# An Efficient Algorithm for Designing Optimal CRCs for Tail-Biting Convolutional Codes

Hengjie Yang, Linfang Wang, Vincent Lau and Richard D. Wesel
Department of Electrical and Computer Engineering
University of California, Los Angeles, Los Angeles, CA 90095, USA
Email: {hengjie.yang, lfwang, vincentlau, wesel}@ucla.edu

***Abstract*—Cyclic redundancy check (CRC) codes combined with convolutional codes yield a powerful concatenated code that can be efficiently decoded using list decoding. To help design such systems, this paper presents an efficient algorithm for identifying the distance-spectrum-optimal (DSO) CRC polynomial for a given tail-biting convolutional code (TBCC) when the target undetected error rate (UER) is small. Lou *et al.* found that the DSO CRC design for a given zero-terminated convolutional code under low UER is equivalent to maximizing the undetected minimum distance (the minimum distance of the concatenated code). This paper applies the same principle to design the DSO CRC for a given TBCC under low target UER. Our algorithm is based on partitioning the tail-biting trellis into several disjoint sets of tail-biting paths that are closed under cyclic shifts. This paper shows that the tail-biting path in each set can be constructed by concatenating the irreducible error events (IEEs) and circularly shifting the resultant path. This motivates an efficient collection algorithm that aims at gathering IEEs, and a search algorithm that reconstructs the full list of error events with bounded distance of interest, which can be used to find the DSO CRC. Simulation results show that DSO CRCs can significantly outperform suboptimal CRCs in the low UER regime.**

## I. Introduction

Tail-biting convolutional codes (TBCCs) are simple and powerful codes in the short blocklength regime. Unlike the conventional zero-terminated convolutional code (ZTCC) whose trellis paths all begin and end in the zero state, a TBCC only requires that each trellis path starts and ends at the same state. This avoids the need for termination bits. TBCCs were first proposed by Ma and Wolf [1] as a modified version of the ZTCC to eliminate the rate loss caused by termination bits. Solomon and Tilborg [2] demonstrated the intriguing relation that any TBCC can be transformed into a quasi-cyclic code and conversely, many quasi-cyclic codes can be viewed as a TBCC with a small constraint length. Subsequently, it was shown that any linear block code can correspond to a tail-biting (TB) trellis representation and the code represented by such trellis is called a TB code [3], [4]. The significance of TB codes lies in the fact that they achieve the best minimum distance of codes in the short-to-medium blocklength regime [1], [5], [6].

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

Since the advent of TBCCs and TB codes, several authors proposed a variety of algorithms to decode a TBCC or a TB code, e.g., [1], [7]–[12]. These algorithms are based on either maximum likelihood (ML) or maximum a posteriori (MAP) criteria. For ML decoding algorithms, the wrap-around Viterbi algorithm (WAVA) [10] achieves the near-ML performance with the minimum complexity.

Cyclic redundancy checks (CRCs) are commonly used to detect whether a codeword is correctly received. Recently, with the development of 5G, CRC-aided list decoding of finite blocklength codes has received increasing popularity. CRC-aided list decoding can significantly help improve the code performance, e.g., [13]–[16]. Lou *et al.* [17] first designed the optimal CRC for a given ZTCC such that the concatenated CRC-ZTCC achieves the minimum undetected error rate (UER) when the target UER is low. The CRC they designed can be referred to as the *distance-spectrum-optimal* (DSO) CRC in the sense that the upper bound of the UER characterized by the full undetected distance spectrum is minimized and the upper bound is close to the true UER when the target UER is set low. However, DSO CRC design for a given TBCC is still missing from the literature. It is remarkable that a simple suboptimal CRC design [16] can nearly achieve the random coding union bound of Polyanskiy *et al.* [18].

The DSO CRC design principle under a low target UER parallels that of Lou *et al.*, which is equivalent to maximizing the *undetectable minimum distance* of the overall concatenated code. To this end, the first step is to gather a sufficient number of error events, i.e., TB paths, of distances less than some threshold. This can be accomplished by the *collection algorithm*. Then, the *search algorithm* is employed to find the DSO CRC polynomial that maximizes the undetectable minimum distance. For TBCCs, a trivial collection algorithm is to perform Viterbi search separately at each possible initial state to find all error events of a bounded distance. However, such an algorithm will be *inefficient* in collecting TB paths for a family of objective trellis lengths. If the objective trellis length changes to a smaller value, one has to redo the above procedure from scratch.

Unlike the trivial algorithm, this paper provides an *efficient* algorithm that supports the DSO CRC design of a given TBCC for a family of objective trellis lengths. The algorithm is based on partitioning the TB trellis into several disjoint sets of TB paths that are closed under cyclic shifts. Specifically, for a

feedforward convolutional encoder with $v$ memory elements and a specified blocklength, its TB trellis can be described as the union of all TB paths of the required length that start and end at any of the $2^v$ states. Each TB path can be categorized by a state through which it traverses. Let $\mathrm{TBP}(0)$ be the set of TB paths that traverse through state 0. Then, recursively define $\mathrm{TBP}(i)$, $1 \leq i \leq 2^v - 1$, as the set of TB paths that traverse through state $i$ but not through $0, 1, \ldots, i-1$. Clearly, the $2^v$ sets are disjoint and collectively contain all TB paths. Next, we introduce the concept of *irreducible error event* (IEE) of $\mathrm{TBP}(i)$, the atomic TB path starting at state $i$ but not passing states $0, 1, \ldots, i-1$ in between. Thus, each path in $\mathrm{TBP}(i)$ can be reconstructed by concatenating the corresponding IEEs and then circular shifting the resultant path. Since the set of IEEs can be reused for equal or smaller trellis lengths, our collection algorithm will be efficient compared to the trivial algorithm.

The paper is organized as follows. Sec. II reviews the preliminaries of the TBCC and TB trellises, and Lou *et al.*'s CRC design for ZTCCs. Sec. III introduces the partition of a TB trellis, IEEs, our DSO CRC design algorithm for TBCCs under low UERs, and a design example. Sec. IV concludes the entire paper.

## II. Preliminaries

### A. Construction of the TBCC

We briefly follow [1] in describing a TBCC. For ease of understanding, consider a feedforward, $(n, 1, v)$ convolutional code of rate $1/n$ and $v$ memory elements, albeit the design approach in this paper can be generalized to any feedforward, $(n, k, v)$ TBCC. For a binary information sequence of length $K$, $K \geq v$, we first use the last $v$ bits to initialize the convolutional encoder and ignore the outputs. Then the entire $K$-bit information sequence is fed into the encoder and the resultant $nK$-bit output is a TB codeword. As can be seen, the initial and final state of the codeword will be the same. In this way the rate loss caused by termination in a ZTCC is eliminated.

### B. Tail-Biting Trellises

We follow [4] in describing the tail-biting trellises. Let $V$ be a set of vertices (or states), $\mathcal{A}$ the set of output alphabet, and $E$ the set of ordered triples or edges $(v, a, v')$, 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, [4])**.** *A tail-biting (TB) 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 into $N$ vertex classes*

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

*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 $V_0$.*

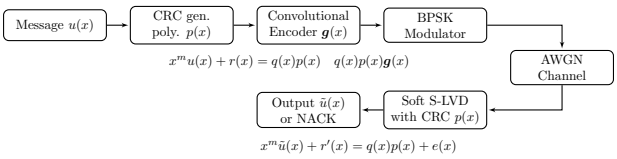


Fig. 1. Block diagram of a system employing CRC and convolutional codes.

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 conventional 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 $\boldsymbol{s} \in V_0 \times V_1 \times \cdots \times V_N$, $\boldsymbol{a} \in \mathcal{A}^N$. The code represented by trellis $T$ is the set of all output sequences $\boldsymbol{a}$ corresponding to all trellis paths $(\boldsymbol{s}, \boldsymbol{a})$ in $T$.

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 $s_0 = s_N$.

In this paper, we only consider the TB trellis $T$ of depth $N$ satisfying $V_0 = V_i$, $i = 1, 2, \ldots, N-1$. Clearly, the TB trellis generated by the feedforward, $(n, 1, v)$ convolutional encoder $\boldsymbol{g}(x)$ meets our condition.

### C. System Model and Lou et al's CRC Design Method

We briefly follow [17] in introducing their DSO CRC design scheme for a given ZTCC under a low target UER.

The basic system model is depicted in Fig. 1. Let us consider a $K$-bit information sequence represented as a binary polynomial $u(x)$ of degree no greater than $K-1$. Then, the $m$ parity check bits are calculated as the remainder $r(x)$ of $x^m u(x)$ divided by a degree-$m$ CRC generator polynomial $p(x)$. Therefore, the $(K+m)$-bit sequence described by $x^m u(x) + r(x)$ is divisible by $p(x)$, i.e., there exists a unique polynomial $q(x)$ such that $x^m u(x) + r(x) = q(x)p(x)$. Hence, the CRC-coded sequence can be concisely expressed as $q(x)p(x)$. Let $\boldsymbol{g}(x) = (g^{(1)}(x), \ldots, g^{(n)}(x))$ be the generator polynomial of a feedforward, rate-$1/n$ convolutional encoder. After feeding the CRC-coded sequence $q(x)p(x)$ into the encoder, the output $q(x)p(x)\boldsymbol{g}(x)$ is the final codeword of the ZTCC. The transmitter sends the BPSK-modulated sequence of $q(x)p(x)\boldsymbol{g}(x)$ through an additive white Gaussian noise (AWGN) channel. After receiving the channel outputs, the serial list Viterbi decoder (S-LVD) produces the most likely message sequence $\tilde{u}(x)$ if a codeword passes the CRC check before reaching the maximum list size. Otherwise, a negative acknowledgement (NACK) is output. Performance analysis of S-LVD can be found in [14]. An undetected error occurs if S-LVD erroneously identified a path corresponding to input

sequence $q(x)p(x)+e(x)$, where $e(x) \neq 0$ and is divisible by $p(x)$, as the maximum-likelihood (ML) path.

The fundamental design challenge is to identify the optimal CRC for the ZTCC generated by $\boldsymbol{g}(x)$ such that the UER is minimized. Lou *et al.* [17] showed that if the target UER is low enough, the UER will be dominated by the smallest-distance undetected error. Therefore, designing the DSO CRC polynomial is equivalent to designing the CRC with the maximum undetectable minimum distance. This is the essential motivation of Lou *et al*'s approach. In their design method, a CRC polynomial is removed from the candidate list if it possesses a smaller undetectable minimum distance or more undetected errors at the same distance. As we proceed to higher distances, the CRC candidate list is refined until only one candidate remains in the list. This candidate is the DSO CRC polynomial for the given ZTCC under the target UER.

Note that the DSO CRC polynomial is always the one that minimizes the upper bound of the UER characterized by the full undetected distance spectrum. If the target UER is not low enough, the above CRC design procedure does not necessarily yield the DSO CRC polynomial.

## III. Optimal CRC Design for the TBCC

In this paper, we consider the same system model as in Fig. 1 except replacing ZTCCs with TBCCs. The primary distinction between the two types of convolutional codes is that a TB error event can start at a nonzero state and remain in nonzero states on the trellis.

The fundamental DSO CRC design principle for a given convolutional code under a low target UER is analogous to that of Lou *et al.*, which is to maximize the minimum distance at which an undetectable TB error event first occurs, (or undetectable minimum distance). Formally speaking, the degree-$m$ DSO CRC design procedure involves two steps. First, the *collection algorithm* gathers a *sufficient* number of error events of distances less than some threshold $\tilde{d}$ and stores them for future use. By "sufficient", we mean that the number of error events is enough to sieve the unique, degree-$m$ CRC polynomial out of $2^{m-1}$ candidates[2]. Next, the *search algorithm* initializes a list of $2^{m-1}$ CRC candidates. Iterating from distance 1 to $\tilde{d}$, a candidate is removed from consideration if it possesses a smaller undetected minimum distance or more undetected errors at the same distance. Eventually, the last one in the list is the DSO CRC polynomial.

For TBCCs, the trivial collection algorithm is to perform Viterbi search at each initial state and then aggregate error events according to increasing distances. However, such an algorithm will be inefficient in designing DSO CRCs for a family of objective trellis lengths. The TB paths of one trellis length found by the trivial collection algorithm cannot be easily adapted to another trellis length.

To enable the design for a family of objective trellis lengths, we propose an efficient collection algorithm that finds sufficient number of IEEs. These IEEs can be reused to reconstruct TB paths of any objective length via concatenation and circularly shifting the resultant path. The motivation of our collection algorithm originates from the following partitioning of TB trellises.

[2]A CRC generator polynomial must have 1 as coefficients for both the scalar term and the degree-$m$ term.

### A. Partitioning of the Tail-Biting Trellis

For a given feedforward, $(n, 1, v)$ convolutional encoder $\boldsymbol{g}(x)$, let us consider the corresponding TB trellis $T = (V, E, \mathcal{A})$ defined on a given sequential time axis $\mathcal{I} = \{0, 1, \ldots, N\}$. Since $T$ can also be represented by the union of TB paths (each corresponding to a TBCC codeword), we categorize each TB path according to the states through which it traverses. Formally speaking, let

$$V_0^{(\pi)} = (\sigma_0, \sigma_1, \ldots, \sigma_{2^v-1}) \tag{2}$$

be a predetermined permutation of $V_0 = \{0, 1, \ldots, 2^v - 1\}$. Define the set of TB paths w.r.t. $V_0^{(\pi)}$ as

$$\begin{aligned}\mathrm{TBP}(\sigma_i) \triangleq \big\{&(\boldsymbol{s}, \boldsymbol{a}) \in V_0^{N+1} \times \mathcal{A}^N : s_0 = s_N; \\ &\exists j \in \mathcal{I} \text{ s.t. } s_j = \sigma_i;\ \forall j \in \mathcal{I},\ s_j \notin \{\sigma_0, \sigma_1, \ldots, \sigma_{i-1}\}\big\}, \\ &\forall i = 0, 1, \ldots, 2^v - 1.\end{aligned} \tag{3}$$

In words, the set of $\mathrm{TBP}(\sigma_0)$ only contains TB paths that traverse through state $\sigma_0$; the set of $\mathrm{TBP}(\sigma_1)$ contains TB paths that traverse through state $\sigma_1$ but not $\sigma_0$; so on and so forth. Clearly, all sets $\mathrm{TBP}(\sigma)$, $\sigma \in V_0^{(\pi)}$, form a partition of the TB trellis $T$, i.e.,

$$\mathrm{TBP}(\sigma_i) \cap \mathrm{TBP}(\sigma_j) = \varnothing, \quad \text{if } \sigma_i \neq \sigma_j \tag{4}$$

$$\bigcup_{\sigma \in V_0^{(\pi)}} \mathrm{TBP}(\sigma) = T. \tag{5}$$

An important property of the above decomposition is that each set $\mathrm{TBP}(\sigma)$ is closed under cyclic shifts.

**Theorem 1.** *Any cyclic shift of a TB path* $(\boldsymbol{s}, \boldsymbol{a}) \in \mathrm{TBP}(\sigma)$ *is also a TB path in* $\mathrm{TBP}(\sigma)$.

*Proof:* Since circularly shifting a TB path $(\boldsymbol{s}, \boldsymbol{a})$ on a TB trellis $T$ defined on a given sequential time axis $\mathcal{I} = \{0, 1, \ldots, N\}$ is equivalent to circularly shifting $\mathcal{I}$ around $T$ defined on a circular time axis, this preserves the sequence of states (or vertices) through which the TB path $(\boldsymbol{s}, \boldsymbol{a})$ traverses. Hence, the statement in Theorem 1 holds. ∎

Inspired by the concepts of basis and linear combination in a vector space, we can consider the set of IEEs starting at state $\sigma$ as a basis from which each TB path of length $N$ in $\mathrm{TBP}(\sigma)$ may be constructed. The next section shows that this is accomplished by concatenating the IEEs and then circularly shifting the resultant TB path.

**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 $\sigma$ w.r.t. $V_0^{(\pi)} = (\sigma_0, \sigma_1, \ldots, \sigma_{2^v-1})$ is defined as*

$$\mathrm{IEE}(\sigma_i) \triangleq \bigcup_{j=1,2,\ldots,N} \overline{\mathrm{IEE}}(\sigma_i, j),\ \forall i = 0, 1, \ldots, 2^v - 1, \tag{6}$$

**Algorithm 1** The Collection Algorithm

**Input:** The TB trellis $T$, threshold $\tilde{d}$, permutation $V_0^{(\pi)}$
**Output:** The list of IEEs $\mathcal{L}_{\rm IEE}(\tilde{d}) = \{(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{u})\}$

Initialize lists $\mathcal{L}_\sigma$ to be empty for all $\sigma \in V_0^{(\pi)}$;
**for** $i \leftarrow 0, 1, \ldots, |V_0^{(\pi)}| - 1$ **do**
    Perform Viterbi search at $\sigma_i$ on $T$ to collect list $\mathcal{L}_{\sigma_i}(\tilde{d})$ of all IEEs of distances less than $\tilde{d}$;
**end for**
**return** $\mathcal{L}_{\rm IEE}(\tilde{d}) \leftarrow \bigcup_{\sigma \in V_0^{(\pi)}} \mathcal{L}_\sigma(\tilde{d})$;

**Algorithm 2** The Search Algorithm

**Input:** The length $N$, degree $m$, list of IEEs $\mathcal{L}_{\rm IEE}(\tilde{d})$
**Output:** The optimal degree-$m$ CRC gen. poly. $p(x)$

Initialize the list $\mathcal{L}_{\rm CRC}$ of $2^{m-1}$ CRC candidates, the empty list $\mathcal{L}_{\rm TBP}(d)$ of TBPs, $d = 0, 1, \ldots, \tilde{d} - 1$;
**for** $d \leftarrow 1, \ldots, \tilde{d} - 1$ **do**
    Construct new TBPs $(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{u})$ from $\mathcal{L}_{\rm IEE}(\tilde{d})$ s.t. $w_H(\boldsymbol{a}) = d$, $|\boldsymbol{s}| = N$, via concatenating or cyclic shifting;
    $\mathcal{L}_{\rm TBP}(d) \leftarrow \mathcal{L}_{\rm TBP}(d) \cup \{(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{u})\}$;
**end for**
$\mathrm{Candi}(1) \leftarrow \mathcal{L}_{\rm CRC}$;
**for** $d \leftarrow 1, \ldots, \tilde{d} - 1$ **do**
    **for** $p_i(x) \in \mathrm{Candi}(d)$ **do**
        Pass all $\boldsymbol{u}(x) \in \mathcal{L}_{\rm TBP}(d)$ to $p_i(x)$;
        $C_i \leftarrow$ the number of divisible $\boldsymbol{u}(x)$ 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**

*where*

$$\overline{\rm IEE}(\sigma_i, j) \triangleq \{(\boldsymbol{s}, \boldsymbol{a}) \in V_0^{j+1} \times \mathcal{A}^j : s_0 = s_j = \sigma_i; \; s_{j'} \notin \{\sigma_0, \sigma_1, \ldots, \sigma_i\} \text{ for all } j', 0 < j' < j\}. \quad (7)$$

**Theorem 2.** *Every TB path $(\boldsymbol{s}, \boldsymbol{a}) \in \mathrm{TBP}(\sigma)$ can be constructed from the IEEs in $\mathrm{IEE}(\sigma)$ via concatenation and cyclic shifting operations.*

*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

$$\boldsymbol{s}^{(1)} = (s_0, s_1, \ldots, s_j), \; \boldsymbol{a}^{(1)} = (a_0, a_1, \ldots, a_{j-1}),$$

$$\boldsymbol{s}^{(2)} = (s_j, s_{j+1}, \ldots, s_N), \; \boldsymbol{a}^{(2)} = (a_j, a_{j+1}, \ldots, a_{N-1}).$$

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 procedures 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 $s_0 = s_{j*} = \sigma$, $s_{j'} \neq \sigma$, $\forall j' \in (0, j^*)$ which is clearly an element of $\mathrm{IEE}(\sigma)$. Thus, we end up obtaining sub-paths that are all elements of $\mathrm{IEE}(\sigma)$. Concatenating them yields the circularly shifted version of TB path $(\boldsymbol{s}^{(0)}, \boldsymbol{a}^{(0)})$. ■

Theorem 2 indicates that collecting IEEs starting at every state $\sigma$ is enough to reconstruct all TB paths in set $\mathrm{TBP}(\sigma)$. This underlies the collection and search algorithm we are about to propose. Note that the collection of IEEs only relies on the distance threshold $\tilde{d}$ assuming sufficiently long search depth. Once we collect *all* IEEs of distance less than $\tilde{d}$, these IEEs can be reused to reconstruct TB path of distance less than $\tilde{d}$ and of any objective length.

### *B. The CRC Design Algorithm for the TBCC*

For the TB trellis $T$ of a feedforward, $(n, 1, v)$ convolutional encoder $\boldsymbol{g}(x)$, let $(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{u})$ denote the triple of states $\boldsymbol{s}$, outputs $\boldsymbol{a}$ and inputs $\boldsymbol{u}$, where the inputs $\boldsymbol{u}$ are uniquely determined by state transitions $s_i \to s_{i+1}$, $i = 0, 1, \ldots, N-1$. Motivated by the partitioning of $T$ and IEEs in Sec. III-A, we propose the collection algorithm and search algorithm to design the degree-$m$ DSO CRC polynomial $p(x)$, as demonstrated in Algorithm 1 and 2, respectively. In the pseudo-code description, we use $\boldsymbol{u}$ and $\boldsymbol{u}(x)$ interchangeably to denote the sequence and the corresponding polynomial, respectively.

To visualize the process of the collection algorithm, consider the state diagram of the convolutional code, where each cycle in the state diagram with a length equal to the trellis depth represents a TB path. For a given ordering of states $V_0^{(\pi)} = (\sigma_0, \sigma_1, \ldots, \sigma_{2^v-1})$, once the algorithm finds all IEEs starting from $\sigma_0$, the state diagram is reduced by removing $\sigma_0$ and the incoming and outgoing edges associated with it. The algorithm then finds the IEEs starting at $\sigma_1$ on the reduced state diagram. Repeating the above procedure, the collection algorithm is able to find all sets of IEEs.

The search algorithm first reconstructs each length-$N$ TB path of bounded distance through concatenation of IEEs and cyclic shift, and then finds the DSO CRC polynomial. The reconstruction step can be accomplished via dynamic programming. Specifically, let $\mathcal{L}(w, l)$ be the list of TB paths of weight $w$ and of length $l$, $0 \le w < \tilde{d}$, $1 \le l \le N$. Thus, given a new IEE $(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{u})$ of weight $w_H(\boldsymbol{a})$ and of length $|\boldsymbol{s}|$ satisfying $w_H(\boldsymbol{a}) \le w$ and $|\boldsymbol{s}| < l$,

$$\mathcal{L}(w, l) = \mathcal{L}(w, l) \cup \{\mathcal{L}(w - w_H(\boldsymbol{a}), l - |\boldsymbol{s}|) + (\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{u})\}, \quad (8)$$

where $+$ denotes the element-wise concatenation. Eventually, the lists $\mathcal{L}(w, N)$, $w < \tilde{d}$ stores all length-$N$ TB paths of distance less than $\tilde{d}$.

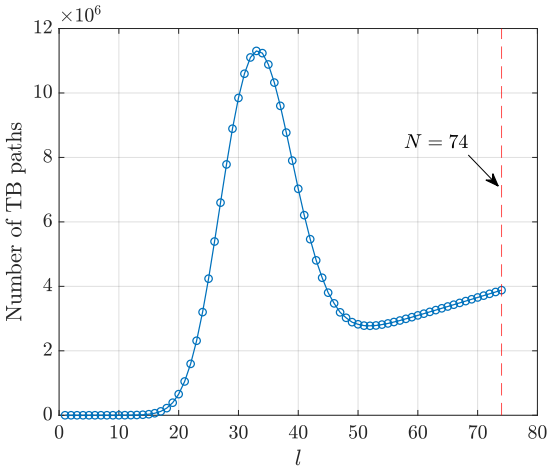


Fig. 2. The number of TB paths of length equal to $l$ vs. length $l$ for TBCC $(133, 171)$ with $\tilde{d} = 22$, $N = 74$.

TABLE I
COMPARISON OF UNDETECTED DISTANCE SPECTRA BETWEEN THE DEGREE-6 DSO CRCS AND THE SUBOPTIMAL CRCS IN [16] TBCC $(13, 17)$ AND $N = 70$ BITS. THE DSO CRCS ARE HIGHLIGHTED. $A_d$ OF DISTANCES BETWEEN 8 TO 10 ARE ALL ZEROS THUS OMITTED.

| $v$ | TBCC | CRC | Undetected Distance Spectra $A_d$ | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | 7 | 11 | 12 | 13 | 14 | 15 | 16 | 17 |
| 3 | $(13, 17)$ | 0x43 | 1 | 8 | 198 | 758 | 1114 | 2814 | 7375 | 18473 |
| | | **0x63** | 0 | 0 | 735 | 0 | 2310 | 0 | 13965 | 0 |

*1) Space complexity of the search algorithm:* The space complexity is proportional to the total number of bits required to represent all TB paths in all lists $\mathcal{L}(w, l)$, $0 \leq w < \tilde{d}$, $1 \leq l \leq N$. If the distance threshold $\tilde{d}$ is much less than the target length $N$, the growth of the number of TB paths of length equal to $l$ eventually becomes polynomial in $l$. Fig. 2 shows the growth of number of TB paths of length $l$ for TBCC $(133, 171)$ with $\tilde{d} = 22$ and $N = 74$. As can be seen, if $l \geq 3\tilde{d}$, the growth then becomes polynomial. This suggests that space complexity is polynomial in $N$ provided that $N > 3\tilde{d}$.

*2) Choices of distance threshold $\tilde{d}$:* In order to design the DSO CRC polynomial for a given TBCC, one has to select an appropriate $\tilde{d}$. Empirically, $\tilde{d}$ ranges from $2d_{\text{free}}$ to $3d_{\text{free}}$ for designing a CRC polynomial of degree $m \leq 10$.

*3) Choices of $V_0^{(\pi)}$:* We note that in practice, the ordering of $V_0^{(\pi)}$ exerts a negligible influence on the space complexity. Hence, the natural ordering suffices for the DSO CRC design.

### C. Example: Degree-6 DSO CRC for TBCC $(13, 17)$

As an example, we design the degree-6 DSO CRC polynomial for TBCC $(13, 17)$ with $K = 64$ bits under target UER $P_e = 10^{-10}$. Since the UER is low enough, the UER in this regime will be dominated by the smallest undetected errors.

Table I presents the undetected distance spectra up to $\tilde{d} = 17$ for the degree-6 suboptimal CRC polynomial 0x43 designed in [16] and our degree-6 DSO CRC polynomial 0x63 for TBCC $(13, 17)$ and $K = 64$ bits with overall trellis length $N = K + m = 70$. With the full undetected distance spectrum, the UER of a given CRC and TBCC can be upper bounded by the union bound of probability, namely,

$$P_e \leq \sum_{d=1}^{d_{\max}} A_d Q\left(\sqrt{\frac{dE_s}{N_0}}\right), \tag{9}$$

where $d_{\max}$ is the maximum possible distance of the finite-length TBCC, and $Q(x)$ is the tail probability function of standard normal distribution. In practice, the full undetected distance spectrum can be computationally expensive. Instead, we will only calculate the bound in (9) up to $\tilde{d}$ and such a bound is known as the *truncated union bound*. Despite the resulting computational inaccuracy, the truncated union bound still serves as a good estimate in the low UER regime.

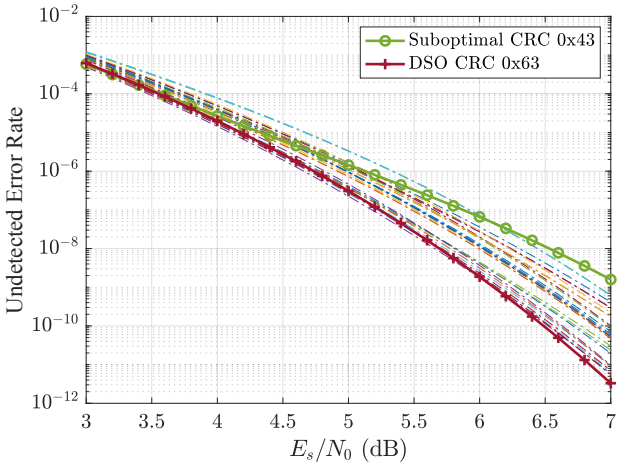


Fig. 3. Assume the target UER $P_e = 10^{-10}$. The truncated union bound vs. SNR for all 32 degree-6 CRC polynomial candidates for TBCC $(13, 17)$ and $K = 64$ bits. The degree-6 DSO CRC polynomial 0x63 and degree-6 suboptimal CRC polynomial 0x43 designed in [16] are highlighted.

Fig. 3 shows the truncated union bounds up to distance $\tilde{d} = 17$ of all 32 degree-6 candidate CRC polynomials for TBCC $(13, 17)$. The curves corresponding to the suboptimal CRC 0x43 and the DSO CRC 0x63 are highlighted. As can be seen, the DSO CRC outperforms the suboptimal CRC by 2 orders of magnitudes at 6.5 dB, the SNR at which the DSO CRC attains the target UER of $10^{-10}$. However at 3 dB, the DSO CRC 0x63 designed for 6.5 dB performs worse than the CRC 0x43 and thus fails to remain optimal. This demonstrates that the DSO condition indeed depends on the operating SNR or target UER.

## IV. CONCLUSION

In this paper, we propose an efficient algorithm for designing DSO CRC polynomials for any specified TBCC for a low target UER. The algorithm is based on decomposing the TB trellis into several disjoint sets of TB paths that are closed under cyclic shifts. We also showed that the TB path in each set can be constructed from the IEEs via concatenation and cyclic shift. The use of IEEs enables the DSO CRC design for a family of trellis lengths (or the corresponding blocklengths). Our results demonstrate that for low target UER, DSO CRCs can significantly outperform suboptimal CRCs.

## REFERENCES


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

[2] G. Solomon and H. C. A. Tilborg, "A connection between block and convolutional codes," *SIAM Journal on Applied Mathematics*, vol. 37, no. 2, pp. 358–369, 1979.

[3] A. R. Calderbank, G. D. Forney, and A. Vardy, "Minimal tail-biting trellises: the golay code and more," *IEEE Trans. Inf. Theory*, vol. 45, no. 5, pp. 1435–1455, July 1999.

[4] 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.

[5] P. Stahl, J. B. Anderson, and R. Johannesson, "Optimal and near-optimal encoders for short and moderate-length tail-biting trellises," *IEEE Trans. Inf. Theory*, vol. 45, no. 7, pp. 2562–2571, Nov 1999.

[6] I. E. Bocharova, R. Johannesson, B. D. Kudryashov, and P. Stahl, "Tailbiting codes: bounds and search results," *IEEE Trans. Inf. Theory*, vol. 48, no. 1, pp. 137–148, Jan 2002.

[7] 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.

[8] 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.

[9] J. B. Anderson and S. M. Hladik, "Tailbiting map decoders," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 2, pp. 297–302, Feb 1998.

[10] 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.

[11] Tsao-Tsen Chen and Shiau-He Tsai, "Reduced-complexity wrap-around viterbi algorithm for decoding tail-biting convolutional codes," in *2008 14th European Wireless Conf.*, June 2008, pp. 1–6.

[12] A. R. Williamson, M. J. Marshall, and R. D. Wesel, "Reliability-output decoding of tail-biting convolutional codes," *IEEE Trans. Commun.*, vol. 62, no. 6, pp. 1768–1778, June 2014.

[13] K. Niu and K. Chen, "CRC-aided decoding of polar codes," *IEEE Commun. Lett.*, vol. 16, no. 10, pp. 1668–1671, October 2012.

[14] 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.

[15] 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 Communication*, vol. 34, pp. 66 – 79, 2019.

[16] 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)*, Dec 2019.

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

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