Double Serially Concatenated Convolutional Codes With Jointly Designed S-Type Permutors

Axel Huebner, *Member, IEEE*, Jörg Kliewer, *Senior Member, IEEE*, and Daniel J. Costello Jr., *Life Fellow, IEEE*

***Abstract*—In this paper, the design of double serially concatenated convolutional codes with S-type permutors, i.e., permutors that provide a nontrivial separation, is considered. Based on a newly introduced parameter, namely, the so-called symbol span, a joint design of the outer and inner permutor is presented and its impact on the minimum distance of the overall code is analyzed. It is shown that a lower bound on the minimum distance that is given by the product of the free distances of all three component codes can be guaranteed. Design tables and simulation results are presented that include comparisons with single serially concatenated convolutional codes. In addition, a comparison with double/generalized repeat accumulate codes is briefly sketched.**

***Index Terms*—Double serially concatenated convolutional codes, S-type permutors, serially concatenated convolutional codes, spreading factor.**

I. Introduction

RECENTLY, double repeat accumulate codes (DRACs) and generalized repeat accumulate codes (GRACs) [2]–[4] have gained in importance as a result of their excellent performance combined with low encoding and decoding complexity. Both are generated by component encoders and permutors that are alternately cascaded in a chain, where rate is only spent on the most outer code. In the case of DRACs, the outer code is a repetition code and for GRACs it is a convolutional code. All other codes have a rate of one, i.e., they are so-called accumulate codes that are generated by memory one recursive convolutional encoders and do not contribute any additional redundancy to the codeword. Such codes are, in fact, very weak when considered by themselves. However, in such a cascade they lead to excellent performance while saving decoding complexity due to their simple structure.

Manuscript received August 26, 2007; revised June 03, 2009. Current version published November 20, 2009. This work was supported in part by the German Research Foundation (DFG) under Grant KL 1080/3-1, the University of Notre Dame Faculty Research Program, the National Science Foundation (NSF) under Grants CCF08-30651, CCF08-30666, and CCF05-15012, and by NASA under Grant NNX07AK536. The material in this paper was presented in part at the IEEE Information Theory Workshop, Punta del Este, Uruguay, March 2006.

A. Huebner was with the Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN 46556 USA. He is now with Infineon Technologies AG, Am Campeon 1-12, 85579 Neubiberg, Germany (e-mail: axel.huebner@infineon.com).

J. Kliewer was with the Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN 46556 USA. He is now with the Klipsch School of Electrical and Computer Engineering, New Mexico State University, Las Cruces, NM 88003 USA (e-mail: jkliewer@nmsu.edu).

D. J. Costello, Jr. is with the Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN 46556 USA (e-mail: costello.2@nd.edu).

Communicated by H.-A. Loeliger, Associate Editor for Coding Techniques.

Digital Object Identifier 10.1109/TIT.2009.2032804

A more general scheme is that of double serially concatenated convolutional codes (DSCCCs) [5] that was introduced as an extension of the idea of serially concatenated convolutional codes (SCCCs) [6]. Compared to single (or conventional) SCCCs, the encoding of DSCCCs benefits from an additional encoder and an additional permutor—similar to DRACs and GRACs, but with an arbitrary rate allocation for the three component codes and an arbitrary choice for the component encoders themselves. While most of the results for DRACs and GRACs in the literature are either ensemble analyses for uniform interleaving, which give insight into the average performance of these codes, or computer simulation results for random permutors, we address the permutor design problem for DSCCCs, with a goal of guaranteeing a certain minimum distance for practically relevant block lengths. Related work was recently proposed in [7], where permutor designs for DRACs are considered. In particular, for an inner random permutor, an explicit construction for the outer permutor is given such that the minimum distance grows linearly with block length. By contrast, in this work we provide explicit constructions for both inner and outer permutors for the more general class of DSCCCs such that the minimum distance is lower bounded by the product of the free distances of the three component codes.

Based on so-called S-type permutors, i.e., permutors that provide a nontrivial separation (also known as the spreading factor [8]), that are known to result in large minimum distances for single SCCCs, a joint permutor design for DSCCCs is presented. Specifically, a new permutor parameter, the so-called symbol span, is introduced that allows an adaptation between the inner and the outer permutor. Using the symbol span properties, a construction method for a permutor pair is given which results in a strong minimum distance lower bound. In addition, lower bounds on the required outer and inner permutor sizes and tables which summarize guaranteed minimum distances for different choices of component codes are presented. Simulation results that confirm the theoretical results and compare the performance of DSCCCs with single SCCCs are given. Finally, a brief comparison with DRACs and GRACs is included.

The paper is organized as follows. Section II briefly introduces the encoding of DSCCCs and some fundamental permutor properties. Then, a joint permutor design is presented and the distance properties of DSCCCs emerging from such a design are discussed. Realization aspects are presented in Section III and design tables are included. In Section IV, simulation results are shown that include comparisons with single SCCCs. In addition, DRACs and GRACs are briefly discussed and they are compared to DSCCCs on the basis of minimum distance bounds

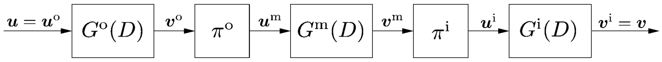

Fig. 1. Block diagram of a double serially concatenated convolutional encoder.

and simulation results. In Section V, we give a short summary of the main results.

II. Encoder Design and Minimum-Distance Bound

Similar to the one permutor in the case of a single SCCC, for a DSCCC the outer and inner permutors together essentially determine the distance spectrum and, in particular, the minimum distance once the component encoders are chosen. Therefore, it is of particular interest to design the outer and inner permutor jointly.

In the following, we briefly summarize the encoding of DSCCCs and the fundamental permutor properties in Section II-A. In addition, a new permutor parameter, the so-called symbol span, is introduced. Then, based on both the symbol separation and the symbol span, in Section II-B a novel joint design of the outer and inner permutor is presented and a lower bound on the minimum distance of a DSCCC with such a permutor pair is given.

A. Encoder and Permutor Fundamentals

In Fig. 1, the encoding of a DSCCC is shown. The outer rate $R^{\mathrm{o}} = b^{\mathrm{o}}/c^{\mathrm{o}}$, memory m^{o} convolutional encoder generates the code sequence $\boldsymbol{v}^{\mathrm{o}}$ from the information sequence $\boldsymbol{u} = \boldsymbol{u}^{\mathrm{o}}$ of length k, where k is a multiple of b^{o}. The encoding of $\boldsymbol{u}^{\mathrm{o}}$ is terminated by appending a tail of length $b^{\mathrm{o}}m^{\mathrm{o}}$. The code sequence $\boldsymbol{v}^{\mathrm{o}}$ of length n^{o} is rearranged by the outer block permutor, which realizes the permutation π^{o} of length $N^{\mathrm{o}} = n^{\mathrm{o}}$, resulting in the input sequence $\boldsymbol{u}^{\mathrm{m}}$ to the middle encoder. The middle encoder is a recursive systematic convolutional encoder of rate $R^{\mathrm{m}} = b^{\mathrm{m}}/c^{\mathrm{m}}$ and memory m^{m}. Assuming n^{o} is a multiple of b^{m}, the middle encoder is terminated by a tail of length $b^{\mathrm{m}}m^{\mathrm{m}}$. Its code sequence $\boldsymbol{v}^{\mathrm{m}}$ of length n^{m} is then mapped to the sequence $\boldsymbol{u}^{\mathrm{i}}$ by the inner block permutor, which realizes the permutation π^{i} of length $N^{\mathrm{i}} = n^{\mathrm{m}}$. The sequence $\boldsymbol{u}^{\mathrm{i}}$ is fed to the inner recursive systematic convolutional encoder of rate $R^{\mathrm{i}} = b^{\mathrm{i}}/c^{\mathrm{i}}$ and memory m^{i}, where, assuming n^{m} is a multiple of b^{i}, the inner encoder is terminated by appending $b^{\mathrm{i}}m^{\mathrm{i}}$ symbols. The output sequence $\boldsymbol{v}^{\mathrm{i}}$ of the inner encoder is the length n code sequence $\boldsymbol{v}$ of the DSCCC. Hence, its rate is given by

$$R = \frac{k}{n} = \frac{kR^{\mathrm{o}}R^{\mathrm{m}}R^{\mathrm{i}}}{k + b^{\mathrm{o}}m^{\mathrm{o}} + R^{\mathrm{o}}b^{\mathrm{m}}m^{\mathrm{m}} + R^{\mathrm{o}}R^{\mathrm{m}}b^{\mathrm{i}}m^{\mathrm{i}}} \quad (1)$$

$$\leq R^{\mathrm{o}}R^{\mathrm{m}}R^{\mathrm{i}} \quad (2)$$

where the *fractional rate loss* due to termination of the component encoders is negligible for code dimensions $k \gg b^{\mathrm{o}}m^{\mathrm{o}} + R^{\mathrm{o}}b^{\mathrm{m}}m^{\mathrm{m}} + R^{\mathrm{o}}R^{\mathrm{m}}b^{\mathrm{i}}m^{\mathrm{i}}$. Furthermore, $d_{\mathrm{free}}^{\mathrm{o}}$, $d_{\mathrm{free}}^{\mathrm{m}}$, and $d_{\mathrm{free}}^{\mathrm{i}}$ denote the minimum distances of the outer, middle, and inner codes, respectively. Note that a similar DSCCC can be generated using outer, middle, and inner encoders of rate $R^{\mathrm{o}} = R^{\mathrm{m}} = R^{\mathrm{i}} = 1/2$ with subsequent uniform puncturing.

In the following, we recall some basic properties of a permutor and its corresponding permutation π, including the symbol separation (also known as spreading factor [8]), and introduce the new concept of symbol span.

A *permutor* is a single-input single-output device that rearranges the temporal order of a sequence in a one-to-one deterministic manner without changing the symbol alphabet. The input symbol x_i to the permutor at time instant i is written to the output symbol $y_{\pi(i)}$ at time instant $\pi(i)$

$$y_{\pi(i)} = x_i \quad (3)$$

where $i \in \{0, 1, \ldots, N-1\}$ and N is the permutation length.

An important parameter describing the relative positions of two symbols before and after permutation is the so-called symbol separation (spreading factor [8]).

Definition 1: A permutation π of length N is said to have *symbol separation* (s, t) *(spreading factor)* if it satisfies

$$|i - i'| < s \quad \Rightarrow \quad |\pi(i) - \pi(i')| \geq t \quad (4)$$

for any $i \neq i'$, $i, i' \in \{0, 1, \ldots, N-1\}$, where $s, t \in \{0, 1, \ldots, N-1\}$ and s is called the *input separation* and t is called the *output separation*.

Definition 2: Let $t(i, s)$ be the maximum output separation t for $i \in \{0, 1, \ldots, N-1\}$ and a given input separation $s \in \{2, 3, \ldots, N-1\}$ such that the condition on the output separation in (4) is satisfied with equality. Then, a separation is called trivial if

$$\min_i \{t(i, 2)\} = 1 \quad (5)$$

and nontrivial otherwise.

A permutation π having nontrivial separation is denoted as an (s, t)*-permutation*, and the corresponding permutor as an (s, t)*-permutor*. In general, an (s, t)-permutation can contain additional randomness unless it is highly structured, such as the permutation of a block interleaver[1] that yields the largest possible separation for a given size N [9]. In the following, we refer to (s, t)-permutors (including the well-known semi-random, or S-random permutors [10], where S is usually given by the minimum of s and t) and block interleavers, i.e., the class of all permutors providing nontrivial separation, as S*-type permutors*.

The output separation t gives a lower bound on the minimum spacing of any two symbols in the output sequence of the permutor that lie in a window of size s in the input sequence to the permutor. In the following, we introduce a parameter pair that considers the opposite, namely, an upper bound on the maximum spacing of any two symbols in the output sequence that lie in a window of size S in the input sequence.

Definition 3: A permutation π of length N is said to have *symbol span* (S, T) if it satisfies

$$|i - i'| < S \quad \Rightarrow \quad |\pi(i) - \pi(i')| \leq T \quad (6)$$

[1]Generally, a device that realizes a row-column permutation, i.e., a permutation where the input sequence is written row-wise into a matrix and read out column-wise, is called a block interleaver.

for any $i \neq i'$, $i, i' \in \{0, 1, \ldots, N-1\}$, where $S, T \in \{0, 1, \ldots, N-1\}$ and S is called the *input span* and T is called the *output span*.

Let π be a permutation with nontrivial separation (s, t). If we choose the input span $S \leq s$ in (6), then it follows immediately that the output span T satisfies $T \geq (S-1)t$. If $S > s$, then T is at least $(s-1)t$, such that in general a lower bound on the output span T is given by

$$T \geq (\min\{s, S\} - 1)t \tag{7}$$

for an (s, t)-permutation (see Fig. 5 in Appendix I). For a permutation with trivial separation (see (5)), $T \geq S - 1$.

In the following, we call a permutation π that has both a symbol separation (s, t) and a symbol span (S, T) a permutation having *span-separation* $(s, t; S, T)$ and an $(s, t; S, T)$*-permutation*, respectively, and the corresponding permutor an $(s, t; S, T)$*-permutor*.

From [11], [12], we know that the separation is of particular interest in the design of SCCCs, since the minimum distance can be guaranteed to be lower bounded by the product of the free distances of the component codes, i.e., $d_{\min} \geq d_{\text{free}}^{\text{o}} d_{\text{free}}^{\text{i}}$ for SCCCs, if the separation of the permutation is sufficiently large. Now, using a permutation with sufficiently large span-separation $(s, t; S, T)$, not only the product distance can be guaranteed for an SCCC, but also the size of the window containing this weight can be bounded. Regarding the concatenation of the outer and middle encoder of a DSCCC as an SCCC, this fact enables us to adapt the inner permutor to the output sequence of the middle encoder, which is essentially determined by the outer permutor, such that a joint permutor design is possible. In the following section, such a joint permutor design is presented, where the inner permutation is adapted to the middle encoder output sequence in order to obtain a guaranteed minimum distance equal to the product of the free distance of the inner code and the minimum distance at the output of the middle encoder. This quantity, in turn, is lower bounded by the product of the free distances of the outer and middle codes if the separation of the outer permutation is chosen to be sufficiently large. Thus, the overall minimum distance is lower bounded by the product of the free distances of the outer, middle, and inner convolutional codes, i.e., $d_{\min} \geq d_{\text{free}}^{\text{o}} d_{\text{free}}^{\text{m}} d_{\text{free}}^{\text{i}}$ for DSCCCs.

B. *Minimum-Distance Properties of DSCCCs*

To effectively design DSCCCs, the inner permutor must be adapted to the outer permutor. Also, both permutors should be adapted to the component encoders, i.e., the outer permutor to the outer and middle encoder and the inner permutor to the middle and inner encoder, respectively. To this end, the permutor parameters, namely, the span-separation, are calculated using specific encoder parameters, in particular, minimum distances and active distances [13], [14]. To simplify the computation of the permutor parameters, we use affine lower bounds on the active distances [15]. Before stating the main result on the minimum distance of DSCCCs with S-type permutors as a theorem, we briefly recall the concept of active distances [13], [14] and affine lower bounds on active distances [15] for convolutional codes.

Consider a convolutional encoder of memory m and rate $R = b/c$ with encoder state sequence $\boldsymbol{\sigma} = \boldsymbol{\sigma}_0 \boldsymbol{\sigma}_1 \ldots$ that corresponds to the sequence of information b-tuples $\boldsymbol{u} = \boldsymbol{u}_0 \boldsymbol{u}_1 \ldots$ and the sequence of code c-tuples $\boldsymbol{v} = \boldsymbol{v}_0 \boldsymbol{v}_1 \ldots$, where a particular state sequence $\boldsymbol{\sigma}_{[m, m+j+1]}$ starting at time m and ending at time $m + j + 1$ corresponds to the code sequence $\boldsymbol{v}_{[m, m+j]}$. Let

$$\begin{aligned} \mathcal{S}_{[t_1, t_2]}^{\boldsymbol{\sigma}_1, \boldsymbol{\sigma}_2} = \{ & \boldsymbol{\sigma}_{[t_1, t_2]} | \boldsymbol{\sigma}_{t_1} = \boldsymbol{\sigma}_1, \boldsymbol{\sigma}_{t_2} = \boldsymbol{\sigma}_2, \\ & \text{with } (\boldsymbol{\sigma}_t, \boldsymbol{\sigma}_{t+1}) \neq (\mathbf{0}, \mathbf{0}) \text{ for } t_1 \leq t < t_2 \} \end{aligned} \tag{8}$$

denote the set of all encoder state sequences $\boldsymbol{\sigma}_{[t_1, t_2]}$ starting at time t_1 in state $\boldsymbol{\sigma}_1$ and ending at time t_2 in state $\boldsymbol{\sigma}_2$ without any transition from the all-zero state to the all-zero state, i.e., such that the encoder does not stay in the all-zero state for consecutive time units.[2]

Definition 4: Consider the convolutional code $\mathcal{C}$ generated by a rational generator matrix $G(D)$ realized in controller canonical form, where the minimum constraint length of the encoder is denoted as $\nu_{\min}$. Then, with the Hamming weight of a sequence denoted by $w_{\text{H}}(\cdot)$, the *jth-order active burst distance* is

$$a_j^{\text{b}} = \min_{\mathcal{S}_{[0, j+1]}^{\mathbf{0}, \mathbf{0}}} \left\{ w_{\text{H}} \left(\boldsymbol{v}_{[0, j]} \right) \right\} \tag{9}$$

where $j \geq \nu_{\min}$. The *jth-order active column distance* is

$$a_j^{\text{c}} = \min_{\mathcal{S}_{[0, j+1]}^{\mathbf{0}, \boldsymbol{\sigma}}} \left\{ w_{\text{H}} \left(\boldsymbol{v}_{[0, j]} \right) \right\} \tag{10}$$

and the *jth-order active reverse column distance* is

$$a_j^{\text{rc}} = \min_{\mathcal{S}_{[m, m+j+1]}^{\boldsymbol{\sigma}, \mathbf{0}}} \left\{ w_{\text{H}} \left(\boldsymbol{v}_{[m, m+j]} \right) \right\} \tag{11}$$

where the minimizations in (10), (11) are over all encoder states $\boldsymbol{\sigma}$. The *jth-order active segment distance* is

$$a_j^{\text{s}} = \min_{\mathcal{S}_{[m, m+j+1]}^{\boldsymbol{\sigma}_1, \boldsymbol{\sigma}_2}} \{ w_{\text{H}} (\boldsymbol{v}_{[m, m+j]}) \} \tag{12}$$

where the minimization is over all pairs $\boldsymbol{\sigma}_1$ and $\boldsymbol{\sigma}_2$ of encoder states.

Lower bounds [15] on the active distances are given by the linear, nondecreasing functions

$$a_j^{\text{b}} \geq f^{\text{b}}(j) = \alpha j + \beta^{\text{b}} \tag{13}$$
$$a_j^{\text{c}} \geq f^{\text{c}}(j) = \alpha j + \beta^{\text{c}} \tag{14}$$
$$a_j^{\text{rc}} \geq f^{\text{rc}}(j) = \alpha j + \beta^{\text{rc}} \tag{15}$$
$$a_j^{\text{s}} \geq f^{\text{s}}(j) = \alpha j + \beta^{\text{s}}. \tag{16}$$

[2]In [13], a transition from the all-zero state to the all-zero state for nonzero input weight is allowed in the definition of the encoder state sequence $\boldsymbol{\sigma}_{[t_1, t_2]}$. However, since we do not need to consider encoders that allow such state transitions, we omitted these cases and used a simplified definition of the encoder state sequence $\boldsymbol{\sigma}_{[t_1, t_2]}$ in (8).

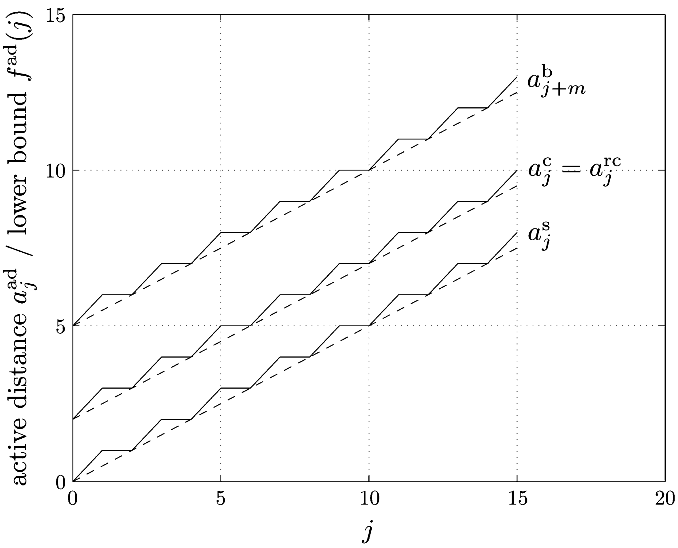

Fig. 2. Example of active distances a_j^{ad} and their lower bounds $f^{\mathrm{ad}}(j)$.

Here, α is the asymptotic (large j) slope of the active distances and the constants satisfy $\beta^{\mathrm{s}} \leq \beta^{\mathrm{c}}, \beta^{\mathrm{rc}} \leq \beta^{\mathrm{b}}$. From these bounds, a condition on the time index j guaranteeing the active distance a_j^{ad} to be at least x, i.e., $a_j^{\mathrm{ad}} \geq x$, denoted as j_x^{ad}, can be easily calculated as

$$j_x^{\mathrm{ad}} \geq \left\lceil \frac{x - \beta^{\mathrm{ad}}}{\alpha} \right\rceil \tag{17}$$

where $\mathrm{ad} = \mathrm{b}$ for the active burst distance, $\mathrm{ad} = \mathrm{c}$ and $\mathrm{ad} = \mathrm{rc}$ for the active column and reverse column distances, respectively, and $\mathrm{ad} = \mathrm{s}$ for the active segment distance.

In Fig. 2, the active distances and their lower bounds for the rate $R = 1/2$ convolutional encoder with generator matrix $G(D) = (1 + D^2, 1 + D + D^2)$ are shown as functions of j. Here, the active burst distance a_j^{b} and its lower bound $f^{\mathrm{b}}(j)$ are shifted by the memory $(m = 2)$ of the encoder, such that $a_j^{\mathrm{b}} = d_{\mathrm{free}}$ for $j = 0$, and thus $\beta^{\mathrm{b}} = 5$; the other parameters in Fig. 2 are given by $\beta^{\mathrm{c}} = \beta^{\mathrm{rc}} = 2$, $\beta^{\mathrm{s}} = 0$, and $\alpha = 1/2$.

The new joint permutor design for DSCCCs as given in the following theorem has two main features. First, the outer permutor has a span-separation constraint, i.e., symbols within a window of size s^{o} are separated after permutation by $t^{\mathrm{o}} - 1$ symbols such that the middle encoder generates an output weight that is lower bounded by the product of the free distances of the outer and middle code, respectively. Assuming that $S^{\mathrm{o}} \geq s^{\mathrm{o}}$, all these symbols from the window of size s^{o} then lie in a window of size T^{o} after permutation, and the whole guaranteed output weight of the middle encoder appears in a window whose size is a function of T^{o}. Second, choosing the input separation s^{i} of the inner permutor to at least equal this window size, and choosing the output separation t^{i} sufficiently large results in an output weight of the inner encoder that is lower bounded by the product of the free distances of all three component codes.

Theorem 1: Consider a DSCCC with an outer permutor with span-separation $(s^{\mathrm{o}}, t^{\mathrm{o}}; S^{\mathrm{o}}, T^{\mathrm{o}})$ satisfying

$$s^{\mathrm{o}} \geq c^{\mathrm{o}} \left(\min \left\{ j^{\mathrm{c}^{\mathrm{o}}}_{d^{\mathrm{o}}_{\mathrm{free}}}, j^{\mathrm{rc}^{\mathrm{o}}}_{d^{\mathrm{o}}_{\mathrm{free}}} \right\} + 1 \right) \tag{18}$$

$$t^{\mathrm{o}} \geq b^{\mathrm{m}} j^{\mathrm{b}^{\mathrm{m}}}_{2d^{\mathrm{m}}_{\mathrm{free}}} \tag{19}$$

$$S^{\mathrm{o}} \geq s^{\mathrm{o}} \tag{20}$$

and an inner permutor with separation $(s^{\mathrm{i}}, t^{\mathrm{i}})$ satisfying

$$s^{\mathrm{i}} \geq \frac{T^{\mathrm{o}}}{R^{\mathrm{m}}} + c^{\mathrm{m}} j^{\mathrm{s}^{\mathrm{m}}}_{d^{\mathrm{m}}_{\mathrm{free}}} \tag{21}$$

$$t^{\mathrm{i}} \geq b^{\mathrm{i}} j^{\mathrm{b}^{\mathrm{i}}}_{2d^{\mathrm{i}}_{\mathrm{free}}}. \tag{22}$$

Then

$$d_{\min} \geq d^{\mathrm{o}}_{\mathrm{free}} d^{\mathrm{m}}_{\mathrm{free}} d^{\mathrm{i}}_{\mathrm{free}} \tag{23}$$

represents a lower bound on the minimum distance of the DSCCC.

Proof: See Appendix I. □

In Theorem 1, a lower bound on the minimum distance given by the product of the free distances of the component codes is proven for DSCCCs whose permutors have sufficiently large span-separation. However, the proof does not give much insight into the existence or construction of permutors with such parameter sets. These particular topics are the subject of the following section.

III. Realization Aspects and Design Tables

In Theorem 1, the input separation s^{i} of the inner permutor depends on the output span T^{o} of the outer permutor, so that all symbols in the corresponding window at the output of the middle encoder are contained in a window of size s^{i}. Now, since we are using an S-type block permutor as the outer permutor, at least one window of size s^{o} exists such that symbols from this window are mapped partly to the beginning and partly to the end of the output sequence. Hence, choosing the input span $S^{\mathrm{o}} = s^{\mathrm{o}}$ results in an output span T^{o} that is approximately N^{o}, the outer permutor size. The output span T^{o} itself then implies that the input separation s^{i} of the inner permutor must be almost N^{i}, i.e., the inner permutor size. Therefore, the output separation t^{i} is approximately one, and (22) cannot be satisfied. Thus, the design constraints of Theorem 1 cannot be satisfied, at least using the definitions of symbol separation and span given in Definitions 1 and 3, respectively.

Therefore, in order to satisfy the design constraints of Theorem 1 that are realizable, Definitions 1 and 3 must be altered. In the following, we show how this can be achieved and how permutations satisfying the design constraints of Theorem 1 can be constructed. In addition, design tables are presented that give the minimum distance lower bound from (23) resulting from different sets of outer, middle, and inner encoders.

As described previously, the output sequences of S-type block permutors, especially block interleavers, exhibit a special structure where all symbols from a relatively small window of size S at the input to the permutor lie either in one window in the output sequence or in two windows very close to the beginning and the end of the output sequence. In addition, block interleavers have the property that these windows are relatively small, due to the fact that symbols are regularly separated, which results in a small δ-separation set [16]. Hence, grouping together the positions of these symbols at the edges of the permutation results in a relatively small window size T. To express this property mathematically, we use the *Lee metric*, defined as

$$d_{\mathrm{L}}(a,b)=\min\{|a-b|,N-|a-b|\}, \tag{24}$$

to modify the definition of the symbol span as follows.

Definition 5: A permutation π of length N is said to have *output-tailbiting symbol span* $(S,T)_{\mathrm{L}}$ if it satisfies

$$|i-i'|<S \quad \Rightarrow \quad d_{\mathrm{L}}(\pi(i),\pi(i'))\leq T \tag{25}$$

for any $i\neq i'$, $i,i'\in\{0,1,\ldots,N-1\}$, where $S,T\in\{0,1,\ldots,N-1\}$.

As a consequence of Definition 5, the output-tailbiting span T can now be restricted to values not larger than $N/2$, i.e., $T\leq N/2$. Therefore, the lower bound on T given in (7) does not hold for arbitrary values of S when considering the output-tailbiting symbol span. However, if S is restricted to

$$S\leq\left\lfloor\frac{N}{2t}\right\rfloor+1, \tag{26}$$

then (7) can also be used to bound the output-tailbiting symbol span, resulting in

$$\frac{N}{2}\geq T\geq(\min\{s,S\}-1)t. \tag{27}$$

Since the separation $(s^{\mathrm{o}},t^{\mathrm{o}})$ of the outer permutor must be chosen in such a way that it guarantees a minimum output weight of at least $d_{\mathrm{free}}^{\mathrm{o}}d_{\mathrm{free}}^{\mathrm{m}}$, and since the component encoders are terminated by appending a tail, there is no need to calculate $(s^{\mathrm{o}},t^{\mathrm{o}})$ using the Lee metric. Moreover, if the input span S^{o} contains all the symbols in the window of size s^{o}, i.e., $S^{\mathrm{o}}=s^{\mathrm{o}}$, then it is sufficient to use the Lee metric only for the calculation of the output-tailbiting span T^{o}, as given in Definition 5.

Since the symbol separation s^{i} of the inner permutor is calculated based on the output-tailbiting span T^{o} of the outer permutor, the definition of symbol separation for the inner permutor must also be modified.

Definition 6: A permutation π of length N is said to have an *input-tailbiting symbol separation* $(s,t)_{\mathrm{L}}$ if it satisfies

$$d_{\mathrm{L}}(i,i')<s \quad \Rightarrow \quad |\pi(i)-\pi(i')|\geq t \tag{28}$$

for any $i\neq i'$, $i,i'\in\{0,1,\ldots,N-1\}$, where $s,t\in\{0,1,\ldots,N-1\}$.

Using Definitions 5 and 6 in (21), Theorem 1 still holds with the same proof. The only difference is that the windows of size T^{o} at the input to the middle encoder and s^{i} at the output of the middle encoder, respectively, are now considered in a tailbiting manner.

Example 1: Using Definition 1 for the calculation of the symbol separation and Definition 5 for the symbol span, we can construct a block interleaver (see [9]) of size $N=3072$ having a span-separation of

$$(s,t;S,T)_{\mathrm{L}}=(256,12;10,108).$$

The input span S satisfies $S\leq\lfloor N/(2t)\rfloor+1=129$, i.e., inequality (26), and hence a lower bound on the output-tailbiting span T can be calculated using (27), yielding $T\geq(\min\{s,S\}-1)t=108$. Since T meets this lower bound, the parameter T is optimal in this case.

In Example 1, we have shown that permutors with large span-separation exist. However, it is not clear what size a permutor must be in order to give some specified span-separation. In the following, we address this issue by giving lower bounds on the sizes of the outer and inner permutors needed to satisfy Theorem 1.

First note that the size of a permutor satisfies the bound

$$N\geq st \tag{29}$$

where equality holds for special types of block interleavers, namely, so-called RlTb- and LrBt-interleavers [9]. Therefore, the sizes of the inner and outer permutors can be chosen as

$$N^{\mathrm{o}}\geq s^{\mathrm{o}}t^{\mathrm{o}} \tag{30}$$

$$N^{\mathrm{i}}\geq s^{\mathrm{i}}t^{\mathrm{i}}. \tag{31}$$

Moreover, the size of the inner permutor can be calculated from the size of the outer permutor using

$$N^{\mathrm{i}}=\frac{N^{\mathrm{o}}}{R^{\mathrm{m}}} \tag{32}$$

where the size of the permutors is assumed to be large compared to the termination tail, so that the effect of the termination tail can be neglected. From Theorem 1, we see that the permutor parameters s^{o}, t^{o}, and t^{i} depend only on the choice of the component encoders, i.e., on the free and active distances of the component encoders. Furthermore, for simplicity, we choose $S^{\mathrm{o}}=s^{\mathrm{o}}$. Hence, only the output span T^{o} of the outer permutor and the input separation s^{i} of the inner permutor must be calculated, where s^{i} is a function of T^{o}.

From (31) and (32), we obtain

$$N^{\mathrm{i}}\geq\max\left\{\frac{s^{\mathrm{o}}t^{\mathrm{o}}}{R^{\mathrm{m}}},s^{\mathrm{i}}t^{\mathrm{i}}\right\} \tag{33}$$

$$\geq\max\left\{\frac{s^{\mathrm{o}}t^{\mathrm{o}}}{R^{\mathrm{m}}},\left(\frac{T^{\mathrm{o}}}{R^{\mathrm{m}}}+c^{\mathrm{m}}j_{d_{\mathrm{free}}^{\mathrm{m}}}^{\mathrm{s^m}}\right)t^{\mathrm{i}}\right\} \tag{34}$$

where we used (21) in the last step. Then from (27) and the fact that $S^{\mathrm{o}}=s^{\mathrm{o}}$, we obtain

$$N^{\mathrm{i}}\geq\max\left\{\frac{s^{\mathrm{o}}t^{\mathrm{o}}}{R^{\mathrm{m}}},\left(\frac{(s^{\mathrm{o}}-1)\,t^{\mathrm{o}}}{R^{\mathrm{m}}}+c^{\mathrm{m}}j_{d_{\mathrm{free}}^{\mathrm{m}}}^{\mathrm{s^m}}\right)t^{\mathrm{i}}\right\} \tag{35}$$

and it follows from (32) that

$$N^{\mathrm{o}}\geq R^{\mathrm{m}}\max\left\{\frac{s^{\mathrm{o}}t^{\mathrm{o}}}{R^{\mathrm{m}}},\left(\frac{(s^{\mathrm{o}}-1)\,t^{\mathrm{o}}}{R^{\mathrm{m}}}+c^{\mathrm{m}}j_{d_{\mathrm{free}}^{\mathrm{m}}}^{\mathrm{s^m}}\right)t^{\mathrm{i}}\right\} \tag{36}$$

$$=\max\left\{s^{\mathrm{o}}t^{\mathrm{o}},\left((s^{\mathrm{o}}-1)t^{\mathrm{o}}+b^{\mathrm{m}}j_{d_{\mathrm{free}}^{\mathrm{m}}}^{\mathrm{s^m}}\right)t^{\mathrm{i}}\right\} \tag{37}$$

where we used $R^{\mathrm{m}}=b^{\mathrm{m}}/c^{\mathrm{m}}$ in the last step. Finally, (37) implies that the outer permutor satisfies (26) and (27) if $t^{\mathrm{i}}\geq 2$, which is always true, since we consider only nontrivial permutors. Hence, we have proved the following theorem.

TABLE I
GUARANTEED MINIMUM DISTANCE OF DSCCCs AND SINGLE SCCCs

m \ $d_{\min}$	DSCCC	SCCC	SCCC+	SCCC++	SCCC+++
1	8	6	10	12	14
2	27	15	18	21	24
3	64	24	28	32	40
4	100	35	40	50	50
5	180	48	60	60	72

Theorem 2: The minimum sizes of the inner and outer permutors required to guarantee a minimum distance for DSCCCs lower bounded by $d_{\min} \geq d_{\text{free}}^{\text{o}} d_{\text{free}}^{\text{m}} d_{\text{free}}^{\text{i}}$ are given by N^{i} in (35) and N^{o} in (37), respectively.

Example 2: We calculate lower bounds on the sizes of the outer and inner permutors for a rate $R = 1/8$ DSCCC with three memory two component encoders with generator matrices $G^{\text{o}}(D) = (1 + D^2, 1 + D + D^2)$ and $G^{\text{m}}(D) = G^{\text{i}}(D) = (1, (1+D^2)/(1+D+D^2))$ (note that these encoders are equivalent, and hence their active distances are all the same and are given in Fig. 2).[3]

We computed $s^{\text{o}} \geq 14$, $t^{\text{o}} \geq 12$, and $t^{\text{i}} \geq 12$ from (18), (19), and (22). From (27), we obtain $T^{\text{o}} \geq 156$, and inequalities (36) and (34) result in $N^{\text{o}} \geq 1992$ and $N^{\text{i}} \geq 3984$. The overall code then has a guaranteed minimum distance of $d_{\text{free}} \geq 125$.[4]

We have shown that permutor pairs can be constructed that satisfy all the conditions of Theorem 1, hence resulting in a DSCCC whose minimum distance is lower bounded by the product of the distances of all three component codes. Furthermore, we have derived lower bounds on the permutor sizes required to guarantee the minimum distance lower bound. In Table I, we compare the guaranteed minimum distances of DSCCCs with permutor pairs jointly designed according to Theorem 1 to SCCCs with a permutor having a sufficiently large symbol separation, where both codes have rate $R = 1/3$. To calculate the product distances for both codes, we have chosen a rate allocation of $R^{\text{o}} = R^{\text{m}} = 2/3$ and $R^{\text{i}} = 3/4$ in the case of DSCCCs and $R^{\text{o}} = 1/2$ and $R^{\text{i}} = 2/3$ in the case of single SCCCs. The component codes were chosen as optimum free distance codes (see [14]) for rate $R = 1/2$ and optimum free distance punctured codes (see [19]) for higher rates, and the component encoders of the DSCCCs and single SCCCs all have the same memory m. However, the decoding complexity of DSCCCs is higher than SCCCs, since three component codes must be decoded during one iteration and, using the decoding scheme described in [5], the middle code is decoded twice per iteration. Hence, in order to have a fairer comparison, we also considered SCCCs with increased memory outer codes, i.e., outer code memories of $m^{\text{o}} = m+1$, $m^{\text{o}} = m + 2$, $m^{\text{o}} = m + 3$, denoted as SCCC+, SCCC++, and SCCC+++, respectively. The decoding complexity (on a normalized decoder-trellis-edge-evaluations/information-bits basis) of the DSCCC is larger than the SCCC++, but smaller than the SCCC+++.

TABLE II
GUARANTEED MINIMUM DISTANCE OF DSCCCs AND SINGLE SCCCs WITH MEMORY ONE MIDDLE AND INNER CODES

m^{o} \ $d_{\min}$	DSCCC	SCCC
1	8	6
2	12	10
3	16	12
4	20	14
5	24	16
6	24	20
7	32	20
8	32	24

In Table II, the guaranteed minimum distances of DSCCCs and single SCCCs are given, where the middle and inner codes of the DSCCCs and the inner code of the SCCCs all have memory one, and only the outer code is varied.

As can be seen from Tables I and II, the guaranteed minimum distance of DSCCCs is considerably larger than for single SCCCs, especially as the memory of the component codes increases. Moreover, the advantage of DSCCCs over single SCCCs still remains if the memory of the outer encoder for single SCCCs is increased.[5]

IV. SIMULATION RESULTS AND CODE COMPARISONS

A. DSCCCs and Single SCCCs

As shown in Example 1, block interleavers are possible candidates for constructing DSCCCs with large minimum distances, since they can have a relatively small output span T for moderate input span S and output separation t. However, block interleavers lead to a poor weight spectrum in serial concatenation, i.e., large multiplicities of low-weight codewords, resulting in

[3]Throughout the paper, we assume the use of recursive systematic middle and inner encoders, since they are needed to achieve good iterative decoding convergence behavior [17].

[4]We note that, even though the guaranteed minimum distance of 125 in this case is quite large, the block length of 7968 is also large and is, in fact, much larger than the minimum required block length needed to achieve a distance of 125 that would be predicted from asymptotic ensemble average bounds such as those presented in [18]. However, we present here a specific construction, guaranteeing a certain achievable minimum distance, whereas ensemble average bounds are not constructive.

[5]Typically, the minimum permutor size required to guarantee the SCCC minimum distances in Tables I and II will be smaller than the minimum permutor sizes needed to guarantee the DSCCC minimum distance. However, these guaranteed distances do not grow with permutor size, so for large block lengths, we expect DSCCCs with S-type permutors to have larger minimum distances than corresponding SCCCs.

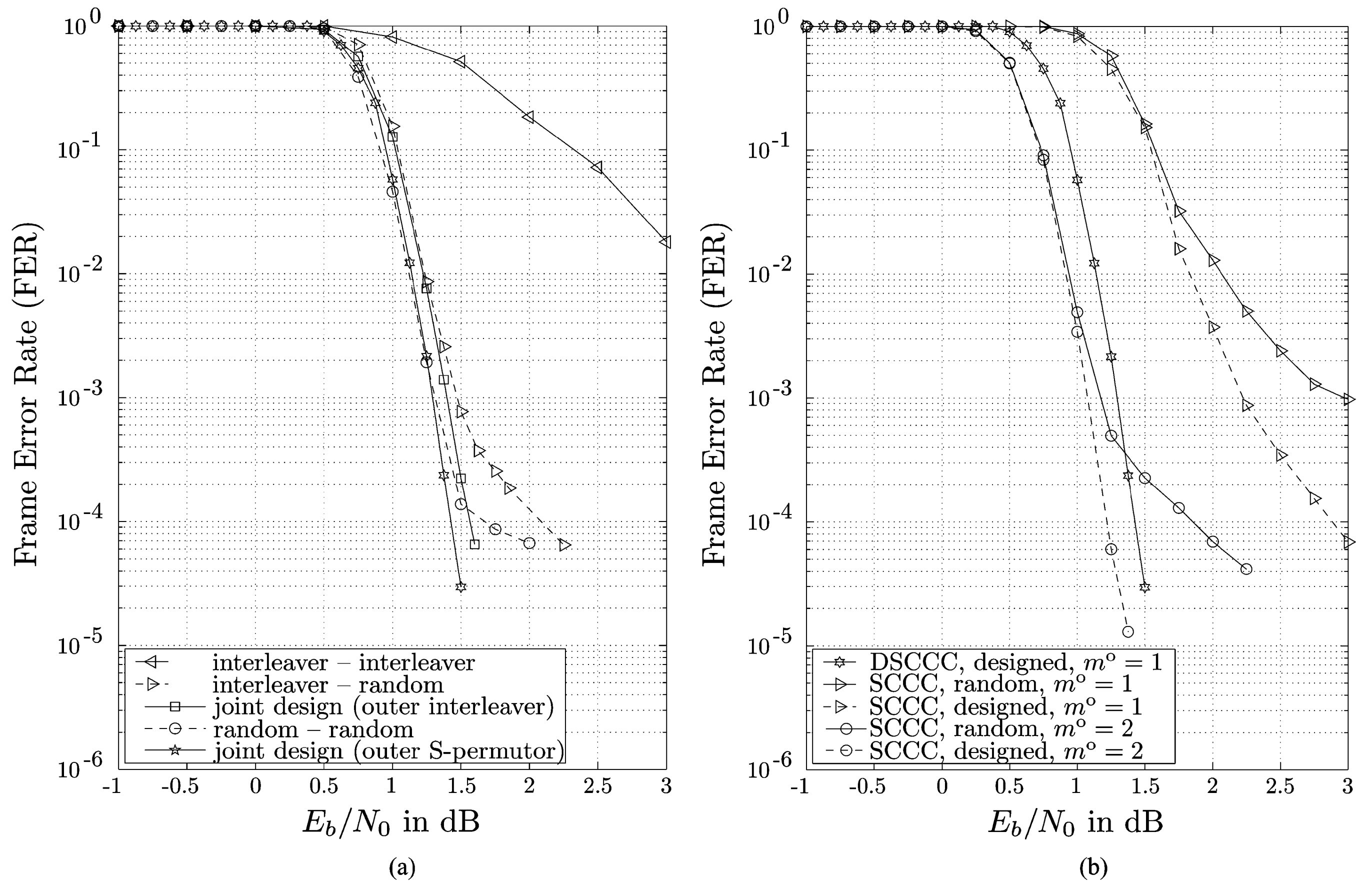

Fig. 3. Simulated FER performance curves of DSCCCs with different permutors and SCCCs with different permutors and different encoders. (a) DSCCCs with different permutor pairs. (b) Comparison of DSCCCs with SCCCs.

inferior error performance [10]. Therefore, we consider a combination of a block interleaver with large span-separation as the outer permutor with an S-random inner permutor having a sufficiently large separation on the one hand and a reduced structure on the other hand. In this way, the weight spectrum problem at the output of the middle encoder, i.e., the large low-weight multiplicities, is significantly improved by the inner permutor, so that the resulting DSCCC exhibits a large minimum distance as well as small low-weight multiplicities.

The joint permutor design procedure described above is not constructive, i.e., an interleaver pair having a certain parameter set must be found by computer search. Therefore, we have modified the algorithm presented in [20] for constructing turbo code permutations to allow us to also construct good $(s,t;S,T)_{\mathrm{L}}$ outer permutations for DSCCCs. Moreover, it is possible to introduce additional randomness, i.e., to reduce the structure, compared to the permutation of a block interleaver. In this way, we obtain outer permutors resulting in good distance spectrum properties at the output of the middle encoder.

Simulated frame error rate (FER) curves for dimension $k = 2047$, rate $R = 1/3$ DSCCCs with outer, middle, and inner encoders given by $G^{\mathrm{o}}(D) = G^{\mathrm{m}}(D) = G^{\mathrm{i}}(D) = (1, 1/(1+D))$ are shown in Fig. 3(a). All encoder outputs are periodically punctured, the outer and middle encoder to rate $R^{\mathrm{o}} = R^{\mathrm{m}} = 2/3$ and the inner encoder to rate $R^{\mathrm{i}} = 3/4$. The decoding scheme presented in [5] was used with $I = 10$ iterations.

The FER curves of DSCCCs with the following permutor pairs are shown:

- the block interleaver from Example 1 as outer permutor and a block interleaver with $(s,t) = (419, 11)$ as inner permutor, not jointly designed, denoted as *interleaver–interleaver*;
- the block interleaver from Example 1 and a random permutor, denoted as *interleaver–random*;
- the block interleaver from Example 1 with symbol separation $(256, 10)$ and output-tailbiting symbol span $(10, 108)_{\mathrm{L}}$, as defined in (25), as outer permutor and a $(180, 10)_{\mathrm{L}}$-permutor with input-tailbiting symbol separation as defined in (28), as inner permutor, denoted as *joint design (outer interleaver)*;
- a $(4, 5; 4, 180)_{\mathrm{L}}$-permutor, where the span was calculated using the Lee metric, as defined in (25), as outer permutor and an inner $(s,t)_{\mathrm{L}}$-permutor with input-tailbiting symbol separation, as defined in (28), of $(s,t)_{\mathrm{L}} = (280, 7)_{\mathrm{L}}$, denoted as *joint design* (*outer* S-*permutor*);
- a permutor pair given by an outer random permutor and an inner random permutor, denoted as *random–random*.

According to Theorem 1, the joint permutor designs, i.e., both *joint design (outer interleaver)* and *joint design* (*outer* S-*permutor*), result in a minimum distance that is lower bounded by the product of the free distances of all the component codes, i.e., $d_{\min} \geq 8$. Compared to that, the guaranteed minimum distance of the DSCCCs with an outer block interleaver and an inner random permutor is given by $d_{\min} \geq 4$, and for outer and inner random permutors the minimum distance is lower bounded by $d_{\min} \geq 2$.

Two main effects can be seen from the FER curves shown Fig. 3(a). First, the new design leads to significant improvement in the error floor region of the FER curves due to the increased minimum distance. Second, the "joint randomness" of the outer and inner permutor makes a significant difference in the iterative decoder's ability to converge. When using block interleavers as both outer and inner permutors, the losses in the waterfall region due to the inherent structure of the permutors, resulting in large multiplicities of low-weight codewords compared to the other designs, are severe. Compared to this *interleaver–interleaver* design, a considerable reduction in the waterfall region losses can be achieved by replacing the inner block interleaver with a permutor that is less structured. As can be seen from Fig. 3(a), also adding randomness to the outer permutor leads to only a small additional improvement in the waterfall region, i.e., about 0.1 dB in this case.

In Fig. 3(b), a performance comparison between DSCCCs with the new permutor design and SCCCs is shown, where all codes have dimension $k = 2047$. The outer encoder of the SCCC has generator matrix $G^{\text{o}}(D) = (1, 1/(1+D))$ in one case and $G^{\text{o}}(D) = (1, (1+D^2)/(1+D+D^2))$ in the other case, i.e., for the SCCC+, and the inner encoder is the recursive memory $m^{\text{i}} = 1$ component encoder punctured to rate $R^{\text{i}} = 2/3$. In each case, both random permutors and designed $(40, 40)$-permutors are employed. Due to the increased minimum distance, the DSCCC considerably outperforms the SCCCs—except for the SCCC with the memory $m^{\text{o}} = 2$ outer encoder and a designed $(40, 40)$-permutor—in the error floor.[6]

B. DRACS—A Special Type of DSCCCs

Compared to the DSCCCs depicted in Fig. 1, GRACs are generated by restricting the choice of the middle and inner encoders to the generator matrices $G^{\text{m}}(D) = G^{\text{i}}(D) = 1/(1+D)$. For DRACs [3], a repetition code is used as outer code. In addition, following [21], DRACs can be designed as so-called irregular DRACs (IDRACs), where the outer repetition code is replaced by a bank of parallel repetition codes of different rates. In both cases, i.e., DRACs and IDRACs, single or conventional SCCCs, i.e., so-called repeat accumulate codes (RACs) and irregular repeat accumulate codes (IRACs), exist that consist of the outer two parts of the DRACs and IDRACs without the inner permutor and inner encoder.

Let d^{o} be the reciprocal of the maximum rate of the parallel repetition codes in the case of the (I)DRAC and the free distance

[6]Since the SCCC+ with a designed permutor is already superior to the DSCCC with respect to the guaranteed minimum distance according to Table I, and it exhibits no discernible error floor, no error floor would be expected either for the SCCC++ and the SCCC+++ in the FER range shown in Fig. 3(b). Moreover, the waterfall performance of the SCCC++ and the SCCC+++ would be worse than that of the SCCC+.

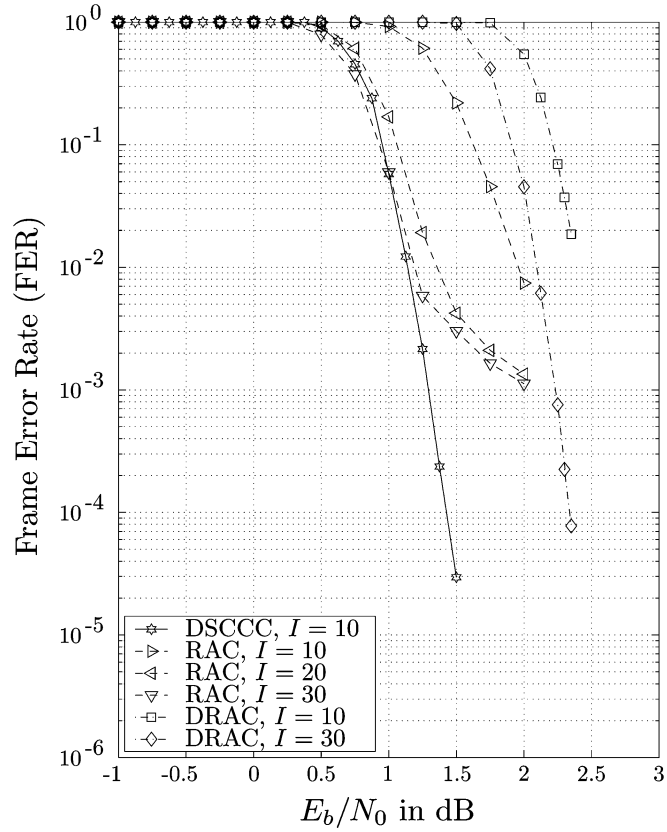

Fig. 4. Comparison of DSCCCs with DRACs and RACs.

of the outer convolutional code in case of the GRAC. Then the minimum distance of the overall code is lower bounded by

$$d_{\min} \geq \left\lceil \frac{d^{\text{o}}}{4} \right\rceil \tag{38}$$

for random permutors (see [4]) and

$$d_{\min} \geq d^{\text{o}} \tag{39}$$

for the design of Theorem 1. For (I)RACs, the corresponding bounds on the minimum distance are given by

$$d_{\min} \geq \left\lceil \frac{d^{\text{o}}}{2} \right\rceil \tag{40}$$

and

$$d_{\min} \geq d^{\text{o}} \tag{41}$$

where, in the latter case, the permutor must provide sufficiently large separation (s, t). (Note that (I)DRACs have better distance spectrum properties than (I)RACs, although the guaranteed minimum distance for random permutors is significantly smaller (see [4])).

In Fig. 4, simulated FER curves for the following codes are shown: a DSCCC with parameters as described in Section IV-A and a joint permutor design, i.e., an outer $(4, 5; 4, 180)_{\text{L}}$-permutor with output-tailbiting symbol span given by (25) and an inner $(280, 7)_{\text{L}}$-permutor with input-tailbiting symbol separation given by (28), a DRAC with an outer repetition code of rate $R^{\text{o}} = 1/3$ along with outer and inner random permutors, and a RAC with an outer repetition code of rate $R^{\text{o}} = 1/3$ and a

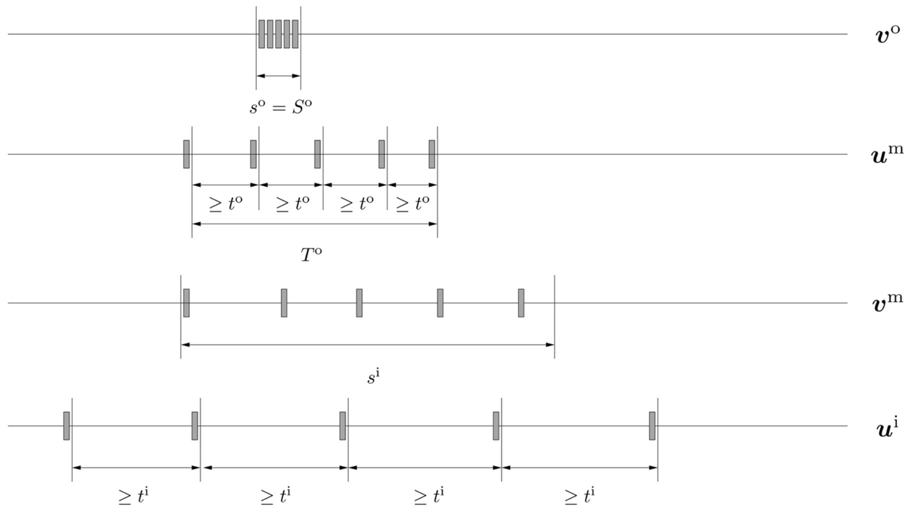

Fig. 5. Illustration of essential joint permutor design characteristics.

random permutor. In the case of the DSCCC, $I = 10$ decoding iterations were used, while for the DRAC and RAC, the number of decoding iterations was chosen to be larger, i.e., $I = 30$.[7] As can be seen from Fig. 4, with the new permutor design the DSCCC combines the advantages of the DRAC and RAC—its performance in the waterfall region is comparable to that of the RAC and its error floor performance is comparable to that of the DRAC. However, the gains in the waterfall region may decrease when considering IDRACs with optimized degree distributions.

V. Conclusion

In this paper, we have introduced a new permutor parameter, the so-called symbol span, and applied it to joint permutor design for DSCCCs. The new parameter enables us to adapt the design of the inner permutor to the outer permutor in such a way that the corresponding permutor design for conventional SCCCs can be generalized. A lower bound on the minimum distance, given by the product of the free distances of all three component codes of DSCCCs with jointly designed permutors was derived. A performance comparison with SCCCs, DRACs, and RACs showed that DSCCCs with jointly designed permutors combine the advantages of good performance in the waterfall region and low error floors resulting from the new joint permutor design.

Appendix I
Proof of Theorem 1

The proof of Theorem 1 is similar to the proof of the corresponding lower bound on the minimum distance of single SCCCs presented in [11] (see also [15] and [22]). However, for the sake of completeness and in order to highlight the structure of the new permutor design, we give a self-contained proof of Theorem 1 here. First, we briefly summarize a preliminary result in the following lemma.

Lemma 1: Consider an input sequence $\boldsymbol{u}$ to a terminated recursive convolutional encoder of rate $R = b/c$ with weight at least x, i.e., $w_{\rm H}(\boldsymbol{u}) \geq x$, where at least x "1's" have a spacing of at least $\gamma = bj^{\rm b}_{2d_{\rm free}}$, i.e., their preceding and succeeding "1's" are at least γ positions away. Then

$$w_{\rm H}(\boldsymbol{v}) \geq x d_{\rm free} \tag{42}$$

lower bounds the weight $w_{\rm H}(\boldsymbol{v})$ of the output sequence $\boldsymbol{v}$.

Proof: Let the output sequence $\boldsymbol{v}$ consist of L so-called bursts (according to the definition of the active burst distance (9)), i.e., segments, where the corresponding encoder state sequence starts in the all-zero state and ends in the all-zero state without a transition from the all-zero state to the all-zero state in between. Since the encoder is recursive, every burst consists of at least two input "1's," and every input "1" belongs to some burst. Each of these bursts consists of exactly x_{i_l}, $l = 1, 2, \ldots, L$, with $x_{i_l} \geq 1$, input "1's" belonging to the at least x input "1's" that have a spacing of at least γ, so that $x_{i_1} + x_{i_2} + \cdots + x_{i_L} = x$. Clearly, for bursts that contain only one of the x input "1's" (and at least one other input "1" not belonging to the x input "1's"), i.e., satisfying $x_{i_l} = 1$, the output weight is at least $d_{\rm free}$. For all other bursts, i.e., those satisfying $x_{i_l} \geq 2$, the output weight is calculated using the active burst distance. Since the x_{i_l} input "1's" from these bursts have a spacing of at least γ and $\gamma = bj^{\rm b}_{2d_{\rm free}}$, it is guaranteed that each of the x_{i_l} input "1's" results in an output weight of at least $d_{\rm free}$. Hence, summing up the output weights of all bursts, we obtain

$$\begin{aligned} w_{\rm H}(\boldsymbol{v}) &\geq x_{i_1} d_{\rm free} + x_{i_2} d_{\rm free} + \cdots + x_{i_L} d_{\rm free} \\ &= d_{\rm free} \sum_{l=1}^{L} x_{i_l} = x d_{\rm free} \end{aligned}$$

which lower-bounds the overall weight of the output sequence $\boldsymbol{v}$ as given by inequality (42), and the proof is complete. □

Using Lemma 1, we now give the proof of Theorem 1.

Proof of Theorem 1: The choice of $s^{\rm o}$ in (18) guarantees that, if the outer encoder generates weight, at least one window of size $s^{\rm o}$ exists that contains at least $d^{\rm o}_{\rm free}$ "1's." After the

[7] We chose the number of decoding iterations for the different schemes so that the decoding complexity (total number of computations) would be approximately the same. Also, a sufficient number of decoding iterations was chosen in each case so that further iterations would only marginally improve performance.

outer permutation, these "1's" then have a spacing of at least $t^{\rm o} \geq b^{\rm m} j^{\rm b^m}_{2d^{\rm m}_{\rm free}}$ (see (19)), so that Lemma 1 can be applied, yielding an output weight for the middle encoder that is at least $d^{\rm m}_{\rm min} \geq d^{\rm o}_{\rm free} d^{\rm m}_{\rm free}$. Due to the choice of $S^{\rm o}$ in (20), all the "1's" from the window of size $s^{\rm o}$, after permutation, lie in a window of size $T^{\rm o}$, and hence, after encoding by the middle encoder, a window of size $s^{\rm i}$ given by (21) exists that contains a weight of at least $d^{\rm o}_{\rm free} d^{\rm m}_{\rm free}$. Here, the span $T^{\rm o}$ is chosen such that all "1's" from the original window of size $s^{\rm o}$ (and the "1's" in between, after encoding) are included, plus an additional tail, guaranteeing that a weight not smaller than $d^{\rm m}_{\rm free}$ is included for the last of the original "1's." Since $t^{\rm i}$ in (22) satisfies the condition of Lemma 1, the output weight of the inner encoder can also be lower bounded using Lemma 1, yielding (23), which completes the proof. □

In Fig. 5, the characteristics of the joint permutor design essential for understanding the proof of Theorem 1 are illustrated. As can be seen, all symbols in a window of size $s^{\rm o} = S^{\rm o}$ in the sequence $\boldsymbol{v}^{\rm o}$ are separated by at least $t^{\rm o}$ symbols in the sequence $\boldsymbol{u}^{\rm m}$ and lie in a window of size $T^{\rm o}$. At the output of the middle encoder, i.e., in the sequence $\boldsymbol{v}^{\rm m}$, these symbols lie in a window of size $s^{\rm i}$ and are separated after permutation, i.e., in the sequence $\boldsymbol{u}^{\rm i}$, by at least $t^{\rm i}$.

REFERENCES

[1] A. Huebner and D. J. Costello, Jr., "On the design of s-type permutors for double serially concatenated convolutional codes," in *Proc. IEEE Information Theory Workshop (ITW)*, Punta del Este, Uruguay, Mar. 2006, pp. 188–192.

[2] H. D. Pfister and P. H. Siegel, "The serial concatenation of rate-1 codes through uniform random interleavers," in *Proc. 37th Annu. Allerton Conf. Communication, Control, and Computing*, Monticello, IL, Oct. 1999, pp. 260–269.

[3] H. D. Pfister and P. H. Siegel, "The serial concatenation of rate-1 codes through uniform random interleavers," *IEEE Trans. Inf. Theory*, vol. 49, no. 6, pp. 1425–1438, Jun. 2003.

[4] H. D. Pfister and P. H. Siegel, "Coding theorems for generalized repeat accumulate codes," in *Proc.Int. Symp. Information Theory and its Applications*, Honoulu, HI, Nov. 2000, pp. 21–25.

[5] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "Analysis, design, and iterative decoding of double serially concatenated codes with interleavers," *IEEE J. Sel. Areas Commun.*, vol. 16, no. 2, pp. 231–244, Feb. 1998.

[6] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "Serial concatenation of interleaved codes: Performance analysis, design and iterative decoding," *IEEE Trans. Inf. Theory*, vol. 44, no. 3, pp. 909–926, May 1998.

[7] V. Guruswami and W. Machmouchi, "Explicit interleavers for a repeat accumulate accumulate (RAA) code construction," in *Proc. Int. Symp. Information Theory*, Toronto, ON, Canada, Jul. 2008, pp. 1968–1972.

[8] J. L. Ramsey, "Realization of optimum interleavers," *IEEE Trans. Inf. Theory*, vol. IT-16, no. 3, pp. 338–345, May 1970.

[9] C. Heegard and S. B. Wicker, *Turbo Coding*. Dordrecht, The Netherlands: Kluwer Academic, 1999.

[10] S. Dolinar and D. Divsalar, "Weight Distributions for Turbo Codes Using Random and Nonrandom Permutations," Jet Propulsion Lab., Pasadena, CA, TDA Progress Rep. 42-122, Aug. 1995.

[11] R. Jordan, S. Höst, M. Bossert, R. Johannesson, and V. V. Zyablov, "Woven convolutional codes II: Decoding aspects," *IEEE Trans. Inf. Theory*, vol. 50, no. 10, pp. 2522–2531, Oct. 2004.

[12] J. Freudenberger, R. Jordan, M. Bossert, and S. Shavgulidze, "Serially concatenated convolutional codes with product distances," in *Proc. 2nd Int. Symp. Turbo Codes and Related Topics*, Brest, France, Sep. 2000, pp. 81–84.

[13] S. Höst, R. Johannesson, K. Sh. Zigangirov, and V. V. Zyablov, "Active distances for convolutional codes," *IEEE Trans. Inf. Theory*, vol. 45, no. 2, pp. 658–669, Mar. 1999.

[14] R. Johannesson and K. Sh. Zigangirov, *Fundamentals of Convolutional Coding*. Piscataway, N.J.: IEEE Press, 1999.

[15] S. Höst, R. Johannesson, and V. V. Zyablov, "Woven convolutional codes I: Encoder properties," *IEEE Trans. Inf. Theory*, vol. 48, no. 1, pp. 149–161, Jan. 2002.

[16] A. Huebner and G. Richter, "On the design of woven convolutional encoders with outer warp row permutors," *IEEE Trans. Commun.*, vol. 54, no. 3, pp. 438–444, Mar. 2006.

[17] J. Kliewer, A. Huebner, and D. J. Costello, Jr., "On the achievable extrinsic information of inner decoders in serial concatenation," in *Proc. IEEE Int. Symp. Information Theory*, Seattle, WA, Jul. 2006, pp. 2680–2684.

[18] C. Koller, J. Kliewer, K. S. Zigangirov, and D. J. Costello, Jr., "Minimum distance bounds for multiple-serially concatenated code ensembles," in *Proc. IEEE Int. Symp. Information Theory*, Toronto, ON, Canada, Jul. 2008, pp. 1888–1892.

[19] J. B. Cain, G. C. Clark, Jr., and J. M. Geist, "Punctured convolutional codes of rate $(n-1)/n$ and simplified maximum likelihood decoding," *IEEE Trans. Inf. Theory*, vol. IT-25, no. 1, pp. 97–100, Jan. 1979.

[20] D. V. Truhachev, M. Lentmaier, and K. Sh. Zigangirov, "Some results concerning design and decoding of turbo-codes," *Probl. Inf. Transm.*, vol. 37, no. 3, pp. 190–205, Jul.–Sep. 2001.

[21] H. Jin, A. Khandekar, and R. J. McEliece, "Irregular repeat accumulate codes," in *Proc. 2nd Int. Symp. Turbo Codes and Related Topics*, Brest, France, Sep. 2000, pp. 1–8.

[22] J. Freudenberger, M. Bossert, S. Shavgulidze, and V. V. Zyablov, "Woven codes with outer warp: Variations, design, and distance properties," *IEEE J. Sel. Areas Commun.*, vol. 19, no. 5, pp. 813–824, May 2001.

Axel Huebner (S'98-M'05) was born in Karlsruhe, Germany, in 1973. He received the Dipl.-Ing. degree in electrical engineering from the Technical University of Karlsruhe, Karlsruhe, Germany, in 1999, and the Ph.D. degree in electrical engineering from the University of Ulm, Ulm, Germany, in 2004.

From 2005 to 2006, he was a Postdoctoral Research Associate in the Coding Research Group at the University of Notre Dame, Notre Dame, IN. During that period, he also was a Visiting Researcher at the Technical University of Munich, Munich, Germany (Summer 2005). In 2006, he joined the Connectivity Systems Group at Philips Research Europe–Aachen, Aachen, Germany, as a Research Scientist. Since 2007, he has been with the Communication Group at Infineon Technologies AG, Munich. His research interest include coding theory, with emphasis on concatenated convolutional coding schemes, and diversity techniques for wireless communication systems.

Jörg Kliewer (S'97–M'99–SM'04) received the Dipl.-Ing. (M.Sc.) degree in electrical engineering from Hamburg University of Technology, Hamburg, Germany, in 1993 and the Dr.-Ing. degree (Ph.D.) in electrical engineering from the University of Kiel, Kiel, Germany, in 1999, respectively.

From 1993 to 1998, he was a Research Assistant at the University of Kiel, and from 1999 to 2004, he was a Senior Researcher and Lecturer with the same institution. In 2004, he visited the University of Southampton, Southampton, U.K., for one year, and from 2005 until 2007, he was with the University of Notre Dame, Notre Dame, IN, as a Visiting Assistant Professor. In August 2007, he joined New Mexico State University, Las Cruces, as an Assistant Professor. His research interests include error-correcting codes, network coding, wireless communications, and communication networks.

Dr. Kliewer was the recipient of a Leverhulme Trust Award and a German Research Foundation Fellowship Award in 2003 and 2004, respectively. He is a Member of the Editorial Board of the *EURASIP Journal on Advances in Signal Processing* and Associate Editor of the IEEE TRANSACTIONS ON COMMUNICATIONS.

Daniel J. Costello Jr. (S'62–M'69–SM'78–F'85–LF'08) was born in Seattle, WA, on August 9, 1942. He received the B.S.E.E. degree from Seattle University, Seattle, WA, in 1964, and the M.S. and Ph.D. degrees in electrical engineering from the University of Notre Dame, Notre Dame, IN, in 1966 and 1969, respectively.

He joined the faculty of the Illinois Institute of Technology, Chicago, IL, in 1969 as an Assistant Professor of Electrical Engineering. He was promoted to Associate Professor in 1973, and to Full Professor in 1980. In 1985, he became Professor of Electrical Engineering at the University of Notre Dame, Notre

Dame, IN, and from 1989 to 1998 served as Chair of the Department of Electrical Engineering. His research interests are in the area of digital communications, with special emphasis on error control coding and coded modulation. He has numerous technical publications in his field, and in 1983 he coauthored a textbook entitled *Error Control Coding: Fundamentals and Applications*, the 2nd edition of which was published in 2004.

In 1991, Dr. Costello was selected as one of 100 Seattle University alumni to receive the Centennial Alumni Award in recognition of alumni who have displayed outstanding service to others, exceptional leadership, or uncommon achievement. In 1999, he received a Humboldt Research Prize from the Alexander von Humboldt Foundation in Germany. In 2000, he was named the Leonard Bettex Professor of Electrical Engineering at Notre Dame. Since 1983, he has been a member of the IEEE Information Theory Society Board of Governors, and in 1986 he served as President of the Board of Governors (BOG) of the Society. He has also served as Associate Editor for Communication Theory for the IEEE TRANSACTIONS ON COMMUNICATIONS, Associate Editor for Coding Techniques for the IEEE TRANSACTIONS ON INFORMATION THEORY, and Co-Chair of the IEEE International Symposia on Information Theory in Kobe, Japan (1988), Ulm, Germany (1997), and Chicago, IL (2004). In 2000, he was selected by the IEEE Information Theory Society as a recipient of a Third-Millennium Medal. He was corecipient of the 2009 IEEE Donald G. Fink Prize Paper Award, which recognizes an outstanding survey, review, or tutorial paper in any IEEE publication issued during the previous calendar year.