

# High-Rate Convolutional Codes with CRC-Aided List Decoding for Short Blocklengths

Wenhui Sui, Hengjie Yang, Brendan Towell, Ava Asmani, and Richard D. Wesel

Department of Electrical and Computer Engineering

University of California, Los Angeles

Email: {wenhui.sui, hengjie.yang, brendan.towell, ava24, wesel}@ucla.edu

**Abstract**—Recently, rate- $1/\omega$  zero-terminated and tail-biting convolutional codes (ZTCCs and TBCCs) with cyclic-redundancy-check (CRC)-aided list decoding have been shown to closely approach the random-coding union (RCU) bound for short blocklengths. This paper designs CRC polynomials for rate- $(\omega - 1)/\omega$  CCs with short blocklengths, considering both the ZT and TB cases. The CRC design seeks to optimize the frame error rate (FER) performance of the code resulting from the concatenation of the CRC code and the CC. Utilization of the dual trellis proposed by Yamada *et al.* lowers the complexity of CRC-aided serial list Viterbi decoding (SLVD) of ZTCCs and TBCCs. CRC-aided SLVD of the TBCCs closely approaches the RCU bound at blocklength of 128.

## I. INTRODUCTION

The structure of concatenating a convolutional code (CC) with a cyclic redundancy check (CRC) code has been a popular paradigm since 1994 when it was proposed in the context of hybrid automatic repeat request (ARQ) [1]. It was subsequently adopted in the cellular communication standards of both 3G [2] and 4G LTE [3]. In general, the CRC code serves as an outer error-detecting code that verifies if a codeword has been correctly received, whereas the CC serves as an inner error-correcting code to combat channel errors.

Recently, there has been a renewed interest in designing powerful short blocklength codes. This renewed interest is mainly driven by the development of finite blocklength information theory by Polyanskiy *et al.*, [4] and the stringent requirement of ultra-reliable low-latency communication (URLLC) for mission-critical IoT (Internet of Things) services [5]. In [4], Polyanskiy *et al.* developed a new achievability bound known as the random-coding union (RCU) bound and a new converse bound, known as the meta-converse (MC) bound. Together, these two bounds characterize the error probability of the best short blocklength code of length  $n$  with  $M$  codewords. The URLLC for mission-critical IoT requires that the time-to-transmit latency is within 500  $\mu$ s while maintaining a frame error rate (FER) less than  $10^{-5}$ .

Several short blocklength code designs have been proposed in literature. Important examples include the tail-biting (TB) convolutional codes decoded with the wrap-around

Viterbi algorithm (WAVA) [6], extended Bose-Chaudhuri-Hocquenghem (BCH) codes under ordered statistics decoding [7], [8], non-binary low-density parity-check (LDPC) codes [9], non-binary turbo codes [10], and polar codes under CRC-aided successive-cancellation list decoding [11]. Recent advances also include the polarization-adjusted convolutional codes by Arkan [12]. As a comprehensive overview, Coşkun *et al.* [7] surveyed most of the contemporary short blocklength code designs in the recent decade. We refer the reader to [7] for additional information.

In [13], Yang *et al.* proposed the CRC-aided CC as a powerful short blocklength code for binary-input (BI) additive white Gaussian noise (AWGN) channels. In [13], the convolutional encoder of interest is either zero-terminated (ZT) or TB with rate- $1/\omega$ . A good CRC-aided CC is constructed by searching for the *distance-spectrum optimal* (DSO) CRC polynomial for a given convolutional encoder and then concatenating the DSO CRC polynomial with the convolutional encoder.

The nature of the concatenation naturally permits the use of serial list Viterbi decoding (SLVD), an efficient algorithm originally proposed by Seshadri and Sundberg [14]. Yang *et al.* showed that the expected list rank  $E[L]$  of the SLVD of the CRC-aided CC is small at a target low error probability, thus achieving a low average decoding complexity. Yang *et al.* demonstrated that several concatenated codes generated by the DSO CRC polynomial and the TBCC, or in short, CRC-TBCCs, approach the RCU bound. In [15], Schiavone extended this line of work by looking at the parallel list Viterbi decoding with a bounded list size. However, these works did not consider rate- $(\omega - 1)/\omega$  CCs. It remains open whether this framework can be extended to CCs of an arbitrary rate such that the resulting concatenated code can approach the RCU bound at a low decoding complexity.

In this paper, we consider designing good CRC-aided CCs for rate- $(\omega - 1)/\omega$  CCs at short blocklength for the BI-AWGN channel, where the CC is either ZT or TB. We consider systematic, rate- $(\omega - 1)/\omega$  convolutional encoders. The resulting concatenated codes are respectively called a CRC-ZTCC and a CRC-TBCC. We assume that the SLVD has a sufficiently large list size such that a codeword whose input sequence passes the CRC verification can always be found. Thus, the SLVD is an implementation of maximum-likelihood decoding. The FER is in fact the undetected error probability. Simulations show that at short blocklength, our rate- $(\omega - 1)/\omega$

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

CRC-TBCCs still perform closely to the RCU bound.

In [16], Karimzadeh and Vu considered designing the optimal CRC polynomial for multi-input CCs. In their framework, the information sequence is first divided into  $(\omega - 1)$  streams, one for each input rail, and they aim at designing an optimal CRC polynomial for each rail. Unlike their architecture, in this paper, the information sequence is first encoded with a single CRC polynomial and is then divided into  $(\omega - 1)$  streams. Simulation results show that our framework can yield better FER performance than that of Karimzadeh and Vu.

For rate- $(\omega - 1)/\omega$  CCs, the SLVD on the *primal trellis* requires high decoding complexity because of the  $2^{\omega-1}$  outgoing branches at each node. SLVD implementation becomes complicated when there exist more than two outgoing branches per state. In order to simplify SLVD implementation and reduce complexity, we utilize the *dual trellis* pioneered by Yamada *et al.* [17]. The dual trellis expands the length of the primal trellis by a factor of  $\omega$ , while reducing the number of outgoing branches at each node from  $2^{\omega-1}$  to at most two. It is also worth noting that in [18], Srinivasan *et al.* derived the dual *a posteriori* probability decoding of a high-rate CC.

The remainder of this paper is organized as follows. Section II reviews systematic encoding for  $(\omega, \omega - 1, v)$  convolutional codes and describes the dual trellis construction. Section III considers CRC-ZTCCs for rate- $(\omega - 1)/\omega$  CCs. It addresses the zero-termination issue, presents DSO CRC design for high-rate ZTCCs, and shows CRC-ZTCC simulation results. Section IV considers CRC-TBCCs for rate- $(\omega - 1)/\omega$  CCs. It addresses how to find the TB initial state over the dual trellis, describes DSO CRC design for TBCCs, and shows CRC-TBCC simulation results. Section V concludes the paper.

*Notation:* Let  $K$  and  $N$  denote the information length and blocklength in bits. Let  $R = K/N$  denote the rate of a code with information length  $K$  and blocklength  $N$ . A degree- $m$  CRC polynomial is of the form  $p(x) = 1 + p_1x + \dots + p_{m-1}x^{m-1} + x^m$ , where  $p_i \in \{0, 1\}$ ,  $i = 1, 2, \dots, m - 1$ . For brevity, a 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 codewords are BPSK modulated. The SNR is defined as  $\gamma_s \triangleq 10 \log_{10}(A^2)$  (dB), where  $A$  represents the BPSK amplitude and the noise follows a standard normal distribution.

## II. SYSTEMATIC ENCODING AND DUAL TRELLIS

This section describes systematic encoding and introduces the dual trellis proposed by Yamada *et al.* [17] for high-rate CCs generated with an  $(\omega, \omega - 1, v)$  convolutional encoder, where  $v$  represents the overall constraint length.

### A. Systematic Encoding

We briefly follow [19, Chapter 11] in describing a systematic  $(\omega, \omega - 1, v)$  convolutional encoder. A systematic  $(\omega, \omega - 1, v)$  convolutional encoder can be represented by its parity check matrix

$$H(D) = [h^{(\omega-1)}(D), h^{(\omega-2)}(D), \dots, h^{(0)}(D)], \quad (1)$$

where each  $h^{(i)}(D)$  is a polynomial of degree up to  $v$  in delay element  $D$  associated with the  $i$ -th code stream, i.e.,

$$h^{(i)}(D) = h_v^{(i)}D^v + h_{v-1}^{(i)}D^{v-1} + \dots + h_0^{(i)}, \quad (2)$$

where  $h_j^{(i)} \in \{0, 1\}$ . For convenience, we represent each  $h^{(i)}(D)$  in octal form. For instance,  $H(D) = [D^3 + D^2 + D + 1, D^3 + D^2 + 1, D^3 + D + 1]$  can be concisely written as  $H = (17, 15, 13)$ . Define  $\mathbf{h}^{(i)} \triangleq [h_v^{(i)}, h_{v-1}^{(i)}, \dots, h_0^{(i)}]$ ,  $i = 0, 1, \dots, \omega - 1$ . The systematic encoding matrix  $G(D)$  associated with  $H(D)$  is given by

$$G(D) = \begin{bmatrix} \frac{h^{(1)}(D)}{h^{(0)}(D)} & 1 & 0 & \dots & 0 \\ \frac{h^{(2)}(D)}{h^{(0)}(D)} & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{h^{(\omega-1)}(D)}{h^{(0)}(D)} & 0 & 0 & \dots & 1 \end{bmatrix}. \quad (3)$$

The first output bit is a coded bit and the remaining output bits are a direct copy of the corresponding input bits.

### B. Dual Trellis

The primal trellis associated with a rate- $(\omega - 1)/\omega$  ZTCC has  $2^{\omega-1}$  outgoing branches per state. Performing SLVD over the primal trellis when  $\omega > 2$  is highly complex. In [13], the low decoding complexity of SLVD for rate- $1/\omega$  convolutional codes relies on the fact that only 2 outgoing branches are associated with each state. In order to efficiently perform SLVD, we consider the dual trellis proposed by Yamada *et al.* [17].

We briefly explain the dual trellis construction for parity check matrix  $H(D) = [h^{(\omega-1)}(D), h^{(\omega-2)}(D), \dots, h^{(0)}(D)]$ . First, we define the maximum instant response order  $\lambda$  as

$$\lambda \triangleq \max\{j \in \{0, 1, \dots, \omega - 1\} : h_0^{(j)} = 1\}. \quad (4)$$

The state of the dual trellis is represented by the partial sums of  $(v + 1)$  adders in the observer canonical form of  $H(D)$ . At time index  $j$ ,  $j = 0, 1, \dots, \omega - 1$ , the state is given by

$$\mathbf{s}^{(j)} = [s_v^{(j)}, s_{v-1}^{(j)}, \dots, s_0^{(j)}]. \quad (5)$$

Next, we show how the state  $\mathbf{s}^{(j)}$  evolves in terms of the output bits  $\mathbf{y}_k = [y_k^{(0)}, y_k^{(1)}, \dots, y_k^{(\omega-1)}]$ ,  $k = 1, 2, \dots, N/\omega$ , so that a dual trellis can be established.

*Dual trellis construction for  $\mathbf{y}_k = [y_k^{(0)}, y_k^{(1)}, \dots, y_k^{(\omega-1)}]$ :*

- 1) At time  $j = 0$ ,  $\mathbf{s}^{(0)} = [0, s_{v-1}^{(j)}, s_{v-2}^{(j)}, \dots, s_0^{(j)}]$ , where  $s_i^{(0)} \in \{0, 1\}$ . Namely, only  $2^v$  states exist at  $j = 0$ .
- 2) At time  $j$ ,  $j < \omega - 1$ , draw branches from each state  $\mathbf{s}^{(j)}$  to state  $\mathbf{s}^{(j+1)}$  by

$$\mathbf{s}^{(j+1)} = \mathbf{s}^{(j)} + y_k^{(j)} \mathbf{h}^{(j)}, \quad y_k^{(j)} = 0, 1. \quad (6)$$

- 3) At time  $j = \omega - 1$ , draw branches from each state  $\mathbf{s}^{(\omega-1)}$  to state  $\mathbf{s}^{(\omega)}$  by

$$\mathbf{s}^{(\omega)} = \left( \mathbf{s}^{(\omega-1)} + y_k^{(\omega-1)} \mathbf{h}^{(\omega-1)} \right)^r, \quad y_k^{(\omega-1)} = 0, 1, \quad (7)$$

where  $(a_v, a_{v-1}, \dots, a_1, a_0)^r = (0, a_v, a_{v-1}, \dots, a_1)$ .

TABLE I  
DSO CRC POLYNOMIALS FOR RATE-3/4 ZTCC AT BLOCKLENGTH  
 $N = 128$  GENERATED BY  $H = (33, 25, 37, 31)$  WITH  $v = 4$ , BY  
 $H = (47, 73, 57, 75)$  WITH  $v = 5$ , AND BY  $H = (107, 135, 133, 141)$   
WITH  $v = 6$

$K$	$m$	$R$	$v = 4$ CRC	$v = 5$ CRC	$v = 6$ CRC
87	3	0.680	0x9	0x9	0xB
86	4	0.672	0x1B	0x15	0x1D
85	5	0.664	0x25	0x25	0x25
84	6	0.656	0x4D	0x7B	0x6F
83	7	0.648	0xF3	0xED	0x97
82	8	0.641	0x1E9	0x1B7	0x1B5
81	9	0.633	0x31B	0x3F1	0x2F1
80	10	0.625	0x5C9	0x66F	0x59F

4) For time  $j = \lambda$ , draw a branch from each state  $s^{(\lambda)}$  according to (6) only for  $y_k^{(\lambda)} = s_0^{(\lambda)}$ .

After repeating the above construction for each  $y_k$ ,  $k = 1, 2, \dots, N/\omega$ , we obtain the dual trellis associated with the  $(\omega, \omega - 1, v)$  convolutional code. Since the primal trellis is of length  $N/\omega$ , whereas the dual trellis is of length  $N$ , the dual trellis can be thought of as expanding the primal trellis length by a factor of  $\omega$ , while reducing the number of outgoing branches per state from  $2^{\omega-1}$  to less than or equal to 2.

### III. ZTCC WITH DSO CRC VIA DUAL TRELLIS SLVD

This section considers CRC-ZTCCs for rate- $(\omega-1)/\omega$  CCs. Section III-A presents a zero termination method over the dual trellis. Section III-B describes our DSO CRC polynomial search procedure. Finally, Section III-C presents simulation results of the CRC-ZTCC compared with the RCU bound. As a case study, this paper mainly focuses on the rate-3/4 systematic feedback convolutional codes in [19, Table 12.1(e)].

#### A. Zero Termination of Dual Trellis

For an  $(\omega, \omega - 1, v)$  CC, zero termination over the dual trellis requires at most  $\omega \lceil v/(\omega - 1) \rceil$  steps. In our implementation, a breadth-first search identifies the zero-termination input and output bit patterns that provide a trajectory from each possible state  $s$  to the zero state. The input and output bit patterns have lengths  $(\omega - 1) \lceil v/(\omega - 1) \rceil$  and  $\omega \lceil v/(\omega - 1) \rceil$  respectively.

#### B. Design of DSO CRCs for High-Rate ZTCCs

In general, a DSO CRC polynomial provides the optimal distance spectrum which minimizes the union bound on the FER at a specified SNR [13]. In this paper, we focus on the low FER regime. Thus, the DSO CRC polynomials identified in this paper simply maximize the minimum distance of the concatenated code. Examples in [13] indicate that DSO CRC polynomials designed in this way can provide optimal or near-optimal performance for a wide range of SNRs.

The design procedure of the DSO CRC polynomial for high-rate ZTCCs essentially follows from the DSO CRC design algorithm for low target error probability in [13]. The first step is to collect the *irreducible error events* (IEEs), which are ZT paths on the trellis that deviate from the zero state once and rejoin it once. IEEs with a very large output Hamming weight do not affect the choice of optimal CRCs. In order to

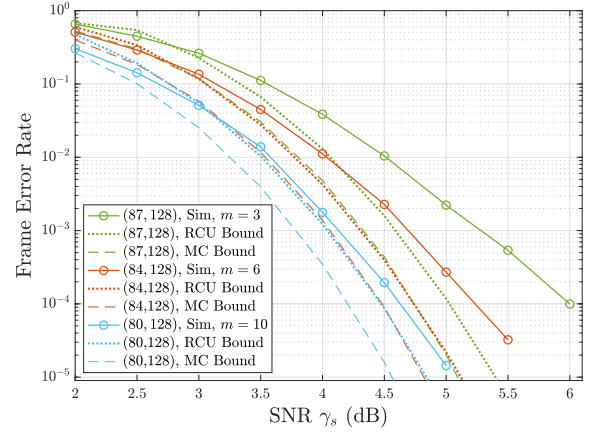


Fig. 1. FER vs. SNR for various CRC-ZTCCs. The ZTCC is generated with the  $(4, 3, 6)$  encoder  $H = (107, 135, 133, 141)$ . The DSO CRC polynomials of degrees 3, 6, and 10 are 0xB, 0x6F, and 0x59F, respectively. Values in parenthesis denote information length  $K$  and blocklength  $N$ , respectively.

reduce the runtime of the CRC optimization algorithm, IEEs with output Hamming weight larger than a threshold  $\tilde{d} - 1$  are not considered. Dynamic programming constructs all ZT paths of length equal to  $N/\omega$  and output weight less than  $\tilde{d}$ . Finally, we use the resulting set of ZT paths to identify the degree- $m$  DSO CRC polynomial for the rate- $(\omega - 1)/\omega$  CC.

Table I presents the DSO CRC polynomials for ZTCCs generated with  $H = (33, 25, 37, 31)$ ,  $H = (47, 73, 57, 75)$ , and  $H = (107, 135, 133, 141)$ . The design assumes a fixed blocklength  $N = 128$  bits. Due to the overhead caused by the CRC bits and by zero termination, the rates of CRC-ZTCCs are less than 3/4. Specifically, for a given information length  $K$ , CRC degree  $m$  and an  $(\omega, \omega - 1, v)$  encoder, the blocklength  $N$  for a CRC-ZTCC is given by  $N = \left( K + m + (\omega - 1) \left\lceil \frac{v}{\omega - 1} \right\rceil \right) \frac{\omega}{\omega - 1}$ , giving  $R = \frac{K}{N} = \frac{\omega - 1}{\omega} \frac{K}{K + m + (\omega - 1) \left\lceil \frac{v}{\omega - 1} \right\rceil}$ . We see from (III-B) that the  $(\omega, \omega - 1, v)$  convolutional encoder can accept any CRC degree  $m$  as long as  $K + m$  is divisible by  $(\omega - 1)$ .

#### C. Results and Comparison with RCU Bound

Fig. 1 shows the performance of CRC-ZTCCs with increasing CRC degrees 3, 6 and 10 and a fixed blocklength  $N = 128$  bits. We see that at the target FER of  $10^{-4}$ , increasing the CRC degree reduces the gap to the RCU bound. With  $m = 10$  and  $v = 6$ , the CRC-ZTCC approaches the RCU bound within 0.25 dB.

In [16], Karimzadeh *et al.* considered designing optimal CRC polynomials for each input rail of a multi-input CC. In their setup, an information sequence for an  $(\omega, \omega - 1, v)$  encoder needs to be split into  $(\omega - 1)$  subsequences before CRC encoding. In contrast, the entire information sequence in our framework is encoded with a single CRC polynomial. Then the resulting sequence is evenly divided into  $(\omega - 1)$  subsequences, one for each rail. To compare the performance between these two schemes, we design three degree-3 optimal CRC polynomials, one for each rail, for ZTCC with  $H =$

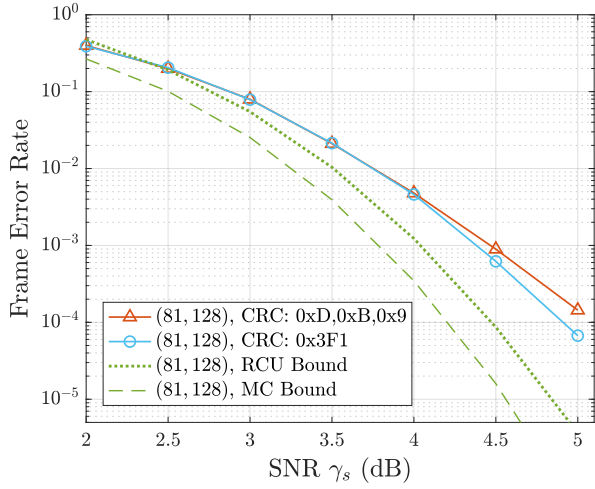


Fig. 2. FER vs. SNR for  $v = 5$  CRC-ZTCCs designed under Karimzadeh *et al.*'s scheme [16] and our scheme. Both CRC-ZTCCs have information length  $K = 81$  and blocklength  $N = 128$ .

(47, 73, 57, 75). The three CRC polynomials jointly maximize the minimum distance of the CRC-ZTCC. For the single-CRC design, we use the single degree-9 DSO CRC polynomial for the same encoder from Table I. Both CRC-ZTCCs have an information length  $K = 81$  and blocklength  $N = 128$ . Fig. 2 shows the performance comparison between these two codes, showing that a single degree-9 DSO CRC polynomial outperforms three degree-3 DSO CRC polynomials, one for each rail. This suggests that a single DSO CRC polynomial may suffice to provide superior protection for each input rail. The decoding complexity is similar regardless of the CRC scheme.

#### IV. TBCC WITH DSO CRC AND DUAL TRELLIS SLVD

The performance of CRC-aided list decoding of ZTCCs relative to the RCU bound is constrained by the termination bits appended to the end of the original message, which are required to bring the trellis back to the all-zero state. TBCCs avoid this overhead by replacing the zero termination condition with the TB condition, which states that the final state of the trellis is the same as the initial state of the trellis [20].

In this section, we apply the SLVD to CRC-TBCCs over the dual trellis. We will discuss how to determine the initial state for the TBCC to ensure that the TB condition is met and demonstrate designs of DSO CRCs for rate  $(\omega - 1)/\omega$  TB codes. Decoding complexity and performance are analyzed at the end of this section.

##### A. List Decoding for TBCC with CRC

There are two primary differences in the development and analysis of list decoding between ZTCCs, as described in Section III, and TBCCs. One difference is that since the ZT condition is replaced with the TB condition, the encoder must determine the initial trellis state so that the TB condition is satisfied. The other difference is that SLVD on the dual trellis must be adapted to handle the TB condition.

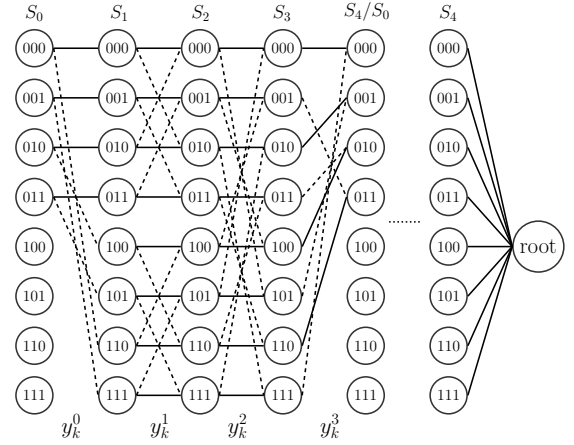


Fig. 3. Dual trellis diagram for rate-3/4 TBCC with a root node at the end for encoder  $H = (2, 5, 7, 6)$  with  $v = 2$ . Solid lines represent 0 paths and dashed lines represent 1 paths.

To satisfy the TB condition, encoding is attempted from every initial state to identify the initial state that satisfies the TB condition. This is required because our recursive encoder cannot simply achieve the TB condition by setting the initial encoder memory to be the final bits of the information sequence.

To adapt SLVD on the dual trellis to handle the TB condition, we propose an efficient way to keep track of the path metrics and find the next path with minimum metric through an additional root node as shown in Fig. 3. The root node connects to all terminating states after forward traversing the dual trellis. The Hamming distance of the branch metric for the branch connecting any state to this root node is zero. This additional root node allows the trellis to end in a single state, so that the basic SLVD approach for a ZTCC may be applied. During SLVD, if the current path does not pass either the CRC or TB check, the minimum value among all remaining path metrics will be selected as the next path to check.

##### B. Design of DSO CRCs for High-Rate TBCCs

The design of DSO CRCs for high-rate TBCCs follows the two-phase design algorithm shown in [13]. This algorithm is briefly explained below.

Consider a TB trellis  $T = (V, E, \mathcal{A})$  of length  $n$ , where  $\mathcal{A}$  denotes the set of output alphabet,  $V$  denotes the set of states, and  $E$  denotes the set of edges described in an ordered triple  $(s, a, s')$  with  $s, s' \in V$  and  $a \in \mathcal{A}$  [21]. Assume  $|V| = 2^v$  and let  $V_0 = \{0, 1, \dots, 2^v - 1\}$ . Define the set of IEEs at state  $\sigma \in V$  as

$$\text{IEE}(\sigma) \triangleq \bigcup_{l=1,2,\dots,n} \overline{\text{IEE}}(\sigma, l), \quad (8)$$

where

$$\overline{\text{IEE}}(\sigma, l) \triangleq \{(\mathbf{s}, \mathbf{a}) \in V_0^{l+1} \times \mathcal{A}^l : s_0 = s_l = \sigma; \forall j, 0 < j < l, s_j \notin \{0, 1, \dots, \sigma\}\}. \quad (9)$$

TABLE II  
DSO CRC POLYNOMIALS FOR RATE-3/4 TBCC AT BLOCKLENGTH  
 $N = 128$  GENERATED BY  $H = (33, 25, 37, 31)$  WITH  $v = 4$ , BY  
 $H = (47, 73, 57, 75)$  WITH  $v = 5$ , AND BY  $H = (107, 135, 133, 141)$   
WITH  $v = 6$

$K$	$m$	$R$	$v = 4$ CRC	$v = 5$ CRC	$v = 6$ CRC
93	3	0.727	0x9	0x9	0xB
92	4	0.719	0x1B	0x1D	0x17
91	5	0.711	0x25	0x3B	0x33
90	6	0.703	0x7D	0x4F	0x41
89	7	0.695	0xF9	0xD1	0xBD
88	8	0.688	0x1CF	0x173	0x111
87	9	0.680	0x38F	0x3BF	0x333
86	10	0.672	0x73F	0x697	0x723

The IEEs at state  $\sigma$  can be thought of as “building blocks” for an arbitrarily long TB path that starts and ends at the same state  $\sigma$ .

The first phase is called the collection phase, during which the algorithm collects  $\text{IEE}(\sigma)$  with output Hamming weight less than the threshold  $\tilde{d}$  over a sufficiently long TB trellis. The second phase is called the search phase, during which the algorithm first reconstructs all TB paths of length  $N/\omega$  and output weight less than  $\tilde{d}$  via concatenation of the IEEs and circular shifting of the resulting path. Then, using these TB paths, the algorithm searches for the degree- $m$  DSO CRC polynomial by maximizing the minimum distance of the undetected TB path.

Table II presents the DSO CRC polynomials for TBCCs generated with  $H = (33, 25, 37, 31)$ ,  $H = (47, 73, 57, 75)$ , and  $H = (107, 135, 133, 141)$ . The design assumes a fixed blocklength  $N = 128$ . TB encoding avoids the rate loss caused by the overhead of the zero termination. Specifically, for a given information length  $K$ , CRC degree  $m$  and an  $(\omega, \omega - 1, v)$  encoder, the blocklength  $N$  for a CRC-TBCC is given by  $N = (K + m) \frac{\omega - 1}{\omega - v}$ , giving  $R = \frac{K}{N} = \frac{\omega - 1}{\omega} \frac{K}{K + m}$ .

### C. Complexity Analysis

In [13], the authors provided the complexity expression for SLVD of CRC-ZTCCs and CRC-TBCCs, where the convolutional encoder is of rate  $1/\omega$ . Observe that the dual trellis has no more than 2 outgoing branches per state, similar to the trellis of a rate- $1/\omega$  CC. Thus, we directly apply their complexity expression to the SLVD over the dual trellis.

As noted in [13], the overall average complexity of the SLVD can be decomposed into three components:

$$C_{\text{SLVD}} = C_{\text{SSV}} + C_{\text{trace}} + C_{\text{list}}, \quad (10)$$

where  $C_{\text{SSV}}$  denotes the complexity of a standard soft Viterbi (SSV),  $C_{\text{trace}}$  denotes the complexity of the *additional* traceback operations required by SLVD, and  $C_{\text{list}}$  denotes the average complexity of inserting new elements to maintain an ordered list of path metric differences.

$C_{\text{SSV}}$  is the complexity of add-compare-select (ACS) operations and the initial traceback operation. For CRC-ZTCCs,

$$C_{\text{SSV}} = (2^{v+1} - 2) + 1.5(2^{v+1} - 2) + 1.5(K + m - v)2^{v+1} + c_1[2(K + m + v) + 1.5(K + m)]. \quad (11)$$

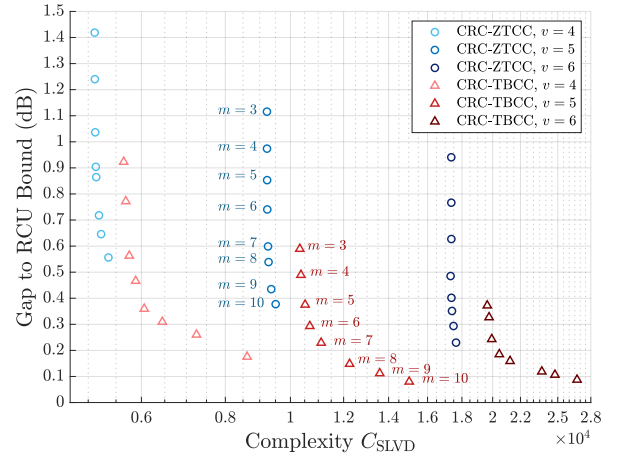


Fig. 4. The SNR gap to the RCU bound vs. the average complexity of SLVD of CRC-ZTCC codes in Table I and CRC-TBCC codes in Table II for target FER of  $10^{-4}$ . Each color represents a specific CRC-aided CC shown in the tables. Markers from top to bottom with the same color correspond to DSO CRC polynomials with  $m = 3, \dots, 10$ .

For CRC-TBCCs, this quantity is given by

$$C_{\text{SSV}} = 1.5(K + m)2^{v+1} + 2^v + 3.5c_1(K + m). \quad (12)$$

The second component  $C_{\text{trace}}$  for CRC-ZTCC is given by

$$C_{\text{trace}} = c_1(E[L] - 1)[2(K + m + v) + 1.5(K + m)]. \quad (13)$$

For CRC-TBCCs,  $C_{\text{trace}}$  is given by

$$C_{\text{trace}} = 3.5c_1(E[L] - 1)(K + m). \quad (14)$$

The third component, which is the same for ZT and TB, is

$$C_{\text{list}} = c_2 E[I] \log(E[I]), \quad (15)$$

where  $E[I]$  is the expected number of insertions to maintain the sorted list of path metric differences. For CRC-ZTCCs,

$$E[I] \leq (K + m)E[L], \quad (16)$$

and for CRC-TBCCs,

$$E[I] \leq (K + m)E[L] + 2^v - 1. \quad (17)$$

In the above expressions,  $c_1$  and  $c_2$  are two computer-specific constants that characterize implementation-specific differences in the implemented complexity of traceback and list insertion (respectively) as compared to the ACS operations of Viterbi decoding. In this paper, we assume that  $c_1 = c_2 = 1$  and use (16) and (17) to estimate  $E[I]$  for CRC-ZTCCs and CRC-TBCCs, respectively.

### D. Results, Analysis, and Expected List Rank of SLVD

Fig. 4 shows the trade-off between the SNR gap to the RCU bound and the average decoding complexity at the target FER  $10^{-4}$  for CRC-ZTCCs designed in Table I and CRC-TBCCs designed in Table II. The average decoding complexity of SLVD is evaluated according to the expressions in Sec. IV-C. We see that for a fixed  $v$  (ZT or TB), increasing the CRC degree  $m$  significantly reduces the gap to the RCU



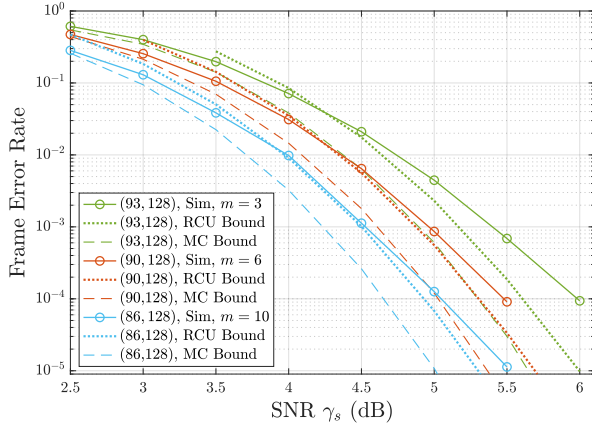


Fig. 5. FER vs. SNR for various CRC-TBCCs. The TBCC is generated with the  $(4, 3, 6)$  encoder  $H = (107, 135, 133, 141)$ . The DSO CRC polynomials of degrees 3, 6, and 10 are 0xB, 0x41, and 0x723, respectively. Values in parenthesis denote information length  $K$  and blocklength  $N$ , respectively.

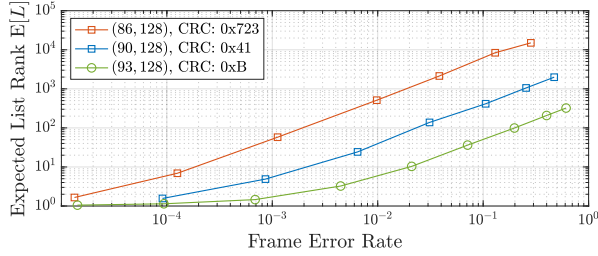


Fig. 6. Expected list rank  $E[L]$  vs. FER for SLVD of CRC-TBCC with blocklength  $N = 128$ . The TBCC is generated with  $H = (107, 135, 133, 141)$ . The DSO CRC polynomials of degrees 3, 6 and 10 are 0xB, 0x41, and 0x723.

bound, at the cost of a small increase in complexity. The minimum gap of 0.08 dB is achieved by the CRC-TBCC with  $v = 5$  and  $m = 10$ . However, for the same CRC degree  $m$ , increasing the overall constraint length  $v$  dramatically increases the complexity, while achieving a minimal reduction in the SNR gap to the RCU bound.

Fig. 5 shows the FER vs. SNR for three CRC-TBCCs at blocklength  $N = 128$ . At the target FER of  $10^{-4}$ , the SNR gap to the RCU bound is reduced to 0.1 dB for the CRC-TBCC with  $m = 10$  and  $v = 6$ . Fig. 6 shows the trade-off between the expected list rank  $E[L]$  and the FER. We see that the expected list rank  $E[L] < 7$  for achieving the target FER of  $10^{-4}$  for  $v = 6$  and  $m \leq 10$ , implying a low average decoding complexity of SLVD.

## V. CONCLUSION

This paper shows that a rate- $(\omega - 1)/\omega$  CC concatenated with a DSO CRC polynomial yields a good high-rate CRC-aided CC that approaches the RCU bound for the BI-AWGN channel. In particular, the best CRC-TBCCs approaches the RCU bound within 0.1 dB for FER  $10^{-4}$  at blocklength  $N = 128$  bits. Adding one bit to the CRC can improve the FER far more than adding an additional memory element to the CC does.

## ACKNOWLEDGMENT

The authors thank Dariush Divsalar for helpful discussions on construction of the dual trellis. We thank Ethan Liang and Linfang Wang for guidance and mentorship. We also thank Mai Vu and Mohammad Karimzadeh for helpful collaboration.

## REFERENCES

- [1] M. Rice, "Comparative analysis of two realizations for hybrid-ARQ error control," in *1994 IEEE Global Commun. Conf.*, 1994, pp. 115–119.
- [2] "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.
- [3] "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.
- [4] 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.
- [5] 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.
- [6] 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.
- [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] 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.
- [9] 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.
- [10] 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.
- [11] I. Tal and A. Vardy, "List decoding of polar codes," *IEEE Trans. Inf. Theory*, vol. 61, no. 5, pp. 2213–2226, 2015.
- [12] E. Arkan, "From sequential decoding to channel polarization and back again," [Online]. Available: <http://arxiv.org/abs/1908.09594>
- [13] H. Yang, E. Liang, M. Pan, and R. D. Wesel, "CRC-aided list decoding of convolutional codes in the short blocklength regime," *IEEE Trans. Inf. Theory*, Feb. 2022, early access. [Online]. Available: <https://doi.org/10.1109/TIT.2022.3150717>
- [14] N. Seshadri and C. E. W. Sundberg, "List Viterbi decoding algorithms with applications," *IEEE Trans. Commun.*, vol. 42, no. 234, pp. 313–323, Feb. 1994.
- [15] R. Schiavone, "Channel coding for massive IoT satellite systems," Master's thesis, Politecnico University of Turin (Polito), 2021.
- [16] M. Karimzadeh and M. Vu, "Optimal CRC design and serial list Viterbi decoding for multi-input convolutional codes," in *2020 IEEE Global Commun. Conf.*, 2020, pp. 1–6.
- [17] T. Yamada, H. Harashima, and H. Miyakawa, "A new maximum likelihood decoding of high rate convolutional codes using a trellis," *Elec. and Commun. in Japan Part I-commun.*, vol. 66, pp. 11–16, 1983.
- [18] S. Srinivasan and S. S. Pietrobon, "Decoding of high rate convolutional codes using the dual trellis," *IEEE Trans. Inf. Theory*, vol. 56, no. 1, pp. 273–295, 2010.
- [19] S. Lin and D. J. Costello, *Error Control Coding: fundamentals and applications*. New Jersey, USA: Pearson Prentice Hall, 2004.
- [20] H. Ma and J. Wolf, "On tail biting convolutional codes," *IEEE Trans. Commun.*, vol. 34, no. 2, pp. 104–111, Feb. 1986.
- [21] 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.