This concludes the proof of Theorem 6. ■

Fig. 4 shows simulation results and approximations for the three probabilities addressed in Theorem 6: $P_{\mathit{NACK},1}$, $P_{e,1}$, and $P_{e,\lambda}$. As SNR increases, all three approximations become asymptotically tight to the respective $P_{e,1}$, $P_{\mathit{NACK},1}$, and $P_{e,\lambda}$. The nearest neighbor approximation of the union bound on $P_{e,\lambda}$ eventually will become asymptotically tight for $P_{e,\lambda}$, but is a tight approximation for $P_{e,1}$ at a much lower SNR.

We remark that improved upper bounds on $P_{\mathit{NACK},1}$ and $P_{e,\lambda}$ can be derived using Gallager's first bounding technique [49], provided that the full distance spectra of $\mathcal{C}_h$ and $\mathcal{C}_l$ are known, respectively. Some classical examples include the tangential bound [50], the tangential sphere (TS) bound [48], [51], and the added-hyperplane bound [52]. These bounds provide a tight estimation at high noise levels and converge to the union bound at low noise levels. As an example, in Fig. 4, we plot the minimum between $(1-2^{-m})$ and the TS bound for $P_{\mathit{NACK},1}$ following [48, Eq. (14)]. It can be seen that the TS bound quickly converges to the TUB as SNR increases. Since this paper mainly focuses on low target error probability, we only consider the TUB for estimating $P_{\mathit{NACK},1}$ and $P_{e,\lambda}$.

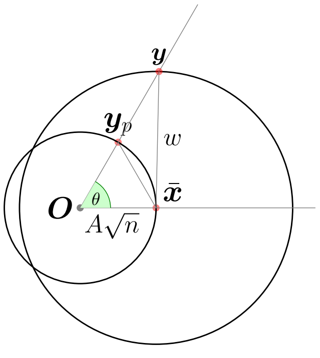

Fig. 5. An illustration of the projection method.

B. Analysis of the Expected List Rank

For a fixed transmitted point $\bar{\boldsymbol{x}}$, observe that $\mathsf{P}(L=s|\boldsymbol{X}=\bar{\boldsymbol{x}})=\sum_{\boldsymbol{c}\in\mathcal{C}_l}\mathsf{P}(Z_s(\boldsymbol{c})|\boldsymbol{X}=\bar{\boldsymbol{x}})$ is independent of Ψ. Combining with the linearity $\mathsf{E}[L]=\mathsf{E}[L|\boldsymbol{X}=\bar{\boldsymbol{x}}]$, it follows that $\mathsf{E}[L]$ is a strictly increasing function in Ψ. In the subsequent analysis, we assume that $\Psi\ge\lambda$ and the terminating list rank L ranges from 1 to λ unless otherwise specified.

Theorem 7. *For a given CRC-aided convolutional code decoded with SLVD,* $\lim_{\gamma_s\to 0}\mathsf{E}[L]=\mathsf{E}[L|\boldsymbol{X}=\boldsymbol{O}]$.

Proof: We use the projection method to show the convergence of $\mathsf{E}[L]$ in the low SNR regime.

For ease of discussion, let $\mathcal{B}(\boldsymbol{a},r)$ denote the *spherical surface* of center $\boldsymbol{a}\in\mathbb{R}^n$ and radius r in $\mathbb{R}^n$. With BPSK modulation, all codewords sit on the *codeword sphere* $\mathcal{B}(\boldsymbol{O},A\sqrt{n})$, whereas the received point $\boldsymbol{y}$ lies on the *noise sphere* $\mathcal{B}(\bar{\boldsymbol{x}},w)$ for some noise vector with Euclidean norm w added to the transmitted point $\bar{\boldsymbol{x}}$. The projection method projects the received point $\boldsymbol{y}$ onto the codeword sphere. Namely, the projected point $\boldsymbol{y}_p$ of $\boldsymbol{y}$ is given by $\boldsymbol{y}_p=(A\sqrt{n}/\|\boldsymbol{y}\|)\boldsymbol{y}$. Fig. 5 illustrates the geometry of the projection method.

The significance of the projection method introduced above lies in the fact that it preserves the order of list decoded codewords. By law of cosines at angle θ in Fig. 5, we obtain

$$\|\boldsymbol{y}_p-\bar{\boldsymbol{x}}\|=\begin{cases}\sqrt{\dfrac{\|\boldsymbol{y}-\bar{\boldsymbol{x}}\|^2-\|\boldsymbol{y}-\boldsymbol{y}_p\|^2}{1+\frac{\|\boldsymbol{y}-\boldsymbol{y}_p\|}{A\sqrt{n}}}}, & \text{if } \boldsymbol{y}_p \text{ in between } \boldsymbol{O},\ \boldsymbol{y}\\ \sqrt{\dfrac{\|\boldsymbol{y}-\bar{\boldsymbol{x}}\|^2-\|\boldsymbol{y}-\boldsymbol{y}_p\|^2}{1-\frac{\|\boldsymbol{y}-\boldsymbol{y}_p\|}{A\sqrt{n}}}}, & \text{otherwise.}\end{cases} \tag{66}$$

Hence, the monotone relation between $\|\boldsymbol{y}_p-\bar{\boldsymbol{x}}\|$ and $\|\boldsymbol{y}-\bar{\boldsymbol{x}}\|$ ensures that performing SLVD using $\boldsymbol{y}$ is equivalent to that using $\boldsymbol{y}_p$. The essential motivation of projecting points onto the codeword sphere is to transfer the computation on the noise sphere to the codeword sphere.

To see how the projection method helps to show the convergence of $\mathsf{E}[L]$, we first decompose the expected list rank $\mathsf{E}[L]$ according to the noise vector norm $W = w$. By linearity of the code,

$$\begin{aligned}\mathsf{E}[L] &= \mathsf{E}[L|\boldsymbol{X} = \bar{\boldsymbol{x}}] \\ &= \int_0^\infty f_W(w)\mathsf{E}[L|W = w, \boldsymbol{X} = \bar{\boldsymbol{x}}]\,\mathrm{d}w, \end{aligned} \tag{67}$$

where $f_W(w)$ denotes the density function of norm $W = w$. To find $f_W(w)$, let

$$\phi_n(w) \triangleq \frac{1}{(\sqrt{2\pi})^n}\exp\left(-\frac{w^2}{2}\right), \tag{68}$$

$$S_{n-1}(w) \triangleq \frac{2\pi^{\frac{n}{2}}}{\Gamma(\frac{n}{2})}w^{n-1} \tag{69}$$

be the n-dimensional standard normal density function and the spherical area of $\mathcal{B}(\bar{\boldsymbol{x}}, w)$ in $\mathbb{R}^n$, respectively. Then,

$$f_W(w) = \phi_n(w)S_{n-1}(w) = \frac{w^{n-1}}{2^{\frac{n-2}{2}}\Gamma(\frac{n}{2})}\exp\left(-\frac{w^2}{2}\right). \tag{70}$$

For a given norm $W = w$, it follows that

$$\mathsf{E}[L|W = w, \boldsymbol{X} = \bar{\boldsymbol{x}}] = \frac{1}{S_{n-1}(w)}\int_{\boldsymbol{y}\in\mathcal{B}(\bar{\boldsymbol{x}},w)\backslash\mathcal{N}} L(\boldsymbol{y})\,\mathrm{d}\boldsymbol{\sigma}, \tag{71}$$

where $\boldsymbol{\sigma}$ denotes the spherical measure on $\mathcal{B}(\bar{\boldsymbol{x}}, w)$. Using the projection method, the integral in (71) can be transformed to the codeword sphere at the cost of introducing an induced density function $g_w(\boldsymbol{y}_p)$. Namely,

$$\mathsf{E}[L|W = w, \boldsymbol{X} = \bar{\boldsymbol{x}}] = \int_{\boldsymbol{y}_p\in\mathcal{B}(\boldsymbol{O},A\sqrt{n})\backslash\mathcal{N}} L(\boldsymbol{y}_p)g_w(\boldsymbol{y}_p)\,\mathrm{d}\boldsymbol{\sigma}. \tag{72}$$

In Appendix A, the induced density function, for $w \geq A\sqrt{n}$, is given by

$$g_w(\boldsymbol{y}_p) = \left(\frac{\|\boldsymbol{y}(\boldsymbol{y}_p)\|}{w}\right)^{n-1}\frac{1}{\cos\angle\bar{\boldsymbol{x}}\boldsymbol{y}(\boldsymbol{y}_p)\boldsymbol{O}}\frac{1}{S_{n-1}(A\sqrt{n})}, \tag{73}$$

where $\boldsymbol{y}(\boldsymbol{y}_p)$ is the pre-image of $\boldsymbol{y}_p$ on the noise sphere $\mathcal{B}(\bar{\boldsymbol{x}}, w)$. Note that $g_w(\boldsymbol{y}_p)$ is rotationally symmetric with respect to axis $\boldsymbol{O}\bar{\boldsymbol{x}}$. Appendix A also shows that

$$g_w(\boldsymbol{y}_p) \geq \frac{1}{S_{n-1}(A\sqrt{n})}\left(1 - \frac{A\sqrt{n}}{w}\right)^{n-1}, \tag{74}$$

$$g_w(\boldsymbol{y}_p) \leq \frac{1}{S_{n-1}(A\sqrt{n})}\left(1 + \frac{A\sqrt{n}}{w}\right)^{n-1}. \tag{75}$$

This implies that for a fixed norm w,

$$\lim_{A\to 0}\frac{g_w(\boldsymbol{y}_p)}{(S_{n-1}(A\sqrt{n}))^{-1}} = 1. \tag{76}$$

Hence, for a fixed norm w, it follows that

$$\begin{aligned}&\lim_{A\to 0}\mathsf{E}[L|W = w, \boldsymbol{X} = \bar{\boldsymbol{x}}] \\ &= \lim_{A\to 0}\int_{\boldsymbol{y}_p\in\mathcal{B}(\boldsymbol{O},A\sqrt{n})\backslash\mathcal{N}} L(\boldsymbol{y}_p)g_w(\boldsymbol{y}_p)\,\mathrm{d}\boldsymbol{\sigma} &(77)\\ &= \lim_{A\to 0}\int_{\boldsymbol{y}_p\in\mathcal{B}(\boldsymbol{O},A\sqrt{n})\backslash\mathcal{N}} L(\boldsymbol{y}_p)\frac{1}{S_{n-1}(A\sqrt{n})}\,\mathrm{d}\boldsymbol{\sigma} &(78)\\ &= \lim_{A\to 0}\mathsf{E}[L|W = A\sqrt{n}, \boldsymbol{X} = \boldsymbol{O}] &(79)\\ &=\mathsf{E}[L|\boldsymbol{X} = \boldsymbol{O}], &(80)\end{aligned}$$

where we have used the fact that $\mathsf{E}[L|W = w, \boldsymbol{X} = \boldsymbol{O}] = \mathsf{E}[L|\boldsymbol{X} = \boldsymbol{O}]$ for all $w > 0$. Similarly, we can also show that, for a fixed amplitude A,

$$\lim_{w\to\infty}\mathsf{E}[L|W = w, \boldsymbol{X} = \bar{\boldsymbol{x}}] = \mathsf{E}[L|\boldsymbol{X} = \boldsymbol{O}]. \tag{81}$$

As a consequence,

$$\begin{aligned}\lim_{\gamma_s\to 0}\mathsf{E}[L] &= \lim_{A\to 0}\int_0^\infty f_W(w)\mathsf{E}[L|W = w, \boldsymbol{X} = \bar{\boldsymbol{x}}]\,\mathrm{d}w \\ &= \int_0^\infty f(w)\lim_{A\to 0}\mathsf{E}[L|W = w, \boldsymbol{X} = \bar{\boldsymbol{x}}]\,\mathrm{d}w \\ &= \int_0^\infty f(w)\mathsf{E}[L|\boldsymbol{X} = \boldsymbol{O}]\,\mathrm{d}w \\ &= \mathsf{E}[L|\boldsymbol{X} = \boldsymbol{O}]. \end{aligned} \tag{82}$$

This completes the proof. ■

The proof above implies that in the low SNR regime, most of the probability will concentrate on the limit of $\mathsf{E}[L|W = w, \boldsymbol{X} = \bar{\boldsymbol{x}}]$ as $w \to \infty$, i.e., $\mathsf{E}[L|\boldsymbol{X} = \boldsymbol{O}]$. In general, $\mathsf{E}[L|\boldsymbol{X} = \boldsymbol{O}]$ depends on the geometric structure of the lower-rate code $\mathcal{C}_l$ and the higher-rate code $\mathcal{C}_h$ on $\mathcal{B}(\boldsymbol{O}, A\sqrt{n})$ and it is not easy to obtain an analytic expression. Still, using a simple random coding argument, we show that a good concatenated code could achieve $\mathsf{E}[L|\boldsymbol{X} = \boldsymbol{O}] \leq 2^m$.

Theorem 8. *For a given higher-rate code $\mathcal{C}_h$ with $|\mathcal{C}_h| = 2^{k+m}$, let $\mathcal{A}_l \triangleq \{\mathcal{C}' \subset \mathcal{C}_h : |\mathcal{C}'| = 2^k\}$. Let $\mathsf{P}(\mathcal{C}') = \frac{1}{|\mathcal{A}_l|}$ be the uniform distribution defined over $\mathcal{A}_l$. Assume $\mathcal{C}'$ is drawn according to $\mathsf{P}(\mathcal{C}')$. Then,*

$$\mathsf{E}_{\mathcal{C}'}\left[\mathsf{E}[L|\boldsymbol{X} = \boldsymbol{O}, \mathcal{C}']\right] \leq 2^m. \tag{83}$$

This implies that there exists a lower-rate code $\mathcal{C}'$ (which may not be a linear code) such that $\mathsf{E}[L|\boldsymbol{X} = \boldsymbol{O}, \mathcal{C}'] \leq 2^m$.

Proof: Let $L(\boldsymbol{y}, \mathcal{C}')$ be the terminating list rank for received point $\boldsymbol{y} \in \mathbb{R}^n$ when a lower-rate code is selected as $\mathcal{C}' \in \mathcal{A}_l$[2]. Hence, we obtain

$$\begin{aligned}&\mathsf{E}_{\mathcal{C}'}\left[\mathsf{E}[L|\boldsymbol{X} = \boldsymbol{O}, \mathcal{C}']\right] \\ &= \sum_{\mathcal{C}'\in\mathcal{A}_l}\mathsf{P}(\mathcal{C}')\frac{1}{S_{n-1}(A\sqrt{n})}\int_{\boldsymbol{y}\in\mathcal{B}(\boldsymbol{O},A\sqrt{n})} L(\boldsymbol{y}, \mathcal{C}')\,\mathrm{d}\boldsymbol{\sigma} \\ &=\frac{1}{S_{n-1}(A\sqrt{n})}\int_{\boldsymbol{y}\in\mathcal{B}(\boldsymbol{O},A\sqrt{n})}\sum_{\mathcal{C}'\in\mathcal{A}_l}\mathsf{P}(\mathcal{C}')L(\boldsymbol{y}, \mathcal{C}')\,\mathrm{d}\boldsymbol{\sigma} \\ &=\frac{1}{S_{n-1}(A\sqrt{n})}\int_{\boldsymbol{y}\in\mathcal{B}(\boldsymbol{O},A\sqrt{n})}\mathsf{E}_{\mathcal{C}'}[L(\boldsymbol{y}, \mathcal{C}')|\boldsymbol{y}]\,\mathrm{d}\boldsymbol{\sigma}. \end{aligned} \tag{84}$$

[2] If there exist two codewords $\boldsymbol{c}_{j_1}$ and $\boldsymbol{c}_{j_2}$ that are equidistant from $\boldsymbol{y}$, the decoder adopts a pre-determined order relation between $\boldsymbol{c}_{j_1}$ and $\boldsymbol{c}_{j_2}$.

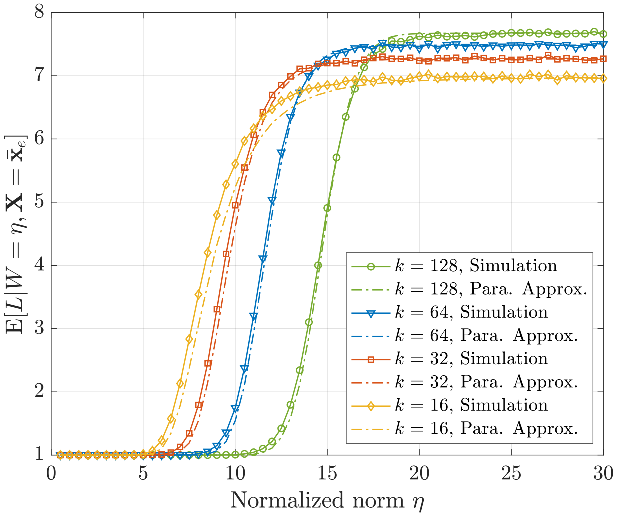

Fig. 6. The conditional expected list rank $\mathsf{E}[L|W=\eta,\boldsymbol{X}=\bar{\boldsymbol{x}}_e]$ vs. the normalized norm η for the CRC-ZTCC generated with the degree-3 DSO CRC polynomial 0x9 and ZTCC $(13,17)$.

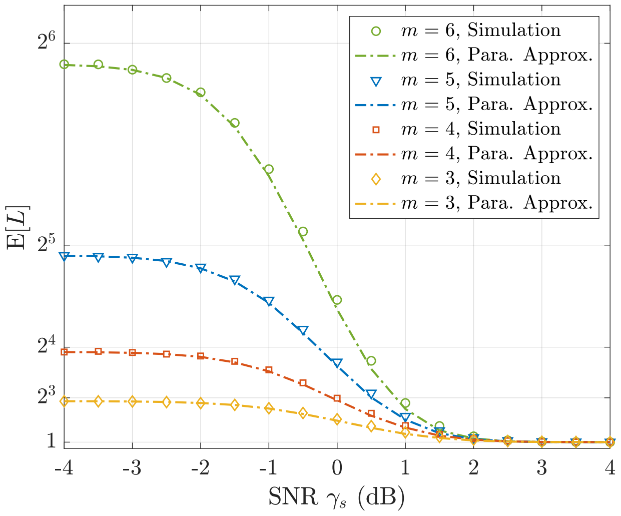

Fig. 7. The expected list rank $\mathsf{E}[L]$ vs. SNR for various CRC-ZTCCs, where ZTCC is $(13,17)$ and the DSO CRC polynomials are from Table II with degree $m=3,4,\ldots,6$. The information length $k=64$.

Next, we show that for any $\boldsymbol{y}\in\mathcal{B}(\boldsymbol{O},A\sqrt{n})$,

$$\mathsf{E}_{\mathcal{C}'}[L(\boldsymbol{y},\mathcal{C}')|\boldsymbol{y}]\le 2^m \tag{85}$$

for $\mathcal{C}'$ uniformly drawn from $\mathcal{A}_l$. Fix a $\boldsymbol{y}\in\mathcal{B}(\boldsymbol{O},A\sqrt{n})$ and let $\boldsymbol{c}_1(\boldsymbol{y}),\boldsymbol{c}_2(\boldsymbol{y}),\ldots,\boldsymbol{c}_{|\mathcal{C}_h|}(\boldsymbol{y})$ be an enumeration of $\mathcal{C}_h$ such that

$$\|\boldsymbol{y}-\boldsymbol{x}(\boldsymbol{c}_1(\boldsymbol{y}))\|\le\cdots\le\|\boldsymbol{y}-\boldsymbol{x}(\boldsymbol{c}_{|\mathcal{C}_h|}(\boldsymbol{y}))\|.$$

Hence, the terminating list rank $L(\boldsymbol{y},\mathcal{C}')$ of $\boldsymbol{y}$ is given by

$$L(\boldsymbol{y},\mathcal{C}')=\min\{s:\boldsymbol{c}_s(\boldsymbol{y})\in\mathcal{C}'\}. \tag{86}$$

For $\mathcal{C}'$ uniformly drawn in $\mathcal{A}_l$, computing $\mathsf{E}_{\mathcal{C}'}[L(\boldsymbol{y},\mathcal{C}')|\boldsymbol{y}]$ is equivalent to solving the following problem: there are $|\mathcal{C}_h|$ balls in a basket, among which $|\mathcal{C}'|$ of them are red and the rest are white. Balls are picked up $|\mathcal{C}_h|$ times without replacement and the time at which the first red ball emerges is marked as the terminating list rank. Since every ordering of ball picking is equiprobable and is bijective with $\mathcal{A}_l$, the expected list rank in ball picking problem is equal to $\mathsf{E}_{\mathcal{C}'}[L(\boldsymbol{y},\mathcal{C}')|\boldsymbol{y}]$. Hence,

$$\begin{aligned}\mathsf{E}_{\mathcal{C}'}[L(\boldsymbol{y},\mathcal{C}')|\boldsymbol{y}]&=\sum_{s=1}^{|\mathcal{C}_h|-|\mathcal{C}'|+1}s\frac{\binom{|\mathcal{C}_h|-s}{|\mathcal{C}'|-1}}{\binom{|\mathcal{C}_h|}{|\mathcal{C}'|}} &(87)\\&=\frac{|\mathcal{C}_h|+1}{|\mathcal{C}'|+1} &(88)\\&\le 2^m,\end{aligned}$$

where (88) follows from a variant of the Chu-Vandermonde identity.

Finally, substituting (85) into (84) proves Theorem 8. ■

In (67), it is shown that $\mathsf{E}[L]$ can be fully characterized by its conditional expectation $\mathsf{E}[L|W=w,\boldsymbol{X}=\bar{\boldsymbol{x}}]$. For a given w and A, let $\bar{\boldsymbol{x}}_e=\bar{\boldsymbol{x}}/A$ be the transmitted point with unit amplitude per dimension. Then it can be shown that

$$\mathsf{E}[L|W=w,\boldsymbol{X}=\bar{\boldsymbol{x}}]=\mathsf{E}[L|W=\eta,\boldsymbol{X}=\bar{\boldsymbol{x}}_e], \tag{89}$$

where $\eta\triangleq w/A$ is called the *normalized norm*. Hence, it suffices to compute $\mathsf{E}[L|W=\eta,\boldsymbol{X}=\bar{\boldsymbol{x}}_e]$. The SNR (equivalently, the BPSK amplitude A) only exhibits a scaling effect. To evaluate $\mathsf{E}[L|W=\eta,\boldsymbol{X}=\bar{\boldsymbol{x}}_e]$, let $\mathcal{C}_l^-\triangleq\mathcal{C}_l\setminus\{\bar{\boldsymbol{c}}\}$ and define the conditional probability of UE conditioned on the sphere $\mathcal{B}(\bar{\boldsymbol{x}}_e,\eta)$ as

$$P_{e,\lambda}(\eta)\triangleq\sum_{\boldsymbol{c}\in\mathcal{C}_l^-}\mathsf{P}(\mathcal{Y}(\boldsymbol{c})|W=\eta,\boldsymbol{X}=\bar{\boldsymbol{x}}_e). \tag{90}$$

In general, it is difficult to know the conditional probability of UE $P_{e,\lambda}(\eta)$. Assuming the knowledge of parametric information $P_{e,\lambda}(\eta)$, we first show an approximation that represents $\mathsf{E}[L|W=\eta,\boldsymbol{X}=\bar{\boldsymbol{x}}_e]$ as a linear combination between $L=1$ and $L=\bar{L}$ with coefficient given by $P_{e,\lambda}(\eta)$.

Approximation 3 (Parametric approximation). *For a CRC-aided convolutional code with corresponding parameters of $\bar{L}$ and $P_{e,\lambda}(\eta)$, where $\bar{L}\triangleq\mathsf{E}[L|\boldsymbol{X}=\boldsymbol{O}]$,*

$$\mathsf{E}[L|W=\eta,\boldsymbol{X}=\bar{\boldsymbol{x}}_e]\approx 1-P_{e,\lambda}(\eta)+P_{e,\lambda}(\eta)\bar{L}. \tag{91}$$

Furthermore, averaging over $W=\eta$ on both sides of (91) *yields the approximation of $\mathsf{E}[L]$, i.e.,*

$$\mathsf{E}[L]\approx 1-P_{e,\lambda}+P_{e,\lambda}\mathsf{E}[L|\boldsymbol{X}=\boldsymbol{O}]. \tag{92}$$

Justification: For ease of discussion, we use the shorthand notation $\mathsf{P}(\cdot|\eta,\bar{\boldsymbol{x}}_e)\triangleq\mathsf{P}(\cdot|W=\eta,\boldsymbol{X}=\bar{\boldsymbol{x}}_e)$ and $\mathsf{P}(\cdot|\boldsymbol{O})=\mathsf{P}(\cdot|\boldsymbol{X}=\boldsymbol{O})$. Let us consider η for which

$P_{e,\lambda}(\eta) > 0$. Hence,

$$
\begin{align*}
&\mathsf{E}[L|W=\eta,\boldsymbol{X}=\bar{\boldsymbol{x}}_e]\\
=&\sum_{s=1}^{\lambda} s\mathsf{P}(L=s|\eta,\bar{\boldsymbol{x}}_e)\\
=&\mathsf{P}(\mathcal{Y}(\bar{\boldsymbol{c}})|\eta,\bar{\boldsymbol{x}}_e)+\sum_{s=1}^{\lambda}s\mathsf{P}(L=s|\eta,\bar{\boldsymbol{x}}_e)-\sum_{s=1}^{\lambda}\mathsf{P}(\mathcal{Z}_s(\bar{\boldsymbol{c}})|\eta,\bar{\boldsymbol{x}}_e)\\
\geq&1-P_{e,\lambda}(\eta)+\sum_{s=1}^{\lambda}s\big(\mathsf{P}(L=s|\eta,\bar{\boldsymbol{x}}_e)-\mathsf{P}(\mathcal{Z}_s(\bar{\boldsymbol{c}})|\eta,\bar{\boldsymbol{x}}_e)\big)\\
=&1-P_{e,\lambda}(\eta)+P_{e,\lambda}(\eta)\left(\sum_{s=1}^{\lambda}s\frac{\sum_{\boldsymbol{c}\in\mathcal{C}_l^-}\mathsf{P}(\mathcal{Z}_s(\boldsymbol{c})|\eta,\bar{\boldsymbol{x}}_e)}{\sum_{\boldsymbol{c}\in\mathcal{C}_l^-}\mathsf{P}(\mathcal{Y}(\boldsymbol{c})|\eta,\bar{\boldsymbol{x}}_e)}\right) \tag{93}\\
\approx&1-P_{e,\lambda}(\eta)+P_{e,\lambda}(\eta)\left(\sum_{s=1}^{\lambda}s\mathsf{P}(L=s|\boldsymbol{O})\right) \tag{94}\\
=&1-P_{e,\lambda}(\eta)+P_{e,\lambda}(\eta)\bar{L},
\end{align*}
$$

where (94) follows from the substitution below. Consider the conditional list rank distribution

$$
\boldsymbol{P}_\eta=\Big(\frac{\sum_{\boldsymbol{c}\in\mathcal{C}_l^-}\mathsf{P}(\mathcal{Z}_1(\boldsymbol{c})|\eta,\bar{\boldsymbol{x}}_e)}{\sum_{\boldsymbol{c}\in\mathcal{C}_l^-}\mathsf{P}(\mathcal{Y}(\boldsymbol{c})|\eta,\bar{\boldsymbol{x}}_e)},\dots,\frac{\sum_{\boldsymbol{c}\in\mathcal{C}_l^-}\mathsf{P}(\mathcal{Z}_\lambda(\boldsymbol{c})|\eta,\bar{\boldsymbol{x}}_e)}{\sum_{\boldsymbol{c}\in\mathcal{C}_l^-}\mathsf{P}(\mathcal{Y}(\boldsymbol{c})|\eta,\bar{\boldsymbol{x}}_e)}\Big). \tag{95}
$$

Using the fact that $\lim_{\eta\to\infty}g_\eta(\boldsymbol{y}_p)=1/S_{n-1}(\sqrt{n})$, the conditional list rank distribution $\boldsymbol{P}_\eta$ will converge to

$$
\begin{align*}
\boldsymbol{P}_\infty&=\left(\frac{\mathsf{P}(\mathcal{Z}_1(\boldsymbol{c})|\boldsymbol{O})}{\mathsf{P}(\mathcal{Y}(\boldsymbol{c})|\boldsymbol{O})},\dots,\frac{\mathsf{P}(\mathcal{Z}_\lambda(\boldsymbol{c})|\boldsymbol{O})}{\mathsf{P}(\mathcal{Y}(\boldsymbol{c})|\boldsymbol{O})}\right) \tag{96}\\
&=\left(\frac{\sum_{\boldsymbol{c}\in\mathcal{C}_l}\mathsf{P}(\mathcal{Z}_1(\boldsymbol{c})|\boldsymbol{O})}{\sum_{\boldsymbol{c}\in\mathcal{C}_l}\mathsf{P}(\mathcal{Y}(\boldsymbol{c})|\boldsymbol{O})},\dots,\frac{\sum_{\boldsymbol{c}\in\mathcal{C}_l}\mathsf{P}(\mathcal{Z}_\lambda(\boldsymbol{c})|\boldsymbol{O})}{\sum_{\boldsymbol{c}\in\mathcal{C}_l}\mathsf{P}(\mathcal{Y}(\boldsymbol{c})|\boldsymbol{O})}\right)\\
&=(\mathsf{P}(L=1|\boldsymbol{O}),\dots,\mathsf{P}(L=\lambda|\boldsymbol{O})), \tag{97}
\end{align*}
$$

where $\boldsymbol{c}$ is any lower-rate codeword in (96). Hence, we directly replace $\boldsymbol{P}_\eta$ with the limit distribution $\boldsymbol{P}_\infty$ in (93). Finally, averaging over $W=\eta$ on both sides of (91) yields (92). ■

Fig. 6 shows the simulation results of the conditional expected list rank $\mathsf{E}[L|W=\eta,\boldsymbol{X}=\bar{\boldsymbol{x}}_e]$ vs. the normalized norm η for CRC-ZTCCs with various information lengths. The corresponding parametric approximation is also provided. We see that the parametric approximation exhibits a remarkable accuracy that improves as k increases. Observe that for large values of k, the convergent $\mathsf{E}[L|W=\eta,\boldsymbol{X}=\bar{\boldsymbol{x}}_e]$ is close to 2^m.

Using (67) and (89), we can produce $\mathsf{E}[L]$ as a function of SNR γ_s. Fig. 7 shows $\mathsf{E}[L]$ vs. SNR along with its parametric approximations for ZTCC $(13,17)$ and various DSO CRC polynomials of degree $m=3,4,\dots,6$. We see that the parametric approximation on $\mathsf{E}[L]$ remains extremely tight.

The parametric approximation provides a practically useful quantitative connection between performance and complexity. Specifically, for CRC-ZTCCs with a target probability of UE $P_{e,\lambda}^*$ and $\bar{L}\approx 2^m$ for CRC degree m, (92) implies that a CRC with degree $m\leq -\log(P_{e,\lambda}^*)$ is sufficient to maintain $\mathsf{E}[L]\leq 2$, which ensures that the average complexity for SLVD to achieve $P_{e,\lambda}^*$ is at most one more traceback than the standard Viterbi decoding.

As an alternative to Approximation 3, we provide a higher-order approximation formula for a good CRC-aided convolutional code that only requires knowledge of $\mathsf{E}[L|\boldsymbol{X}=\boldsymbol{O}]$. This alternative approximation is motivated by Shannon's observation [19] that an optimal (n,M) code places its codewords on the surface of a sphere such that the total solid angle Ω_0 is evenly divided among the M Voronoi regions, one for each codeword, and that each Voronoi region is a circular cone. Hence, if the CRC-aided convolutional code is good enough, the union of order-1 to order-μ decoding regions $\mathcal{Z}_s(\boldsymbol{c})$ for a lower-rate codeword $\boldsymbol{c}\in\mathcal{C}_l$ should resemble circular cones, where μ is a parameter to be optimized. From this perspective, we propose the *onion model* for the order-1 decoding region to the order-μ decoding region based on the following assumptions.

1) The union $\bigcup_{i=1}^{s}\mathcal{Z}_i(\boldsymbol{c})$ of order-1 to order-s decoding regions, $1\leq s\leq\mu$, is a circular cone with half-angle α_s. This implies that each order-s decoding region, $2\leq s\leq\mu$ is an *annulus* in between two circular cones.
2) The solid angle $\Omega(\alpha_s)$ of $\bigcup_{i=1}^{s}\mathcal{Z}_i(\boldsymbol{c})$ is equal to $\frac{s}{2^{k+m}}\Omega_0$, $1\leq s\leq\mu$, where Ω_0 is the total solid angle (i.e., the area of a unit sphere in $\mathbb{R}^n$).
3) The conditional expected list rank beyond $\bigcup_{i=1}^{\mu}\mathcal{Z}_i(\bar{\boldsymbol{c}})$ is equal to $\bar{L}$ (i.e., $\mathsf{E}[L|\boldsymbol{X}=\boldsymbol{O}]$).

Approximation 4 (Higher-order approximation)**.** *For a given CRC-aided convolutional code, let $\bar{L}=\mathsf{E}[L|\boldsymbol{X}=\boldsymbol{O}]$. With the onion model assumptions and parameter μ, $\mu\in\mathbb{N}$, $\mathsf{E}[L|W=\eta,\boldsymbol{X}=\bar{\boldsymbol{x}}_e]$ is approximated by*

$$
\begin{align*}
&\mathsf{E}[L|W=\eta,\boldsymbol{X}=\bar{\boldsymbol{x}}_e]\\
&\approx\begin{cases}1, & \textit{if } \eta<\sqrt{n}\sin\alpha_1\\ \cdots\\ s-\sum_{i=1}^{s-1}F_{\bar{\boldsymbol{x}}_e}(i), & \textit{if } \sqrt{n}\sin\alpha_{s-1}\leq\eta<\sqrt{n}\sin\alpha_s\\ \cdots\\ \bar{L}-(\bar{L}-\mu)F_{\bar{\boldsymbol{x}}_e}(\mu)-\sum_{i=1}^{\mu-1}F_{\bar{\boldsymbol{x}}_e}(i), & \textit{if } \eta\geq\sqrt{n}\sin\alpha_\mu,\end{cases} \tag{98}
\end{align*}
$$

where assuming $\eta\geq\sqrt{n}\sin\alpha_s$,

$$
\begin{align*}
F_{\bar{\boldsymbol{x}}_e}(s)=&\frac{\Gamma\big(\frac{n}{2}\big)}{\sqrt{\pi}\Gamma\big(\frac{n-1}{2}\big)}\\
&\cdot\left(\int_0^{\beta_{s,1}}\sin^{n-2}\theta\,\mathrm{d}\theta+\int_0^{\beta_{s,2}}\sin^{n-2}\theta\,\mathrm{d}\theta\right), \tag{99}\\
\beta_{s,1}=&\frac{\pi}{2}+\alpha_s-\arcsin\left(\frac{\sqrt{\eta^2-n\sin^2\alpha_s}}{\eta}\right), \tag{100}\\
\beta_{s,2}=&\left(\frac{\pi}{2}-\alpha_s-\arcsin\left(\frac{\sqrt{\eta^2-n\sin^2\alpha_s}}{\eta}\right)\right)\mathbf{1}_{\{\eta\leq\sqrt{n}\}}, \tag{101}
\end{align*}
$$

and α_s is the half-angle for which

$$
\frac{\Omega(\alpha_s)}{\Omega_0}=\frac{\Gamma\big(\frac{n}{2}\big)}{\sqrt{\pi}\Gamma\big(\frac{n-1}{2}\big)}\int_0^{\alpha_s}\sin^{n-2}\theta\,\mathrm{d}\theta=\frac{s}{2^{k+m}}. \tag{102}
$$

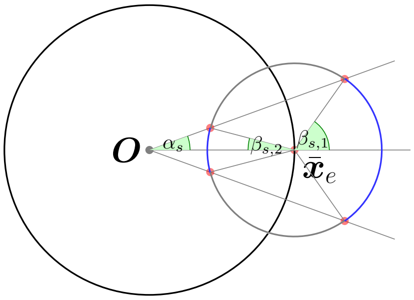

Fig. 8. The geometry of the cumulative probability function $F_{\bar{\boldsymbol{x}}_e}(s)$, assuming that $\sqrt{n}\sin\alpha_s \le \eta \le \sqrt{n}$.

Justification: The onion model assumptions implies that each higher order decoding region $\mathcal{Z}_s(\boldsymbol{c})$, $2 \le s \le \mu$, is an annulus in between two circular cones. Hence, $\mathsf{P}(L = s|W = \eta, \boldsymbol{X} = \bar{\boldsymbol{x}}_e)$ is simply the spherical area of $\mathcal{B}(\bar{\boldsymbol{x}}_e, \eta)$ cut out by the annulus. To evaluate this quantity, consider the cumulative probability function of $L = s$,

$$F_{\bar{\boldsymbol{x}}_e}(s) \triangleq \mathsf{P}(L \le s, \boldsymbol{X} = \bar{\boldsymbol{x}}_e). \tag{103}$$

Thus,

$$\mathsf{P}(L = s|W = \eta, \boldsymbol{X} = \bar{\boldsymbol{x}}_e) = F_{\bar{\boldsymbol{x}}_e}(s) - F_{\bar{\boldsymbol{x}}_e}(s-1). \tag{104}$$

By the onion model assumptions, for $\eta \ge \sqrt{n}\sin\alpha_\mu$,

$$\begin{aligned}
&\mathsf{E}[L|W = \eta, \boldsymbol{X} = \bar{\boldsymbol{x}}_e] && (105)\\
\approx &\sum_{i=1}^{\mu} i(F_{\bar{\boldsymbol{x}}_e}(i) - F_{\bar{\boldsymbol{x}}_e}(i-1)) + \bar{L}(1 - F_{\bar{\boldsymbol{x}}_e}(\mu)) && (106)\\
= &\bar{L} - (\bar{L} - \mu)F_{\bar{\boldsymbol{x}}_e}(\mu) - \sum_{i=1}^{\mu-1} F_{\bar{\boldsymbol{x}}_e}(i). && (107)
\end{aligned}$$

In the similar fashion, for $\sqrt{n}\sin\alpha_{s-1} \le \eta < \sqrt{n}\sin\alpha_s$, $1 \le s \le \mu$,

$$\mathsf{E}[L|W = \eta, \boldsymbol{X} = \bar{\boldsymbol{x}}_e] \approx s - \sum_{i=1}^{s-1} F_{\bar{\boldsymbol{x}}_e}(i). \tag{108}$$

Next, we derive the cumulative probability function $F_{\bar{\boldsymbol{x}}_e}(s)$. Geometrically, $F_{\bar{\boldsymbol{x}}_e}(s)$ is the fraction of the spherical area of $\mathcal{B}(\bar{\boldsymbol{x}}_e, \eta)$ cut out by the circular cone $\bigcup_{i=1}^{s} \mathcal{Z}_s(\bar{\boldsymbol{c}})$ with half-angle α_s to the total noise spherical area. Assume that $\sqrt{n}\sin\alpha_s \le \eta \le \sqrt{n}$. Fig. 8 shows the side view of this scenario in $\mathbb{R}^3$, in which the blue arc represents the spherical area contained in $\bigcup_{i=1}^{s} \mathcal{Z}_s(\bar{\boldsymbol{c}})$. It can be seen that α_s will induce two possible half-angles $\beta_{s,1}$ and $\beta_{s,2}$. By law of cosines,

$$\begin{aligned}
\beta &= \frac{\pi}{2} \pm \alpha_s - \arcsin\left(\frac{r_2 - r_1}{2\eta}\right) && (109)\\
&= \frac{\pi}{2} \pm \alpha_s - \arcsin\left(\frac{\sqrt{\eta^2 - n\sin^2\alpha_s}}{\eta}\right), && (110)
\end{aligned}$$

where r_1, r_2 are solutions to

$$r^2 - (2\sqrt{n}\cos\alpha_s)r + (n - \eta^2) = 0. \tag{111}$$

The induced half-angle β becomes unique once $\eta > \sqrt{n}$.

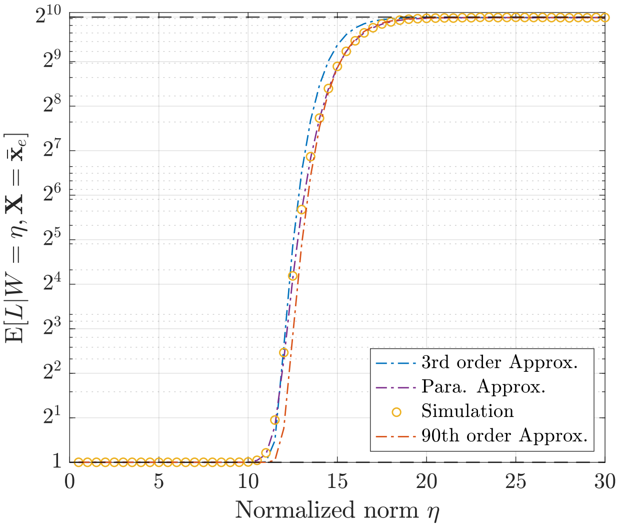

Fig. 9. The parametric and higher-order approximations of $\mathsf{E}[L|W = \eta, \boldsymbol{X} = \bar{\boldsymbol{x}}_e]$ for ZTCC $(561, 753)$ used with the degree-10 DSO CRC polynomial 0x4CF at $k = 64$. Both higher-order approximations assume the knowledge of $\bar{L} = 1017$.

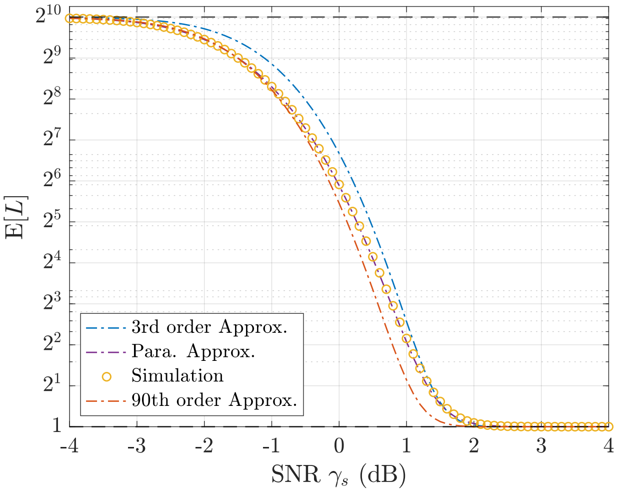

Fig. 10. The expected list rank $\mathsf{E}[L]$ vs. SNR via (67) and (89) for ZTCC $(561, 753)$, degree-10 DSO CRC polynomial 0x4CF at $k = 64$.

From [19, Eq. (21)], the solid angle $\Omega(\alpha)$ of a circular cone with center $\boldsymbol{O}$ and half-angle α in n-dimensional Euclidean space is given by

$$\Omega(\alpha) = \frac{2\pi^{\frac{n-1}{2}}}{\Gamma\left(\frac{n-1}{2}\right)} \int_0^\alpha \sin^{n-2}\theta \,\mathrm{d}\theta. \tag{112}$$

The total solid angle Ω_0 in n-dimensional Euclidean space is given by

$$\Omega_0 = \frac{2\pi^{\frac{n}{2}}}{\Gamma\left(\frac{n}{2}\right)}. \tag{113}$$

Thus, using (112), (113), we can solve α_s from assumption 2 of the onion model. Namely, α_s is the solution to

$$\frac{\Omega(\alpha)}{\Omega_0} = \frac{\Gamma\left(\frac{n}{2}\right)}{\sqrt{\pi}\Gamma\left(\frac{n-1}{2}\right)} \int_0^\alpha \sin^{n-2}\theta \,\mathrm{d}\theta = \frac{s}{2^{k+m}}. \tag{114}$$

By geometry in Fig. 8, $F_{\bar{\boldsymbol{x}}_e}(s)$ in (103) is given by

$$F_{\bar{\boldsymbol{x}}_e}(s) = \frac{\Omega(\beta_{s,1}) + \Omega(\beta_{s,2})}{\Omega_0}. \tag{115}$$

This concludes the justification of Approximation 4. ■

To demonstrate the tightness of the proposed approximation for good enough CRC-aided convolutional codes, Fig. 9 shows the approximations of $\mathsf{E}[L|W = \eta, \boldsymbol{X} = \bar{\boldsymbol{x}}_e]$ for ZTCC $(561, 753)$ used with the degree-10 DSO CRC polynomial 0x4CF at $k = 64$ with $\mu = 3$ and 90. This concatenated code has a minimum distance $d^l_{\text{min}} = 20$ and thus can be deemed as good enough. When $\mu = 3$, our approximation accurately gives the smaller values of the actual conditional expected list rank. As μ increases, the accuracy of the approximation will shift towards large values of conditional expected list rank. Fig. 10 illustrates the approximation of $\mathsf{E}[L]$ vs. SNR via (67) and (89). The 3rd-order and 90-th order approximations still behave in the similar fashion as in Fig. 9.

C. Complexity Analysis

There are a variety of implementations of list decoding of convolutional codes as described in, e.g., [53]–[57]. In this paper, the SLVD implementation maintains a list of path metric differences by using a red-black tree as described in [55], which provides the fastest runtime we found among the data structures that support full floating-point precision. The literature mentioned above also analyzed the number of bit operations or the asymptotic complexity of the algorithms presented, but those complexity metrics are not directly connected with actual runtime. To explore how the additional complexity of SLVD of CRC-ZTCCs relative to the standard soft Viterbi (SSV) decoding, we develop an average complexity expression that closely approximates our empirical runtimes.

For our specific implementation, three components comprise the average complexity of SLVD, given by

$$C_{\text{SLVD}} = C_{\text{SSV}} + C_{\text{trace}} + C_{\text{list}}. \tag{116}$$

The first component C_{SSV} is the complexity required to perform the add-compare-select (ACS) operations on the trellis of the given convolutional code and perform the initial traceback associated with SSV. Specifically, for CRC-ZTCCs, this quantity is given by

$$\begin{aligned} C_{\text{SSV}} = (2^{\nu+1} - 2) + 1.5(2^{\nu+1} - 2) + 1.5(k + m - \nu)2^{\nu+1} \\ + c_1[2(k + m + \nu) + 1.5(k + m)]. \end{aligned} \tag{117}$$

For CRC-TBCCs, this quantity is given by

$$C_{\text{SSV}} = 1.5(k + m)2^{\nu+1} + 2^{\nu} + 3.5c_1(k + m). \tag{118}$$

In order to measure the decoding complexity, define 1 unit of complexity as the complexity required by performing one addition. In (117) and (118), we assign 1 unit of complexity to each addition per branch and 0.5 units of complexity to each compare-select operation per branch. In the first and second terms of (117), $(2^{\nu+1} - 2)$ counts the number of edges in the initial ν sections and the final ν termination sections of a ZT trellis. In the third term of (117), $(k+m-\nu)2^{\nu+1}$ counts the number of edges in the middle $(k + m - \nu)$ sections of a ZT trellis. The fourth term in (117) approximates the complexity of the traceback operation, assigning 2 units of complexity for accessing the parent node per trellis stage and 1.5 units of complexity per codeword symbol for the CRC verification on the decoded sequence $\hat{\boldsymbol{v}}$. In (118), the second term is because it takes 2^{ν} operations to identify the optimal termination state with minimum metric before the first traceback.

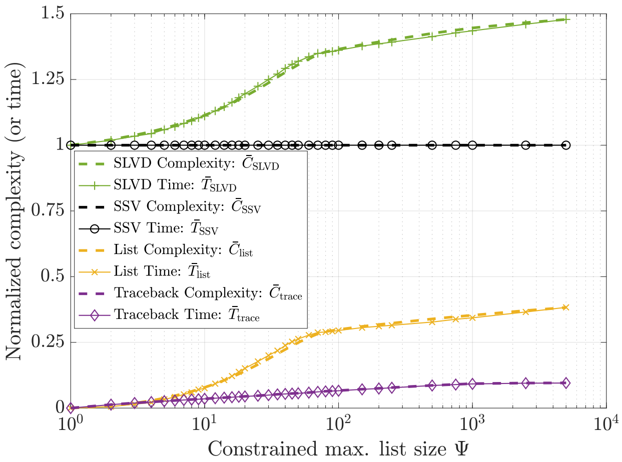

Fig. 11. The complexity of SLVD with different constrained maximum list sizes for ZTCC $(27, 31)$, and degree-10 DSO CRC polynomial 0x709, with $k = 64$ at SNR $\gamma_s = 2$ dB. All variables are normalized by the time or complexity of the SSV algorithm. In the simulation, $c_1 = 1.5$ and $c_2 = 2.2$.

The second component C_{trace} represents the complexity of the *additional* traceback operations required by SLVD. Specifically, for a given CRC-ZTCC,

$$C_{\text{trace}} = c_1(\mathsf{E}[L] - 1)[2(k + m + \nu) + 1.5(k + m)]. \tag{119}$$

For CRC-TBCCs,

$$C_{\text{trace}} = 3.5c_1(\mathsf{E}[L] - 1)(k + m). \tag{120}$$

The third component C_{list} represents the average complexity of inserting new elements to maintain an ordered list of path metric differences. For both CRC-ZTCCs and CRC-TBCCs,

$$C_{\text{list}} = c_2\mathsf{E}[I]\log(\mathsf{E}[I]), \tag{121}$$

where $\mathsf{E}[I]$ is the expected number of insertions to maintain the sorted list of path metric differences. According to the mechanism of insertion, for CRC-ZTCCs,

$$\mathsf{E}[I] \leq (k + m)\mathsf{E}[L], \tag{122}$$

and for CRC-TBCCs,

$$\mathsf{E}[I] \leq (k + m)\mathsf{E}[L] + 2^{\nu} - 1, \tag{123}$$

where $2^{\nu} - 1$ denotes the number of path metric differences between the optimal terminating state and any other of the $2^{\nu} - 1$ terminating states.

In (117), (118), (119), (120), and (121) the constants c_1 and c_2 characterize implementation-specific differences in the implemented complexity of traceback and list insertion, respectively, as compared to the ACS operations of Viterbi decoding. For our implementation, we found $c_1 = 1.5$ and $c_2 = 2.2$.

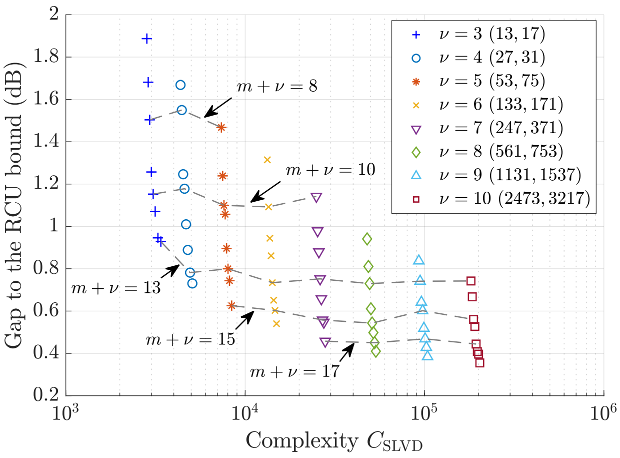

Fig. 12. The SNR gap to the RCU bound vs. the average complexity of SLVD for the family of CRC-ZTCCs in Table II at target $P_{e,\lambda} = 10^{-4}$. Each color represents a specific ZTCC shown in parenthesis. Markers from top to bottom with the same color correspond to the DSO CRC polynomials with $m = 3, 4, \ldots, 10$ in Table II. The information length and blocklength are given by $k = 64$ and $n = 2(64 + m + \nu)$, respectively.

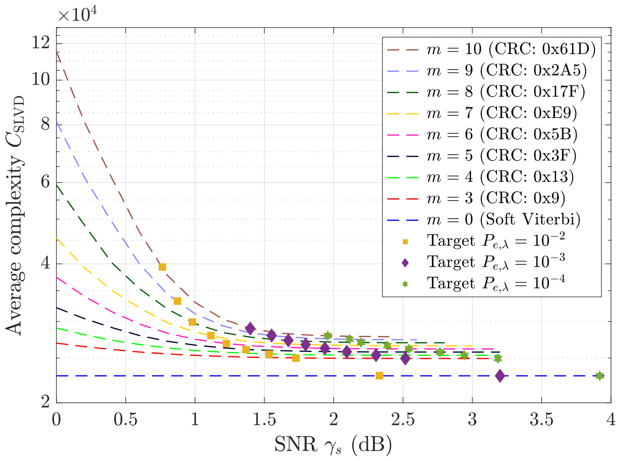

Fig. 13. The average complexity vs. SNR for ZTCC $(247, 371)$ used with its DSO CRC polynomials. The ZTCC with no CRC using soft Viterbi decoding is also given as a reference.

The additional complexity of the SLVD over SSV decoding is completely characterized by the additional tracebacks along the trellis and the maintenance of an ordered list of path metric differences. We define the *normalized complexity* $\bar{C}_{\text{SLVD}}$ as the complexity of SLVD divided by the complexity of SSV decoding, i.e.,

$$\bar{C}_{\text{SLVD}} = \frac{C_{\text{SLVD}}}{C_{\text{SSV}}} = 1 + \bar{C}_{\text{trace}} + \bar{C}_{\text{list}}. \tag{124}$$

The normalized complexity provides a measure for the additional complexity of operations associated with the SLVD relative to the complexity of the SSV algorithm.

We recorded the runtime T_{SLVD}, T_{SSV}, T_{trace}, and T_{list} on an Intel i7-4720HQ using Visual C++. We then divided all of these terms by T_{SSV} to compute a normalized runtime $\bar{T}$. Fig. 11 shows normalized complexity based on equation (124) and normalized runtime. In both cases, the normalization is computed by dividing by the complexity or run-time associated with SSV, i.e., performing all ACS operations on the trellis and a traceback from the state with the best metric. The normalized complexity and normalized runtime curves are indistinguishable. Fig. 11 also shows that the additional complexity of SLVD is primarily from maintaining an ordered list of path metric differences.

V. Simulation Results

In this section, we present our simulation results of CRC-ZTCCs in Table II and CRC-TBCCs in Table III for $k = 64$. Finally, we compare the $(128, 64)$ punctured CRC-TBCC designed in our precursor conference paper [2] with several $(128, 64)$ short blocklength codes presented in [7].

A. Simulation Results for CRC-ZTCCs

Fig. 12 shows the trade-off between the SNR gap to the RCU bound and the average decoding complexity computed using (116) for target probability of UE $P_{e,\lambda} = 10^{-4}$. It is shown that for a given ZTCC, increasing the degree m of DSO CRC polynomials can significantly diminish the SNR gap to the RCU bound at a relatively small complexity increase. This SNR gap reduction is especially considerable when ν is small and becomes less significant as ν becomes large. For all ZTCCs, the complexity cost of increasing m from 3 to 10 is within a factor of 2. This is consistent with Fig. 11 in which the complexity increases by a factor less than 1.5 even for a very large constrained maximum list size Ψ.

A CRC-ZTCC could be decoded using Viterbi alone, without list decoding, on a trellis with $2^{m+\nu}$ states per trellis stage. The dashed lines in Fig. 12 show that the gap to the RCU bound remains roughly constant for a constant value of $m+\nu$. However, list decoding with a well chosen (m, ν) pair achieves this performance with a minimum complexity C_{SLVD}. Thus, for a given target $P_{e,\lambda}$ and a fixed value of $m + \nu$, the inclusion of CRC-aided list decoding will generally reduce complexity over using Viterbi decoding alone on a convolutional code with $2^{m+\nu}$ states per trellis stage.

Fig. 13 shows the complexity C_{SLVD} computed using (116) as a function of SNR for ZTCC $(247, 371)$ and its DSO CRC polynomials with degree m from 3 to 10 from Table II. The ZTCC using soft Viterbi decoding with no CRC is also shown. Here, the target probabilities of UE at $10^{-2}, 10^{-3}, 10^{-4}$ for each CRC-ZTCC are marked by squares, diamonds, and stars, respectively. For each target probability of UE, the corresponding complexity is within a factor of 2 compared to the soft Viterbi decoding of ZTCC $(247, 371)$.

The termination overhead associated with ZTCC induces a gap from the RCU bound, which can be closed by using the corresponding TBCC as we will see below.

B. Simulation Results for CRC-TBCCs

In Section II we use the fact that for a CRC-ZTCC, each SLVD operation yields a valid higher-rate codeword, i.e., a ZT codeword. However, for a CRC-TBCC, SLVD operations do not always yield a valid higher rate codeword, i.e., a TB codeword, because the TB condition is often not met. Because

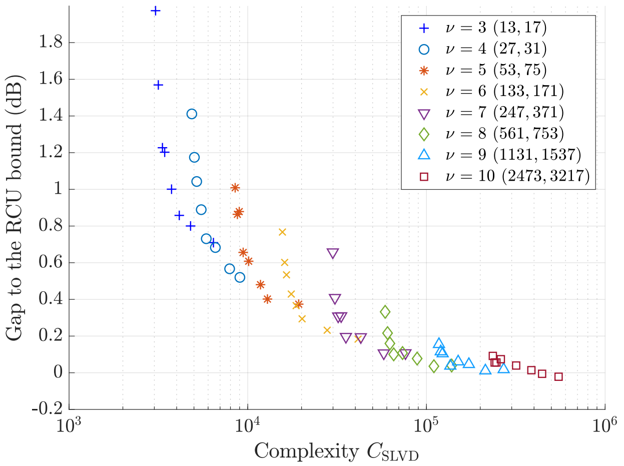

Fig. 14. The SNR gap to the RCU bound vs. the average complexity of SLVD for the family of CRC-TBCCs in Table III at target $P_{e,\lambda} = 10^{-4}$. Each color represents a specific TBCC shown in parenthesis. Markers from top to bottom with the same color correspond to the DSO CRC polynomials with $m = 3, 4, \ldots, 10$ in Table III. The information length and blocklength are given by $k = 64$ and $n = 2(64 + m)$, respectively.

of this, we can no longer assume that $\bar{L} \approx 2^m$. Nevertheless, Approximations 3 and 4 still apply for an accurate value of $\bar{L}$ which can be obtained from simulation.

The increased value of $\bar{L}$ may be understood by considering the higher-rate code $\mathcal{C}_h$ to be the pseudo code represented by all paths on the trellis regardless of whether they meet the TB condition. Due to the additional complexity required to check the TB condition, $\mathsf{E}[I]$ is significantly increased compared to the CRC-ZTCC. While we identified the empirical value of $\mathsf{E}[I]$ for CRC-ZTCCs, in this section we simply assume $\mathsf{E}[I]$ attains the upper bound in (123) for CRC-TBCCs. Hence, using (118), (120) with $c_1 = 1.5$, (121) with $c_2 = 2.2$, together with the aforementioned assumption on $\mathsf{E}[I]$, we can compute an estimate of the average complexity C_{SLVD} of our implementation of SLVD of CRC-TBCCs.

Fig. 14 shows the SNR gap to the RCU bound vs. the average complexity for target probability of UE $P_{e,\lambda} = 10^{-4}$ for all CRC-TBCCs designed in Table III. Compared to Fig. 12, TB encoding significantly reduces the SNR gap to the RCU, because the overhead of termination is avoided. However, this reduction of the gap comes at the expense of a slight increase in average complexity for checking the TB condition. Note the exciting result that some CRC-TBCCs outperform the RCU bound for $\nu = 9$ and 10. Another phenomenon distinct from CRC-ZTCCs is that for TBCCs with large ν, increasing the DSO CRC polynomial degree from $m = 3$ to 10 only provides a small benefit. Note, however, that the degree-3 DSO CRC polynomial does provide a benefit over a TBCC used with no CRC at all.

To illustrate the performance of the best CRC-TBCCs designed in Table III, we select $\nu = 9$ and $\nu = 10$ TBCCs as an example. Fig. 15 shows two cases: $R = 64/134$ corresponding to $m = 3$ and $R = 64/146$ corresponding to $m = 9$. The MC bound and the RCU bounds for these rates are plotted using the saddlepoint approximations provided in Approximations 1 and 2, respectively. We see that in these two cases, the CRC-TBCCs in Fig. 15 beat the RCU bound at low SNR values. However, this superiority gradually fades away as SNR increases, although for $m = 9$, the performance is very close to the RCU bound even at $P_{e,\lambda} = 10^{-5}$. Simulations also suggest that it is extremely difficult to further improve the code performance once beyond the RCU bound at low probability of UE.

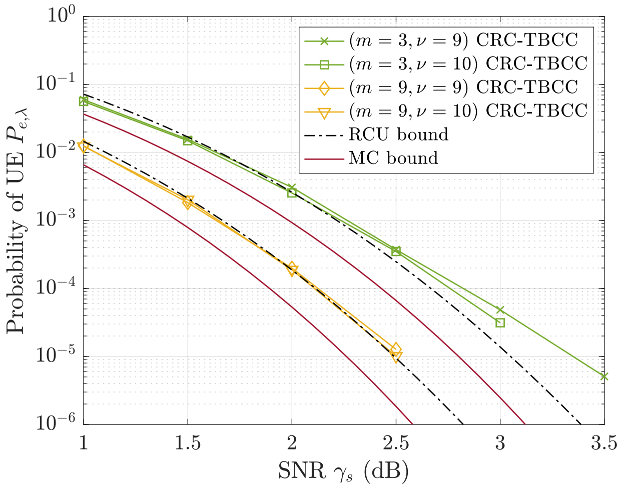

Fig. 15. Comparison between $P_{e,\lambda}$ and RCU and MC bounds at rates $R = 64/134$ $(m = 3)$ and $R = 64/146$ $(m = 9)$ for the CRC-TBCCs designed in Table III. For the sake of clarity, only $\nu = 9, 10$ TBCCs are displayed.

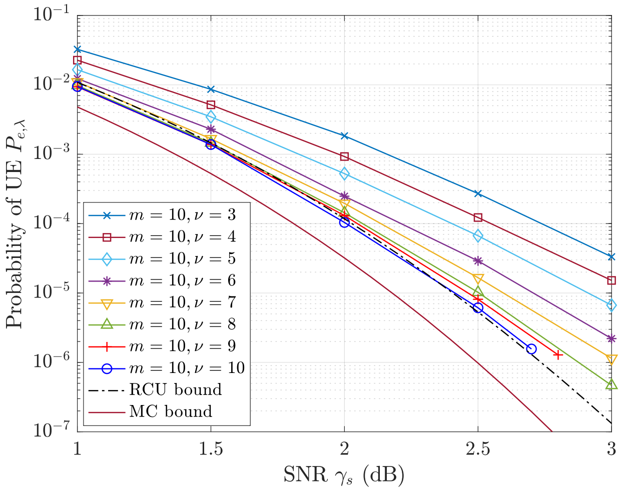

Fig. 16. Comparison between $P_{e,\lambda}$ and RCU and MC bounds at rate $R = 64/148$ (i.e., $m = 10$) for the CRC-TBCCs designed in Table III.

Fig. 16 shows the family of CRC-TBCCs with $k = 64$ and $n = 148$ (corresponding to $m = 10$). For small ν, we see a visible improvement as ν increases. However, once performance reaches the RCU bound, further increases in ν provide little benefit. For example with $m = 10$, the CRC-TBCC with $\nu = 9$ attains similar performance to that with $\nu = 10$.

C. Comparison of $(128, 64)$ *Linear Block Codes*

Direct comparison of CRC-TBCCs with other codes often requires puncturing to match rates. For simplicity, we have

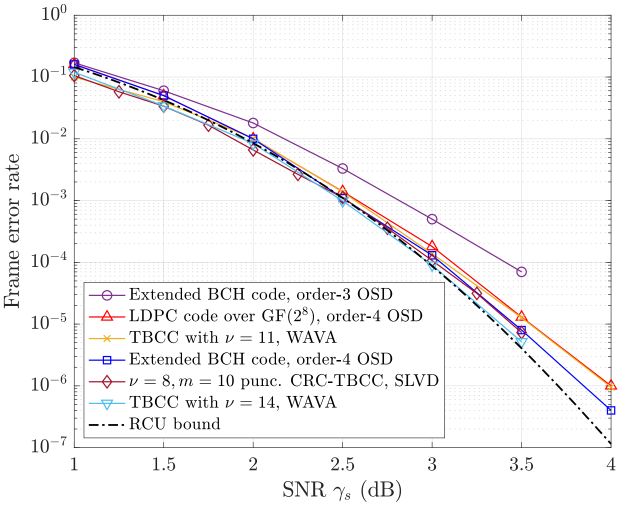

Fig. 17. Comparison of $(128, 64)$ linear block codes.

excluded puncturing from analysis in this paper. However, our precursor conference paper [2] designed a $v = 8$, $m = 10$ punctured CRC-TBCC with $k = 64$ and $n = 128$ whose FER performance can be directly compared to the $(128, 64)$ linear block codes presented in [7], as shown in Fig. 17.

At SNR of 3 dB, the $v = 8$, $m = 10$ punctured CRC-TBCC in [2] and the best codes studied in [7] all perform similarly. Specifically, the four codes in [7] with similar performance at 3 dB to the $v = 8$, $m = 10$ punctured CRC-TBCC are the following: the $v = 14$ and $v = 11$ TBCCs decoded with WAVA, the extended BCH code with order-4 OSD, and a non-binary LDPC code over $\mathbb{F}_{256}$ with order-4 OSD. As shown in Fig. 17, at higher SNR, the FER performance is more differentiated with the best performance provided by the $v = 14$ TBCC, slightly worse performance provided by the $v = 8$, $m = 10$ punctured CRC-TBCC and the extended BCH code with order-4 OSD and further degraded performance by the $v = 11$ TBCC and the non-binary LDPC code over $\mathbb{F}_{256}$ with order-4 OSD.

We now consider the decoding complexity of the three best codes described above at 3 dB, excluding the discussion of the non-binary LDPC code due to its further degraded performance. Actual complexity depends on specific implementation choices, here we consider the total number of computations per codeword as a way to give some flavor of the complexity differences between these approaches. At SNR of 3 dB, simulation shows that $\mathsf{E}[L] = 44.41$ for the $v = 8$, $m = 10$ punctured CRC-TBCC. Using (118), (120), (121) together with (123), we obtain $C_{\text{SLVD}} \leq 1.67 \times 10^5$.

In terms of WAVA complexity, let I be the number of iterations in WAVA. By assuming 0.5 units of complexity for compare/select operation per branch and 1 unit of complexity for one addition, the WAVA complexity for a rate-$1/\omega$ TBCC with ν memory elements at information length k is given by

$$C_{\text{WAVA}} = kI(0.5 \cdot 2^{\nu} + 2^{\nu+1}). \tag{125}$$

Using (125), the complexity of 3-round WAVA for $v = 11$ TBCC in [7] is 9.83×10^5, which is higher than for the $v = 8, m = 10$ punctured CRC-TBCC. The best $v = 14$ TBCC in [7] under 3-round WAVA achieves a complexity of 7.86×10^6.

A direct complexity comparison of SLVD with OSD is more difficult, but Table V in [8] indicates that at 3 dB, the order-3 OSD of the $(128, 64)$ extended BCH code requires 2.83×10^5 operations per codeword on average, which indicates that the order-4 OSD would likely have a higher complexity than the SLVD of $v = 8$, $m = 10$ punctured CRC-TBCC. Based on this analysis, the CRC-TBCC paradigm appears to be competitive with the existing approaches that provide similarly excellent FER performance at short blocklength.

VI. Conclusion

In this paper, we consider the CRC-aided convolutional code as a promising short blocklength code. The concatenated nature permits the use of SLVD that allows the code to attain the ML decoding performance at low complexity. For $k = 64$, we identified the DSO CRC polynomial for a family of ZTCCs and TBCCs generated with the optimum rate-$1/2$ convolutional encoders identified by [42] at sufficiently low target probability of UE. Several CRC-TBCCs beat the RCU bound at practically interesting values of SNR. In a recent work [58], Schiavone confirmed that the CRC-TBCC is indeed a powerful short blocklength code by showing that its performance matches the expurgated ensemble.

All CRC-aided convolutional codes considered in this paper are designed based on an optimum convolutional encoder. It would be interesting to investigate whether a suboptimal convolutional code used with the DSO CRC polynomial can also lead to a good concatenated code. Another interesting direction is to explore the performance of CRC-aided convolutional codes in the moderately short blocklength regime, e.g., $256 \leq n < 1000$. If puncturing is introduced in the code design, it remains open as to how to jointly design the puncturing pattern and the optimal CRC polynomial for a given convolutional code.

The beauty of SLVD lies in the fact that its average complexity is governed by its expected list rank $\mathsf{E}[L]$, a quantity that is inversely proportional to the SNR value. This allows a huge complexity reduction at interesting operating SNR values that guarantee a low target probability of UE. In particular, the parametric approximation of $\mathsf{E}[L]$ provides an explicit characterization of the performance-complexity trade-off. It shows that for CRC-ZTCCs with a target error probability $P^*_{e,\lambda}$ and $\bar{L} \approx 2^m$, a CRC degree $m \leq -\log(P^*_{e,\lambda})$ is sufficient to maintain $\mathsf{E}[L] \leq 2$. However, several problems are still open, for instance, how to upper bound $\mathsf{E}[L|\boldsymbol{X} = \boldsymbol{O}]$, and how to upper bound $P_{e,1}$ using the weight spectrum. In addition, the behavior of the supremum list rank λ is also less understood and is worth future investigation.

Appendix A
Derivation of the Induced Density Function

Let $\mathcal{B}(\boldsymbol{a}, r)$ denote the spherical surface of center $\boldsymbol{a} \in \mathbb{R}^n$ and radius r in $\mathbb{R}^n$. In this section, we derive the induced density function $g_w(\boldsymbol{y}_p)$ incurred when projecting a received point $\boldsymbol{y}$ uniformly distributed on $\mathcal{B}(\bar{\boldsymbol{x}}, w)$ to point $\boldsymbol{y}_p = (r/\|\boldsymbol{y}\|)\boldsymbol{y}$

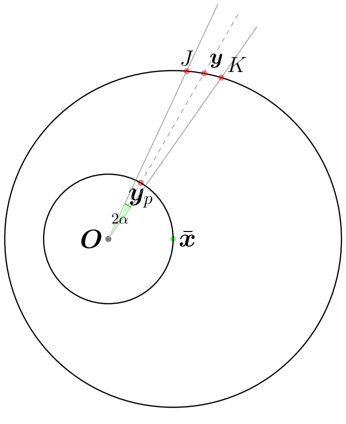

Fig. 18. Derivation of the induced density function $g_w(\boldsymbol{y_p})$ in $\mathbb{R}^n$.

that lies on the codeword sphere $\mathcal{B}(\boldsymbol{O}, A\sqrt{n})$ in $\mathbb{R}^n$. As an illustration, Fig. 18 depicts this scenario in $\mathbb{R}^2$. For our purposes, we assume that $w \geq A\sqrt{n}$ to ensure the bijective relationship between $\boldsymbol{y}$ and $\boldsymbol{y}_p$.

Let us consider a circular cone Q_α in $\mathbb{R}^n$ with apex at the origin $\boldsymbol{O}$, axis along $\boldsymbol{O}\boldsymbol{y}_p$ and half-angle α. Algebraically, define the direction vectors

$$\boldsymbol{y}_e \triangleq \frac{\boldsymbol{y}}{\|\boldsymbol{y}\|}, \tag{126}$$

$$\boldsymbol{z}_e \triangleq \frac{\boldsymbol{y}-\bar{\boldsymbol{x}}}{\|\boldsymbol{y}-\bar{\boldsymbol{x}}\|}. \tag{127}$$

Hence, the circular cone Q_α is given by

$$\begin{aligned} Q_\alpha &= \left\{\boldsymbol{r} \in \mathbb{R}^n : \frac{\boldsymbol{r}^\top \boldsymbol{y}_e}{\|\boldsymbol{r}\|} \geq \cos\alpha\right\} \\ &= \left\{\boldsymbol{r} \in \mathbb{R}^n : (\boldsymbol{r}-\boldsymbol{0})^\top (I - \epsilon^2(\alpha)\boldsymbol{y}_e\boldsymbol{y}_e^\top)(\boldsymbol{r}-\boldsymbol{0}) \leq 0\right\}, \end{aligned} \tag{128}$$

where $\epsilon(\alpha) \triangleq 1/\cos\alpha$ denotes the eccentricity of the cone. Cone Q_α intersects with the noise sphere $\mathcal{B}(\bar{\boldsymbol{x}}, w)$, thus producing a surface area $Q_\alpha \cap \mathcal{B}(\bar{\boldsymbol{x}}, w)$ delimited by J and K on Fig. 18. Thus, the induced density at $\boldsymbol{y}_p$ is given by

$$g_w(\boldsymbol{y_p}) = \lim_{\alpha\to 0} \frac{S(Q_\alpha \cap \mathcal{B}(\bar{\boldsymbol{x}}, w))/S_{n-1}(w)}{S(Q_\alpha \cap \mathcal{B}(\boldsymbol{O}, A\sqrt{n}))}, \tag{129}$$

where $S(\cdot)$ denotes the surface area in $\mathbb{R}^n$. Note that for sufficiently small α, the spherical surface around $\boldsymbol{y}$ is equivalent to the tangent hyperplane at $\boldsymbol{y}$, given by

$$\begin{aligned} H &= \left\{\boldsymbol{r} \in \mathbb{R}^n : \boldsymbol{z}_e^\top(\boldsymbol{r}-\boldsymbol{y}) = 0\right\} \\ &= \left\{\boldsymbol{r} \in \mathbb{R}^n : \boldsymbol{z}_e^\top(\boldsymbol{r}-\boldsymbol{0}) = \hat{h}\right\}, \end{aligned} \tag{130}$$

where $\hat{h} \triangleq \boldsymbol{z}_e^\top \boldsymbol{y}$. Define $\rho \triangleq \sqrt{1-(\boldsymbol{z}_e^\top\boldsymbol{y}_e)^2}$. Thus, using the result by Dearing [59, Eq. (15)], if $\epsilon(\alpha)\rho < 1$, the intersection of hyperplane H and cone Q_α is an ellipsoid of dimension $(n-1)$, which, after proper rotation T around $\boldsymbol{O}$, can be written as

$$\begin{aligned} &T(Q_\alpha) \cap T(H) \\ &= \left\{(r_1, \ldots, r_{n-1}, \hat{h}) : \frac{(r_1-\hat{c}_1)^2}{\hat{a}^2} + \frac{\sum_{j=2}^{n-1}(r_j-\hat{c}_j)^2}{\tilde{b}} = 1\right\}, \end{aligned}$$

where

$$\sigma = \boldsymbol{z}_e^\top \boldsymbol{y}_e, \tag{131}$$

$$\hat{c}_1 = \frac{\epsilon^2(\alpha)\rho\sigma\hat{h}}{1-\epsilon^2(\alpha)\rho^2}, \quad \hat{c}_j = 0,\ j = 2, \ldots, n-1, \tag{132}$$

$$\hat{a}^2 = \frac{(\epsilon^2(\alpha)-1)\hat{h}^2}{(1-\epsilon^2(\alpha)\rho^2)^2}, \tag{133}$$

$$\tilde{b} = \hat{a}^2(1-\epsilon^2(\alpha)\rho^2). \tag{134}$$

Since $\boldsymbol{z}_e$ and $\boldsymbol{y}_e$ are non-orthogonal, $1/\rho > 1$. Hence, for sufficiently small α, $\epsilon(\alpha) < 1/\rho$ and thus Dearing's result follows. Summarizing the analysis above, we obtain

$$\begin{aligned} &\lim_{\alpha\to 0} S\big(Q_\alpha \cap \mathcal{B}(\bar{\boldsymbol{x}}, w)\big) \\ &= \lim_{\alpha\to 0} S\big(T(Q_\alpha) \cap T(H)\big) &(135)\\ &= \lim_{\alpha\to 0} \frac{\pi^{\frac{n-1}{2}}}{\Gamma(\frac{n+1}{2})} \hat{a}\Big(\sqrt{\tilde{b}}\Big)^{n-2} &(136)\\ &= \lim_{\alpha\to 0} \frac{\pi^{\frac{n-1}{2}}}{\Gamma(\frac{n+1}{2})} \left(\frac{(\epsilon^2(\alpha)-1)\hat{h}^2}{(1-\epsilon^2(\alpha)\rho^2)^2}\right)^{\frac{n-1}{2}} \big(1-\epsilon^2(\alpha)\rho^2\big)^{\frac{n-2}{2}} &(137)\\ &= \lim_{\alpha\to 0} \frac{\pi^{\frac{n-1}{2}}}{\Gamma(\frac{n+1}{2})} 2^{\frac{n-1}{2}} (\epsilon(\alpha)-1)^{\frac{n-1}{2}} \hat{h}^{n-1} (\boldsymbol{z}_e^\top \boldsymbol{y}_e)^{-n} &(138)\\ &= \lim_{\alpha\to 0} \frac{\pi^{\frac{n-1}{2}}}{\Gamma(\frac{n+1}{2})} 2^{\frac{n-1}{2}} \Big(\frac{1-\cos\alpha}{\cos\alpha}\Big)^{\frac{n-1}{2}} \Big(\frac{\boldsymbol{z}_e^\top \boldsymbol{y}}{\boldsymbol{z}_e^\top \boldsymbol{y}_e}\Big)^{n-1} \frac{1}{\boldsymbol{z}_e^\top \boldsymbol{y}_e} &(139)\\ &= \lim_{\alpha\to 0} \frac{\pi^{\frac{n-1}{2}}}{\Gamma(\frac{n+1}{2})} 2^{\frac{n-1}{2}} \Big(2\sin^2\Big(\frac{\alpha}{2}\Big)\Big)^{\frac{n-1}{2}} \frac{\|\boldsymbol{y}\|^{n-1}}{\cos\angle\bar{\boldsymbol{x}}\boldsymbol{y}\boldsymbol{O}} &(140)\\ &= \lim_{\alpha\to 0} \frac{\pi^{\frac{n-1}{2}}}{\Gamma(\frac{n+1}{2})} \alpha^{n-1} \frac{\|\boldsymbol{y}\|^{n-1}}{\cos\angle\bar{\boldsymbol{x}}\boldsymbol{y}\boldsymbol{O}}, &(141) \end{aligned}$$

where (135) follows since for sufficiently small half-angle, the spherical surface around $\boldsymbol{y}$ is equivalent to that of the tangent hyperplane H at $\boldsymbol{y}$. From [19, Eq. (21)], the area of the spherical cap $S(Q_\alpha \cap \mathcal{B}(\boldsymbol{O}, A\sqrt{n}))$ is given by

$$\begin{aligned} &S(Q_\alpha \cap \mathcal{B}(\boldsymbol{O}, A\sqrt{n})) \\ &\quad = \frac{(n-1)\pi^{\frac{n-1}{2}}(A\sqrt{n})^{n-1}}{\Gamma\big(\frac{n+1}{2}\big)} \int_0^\alpha \sin^{n-2}\theta \,\mathrm{d}\theta. \end{aligned} \tag{142}$$

Substituting (141), (142) into (129), we obtain

$$\begin{aligned} g_w(\boldsymbol{y}_p) &= \lim_{\alpha\to 0} \frac{S(Q_\alpha \cap \mathcal{B}(\bar{\boldsymbol{x}}, w))}{S(Q_\alpha \cap \mathcal{B}(\boldsymbol{O}, A\sqrt{n}))} \frac{S_{n-1}(A\sqrt{n})}{S_{n-1}(w)S_{n-1}(A\sqrt{n})} \\ &= \lim_{\alpha\to 0} \frac{\alpha^{n-1}\frac{\|\boldsymbol{y}(\boldsymbol{y}_p)\|^{n-1}}{\cos\angle\bar{\boldsymbol{x}}\boldsymbol{y}(\boldsymbol{y}_p)\boldsymbol{O}}}{(n-1)\int_0^\alpha \theta^{n-2}\,\mathrm{d}\theta} \frac{1}{w^{n-1}} \frac{1}{S_{n-1}(A\sqrt{n})} \\ &= \Big(\frac{\|\boldsymbol{y}(\boldsymbol{y}_p)\|}{w}\Big)^{n-1} \frac{1}{\cos\angle\bar{\boldsymbol{x}}\boldsymbol{y}(\boldsymbol{y}_p)\boldsymbol{O}} \frac{1}{S_{n-1}(A\sqrt{n})}, \end{aligned} \tag{143}$$

where $\boldsymbol{y}(\boldsymbol{y}_p)$ is the pre-image of $\boldsymbol{y}_p$ on the noise sphere $\mathcal{B}(\bar{\boldsymbol{x}}, w)$. Here, (143) is the induced density function of $\boldsymbol{y}_p \in \mathcal{B}(\boldsymbol{O}, A\sqrt{n})$. Observe that it is rotationally symmetric with respect to axis $\boldsymbol{O}\bar{\boldsymbol{x}}$.

Next, we give an alternative expression of $g_w(\boldsymbol{y}_p)$ to derive its upper bound and lower bound. First, we rotate the coordinate system such that axis $\boldsymbol{O}\bar{\boldsymbol{x}}$ is the first coordinate and the remaining $(n-1)$ coordinates are orthogonal to $\boldsymbol{O}\bar{\boldsymbol{x}}$. In the new coordinate system, let $\bar{\boldsymbol{x}} = (A\sqrt{n}, 0, \ldots, 0) \in \mathbb{R}^n$. Hence, for an arbitrary projected point $\boldsymbol{y}_p = (y_1, y_2, \ldots, y_n) \in \mathcal{B}(\boldsymbol{O}, A\sqrt{n})$, assume that $\rho \triangleq \|\boldsymbol{y}(\boldsymbol{y}_p)\|$. Thus,

$$\boldsymbol{y}(\boldsymbol{y}_p) = \frac{\rho}{A\sqrt{n}}(y_1, y_2, \ldots, y_n). \tag{144}$$

Since $\boldsymbol{y}(\boldsymbol{y}_p) \in \mathcal{B}(\bar{\boldsymbol{x}}, w)$,

$$\left(\frac{\rho}{A\sqrt{n}}y_1 - A\sqrt{n}\right)^2 + \left(\frac{\rho}{A\sqrt{n}}\right)^2 \sum_{i=2}^{n} y_i^2 = w^2. \tag{145}$$

Solving for ρ yields

$$\rho = y_1 + \sqrt{y_1^2 + w^2 - A^2 n}. \tag{146}$$

By law of cosines, it is shown that

$$\cos\angle\bar{\boldsymbol{x}}\boldsymbol{y}\boldsymbol{O} = \frac{\rho^2 + w^2 - A^2 n}{2\rho w} = \frac{\sqrt{y_1^2 + w^2 - A^2 n}}{w}. \tag{147}$$

Hence, substituting (146) and (147) into (143) and expressing $g_w(\boldsymbol{y}_p)$ in terms of $y_1 \in [-A\sqrt{n}, A\sqrt{n}]$, we obtain

$$g_w(y_1) = \frac{1}{S_{n-1}(A\sqrt{n})} \frac{(y_1 + \sqrt{y_1^2 + w^2 - A^2 n})^{n-2}}{w^{n-2}} \cdot \left(1 + \frac{y_1}{\sqrt{y_1^2 + w^2 - A^2 n}}\right). \tag{148}$$

Clearly, $g_w(y_1)$ is monotonically increasing in y_1. Hence,

$$g_w(y_1) \geq g_w(-A\sqrt{n}) = \frac{1}{S_{n-1}(A\sqrt{n})}\left(1 - \frac{A\sqrt{n}}{w}\right)^{n-1} \tag{149}$$

$$g_w(y_1) \leq g_w(A\sqrt{n}) = \frac{1}{S_{n-1}(A\sqrt{n})}\left(1 + \frac{A\sqrt{n}}{w}\right)^{n-1}. \tag{150}$$

Geometrically, this implies that the maximum induced density is attained at the transmitted point $\bar{\boldsymbol{x}}$, whereas the minimum induced density is attained at $-\bar{\boldsymbol{x}}$.

Acknowledgment

The authors would like to thank the anonymous reviewers for their careful reading and constructive comments that improved this paper.

References

[1] H. Yang, S. V. S. Ranganathan, and R. D. Wesel, "Serial list Viterbi decoding with CRC: Managing errors, erasures, and complexity," in *2018 IEEE Global Commun. Conf. (GLOBECOM)*, Dec 2018, pp. 1–6.

[2] E. Liang, H. Yang, D. Divsalar, and R. D. Wesel, "List-decoded tail-biting convolutional codes with distance-spectrum optimal CRCs for 5G," in *2019 IEEE Global Commun. Conf. (GLOBECOM)*, 2019, pp. 1–6.

[3] H. Yang, L. Wang, V. Lau, and R. D. Wesel, "An efficient algorithm for designing optimal CRCs for tail-biting convolutional codes," in *Proc. 2020 IEEE Int. Symp. Inf. Theory (ISIT)*, 2020, pp. 292–297.

[4] H. Ji, S. Park, J. Yeo, Y. Kim, J. Lee, and B. Shim, "Ultra-reliable and low-latency communications in 5G downlink: Physical layer aspects," *IEEE Wireless Commun. Mag.*, vol. 25, no. 3, pp. 124–130, 2018.

[5] Y. Polyanskiy, H. V. Poor, and S. Verdú, "Channel coding rate in the finite blocklength regime," *IEEE Trans. Inf. Theory*, vol. 56, no. 5, pp. 2307–2359, May 2010.

[6] J. Font-Segura, G. Vazquez-Vilar, A. Martinez, A. Guillén i Fàbregas, and A. Lancho, "Saddlepoint approximations of lower and upper bounds to the error probability in channel coding," in *2018 52nd Annual Conf. Inf. Sci. Syst. (CISS)*, 2018, pp. 1–6.

[7] M. C. Coşkun, G. Durisi, T. Jerkovits, G. Liva, W. Ryan, B. Stein, and F. Steiner, "Efficient error-correcting codes in the short blocklength regime," *Physical Commun.*, vol. 34, pp. 66 – 79, 2019.

[8] M. P. C. Fossorier and S. Lin, "Soft-decision decoding of linear block codes based on ordered statistics," *IEEE Trans. Inf. Theory*, vol. 41, no. 5, pp. 1379–1396, 1995.

[9] C. Yue, M. Shirvanimoghaddam, B. Vucetic, and Y. Li, "A revisit to ordered statistics decoding: Distance distribution and decoding rules," *IEEE Trans. Inf. Theory*, vol. 67, no. 7, pp. 4288–4337, 2021.

[10] L. Gaudio, T. Ninacs, T. Jerkovits, and G. Liva, "On the performance of short tail-biting convolutional codes for ultra-reliable communications," in *SCC 2017; 11th Int. ITG Conf. Syst., Commun., and Coding*, Feb 2017, pp. 1–6.

[11] L. Dolecek, D. Divsalar, Y. Sun, and B. Amiri, "Non-binary protograph-based LDPC codes: Enumerators, analysis, and designs," *IEEE Trans. Inf. Theory*, vol. 60, no. 7, pp. 3913–3941, 2014.

[12] S. V. S. Ranganathan, D. Divsalar, and R. D. Wesel, "Quasi-cyclic protograph-based raptor-like LDPC codes for short block-lengths," *IEEE Trans. Inf. Theory*, vol. 65, no. 6, pp. 3758–3777, 2019.

[13] G. Liva, E. Paolini, B. Matuz, S. Scalise, and M. Chiani, "Short turbo codes over high order fields," *IEEE Trans. Commun.*, vol. 61, no. 6, pp. 2201–2211, 2013.

[14] T. Jerkovits and B. Matuz, "Turbo code design for short blocks," in *2016 8th Advanced Satellite Multimedia Syst. Conf. and the 14th Signal Processing for Space Commun. Workshop (ASMS/SPSC)*, 2016, pp. 1–6.

[15] E. Arıkan, "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, 2009.

[16] I. Tal and A. Vardy, "List decoding of polar codes," *IEEE Trans. Inf. Theory*, vol. 61, no. 5, pp. 2213–2226, 2015.

[17] E. Arıkan, "From sequential decoding to channel polarization and back again." [Online]. Available: http://arxiv.org/abs/1908.09594

[18] H. Yao, A. Fazeli, and A. Vardy, "List decoding of Arıkan's PAC codes," in *2020 IEEE Int. Symp. Inf. Theory (ISIT)*, 2020, pp. 443–448.

[19] C. E. Shannon, "Probability of error for optimal codes in a Gaussian channel," *Bell Syst. Tech. J.*, vol. 38, no. 3, pp. 611–656, May 1959.

[20] P. Elias, "Coding for noisy channels," *Proc. IRE. Conv. Rec. pt. 4*, vol. 3, pp. 37 – 46, 1955.

[21] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Trans. Inf. Theory*, vol. 13, no. 2, pp. 260–269, 1967.

[22] G. D. Forney, "The Viterbi algorithm," *Proc. IEEE*, vol. 61, no. 3, pp. 268–278, 1973.

[23] ——, "Convolutional codes II. maximum-likelihood decoding," *Inf. Contr.*, vol. 25, no. 3, pp. 222–266, 1974.

[24] T. Hehn and J. B. Huber, "LDPC codes and convolutional codes with equal structural delay: a comparison," *IEEE Trans. Commun.*, vol. 57, no. 6, pp. 1683–1692, 2009.

[25] S. V. Maiya, D. J. Costello, and T. E. Fuja, "Low latency coding: Convolutional codes vs. LDPC codes," *IEEE Trans. Commun.*, vol. 60, no. 5, pp. 1215–1225, 2012.

[26] W. W. Peterson and D. T. Brown, "Cyclic codes for error detection," *Proc. IRE*, vol. 49, no. 1, pp. 228–235, 1961.

[27] T. Baicheva and P. Kazakov, "CRC selection for decoding of CRC-Polar concatenated codes," in *Proc. the 9th Balkan Conf. Inf.* New York, NY, USA: ACM, 2019.

[28] M. Rice, "Comparative analysis of two realizations for hybrid-ARQ error control," in *1994 IEEE Global Commun. Conf. (GLOBECOM)*, 1994, pp. 115–119.

[29] "Universal mobile telecommunications system (UMTS); multiplexing and channel coding (FDD); 3GPP TS 25.212 version 7.0.0 release 7," European Telecommunications Standards Institute, Tech. Rep., 2006.

[30] "LTE; evolved universal terrestrial radio access (E-UTRA); multiplexing and channel coding; 3GPP TS 36.212 version 15.2.1 release 15," European Telecommunications Standards Institute, Tech. Rep., 2018.

[31] N. Seshadri and C. W. Sundberg, "List Viterbi decoding algorithms with applications," *IEEE Trans. Commun.*, vol. 42, no. 234, pp. 313–323, Feb 1994.

[32] P. Elias, "List decoding for noisy channels," *Proc. IRE WESCON Conf. Rec.*, vol. 2, pp. 94–104, 1957.

[33] I. E. Bocharova, R. Johannesson, B. D. Kudryashov, and M. Loncar, “An improved bound on the list error probability and list distance properties,” *IEEE Trans. Inf. Theory*, vol. 54, no. 1, pp. 13–32, 2008.

[34] E. Hof, I. Sason, and S. Shamai, “Performance bounds for erasure, list, and decision feedback schemes with linear block codes,” *IEEE Trans. Inf. Theory*, vol. 56, no. 8, pp. 3754–3778, 2010.

[35] H. Ma and J. Wolf, “On tail biting convolutional codes,” *IEEE Trans. Commun.*, vol. 34, no. 2, pp. 104–111, February 1986.

[36] C. Lou, B. Daneshrad, and R. D. Wesel, “Convolutional-code-specific CRC code design,” *IEEE Trans. Commun.*, vol. 63, no. 10, pp. 3459–3470, 2015.

[37] Q. Wang and V. K. Bhargava, “An efficient maximum likelihood decoding algorithm for generalized tail biting convolutional codes including quasicyclic codes,” *IEEE Trans. Commun.*, vol. 37, no. 8, pp. 875–879, Aug 1989.

[38] R. V. Cox and C. E. W. Sundberg, “An efficient adaptive circular Viterbi algorithm for decoding generalized tailbiting convolutional codes,” *IEEE Trans. Veh. Technol.*, vol. 43, no. 1, pp. 57–68, Feb 1994.

[39] J. B. Anderson and S. M. Hladik, “Tailbiting MAP decoders,” *IEEE J. Sel. Areas Commun.*, vol. 16, no. 2, pp. 297–302, Feb 1998.

[40] R. Y. Shao, Shu Lin, and M. P. C. Fossorier, “Two decoding algorithms for tailbiting codes,” *IEEE Trans. Commun.*, vol. 51, no. 10, pp. 1658–1665, Oct 2003.

[41] P. Shankar, P. N. A. Kumar, K. Sasidharan, B. S. Rajan, and A. S. Madhu, “Efficient convergent maximum likelihood decoding on tail-biting trellises.” [Online]. Available: https://arxiv.org/abs/cs/0601023

[42] S. Lin and D. J. Costello, *Error Control Coding*. New Jersey, USA: Pearson Education, 2004.

[43] R. Johannesson and K. S. Zigangirov, *Fundamentals of Convolutional Coding*, J. B. Anderson, Ed. New Jersey, USA: IEEE Press, 1999.

[44] R. Koetter and A. Vardy, “The structure of tail-biting trellises: minimality and basic principles,” *IEEE Trans. Inf. Theory*, vol. 49, no. 9, pp. 2081–2105, Sep. 2003.

[45] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed. New Jersey, USA: Wiley, 2006.

[46] H. Yang, “Github repository: CRC design for ZTCCs.” [Online]. Available: https://github.com/hengjie-yang/DSO_CRC_Design_for_ZTCCs

[47] ——, “Github repository: CRC design for TBCCs.” [Online]. Available: https://github.com/hengjie-yang/DSO_CRC_Design_for_TBCCs

[48] S. Yousefi and A. Khandani, “Generalized tangential sphere bound on the ML decoding error probability of linear binary block codes in AWGN interference,” *IEEE Trans. Inf. Theory*, vol. 50, no. 11, pp. 2810–2815, 2004.

[49] R. G. Gallager, *Low Density Parity Check Codes*. Cambridge, MA, USA: MIT Press, 1963.

[50] E. Berlekamp, “The technology of error-correcting codes,” *Proc. IEEE*, vol. 68, no. 5, pp. 564–593, 1980.

[51] G. Poltyrev, “Bounds on the decoding error probability of binary linear codes via their spectra,” *IEEE Trans. Inf. Theory*, vol. 40, no. 4, pp. 1284–1292, 1994.

[52] S. Yousefi and A. Khandani, “A new upper bound on the ML decoding error probability of linear binary block codes in AWGN interference,” *IEEE Trans. Inf. Theory*, vol. 50, no. 12, pp. 3026–3036, 2004.

[53] C. Bai, B. Mielczarek, W. A. Krzymien, and I. J. Fair, “Efficient list decoding for parallel concatenated convolutional codes,” in *2004 IEEE 15th Int. Symp. Personal, Indoor and Mobile Radio Commun.*, vol. 4, 2004, pp. 2586–2590.

[54] L. Lijofi, D. Cooper, and B. Canpolat, “A reduced complexity list single-wrong-turn (SWT) Viterbi decoding algorithm,” in *2004 IEEE 15th Int. Symp. Personal, Indoor and Mobile Radio Commun.*, vol. 1, 2004, pp. 274–279.

[55] M. Roder and R. Hamzaoui, “Fast tree-trellis list Viterbi decoding,” *IEEE Trans. Commun.*, vol. 54, no. 3, pp. 453–461, Mar. 2006.

[56] J. Kim, J. Tak, H. Kwak, and J. No, “A new list decoding algorithm for short-length TBCCs with CRC,” *IEEE Access*, vol. 6, pp. 35 105–35 111, 2018.

[57] M. Shirvanimoghaddam, M. S. Mohammadi, R. Abbas, A. Minja, C. Yue, B. Matuz, G. Han, Z. Lin, W. Liu, Y. Li, S. Johnson, and B. Vucetic, “Short block-length codes for ultra-reliable low latency communications,” *IEEE Commun. Mag.*, vol. 57, no. 2, pp. 130–137, 2019.

[58] R. Schiavone, “Channel coding for massive IoT satellite systems,” Master’s thesis, Politechnic University of Turin (Polito), 2021.

[59] P. M. Dearing, “Intersections of hyperplanes and conic sections in $\mathbf{R}^n$.” [Online]. Available: http://arxiv.org/abs/1702.03205