Combinatorial Analysis of the Minimum Distance of Turbo Codes

Marco Breiling and Johannes Huber
Lehrstuhl für Nachrichtentechnik II, Universität Erlangen-Nürnberg
Cauerstr. 7, D-91058 Erlangen, Germany
Tel.: +49-9131-85-27668, Fax: +49-9131-85-28919
Email: breiling@LNT.de
http://www.LNT.de/∼breiling

Submitted 06/03/2000, revised 29/03/2001

Abstract

In this paper, new upper bounds on the maximum attainable minimum Hamming distance of Turbo codes with arbitrary — including the best — interleavers are established using a combinatorial approach. These upper bounds depend on the interleaver length, on the code rate and on the scramblers employed in the encoder. Examples of the new bounds for particular Turbo codes are given and discussed. The new bounds are tighter than all existing ones and prove that the minimum Hamming distance of Turbo codes cannot asymptotically grow at a rate more than the third root of the codeword length.

Keywords: Turbo code, minimum Hamming distance bounds, interleaver design

The authors would like to thank the Fraunhofer Gesellschaft – Institut für Integrierte Schaltungen, Erlangen, for supporting this work.

I. Introduction

In contrast to most conventional channel codes, which were constructed starting from a theoretical viewpoint (e.g. BCH codes), the invention of Turbo codes was mainly founded on an experimental basis [1]. More specifically, the inventors of Turbo codes investigated encoder structures generating codes, that could be simply decoded using soft-output decoders. This led to the Turbo encoder structure of [2], which contains in its very core an interleaver. Turbo codes are still — besides the more recently discovered *irregular low density parity check codes* [3] — the most powerful channel codes known up to date as they exhibit a remarkable power-efficiency [4]. Only very low signal-to-noise ratios, which are in the magnitudes predicted by information theory, are needed for achieving Bit Error Rates (BER) down to typically 10^{-7}. However nothing was known about the code structure, e.g. the distance spectrum, at the beginning, nor was anything known, why Turbo codes perform so well. Soon it was discovered that the minimum Hamming distance of a Turbo code is very low for practically every randomly chosen interleaver [5]. The low minimum distance explained the fact that Turbo codes possess an *error floor*, where the BER curve suddenly flares out at a low BER. To solve the problem of the undesired error floor, several algorithms were proposed (e.g. [6],[7],[8],[9]) for constructing the interleaver rather than choosing it randomly.

The following years of research brought a wealth of contributions focussing on different aspects of the Turbo code structure, e.g. [10],[11],[12],[13]. There are several papers giving bounds on the minimum distance of Turbo codes: [14] introduced a hypothetical "fully optimal interleaver" to give upper bounds on the BER performance. In [15], it was shown that for a randomly chosen interleaver, the minimum distance does not grow with the codeword length with asymptotically probability 1. The paper [16] stated an upper bound on the minimum distance for a *given* interleaver. To derive this bound, only Turbo encoder input words of weight 2 were taken into consideration. Only these weight-2 input words were considered in [17], too, but here an upper bound on *all* possible interleavers of an arbitrary length was given.

The present paper sheds some more light on the code structure and the distance spectrum of Turbo codes, particularly on the minimum distance. Like [17] it shows, how the minimum distance can be upper-bounded for *all* possible interleavers of an arbitrary length, i.e. it gives upper bounds on the minimum distance also for the *best* of all possible interleavers of that length. However, in contrast to the *geometric* considerations performed in [17], we use a *combinatorial* approach here, and we consider input words of weight 2 as well as 4. This approach reveals the fact that the minimum distance cannot grow more quickly than the third root of the codeword length, such that unattainable minimum distances are identified for Turbo codes. The derived new bounds are significantly tighter

than that of [17], and they are by far lower for medium and large codewords than conventional upper bounds for general linear binary block codes, e.g. the Hamming or the McEliece/Rodemich/Rumsey/Welch bound, and the difference increases with the codeword length. Thus, the new upper bounds can be used as a benchmark for assessing the performance of interleaver design algorithms like the *s-random* design of [6] and of other designed Turbo code interleavers. Turbo codes are often referred to as "Shannon codes" due to their almost optimum power-efficiency and their random-like structure. However, the paper reveals that in fact Turbo codes differ from "Shannon codes", since their minimum distance grows, at best, at a rate much less than that of Shannon's genuine random codes.

The paper is organized as follows: At first, we present a model of the Turbo encoder in Section II, which we are considering throughout the paper, and introduce the notation and terminology used in the paper. In Section III we first introduce some basic concepts and tools, which we will need to upper-bound the minimum distance. Then the new upper bounds are derived in this section. We then show examples of the newly derived bounds for particular Turbo codes in Section IV, compare them to existing bounds and discuss the implications. We close with a short summary in Section V.

II. Model of the Turbo encoder

The Turbo codes considered in this paper are parallel concatenated convolutional codes, whose encoder model is shown in Fig. 1. The encoder accepts a binary input word $\boldsymbol{u} = (u_0, \ldots, u_{K-1}) \in [\mathrm{GF}(2)]^K$ of length K, where GF(2) represents the binary Galois field, and propagates it to three parallel branches. The upper branch passes $\boldsymbol{u}$ to the systematic part of the Turbo codeword $\boldsymbol{c}$. The middle and the lower branches of the encoder produce the two parity parts of $\boldsymbol{c}$, which we refer to as the first and the second component parity word $\boldsymbol{c}^{(1)}$ and $\boldsymbol{c}^{(2)}$, respectively. These two branches are consequently termed the first and the second *component*, respectively, and all quantities $\bullet$ associated with the i-th component ($i = 1, 2$) will be denoted by $\bullet^{(i)}$. The Turbo codeword consists hence of the three parts $\boldsymbol{c} = (\boldsymbol{u}, \boldsymbol{c}^{(1)}, \boldsymbol{c}^{(2)})$.

The input to the first component is the input word $\boldsymbol{u}^{(1)} = \boldsymbol{u}$ in its natural order, which is fed into a scrambler. A scrambler ("scr" in the figure) is a linear time-invariant filter with an infinite impulse response (IIR), where all quantities and operations are taken from GF(2). It consists of a shift-register with a feedback- and a feedforward-branch. The output of the scrambler is then possibly punctured ("pct") in order to discard some of the parity symbols produced in the first component and hence to increase the rate of the Turbo code adequately.

The operation in the second component is similar. The main difference to the first component is that not $\boldsymbol{u}$ but an *interleaved* input word $\boldsymbol{u}^{(2)}$ is scrambled and punctured. The

interleaver performs a permutation of the elements of $\boldsymbol{u}$ to those of $\boldsymbol{u}^{(2)}$. The performed displacements of elements can be described by a vector $\boldsymbol{\pi} = (\pi_0, \ldots, \pi_{K-1})$, that defines a mapping from the indices of $\boldsymbol{u}^{(1)}$ to those of $\boldsymbol{u}^{(2)}$, such that $u^{(2)}_{\pi_i} = u_i,\ i = 0, \ldots, K-1$. Note that for our encoder model, the interleaver length is identical to the input word length K.

Besides the notation $\boldsymbol{x} = (x_0, \ldots, x_j)$ we also use the D-transform $x(D) = \bigoplus_{i=0}^{j} x_i \cdot D^i$ to represent a vector $\boldsymbol{x}$ in this paper, if this representation is more favourable. Here $\oplus$ and $\bigoplus$ represent the addition in GF(2). In this paper we restrict ourselves without loss of generality to the case that the scramblers and puncturers in both components are identical. A generalization of the derived bounds to differing scramblers and/or puncturers is however straightforward. We do not rule out the case that the input word $\boldsymbol{u}$ contains redundant symbols in order to terminate one or both component scramblers in the zero-state at the end of the input word [18]. Since $\boldsymbol{u}$ does not necessarily contain exclusively information symbols, we use the term *input* word rather than *information* word. However, the bounds established and the conclusions drawn in this paper are valid for the cases that neither, only one, or both scramblers are terminated in the zero-state by $\boldsymbol{u}$.

III. Derivation of the new upper bounds

In this section, we will develop two different combinatorial approaches to establish upper bounds on the minimum Hamming distance of Turbo codes. At first, some simple reflections are performed on how we can upper-bound the weight of a Turbo codeword, and we motivate the approach chosen. Next we present an index partition strategy, which we will then use as a tool to pursue the aforementioned two combinatorial approaches.

A. Basic considerations on the minimum distance

Since a Turbo code is linear, its distance spectrum is identical to its weight spectrum. For determining the minimum distance, we can hence identify the minimum of the codeword weights associated with all possible input words $\boldsymbol{u}$ other than the all zero-word $\mathbf{0}$. The set of possible input words is denoted by $\mathcal{U}$. Furthermore, we are interested in the maximum possible minimum distance for *all* possible interleavers $\boldsymbol{\pi}$ of a given length K. Thus, the maximum attainable minimum distance $\delta_{\min,\max}$ of a Turbo code, for which we want to establish an upper bound in this paper, can be defined as follows:

$$\delta_{\min,\max} = \max_{\boldsymbol{\pi}} \{ \min_{\boldsymbol{u} \in \mathcal{U} \setminus \{\mathbf{0}\}} \{ w(\mathrm{enc}_{\boldsymbol{\pi}}(\boldsymbol{u})) \} \}. \tag{1}$$

In this equation, the function $\mathrm{enc}_{\boldsymbol{\pi}} : \boldsymbol{u} \mapsto \boldsymbol{c} = \mathrm{enc}_{\boldsymbol{\pi}}(\boldsymbol{u})$ represents the Turbo encoder function, i.e. the mapping from input words $\boldsymbol{u}$ to codewords $\boldsymbol{c}$ using the particular interleaver $\boldsymbol{\pi}$, and $w(\boldsymbol{x})$ is the Hamming weight of a vector $\boldsymbol{x}$.

To make the derivation of an upper bound on $\delta_{\min,\max}$ feasible, we will not consider the complete set $\mathcal{U} \setminus \{\mathbf{0}\}$ of input words other than the zero-word in this paper. Instead, we will show that for *every* interleaver $\boldsymbol{\pi}$, there exists an input word $\boldsymbol{u}(\boldsymbol{\pi}) \in \mathcal{U} \setminus \{\mathbf{0}\}$, which depends on the particular interleaver $\boldsymbol{\pi}$ and whose associated codeword has low weight $w(\mathrm{enc}_{\boldsymbol{\pi}}(\boldsymbol{u}(\boldsymbol{\pi}))) \leq \delta_{\min,\mathrm{UB}}$, but where the upper bound $\delta_{\min,\mathrm{UB}}$ does not depend on the interleaver. Using Eq. (1), we obtain therefore an upper bound as follows

$$\begin{aligned} \delta_{\min,\max} &\leq \max_{\boldsymbol{\pi}}\{w(\mathrm{enc}_{\boldsymbol{\pi}}(\boldsymbol{u}(\boldsymbol{\pi})))\} && (2)\\ &\leq \delta_{\min,\mathrm{UB}}, && (3) \end{aligned}$$

since the specific input word (or words) of $\mathcal{U}$, for which the minimum is taken in Eq. (1) for a given interleaver $\boldsymbol{\pi}$, might be different from $\boldsymbol{u}(\boldsymbol{\pi})$. Thus, what we derive in this paper, is not an equation for the actually attainable $\delta_{\min,\max}$ but upper bounds $\delta_{\min,\mathrm{UB}}$ on this quantity.

For upper-bounding the minimum distance $\delta_{\min,\max}$ of Turbo codes, we will at first concentrate on Turbo encoder input words $\boldsymbol{u}(\boldsymbol{\pi})$ of weight 2 for deriving our first upper bounds in Section III-E. The reason is that the behaviour of the Turbo encoder is analytically tractable for these input words. In a second step, we will extend the considerations to input words $\boldsymbol{u}(\boldsymbol{\pi})$ of weight 2 *and* 4 in order to derive tighter upper bounds in Section III-F.

Since the Turbo codeword $\boldsymbol{c}$ is composed of the three parts $(\boldsymbol{u}, \boldsymbol{c}^{(1)}, \boldsymbol{c}^{(2)})$, its weight can be split into three terms:

$$w(\boldsymbol{c}) = w(\boldsymbol{u}) + w(\boldsymbol{c}^{(1)}) + w(\boldsymbol{c}^{(2)}), \tag{4}$$

i.e. the weight of the Turbo encoder input word $\boldsymbol{u}$ and the outputs of the two component scrambler/puncturer combinations. To start with the simple, let us consider a *single* component at first by examining the reaction of the component scramblers on weight-2 input words.

B. Bounding the component parity weight for weight-2 input words

It is well known, e.g. from [19], that a binary shift-register containing a feedback-branch, as used in our component scramblers, responds to an impulse (a single "1" followed by "0"s) at its input by periodically cycling through p distinct shift-register states (p is called the *period* of the shift-register). Furthermore, an input sequence "1 $0^{\kappa \cdot p-1}$ 1", $\kappa \in \mathbb{N}^{+1}$, i.e. two "1"s spaced a positive integer multiple of p positions apart from each other, leads to an *error event* (with respect to the all zero-path in the scrambler trellis) of finite length $\kappa \cdot p + 1$, i.e. a deviation from the zero-path. The scrambler is driven out of the zero-state by the first "1", then it performs $\kappa - 1$ complete periods, and it is driven back to the

[1] N^{+} denotes the set of positive integers.

zero-state by the second "1" just before the κ-th period would be complete. The following Lemma, which repeats Lemma 1 from [17], uses this fact:

Lemma 1 *Let the function* scrp *denote the mapping from input to output of a given combination of a scrambler of memory ν and a puncturer. The scrambler consists of a shift-register with a feedback-branch described by a monic feedback-polynomial of degree ν and a feedforward-branch. Let the scrambler's period be denoted by p. Then for every* scrp*, there exist two numbers $\alpha, \beta \in \mathbb{N} \cup \{0\}$ such that*

$$w(\mathrm{scrp}(u(D))) \leq \alpha \cdot \kappa + \beta \qquad \text{for } u(D) = D^i \oplus D^{i+\kappa \cdot p} \tag{5}$$

holds $\forall i, \kappa \in \mathbb{N} \cup \{0\}$. The proof is straightforward: Since the error event is of finite length $\kappa \cdot p + 1$, the associated output weight cannot surpass $\kappa \cdot p + 1$, and we obtain the upper bound $w(\mathrm{scrp}(u(D))) \leq \kappa \cdot p + 1$ by setting $\alpha = p$ and $\beta = 1$. ■

The above Lemma provides hence an upper bound on the weight of a component parity word, if the component input word contains – besides zeros – only two "1"s, whose associated indices $i, i + \kappa \cdot p$ belong to the same *equivalence class* with respect to the modulo p-operation: $i \bmod p = (i + \kappa \cdot p) \bmod p \in \{0, \ldots, p-1\}$. Let us introduce the term *weight-2 error word* for such a component input word $u(D) = D^i \oplus D^{i+\kappa \cdot p}$ resulting in a finite-length error event in the trellis of the component scrambler. Note that the bound of Lemma 1 grows with an increasing difference $\kappa \cdot p$ between the indices of the two "1"s. Observe the relationship to active distances known from woven codes [20]. The minimum value of α can similarly be interpreted as the *asymptotic slope* [20] of the scrambler/puncturer output weight for an input of two "1"s spaced a multiple of p positions apart from each other. Table I displays examples for scramblers of memory $\nu = 2, 3, 4$ with an unpunctured output (resulting in an overall Turbo code rate $R = 1/3$) and lists the associated period p and the minimum values of α and β. The polynomials describing the feedforward- and the feedback-branch (numerator and denominator of the transfer function) of the component scramblers are represented in octal notation. All three scramblers of Table I have the maximum possible period $p = 2^\nu - 1$, since the feedback-polynomials are primitive.

TABLE I

(α, β) BOUNDING VALUES AND PERIOD p FOR COMPONENT SCRAMBLERS OF DIFFERENT MEMORY ν

ν	Transf.fct.	p	(α, β) f. $R = 1/3$
2	$5_8/7_8$	3	(2,2)
3	$17_8/13_8$	7	(4,2)
4	$35_8/23_8$	15	(8,2)

C. The Turbo codeword weight for weight-2 input words

Having examined the *component* scramblers, we shall now focus on the reaction of the *complete* Turbo encoder on input words $\boldsymbol{u}$ of weight $w(\boldsymbol{u}) = 2$. A Turbo codeword of low weight occurs, if all three terms of Eq. (4) have low weight. We know from the preceding subsection that the component parity words $\boldsymbol{c}^{(1)}$ and $\boldsymbol{c}^{(2)}$, respectively, have low weight, if both $\boldsymbol{u}$ and $\boldsymbol{u}^{(2)}$ are weight-2 error words, and if the two "1" elements of $\boldsymbol{u}$ and $\boldsymbol{u}^{(2)}$, respectively, are in the proximity of each other. For a *given* interleaver $\boldsymbol{\pi}$, let us consider weight-2 error words $\boldsymbol{u}$ at the input of the first component, which are permuted by the interleaver $\boldsymbol{\pi}$ to weight-2 error words $\boldsymbol{u}^{(2)}$ at the input of the second component.

In [7] it is shown that due to the *pigeonhole principle* [21], at least one such pair $(\boldsymbol{u}, \boldsymbol{u}^{(2)})$ of weight-2 error words exists for every interleaver, if only the interleaver length satisfies $K > p^2$. The *simple form* of the pigeonhole principle states that if a set of $> L$ elements (the *pigeons*) is partitioned into L subsets (the *pigeonholes*), then at least one subset contains ≥ 2 elements (i.e. there will be more than one pigeon in at least one pigeonhole) [21]. In [7], this principle is employed as follows: Partition all K indices $i \in \{0, \ldots, K-1\}$ of the first component input word into subsets, such that all elements i of a subset have identical residues $i \bmod p$ *and* $\pi_i \bmod p$ in the first and in the second component, respectively. There is hence a total of p^2 subsets. If $K > p^2$, then the pigeonhole principle states that there exists at least one pair (i_1, i_2) with $i_1 \bmod p = i_2 \bmod p$ *and* $\pi_{i_1} \bmod p = \pi_{i_2} \bmod p$, hence both $u(D) = D^{i_1} \oplus D^{i_2}$ *and* the associated $u^{(2)}(D) = D^{\pi_{i_1}} \oplus D^{\pi_{i_2}}$ are weight-2 error words. Note that the set of these particular input words $\boldsymbol{u}$ and $\boldsymbol{u}^{(2)}$, respectively, depends on the specific interleaver $\boldsymbol{\pi}$, but we will leave out this dependence in our notation for the sake of readability. Let us consider one of these weight-2 error words $u(D) = D^{i_1} \oplus D^{i_2}$ with $i_2 = i_1 + \kappa \cdot p,\ \kappa \in \mathbb{N}^+$ for a given interleaver. For such an input word, we can state an upper bound on the associated codeword weight using Eq. (4) and Lemma 1:

$$w(\boldsymbol{c}) \leq 2 + \alpha \cdot \frac{|i_1 - i_2|}{p} + \beta + \alpha \cdot \frac{|\pi_{i_1} - \pi_{i_2}|}{p} + \beta \tag{6}$$

$$= 2 + \alpha \cdot \frac{|i_1 - i_2| + |\pi_{i_1} - \pi_{i_2}|}{p} + 2\beta. \tag{7}$$

Since we consider only those $\boldsymbol{u}$ here, whose associated $\boldsymbol{u}^{(2)}$ is a weight-2 error word, the term $|\pi_{i_1} - \pi_{i_2}|$ is a multiple of p.

An interleaver $\boldsymbol{\pi}$, that is optimum in the maximization of Eq. (1), must avoid low weight codewords. The optimum interleaver must distribute all pairs of indices belonging to the same equivalence class ($\bmod p$) in $\boldsymbol{u}$ (i.e. forming weight-2 error words), which are in the proximity of each other, to either (a) distinct equivalence classes ($\bmod p$), i.e. not forming a weight-2 error word in the second component, or (b) sufficiently distant indices in $\boldsymbol{u}^{(2)}$.

As mentioned above, [7] shows that no interleaver can enable (a) for all pairs $\boldsymbol{u}, \boldsymbol{u}^{(2)}$ if $K > p^2$, i.e. there is at least one weight-2 error word $\boldsymbol{u}$ permuted to a weight-2 error word $\boldsymbol{u}^{(2)}$ for every interleaver of length $K > p^2$. Our first upper bound $\delta_{\min,\mathrm{UB}}$ on the maximum attainable minimum Hamming distance $\delta_{\min,\max}$ of Turbo codes will elaborate on (b) by showing that there exist further restrictions on the possible index differences between the two "1" elements in the input words $\boldsymbol{u}$ and $\boldsymbol{u}^{(2)}$. To motivate the approaches taken for our first bound, let us state more precisely, what we want to show:

Statement for input weight 2: For any interleaver $\boldsymbol{\pi}$ of length K, there exists a pair of indices (i_1, i_2) of $\boldsymbol{u}$ with the following property: i_1, i_2 belong to an identical equivalence class ($\bmod p$) and are in the proximity of each other, i.e. $|i_1 - i_2| \leq s^{(1)}$ with some small $s^{(1)}$, *and* the associated indices π_{i_1}, π_{i_2} (after the permutation) belong to an identical equivalence class and are in the proximity of each other, i.e. $|\pi_{i_1} - \pi_{i_2}| \leq s^{(2)}$ with some small $s^{(2)}$.

We still have to define, what "proximity" exactly means in this context, i.e. we want to derive an upper bound on the sum of the index differences $|i_1 - i_2| + |\pi_{i_1} - \pi_{i_2}| \leq s^{(1)} + s^{(2)}$ (cf. Ineq. (7)), where $s^{(1)} + s^{(2)}$ is independent of the particular interleaver $\boldsymbol{\pi}$.

A similar task was solved in [17]. Let us shortly summarize the conclusions of this paper, in order to motivate the new combinatorial approach chosen in the present paper. [17] shows that for every interleaver of a given length K, there exists at least one pair of indices (i_1, i_2) with the following two properties resembling those given in the statement above:

(a) i_1 and i_2 belong to an identical equivalence class ($\bmod p$), i.e. $i_1 \bmod p = i_2 \bmod p$, and π_{i_1} and π_{i_2} belong to an identical equivalence class, i.e. $\pi_{i_1} \bmod p = \pi_{i_2} \bmod p$;

(b) For any interleaver length K and employed component scramblers/puncturers, a quantity l can be found with the following property: if i_2 is within a neighbourhood $\{i_1 - s^{(1)}, \ldots, i_1 + s^{(1)}\}$ of i_1, then π_{i_2} is within a neighbourhood $\{\pi_{i_1} - s^{(2)}(s^{(1)}), \ldots, \pi_{i_1} + s^{(2)}(s^{(1)})\}$ of π_{i_1}, where $s^{(2)}(s^{(1)}) = l - s^{(1)}$. Hence the sum $s^{(1)} + s^{(2)}(s^{(1)}) = l$ depends only on K and the scramblers/puncturers, but not on the particular interleaver $\boldsymbol{\pi}$.

From this sum $s^{(1)} + s^{(2)}$, [17] deduces an interleaver-independent upper bound $\delta_{\min,\mathrm{UB}}$ on the associated codeword weight using Ineq. (7). The bound of [17] was based on geometric considerations. In the following elaborations, we will develop a different bounding method using a combinatorial approach to conceive new two upper bounds, which are tighter than the bound of [17]. Let us now introduce the tool, which we will need in our combinatorial approach.

D. Index partition strategy

In this subsection we will present a strategy for partitioning all input word indices into subsets. Using this partition strategy in our combinatorial approach, we can prove the

statement for input weight 2 given in the preceding subsection. Hence there exists a pair (i_1, i_2) with property *(a)* as stated above. In contrast to [17], we show that these indices i_1, i_2 have additionally the following property replacing *(b)*:

($\tilde{b}$) i_1 is in the same index subset of *width* $s^{(1)}$ in $\boldsymbol{u}^{(1)}$ as i_2, and π_{i_1} is in the same index subset of width $s^{(2)}$ in $\boldsymbol{u}^{(2)}$ as π_{i_2}, where the *width* of an index subset represents the maximum of the differences between all the indices belonging to this index subset.

In order to show *($\tilde{b}$)*, let us partition the set of indices $\{0, \ldots, K-1\}$ into Λ disjoint subsets $\mathcal{I}_l,\ l = 0, \ldots, \Lambda - 1$, such that (a) every $\mathcal{I}_l$ is a subset of an equivalence class $(\bmod p)$, and (b) the maximum width (see above) of the Λ index subsets is minimized. This leads to the following partition rules:

(A) Each subset $\mathcal{I}_l,\ l = 0, \ldots, \Lambda - 1$ is composed of *consecutive* indices belonging to an identical equivalence class $(\bmod p)$. That is, if ι_l denotes the smallest index belonging to subset $\mathcal{I}_l$, then the subset also contains $\iota_l + p, \iota_l + 2p, \ldots, \iota_l + (|\mathcal{I}_l| - 1) \cdot p$, where $|\mathcal{I}_l|$ denotes the cardinality of $\mathcal{I}_l$.

(B) The cardinalities of the Λ subsets $\mathcal{I}_l,\ l = 0, \ldots, \Lambda - 1$ differ at most by 1, such that their widths differ at most by p (see *(A)*). Hence, each subset contains at least

$$\Omega \stackrel{\triangle}{=} \lfloor K/\Lambda \rfloor \tag{8}$$

elements, where $\lfloor x \rfloor$ represents the largest integer $\leq x$, i.e. the *floor* function of the quantity x. Without loss of generality, we partition the set of indices such that the first $\Psi \stackrel{\triangle}{=} K \bmod \Lambda$ subsets $\mathcal{I}_l,\ l = 0, \ldots, \Psi - 1$ contain $\Omega + 1$ elements each. The remaining $\Lambda - \Psi$ subsets $\mathcal{I}_l,\ l = \Psi, \ldots, \Lambda - 1$ contain Ω elements each. Together with *(A)* we obtain therefore the following subsets:

$$\mathcal{I}_l \stackrel{\triangle}{=} \begin{cases} \{\iota_l, \iota_l + p, \iota_l + 2p, \ldots, \iota_l + (\Omega - 1) \cdot p, \iota_l + \Omega \cdot p\} & \text{for } 0 \leq l < \Psi \\ \{\iota_l, \iota_l + p, \iota_l + 2p, \ldots, \iota_l + (\Omega - 1) \cdot p\} & \text{for } \Psi \leq l < \Lambda. \end{cases} \tag{9}$$

(C) Every subset $\mathcal{I}_l,\ l = 0, \ldots, \Lambda - 1$ must contain at least 2 indices. Thus, we have $\Omega \geq 2$ or equivalently $\Lambda \leq K/2$. This condition *(C)* is necessary since we need *two distinct* input word indices belonging to an identical subset in the following proofs.

Note that $\Omega + 1 = \lceil K/\Lambda \rceil$ for $\Psi > 0$ and $\Omega = \lfloor K/\Lambda \rfloor = \lceil K/\Lambda \rceil$ for $\Psi = 0$. Here $\lceil x \rceil$ denotes the smallest integer $\geq x$, i.e. the *ceiling* function. Thus, the maximum number of elements of all index subsets is $\max_{l=0,\ldots,\Lambda-1}\{|\mathcal{I}_l|\} = \lceil K/\Lambda \rceil$ regardless of Ψ, and the maximum width of all subsets, i.e. the maximum difference between any two indices belonging to an identical subset, is

$$\max_{l=0,\ldots,\Lambda-1} \Big\{ \max_{i_1, i_2 \in \mathcal{I}_l} \{|i_1 - i_2|\} \Big\} = \left(\left\lceil \frac{K}{\Lambda} \right\rceil - 1 \right) \cdot p. \tag{10}$$

We will utilize these rules for partitioning all indices of the input word in the forthcoming proofs. For the case that the number of subsets Λ is a multiple of the period length p, i.e.

for $\Lambda \bmod p = 0$ with $p \leq \Lambda \leq K/2$ (which immediately implies the necessary condition $K \geq 2p$), there does actually exist at least one index partitioning according to these rules for every input word length K.

An example of such a partition strategy is illustrated in Fig. 2, where we assume $p = 3$ and we partition the $K = 29$ indices $\{0, \ldots, 28\}$ into $\Lambda = 9$ subsets. Here we display the set of indices in an array with columns of height $p = 3$ and with rows of width $\lceil K/p \rceil = 10$ for the $K \bmod p = 2$ up-most rows and width $\lfloor K/p \rfloor = 9$ for the remaining $p - (K \bmod p) = 1$ down-most rows, respectively. The indices are *columnwise* written into the array elements starting from the left-upper corner, i.e. the left-most column contains the indices 0, 1, 2 (read from top to bottom) and the top row contains the indices $0,\ p = 3,\ 2p = 6, \ldots$ (read from left to right). Obviously, each row contains all elements belonging to an identical equivalence class. We find that $\Omega = 3$. The subsets $\mathcal{I}_l$ are therefore blocks of $\Omega + 1 = 4$ or $\Omega = 3$ *consecutive* array elements in the same row, where the boundaries between subsets are displayed as thick lines in the figure. As Fig. 2 shows, the subsets $\mathcal{I}_l$ are placed columnwise in the array, starting in the left-upper corner.

In general, the array is filled up with the subsets $\mathcal{I}_l$ in the following manner. The first p subsets $\mathcal{I}_0, \ldots, \mathcal{I}_{p-1}$ can be successively put below each other starting from the left-upper corner. Then subset $\mathcal{I}_p$ is put immediately right of subset $\mathcal{I}_0$. The next $p - 1$ subsets are successively put below subset $\mathcal{I}_p$. Then the next subset (subset $\mathcal{I}_{2p}$) is placed immediately right of subset $\mathcal{I}_p$ and so on. The first Ψ subsets (in Fig. 2, we have $\Psi = 2$) are wider by one element than the remaining $\Lambda - \Psi$ subsets ($\Omega + 1$ elements compared to Ω). The first subset with only Ω elements is subset $\mathcal{I}_\Psi$. Hence the subsets $\mathcal{I}_{\Psi+p}, \mathcal{I}_{\Psi+2p}, \mathcal{I}_{\Psi+3p}, \ldots$ are shifted left by one element compared to the subset directly above them as can be viewed in Fig. 2.

We can state this specific index partitioning into Λ subsets in a more formal way, supposed that Λ is a multiple of p. To obtain an arrangement of the corresponding subsets as described above and exemplified in Fig. 2, the smallest index ι_l in subset $\mathcal{I}_l$ can be recursively determined as follows

$$\iota_l = \begin{cases} l & \text{for } 0 \leq l < p \\ \iota_{l-p} + p \cdot (\Omega + 1) & \text{for } 0 \leq l - p < \Psi \\ \iota_{l-p} + p \cdot \Omega & \text{for } \Psi \leq l - p < \Lambda - p, \end{cases} \tag{11}$$

i.e. we calculate the smallest index ι_l of subset $\mathcal{I}_l$ by referring to subset $\mathcal{I}_{l-p}$, which is adjacent to the left of subset $\mathcal{I}_l$ (cf. Fig. 2). We can hence compute ι_l from ι_{l-p} by taking into account the width — and thereby the cardinality — of subset $\mathcal{I}_{l-p}$. In the rest of the paper, we will exclusively use the partitioning defined above.

We see from the above partition rules *(A)* and *(B)*, that if two indices belong to an identical subset, then they belong to the same equivalence class and they are in the proximity

of each other. As shown above, the maximum width of all index subsets is $(\lceil K/\Lambda \rceil - 1) \cdot p$, which is the maximum possible difference between any two indices provided they belong to an identical index subset. If the input word to a scrambler/puncturer combination consists of "1"s in two positions belonging to an identical subset and otherwise only "0"s, then the scrambler output is of a low weight and can be upper-bounded by applying Lemma 1. This fact will now be exploited in the first combinatorial approach, which we present in the following subsection.

E. Upper bound for input words of weight 2

In this subsection we restrict ourselves to taking into consideration only input words of weight 2. Fig. 3a displays an example, where such a first component input word $u(D) = D^i \oplus D^{i+\kappa_1 \cdot p},\ \kappa_1 \in \mathbb{N}^+$ is permuted by the interleaver $\boldsymbol{\pi}$ to a second component input word $u^{(2)}(D) = D^j \oplus D^{j+\kappa_2 \cdot p},\ \kappa_2 \in \mathbb{N}^+$ (for this example, we have $p = 3$ and $\kappa_1 = \kappa_2 = 1$). The arrows in the figure denote the displacements of the "1" elements from $\boldsymbol{u}$ to $\boldsymbol{u}^{(2)}$ by the interleaver. Our aim in this subsection is to derive an upper bound $\delta_{\min,\mathrm{UB}}$ (cf. Ineq. (3)) on the minimum distance, which is independent of the particular interleaver, by using the partition strategy introduced in the preceding subsection.

Let us use our partition strategy on both the indices of $\boldsymbol{u}$ as well as the indices of $\boldsymbol{u}^{(2)}$. Let the number of subsets in the first and second component be denoted by $\Lambda^{(1)}$ and $\Lambda^{(2)}$, respectively (we assume hence that $\Lambda^{(1)} \bmod p = \Lambda^{(2)} \bmod p = 0$). The subsets are referred to as $\mathcal{I}_l^{(1)},\ l \in \{0, \ldots, \Lambda^{(1)} - 1\}$ and $\mathcal{I}_m^{(2)},\ m \in \{0, \ldots, \Lambda^{(2)} - 1\}$, and let the quantities $\Psi^{(1)}$, $\Psi^{(2)}$, $\Omega^{(1)}$, and $\Omega^{(2)}$ be defined in analogy to Section III-D.

We know that an error event of finite length occurs in the first component, and that the associated parity word $\boldsymbol{c}^{(1)}$ has relatively low weight, if both "1"s of $\boldsymbol{u}$ are in positions belonging to an identical subset $\mathcal{I}_l^{(1)}$. When we use the partition strategy as exhibited above, then at least the subset $\mathcal{I}_0^{(1)}$ has cardinality $|\mathcal{I}_0^{(1)}| = \lceil K/\Lambda^{(1)} \rceil$. The elements of $\mathcal{I}_0^{(1)}$, which are indices of $\boldsymbol{u}$, are mapped by the interleaver to indices of $\boldsymbol{u}^{(2)}$. These indices of $\boldsymbol{u}^{(2)}$ are partitioned into $\Lambda^{(2)}$ subsets $\mathcal{I}_m^{(2)}$ according to the above partition strategy. If we choose the pair $(\Lambda^{(1)}, \Lambda^{(2)})$, such that $\lceil K/\Lambda^{(1)} \rceil > \Lambda^{(2)}$, then we have $|\mathcal{I}_0^{(1)}| = \lceil K/\Lambda^{(1)} \rceil > \Lambda^{(2)}$. By virtue of the aforementioned pigeonhole principle, it is not possible for *any* interleaver $\boldsymbol{\pi}$ that all elements of $\mathcal{I}_0^{(1)}$ are mapped to *distinct* subsets $\mathcal{I}_m^{(2)}$ for this pair $(\Lambda^{(1)}, \Lambda^{(2)})$. For *every* interleaver $\boldsymbol{\pi}$, there is at least one pair of indices (i_1, i_2) belonging to $\mathcal{I}_0^{(1)}$, which are mapped by the interleaver to an *identical* subset $\mathcal{I}_{m_0}^{(2)}$. Because of the partition strategy used, the indices i_1 and i_2 of $\boldsymbol{u}$ belong to the same equivalence class $(\bmod p)$ and are separated by $\leq \lceil K/\Lambda^{(1)} \rceil - 1$ scrambler periods. The associated indices π_{i_1} and π_{i_2} of $\boldsymbol{u}^{(2)}$ also belong to the same equivalence class and are separated by $\leq \lceil K/\Lambda^{(2)} \rceil - 1$ periods. Let us consider the input word $u(D) = D^{i_1} \oplus D^{i_2}$ with "1"s in these two positions. Then

an upper bound for the associated codeword weight can be calculated from Ineq. (7):

$$w(\mathrm{enc}_{\boldsymbol{\pi}}(u(D)) \leq 2 + \alpha \cdot \left(\left\lceil \frac{K}{\Lambda^{(1)}} \right\rceil - 1 + \left\lceil \frac{K}{\Lambda^{(2)}} \right\rceil - 1 \right) + 2\beta. \tag{12}$$

Although the particular pair of indices (i_1, i_2) and the associated input word $u(D)$ and codeword are dependent on the considered interleaver $\boldsymbol{\pi}$, we see that this upper bound depends only on the pair $(\Lambda^{(1)}, \Lambda^{(2)})$ and is *independent* of the particular interleaver. At least one codeword of a weight less or equal than the given upper bound exists for *every* possible interleaver $\boldsymbol{\pi}$. To make the above upper bound as low as possible, we have to make $\lceil K/\Lambda^{(1)} \rceil$ and $\lceil K/\Lambda^{(2)} \rceil$ as small as possible, i.e. $\Lambda^{(1)}$ and $\Lambda^{(2)}$ large. Note that the above condition $\lceil K/\Lambda^{(1)} \rceil > \Lambda^{(2)}$ translates into $K/\Lambda^{(1)} > \Lambda^{(2)}$, since $\Lambda^{(2)}$ is integer. Following the argumentation leading to Ineq. (3), we obtain therefore the following upper bound on $\delta_{\min,\max}$:

Theorem 1 *The minimum distance $\delta_{\min,\max}(K, \alpha, \beta, p)$ of a parallel concatenated convolutional code with interleaver length K, and parameters α, β and p describing the component scramblers as stated in Lemma 1, is upper-bounded by*

$$\delta_{\min,\max} \leq 2 + \alpha \cdot \left(\min_{(\Lambda^{(1)},\Lambda^{(2)}) \in \mathcal{L}_2} \left\{ \left\lceil \frac{K}{\Lambda^{(1)}} \right\rceil + \left\lceil \frac{K}{\Lambda^{(2)}} \right\rceil \right\} - 2 \right) + 2\beta, \tag{13}$$

where the minimization is performed over the set $\mathcal{L}_2$ of values $\{(\Lambda^{(1)}, \Lambda^{(2)})\} \subseteq \{1, \ldots, K/2\}^2$, for which both of the following conditions are true

(1) $\Lambda^{(1)} \bmod p = \Lambda^{(2)} \bmod p = 0$

(2) $\frac{K}{\Lambda^{(1)}} > \Lambda^{(2)}$.

We have hence transformed the complex maximin-problem of Eq. (1) into a discrete optimization problem, which is solvable even by a brute-force method (considering every pair $(\Lambda^{(1)}, \Lambda^{(2)}) \in \mathcal{L}_2$). However, we can simplify Theorem 1 as follows. The minimization in Ineq. (13) can be rewritten:

$$\min_{(\Lambda^{(1)},\Lambda^{(2)}) \in \mathcal{L}_2} \left\{ \left\lceil \frac{K}{\Lambda^{(1)}} \right\rceil + \left\lceil \frac{K}{\Lambda^{(2)}} \right\rceil \right\} = \min_{\Lambda^{(1)}} \left\{ \left\lceil \frac{K}{\Lambda^{(1)}} \right\rceil + \min_{\Lambda^{(2)} < \frac{K}{\Lambda^{(1)}}} \left\{ \left\lceil \frac{K}{\Lambda^{(2)}} \right\rceil \right\} \right\}, \tag{14}$$

where we minimize over $\Lambda^{(1)}, \Lambda^{(2)} \in \{1, \ldots, K/2\}$ with $\Lambda^{(1)}, \Lambda^{(2)} \bmod p = 0$. Clearly the inner expression $\lceil K/\Lambda^{(2)} \rceil$ becomes minimal, when $\Lambda^{(2)} < K/\Lambda^{(1)}$ becomes maximal. In the further elaborations, we will frequently make use of the following inequalities for the ceiling function:

$$\begin{array}{rccccc} & \lceil x \rceil - 1 & < & x & \leq & \lceil x \rceil \\ \Leftrightarrow & x & \leq & \lceil x \rceil & < & x + 1 \end{array} \quad \forall x \in \mathbb{R}. \tag{15}$$

The optimum $\Lambda^{(2)}$ for a given $\Lambda^{(1)}$ is therefore $\Lambda^{(2)} = p \cdot \left(\lceil K/(\Lambda^{(1)} \cdot p) \rceil - 1 \right) < K/\Lambda^{(1)}$, since for the next-greater multiple of p we find that $p \cdot \lceil K/(\Lambda^{(1)} \cdot p) \rceil \geq K/\Lambda^{(1)}$. Then we obtain the following simplified upper bound, which is as tight as that of Theorem 1:

Theorem 2 *The minimum distance of a parallel concatenated convolutional code with parameters (K, α, β, p) is upper bounded by*

$$\delta_{\min,\max} \leq 2 + \alpha \cdot \left(\min_{\Lambda^{(1)}} \left\{ \left\lceil \frac{K}{\Lambda^{(1)}} \right\rceil + \left\lceil \frac{K}{p \cdot (\lceil K/(\Lambda^{(1)} \cdot p) \rceil - 1)} \right\rceil \right\} - 2 \right) + 2\beta, \qquad (16)$$

where the minimization is over $\Lambda^{(1)} \in \{1, \ldots, \lceil K/p \rceil - 1\}$ with $\Lambda^{(1)} \bmod p = 0$. The bound is valid for all $K > p^2$.

In the above Theorem, we have to restrict $\Lambda^{(1)}$ to $\leq \lceil K/p \rceil - 1$ in order to assure that $\Lambda^{(2)} \geq p$. Moreover, we require $K > \Lambda^{(1)} \cdot \Lambda^{(2)} \geq p^2$, since $\Lambda^{(1)}, \Lambda^{(2)} \geq p$.

We can state another simplification of Ineq. (13) by slightly relaxing the tightness of this upper bound. Let us choose $(\Lambda^{(1)}, \Lambda^{(2)}) = (\Lambda_0, \Lambda_0)$ with $\Lambda_0 = p \cdot \left(\lceil \sqrt{K}/p \rceil - 1 \right)$. We can easily verify that this pair (Λ_0, Λ_0) belongs to $\mathcal{L}_2$, since both Λ_0 is a multiple of p and $\Lambda_0 \cdot \Lambda_0 < K$. The last statement is true, because we find that $\Lambda_0 < p \cdot (\sqrt{K}/p) = \sqrt{K}$ by employing Ineq. (15). From Ineq. (13), we obtain hence:

$$\delta_{\min,\max} \leq 2 + \alpha \cdot \left(\left\lceil \frac{K}{\Lambda_0} \right\rceil + \left\lceil \frac{K}{\Lambda_0} \right\rceil - 2 \right) + 2\beta. \qquad (17)$$

Furthermore, we recognize that $\Lambda_0 \geq p \cdot \left(\sqrt{K}/p - 1 \right) = \sqrt{K} - p$, from where we can set up the following chain of inequalities:

$$\left\lceil \frac{K}{\Lambda_0} \right\rceil \leq \left\lceil \frac{K}{\sqrt{K} - p} \right\rceil \qquad (18)$$

$$< \frac{K}{\sqrt{K} - p} + 1. \qquad (19)$$

Using these inequalities, we can finally state the following upper bound, which is less tight yet simpler than the bounds of the above Theorems 1 and 2.

Theorem 3 *The minimum distance of a parallel concatenated convolutional code with parameters (K, α, β, p) is upper-bounded by*

$$\delta_{\min,\max} < 2 + 2\alpha \frac{K}{\sqrt{K} - p} + 2\beta. \qquad (20)$$

Asymptotically, this upper bound grows like $2\alpha \cdot \sqrt{K}$, which is a factor $\sqrt{2}$ larger than the asymptotic function $\alpha \cdot \sqrt{2K}$ of the geometric bound of [17]. A comparison of the above new upper bounds with the aforementioned existing upper bound of [17], which is based

on geometric considerations, will be given in Section IV. Up to now we have considered input words of weight 2 only. In the following subsection, we shall derive another upper bound on the minimum distance of Turbo codes by also taking into account input words of weight 4.

F. Upper bound for input words of weight 4

In this subsection, we consider primarily a situation as displayed in Fig. 3b with the following input words of weight 4: the first component input word $u(D) = D^{i_{1,1}} \oplus D^{i_{1,2}} \oplus D^{i_{2,1}} \oplus D^{i_{2,2}}$, where $i_{1,2} = i_{1,1} + \kappa_1 \cdot p$ and $i_{2,2} = i_{2,1} + \kappa_2 \cdot p,\ \kappa_1, \kappa_2 \in \mathbb{N}^+$ is permuted to the second component input word $u^{(2)}(D) = D^{j_{1,1}} \oplus D^{j_{1,2}} \oplus D^{j_{2,1}} \oplus D^{j_{2,2}}$, where $j_{1,2} = j_{1,1} + \kappa_3 \cdot p$ and $j_{2,2} = j_{2,1} + \kappa_4 \cdot p,\ \kappa_3, \kappa_4 \in \mathbb{N}^+$ (in the example of Fig. 3b, the parameters are $p = 3$ and $\kappa_1 = \kappa_2 = \kappa_3 = \kappa_4 = 1$). Both $\boldsymbol{u}$ and $\boldsymbol{u}^{(2)}$ are hence the (binary) sum of two weight-2 error words, respectively. Let us use the term *weight-4 error word* for such a sum of two weight-2 error words. In analogy to our proceeding for weight-2 error words in Section III-C (cf. Ineq. (7)), we can use Lemma 1 in order to upper bound the corresponding codeword weight in this case:

$$w(\boldsymbol{c}) \leq 4 + \alpha \cdot \frac{|i_{1,1} - i_{1,2}| + |i_{2,1} - i_{2,2}| + |j_{1,1} - j_{1,2}| + |j_{2,1} - j_{2,2}|}{p} + 4\beta. \tag{21}$$

We will show that taking into account these weight-4 error words provides a means to derive an upper bound on $\delta_{\min,\max}$, which is tighter for large K than both the geometric bound of [17] and the combinatorial bound of Section III-E.

We can again use our partition strategy of Section III-D here, where we partition the indices of the first component into $\Lambda^{(1)}$ subsets and those of the second component into $\Lambda^{(2)}$ subsets. We must accordingly impose the following condition *(1)* on the number of subsets: $\Lambda^{(1)} \bmod p = \Lambda^{(2)} \bmod p = 0$. As in the preceding subsection, we will need a second condition on $\Lambda^{(1)}$ and $\Lambda^{(2)}$ in our forthcoming proofs. In the following paragraphs, we will motivate and eventually deduce this condition *(2)*. Let us consider an *arbitrary* given interleaver $\boldsymbol{\pi}$ of length K. For this $\boldsymbol{\pi}$, two cases are possible, of which exactly one applies for *any* pair $(\Lambda^{(1)}, \Lambda^{(2)})$ satisfying condition *(1)*:

Case 1: There exists a subset $\mathcal{I}^{(1)}_{l_0}$ of the first component, of which at least two indices i_1, i_2 are mapped by the interleaver $\boldsymbol{\pi}$ to an identical associated subset $\mathcal{I}^{(2)}_{m_0}$ of the second component.

Then we have the same situation as in Theorem 1 that two "1"s in positions i_1 and i_2 of $\boldsymbol{u}$ generate a rather low weight in both the first and second component parity words $\boldsymbol{c}^{(1)}$ and $\boldsymbol{c}^{(2)}$, respectively. We can hence give the same upper bound as in Ineq. (12) for this input word $u(D) = D^{i_1} \oplus D^{i_2}$:

$$w(\mathrm{enc}_{\boldsymbol{\pi}}(\boldsymbol{u})) \leq 2 + \alpha \cdot \left(\left\lceil \frac{K}{\Lambda^{(1)}} \right\rceil + \left\lceil \frac{K}{\Lambda^{(2)}} \right\rceil - 2 \right) + 2\beta, \tag{22}$$

Observe that we do explicitly consider a weight-2 error word here, as at least one such word exists in this Case 1, and we do not need to fall back on weight-4 error words. Since Ineqs. (22) and (12) are identical, it might appear as if we are going to obtain exactly the same bound in this subsection as in the preceding subsection, where we exclusively considered input words of weight 2. We will however see later that the bound obtained in this subsection is tighter than that of Section III-E. The reason is that the valid pairs $(\Lambda^{(1)}, \Lambda^{(2)})$ used in Ineq. (22) can be chosen from a superset of the set $\mathcal{L}_2$, which is valid for Ineq. (12). This will be shown below.
End of Case 1.

Case 2: For *every* subset $\mathcal{I}_l^{(1)}, l \in \{0, \ldots, \Lambda^{(1)} - 1\}$ of the first component, *all* indices of the subset are mapped by the interleaver $\boldsymbol{\pi}$ to $|\mathcal{I}_l^{(1)}|$ *distinct* subsets $\mathcal{I}_m^{(2)}$ of the second component.

In Case 2, we focus on weight-4 error words $u(D) = D^{i_{1,1}} \oplus D^{i_{1,2}} \oplus D^{i_{2,1}} \oplus D^{i_{2,2}}$ and $u^{(2)}(D) = D^{j_{1,1}} \oplus D^{j_{1,2}} \oplus D^{j_{2,1}} \oplus D^{j_{2,2}}$, which each contain *two* pairs of indices. In each of these pairs $(i_{1,1}, i_{1,2})$ and $(i_{2,1}, i_{2,2})$, or $(j_{1,1}, j_{1,2})$ and $(j_{2,1}, j_{2,2})$, both indices belong to an identical subset of the first component, or of the second component, respectively. We assume without loss of generality that the interleaver performs the mappings $j_{1,1} = \pi_{i_{1,1}}$, $j_{1,2} = \pi_{i_{2,1}}$, $j_{2,1} = \pi_{i_{1,2}}$ and $j_{2,2} = \pi_{i_{2,2}}$. Such a situation is depicted in Fig. 4, which resembles Fig. 3b. In Fig. 4, the variables $\mathcal{I}_{l_1}^{(1)}$, $\mathcal{I}_{l_2}^{(1)}$, $\mathcal{I}_{m_1}^{(2)}$ and $\mathcal{I}_{m_2}^{(2)}$ represent the subsets containing the indices of the four pairs, and the filled elements (shaded or black) of $\boldsymbol{u}$ and $\boldsymbol{u}^{(2)}$ represent the indices belonging to these subsets (in the example of this figure, the period is $p = 3$). The black elements represent the indices belonging to the four pairs, i.e. the "1" elements of the input words $\boldsymbol{u}$ and $\boldsymbol{u}^{(2)}$ given above — the remaining indices (the "0" elements) of the four considered subsets are marked by shaded elements. As in Section III-D, we can immediately state the following upper bounds on the index differences within the two component input words:

$$|i_{1,1} - i_{1,2}|, |i_{2,1} - i_{2,2}| \leq p \cdot \left(\left\lceil \frac{K}{\Lambda^{(1)}} \right\rceil - 1 \right) \qquad \text{and} \tag{23}$$

$$|j_{1,1} - j_{1,2}|, |j_{2,1} - j_{2,2}| \leq p \cdot \left(\left\lceil \frac{K}{\Lambda^{(2)}} \right\rceil - 1 \right). \tag{24}$$

We can therefore reformulate the upper bound on the associated codeword weight using Ineq. (21) as follows:

$$w(\mathrm{enc}_{\boldsymbol{\pi}}(\boldsymbol{u})) \leq 4 + \alpha \cdot \left(2 \left\lceil \frac{K}{\Lambda^{(1)}} \right\rceil - 2 + 2 \left\lceil \frac{K}{\Lambda^{(2)}} \right\rceil - 2 \right) + 4\beta. \tag{25}$$

Our goal here is to find a condition *(2)* on the numbers $\Lambda^{(1)}$ and $\Lambda^{(2)}$, for which we can show that at least one $\boldsymbol{u}$ is permuted to a $\boldsymbol{u}^{(2)}$ as exhibited above for *every* interleaver $\boldsymbol{\pi}$. This condition *(2)* is found by the following argumentation.

Let $\mathcal{I}^{(2)}(j)$ denote the subset containing an index j of the second component input word $\boldsymbol{u}^{(2)}$: $j \in \mathcal{I}^{(2)}(j)$. Any index i in the first component is mapped by the interleaver $\boldsymbol{\pi}$ to an associated index π_i in the second component, and there exists an associated subset $\mathcal{I}^{(2)}(\pi_i)$ of the second component.

Next, let us define for any subset $\mathcal{I}_l^{(1)}$ of the first component the corresponding set of index *subsets* of the second component, which the elements of $\mathcal{I}_l^{(1)}$ (i.e. *indices* of the first component) are associated with by the mapping $\mathcal{I}^{(2)}(\pi_i)$:

$$\mathcal{S}_l = \left\{ \mathcal{I}_m^{(2)} \quad | \quad \mathcal{I}_m^{(2)} = \mathcal{I}^{(2)}(\pi_i),\ i \in \mathcal{I}_l^{(1)} \right\}. \tag{26}$$

Thus, the elements of $\mathcal{S}_l$ are from the set $\left\{ \mathcal{I}_0^{(2)}, \ldots, \mathcal{I}_{\Lambda^{(2)}-1}^{(2)} \right\}$. For the present Case 2 and any given $l \in \{0, \ldots, \Lambda^{(1)} - 1\}$, the interleaver-dependent mapping $\mathcal{I}^{(2)}(\pi_i)$ represents a bijection from $\mathcal{I}_l^{(1)}$ to $\mathcal{S}_l$, since all indices $i \in \mathcal{I}_l^{(1)}$ are mapped to distinct index subsets of the second component, such that the cardinality is $|\mathcal{S}_l| = |\mathcal{I}_l^{(1)}|$.

Now let us construct *two-element* subsets $\mathcal{T}_l$ of $\mathcal{S}_l$, i.e. $\mathcal{T}_l = \left\{ \mathcal{I}_{m_1}^{(2)}, \mathcal{I}_{m_2}^{(2)} \right\} \subseteq \mathcal{S}_l \quad \wedge \quad |\mathcal{T}_l| = 2$ with $l \in \{0, \ldots, \Lambda^{(1)} - 1\}$. These subsets will be used as the pigeons, when we soon utilize the pigeonhole principle in our argumentation again. We need to count, how many (not necessarily *distinct*) such sets $\mathcal{T}_l$ can be constructed for $l = 0, \ldots, \Lambda^{(1)} - 1$. We find that for every subset $\mathcal{S}_l,\ l \in \{0, \ldots, \Lambda^{(1)} - 1\}$, there exist $\phi_l = \binom{|\mathcal{S}_l|}{2}$ *distinct* subsets $\mathcal{T}_{l,j},\ j = 0, \ldots, \phi_l - 1$. As exhibited in Section III-D, there are $\Psi^{(1)} = K \bmod \Lambda^{(1)}$ subsets $\mathcal{I}_l^{(1)},\ l = 0, \ldots, \Psi^{(1)} - 1$ with $|\mathcal{I}_l^{(1)}| = \Omega^{(1)} + 1$ elements and $\Lambda^{(1)} - \Psi^{(1)}$ subsets $\mathcal{I}_l^{(1)},\ l = \Psi^{(1)}, \ldots, \Lambda^{(1)} - 1$ with $|\mathcal{I}_l^{(1)}| = \Omega^{(1)}$ elements, where $\Omega^{(1)} = \lfloor K/\Lambda^{(1)} \rfloor$. Since $|\mathcal{S}_l| = |\mathcal{I}_l^{(1)}|$, we have $\phi_l = \binom{\Omega^{(1)}+1}{2}$ for $l = 0, \ldots, \Psi^{(1)} - 1$ and $\phi_l = \binom{\Omega^{(1)}}{2}$ for $l = \Psi^{(1)}, \ldots, \Lambda^{(1)} - 1$. There is hence a total of

$$\Psi^{(1)} \cdot \binom{\Omega^{(1)} + 1}{2} + (\Lambda^{(1)} - \Psi^{(1)}) \cdot \binom{\Omega^{(1)}}{2} = \Psi^{(1)} \cdot \Omega^{(1)} + \Lambda^{(1)} \cdot \binom{\Omega^{(1)}}{2} \tag{27}$$

two-element sets $\mathcal{T}_{l,j}$, that can be constructed from the sets $\mathcal{S}_l,\ l = 0, \ldots, \Lambda^{(1)} - 1$.

On the other hand, the two distinct elements of any set $\mathcal{T}_{l,j}$ stem from the set $\left\{ \mathcal{I}_0^{(2)}, \ldots, \mathcal{I}_{\Lambda^{(2)}-1}^{(2)} \right\}$ of index subsets of the second component. Since $\mathcal{T}_{l,j} \subseteq \left\{ \mathcal{I}_0^{(2)}, \ldots, \mathcal{I}_{\Lambda^{(2)}-1}^{(2)} \right\}$ and $|\mathcal{T}_{l,j}| = 2$, only at most $\binom{\Lambda^{(2)}}{2}$ *distinct* sets $\mathcal{T}_{l,j}$ are possible. Now we can make use of the pigeonhole principle: If the following condition on $\Lambda^{(1)}$ and $\Lambda^{(2)}$ is satisfied

$$\Lambda^{(1)} \binom{\Omega^{(1)}}{2} + \Omega^{(1)} \cdot \Psi^{(1)} > \binom{\Lambda^{(2)}}{2}, \tag{28}$$

then we can be certain that among all the sets $\mathcal{T}_{l,j}$, which can be constructed for $l \in \{0, \ldots, \Lambda^{(1)} - 1\}$, there exist two identical sets $\mathcal{T}_{l_1,j_1} = \mathcal{T}_{l_2,j_2} = \left\{ \mathcal{I}_{m_1}^{(2)}, \mathcal{I}_{m_2}^{(2)} \right\}$ with $\mathcal{I}_{m_1}^{(2)} \neq \mathcal{I}_{m_2}^{(2)}$. Moreover, we know that $l_1 \neq l_2$ for these identical sets $\mathcal{T}_{l_1,j_1} = \mathcal{T}_{l_2,j_2}$, since the ϕ_{l_1} sets $\mathcal{T}_{l_1,j},\ j \in \{0, \ldots, \phi_{l_1} - 1\}$ are *distinct* for any $l_1 \in \{0, \ldots, \Lambda^{(1)} - 1\}$ (see above). Thus,

we find that $\left\{\mathcal{I}_{m_1}^{(2)}, \mathcal{I}_{m_2}^{(2)}\right\} \subseteq \mathcal{S}_{l_1}$ and $\left\{\mathcal{I}_{m_1}^{(2)}, \mathcal{I}_{m_2}^{(2)}\right\} \subseteq \mathcal{S}_{l_2}$, and that there exist two indices $i_{1,1}, i_{1,2} \in \mathcal{I}_{l_1}^{(1)}$ and two indices $i_{2,1}, i_{2,2} \in \mathcal{I}_{l_2}^{(1)}$ with $\pi_{i_{1,1}}, \pi_{i_{2,1}} \in \mathcal{I}_{m_1}^{(2)}$ and $\pi_{i_{1,2}}, \pi_{i_{2,2}} \in \mathcal{I}_{m_2}^{(2)}$. This is exactly the situation, that was described at the start of Case 2 and is displayed in Fig. 4.

Therefore, Ineq. (28) represents the condition *(2)* on $\Lambda^{(1)}$ and $\Lambda^{(2)}$, which we have been looking for. An important fact is that this result is valid independently of the particular interleaver $\boldsymbol{\pi}$ considered. Let us define a set $\mathcal{L}_4$ of pairs $(\Lambda^{(1)}, \Lambda^{(2)})$ through the two conditions *(1)* $\Lambda^{(1)} \bmod p = \Lambda^{(2)} \bmod p = 0$ (as already given above) and *(2)* the above Ineq. (28). If we consider only pairs $(\Lambda^{(1)}, \Lambda^{(2)})$ from this set $\mathcal{L}_4$, then for any interleaver, there exists at least one weight-4 error word $\boldsymbol{u}$ generating a (low) codeword weight, which can be upper-bounded by Ineq. (25)
End of Case 2.

For each of the two possible cases discussed above, we have obtained an upper bound on the minimum codeword weight, which is independent of the considered interleaver $\boldsymbol{\pi}$. We can use these results to state a new Theorem as follows:

Theorem 4 *The minimum distance of a parallel concatenated convolutional code with parameters (K, α, β, p) is upper-bounded by*

$$\delta_{\min,\max} \leq 4 + 2\alpha \cdot \left(\min_{(\Lambda^{(1)},\Lambda^{(2)}) \in \mathcal{L}_4} \left\{ \left\lceil \frac{K}{\Lambda^{(1)}} \right\rceil + \left\lceil \frac{K}{\Lambda^{(2)}} \right\rceil \right\} - 2 \right) + 4\beta, \tag{29}$$

where the minimization is performed over the set $\mathcal{L}_4$ of values $\{(\Lambda^{(1)}, \Lambda^{(2)})\} \subseteq \{1, \ldots, K/2\}^2$, for which both of the following conditions are true
(1) $\Lambda^{(1)} \bmod p = \Lambda^{(2)} \bmod p = 0$
(2) $\Lambda^{(1)} \binom{\Omega^{(1)}}{2} + \Omega^{(1)} \cdot \Psi^{(1)} > \binom{\Lambda^{(2)}}{2}$,
where $\Omega^{(1)} = \lfloor K/\Lambda^{(1)} \rfloor$ and $\Psi^{(1)} = K \bmod \Lambda^{(1)}$.

Proof: Ineq. (25) is valid for Case 1, too, since this upper bound is always above that of Ineq. (22). Thus, for every $(\Lambda^{(1)}, \Lambda^{(2)}) \in \mathcal{L}_4$ and any interleaver $\boldsymbol{\pi}$, the upper bound of Ineq. (25) applies. Taking into account that Ineq. (25) is independent of the particular interleaver $\boldsymbol{\pi}_0$, we obtain the following upper bound by applying Ineq. (3):

$$\delta_{\min,\max} \leq 4 + 2\alpha \cdot \left(\left\lceil \frac{K}{\Lambda^{(1)}} \right\rceil + \left\lceil \frac{K}{\Lambda^{(2)}} \right\rceil - 2 \right) + 4\beta. \tag{30}$$

Theorem 4 follows immediately, when trying to minimize this upper bound over all pairs $(\Lambda^{(1)}, \Lambda^{(2)})$, for which the above derivation is valid. ■

Although the above Case 1 might seem superfluous for the derivation of Theorem 4 (since the associated upper bound of Ineq. (22) is itself upper-bounded by Ineq. (25) of

Case 2), it is actually needed in the proof. Consider as an example the *identity interleaver* $\pi_i = i,\ i = 0, \ldots, K-1$. Obviously, if both indices of a pair (i_1, i_2) belong to an identical subset in the first component and if the partitioning is identical in both components, i.e. $\Lambda^{(2)} = \Lambda^{(1)}$, then the associated two indices π_{i_1} and π_{i_2} also belong to an identical subset in the second component. A situation as in Fig. 4, where there exist cross-connections from each of the two subsets in the first component to *both* subsets in the second component, does not arise for the identity interleaver, and hence the valid case is Case 1.

Comparing the bounds of Theorems 1 and 4, we find that for identical $(\Lambda^{(1)}, \Lambda^{(2)})$ pairs, the latter upper bound is a factor 2 as large as the first upper bound. The bound of Theorem 4 can nevertheless be tighter for given parameters (K, α, β, p) than the bound of Theorem 1 as the underlying sets of $(\Lambda^{(1)}, \Lambda^{(2)})$ pairs are not identical: $\mathcal{L}_2 \neq \mathcal{L}_4$. In fact, it can be shown [22] that $\mathcal{L}_2 \subseteq \mathcal{L}_4$ for $K \geq 9$ and $p \geq 3$. The bound of Theorem 4 can therefore actually be tighter than that of Theorem 1, if the optimum $(\Lambda^{(1)}, \Lambda^{(2)})$ pair is in the subset $\mathcal{L}_4 \setminus \mathcal{L}_2$.

As in Theorem 1, we can simplify Theorem 4 by turning the two-dimensional optimization problem of Ineq. (29) (minimization over $(\Lambda^{(1)}, \Lambda^{(2)})$-pairs) into a minimization with only one degree of freedom, since Ineq. (29) can be rewritten:

$$\delta_{\min,\max} \leq 4 + 2\alpha \cdot \left(\min_{\Lambda^{(1)}} \left\{ \left\lceil \frac{K}{\Lambda^{(1)}} \right\rceil + \min_{\substack{\Lambda^{(2)} \\ (\Lambda^{(1)}, \Lambda^{(2)}) \in \mathcal{L}_4}} \left\{ \left\lceil \frac{K}{\Lambda^{(2)}} \right\rceil \right\} \right\} - 2 \right) + 4\beta. \tag{31}$$

The minimization is performed over those $\Lambda^{(1)} \in \{1, \ldots, K/2\}$, which are multiples of p, i.e. $\Lambda^{(1)} \bmod p = 0$, and for which there exist values $\Lambda^{(2)}$, such that $(\Lambda^{(1)}, \Lambda^{(2)}) \in \mathcal{L}_4$. We can rewrite the condition *(2)* of Theorem 4 as follows

$$(\Lambda^{(2)})^2 - \Lambda^{(2)} - 2\left(\Lambda^{(1)} \binom{\Omega^{(1)}}{2} + \Omega^{(1)} \cdot \Psi^{(1)} \right) < 0, \tag{32}$$

and solving this inequality for $\Lambda^{(2)}$ for a given $\Lambda^{(1)}$ value gives the condition that $\Lambda_1 < \Lambda^{(2)} < \Lambda_2$, where

$$\Lambda_1, \Lambda_2 = \frac{1}{2} \mp \sqrt{\frac{1}{4} + 2\left(\Lambda^{(1)} \binom{\Omega^{(1)}}{2} + \Omega^{(1)} \cdot \Psi^{(1)} \right)}. \tag{33}$$

The optimum $\Lambda^{(2)}$ is the largest valid $\Lambda^{(2)}$, since it minimizes $\lceil K/\Lambda^{(2)} \rceil$ in Ineq. (31) for a given $\Lambda^{(1)}$. The optimum value is hence $\Lambda^{(2)} = p \cdot (\lceil \Lambda_2/p \rceil - 1)$, since for this multiple of p, we have $\Lambda^{(2)} < \Lambda_2$ and the next-greater multiple of p is $\Lambda^{(2)} + p \geq \Lambda_2$ (cf. Ineq. (15)). Thus, an upper bound, which contains only one minimization variable and which nevertheless is equivalent to that of Theorem 4, can be stated as follows:

Theorem 5 *The minimum distance of a parallel concatenated convolutional code with parameters* (K, α, β, p) *is upper-bounded by*

$$\delta_{\min,\max} \leq 4+2\alpha\cdot\left(\min_{\Lambda^{(1)}}\left\{\left\lceil\frac{K}{\Lambda^{(1)}}\right\rceil+\left\lfloor\frac{K}{p\cdot\left(\left\lceil\frac{\frac{1}{2}+\sqrt{\frac{1}{4}+2\left(\Lambda^{(1)}\binom{\Omega^{(1)}}{2}+\Omega^{(1)}\cdot\Psi^{(1)}\right)}}{p}\right\rceil-1\right)}\right\rfloor\right\}-2\right)+4\beta, \tag{34}$$

where the minimization is over those $\Lambda^{(1)} \in \{1,\ldots,K/2\}$*, which are multiples of* p*, and for which the quantity* $\Lambda_2 \stackrel{\triangle}{=} 1/2+\sqrt{1/4+2\left(\Lambda^{(1)}\binom{\Omega^{(1)}}{2}+\Omega^{(1)}\cdot\Psi^{(1)}\right)}$ *lies in the interval* $(p; K/2]$*. It can be shown [22] that there exists at least one valid value* $\Lambda^{(1)}$*, which satisfies* $\Lambda_2 > p$*, if* $K \geq K_0 \stackrel{\triangle}{=} p\cdot(\lfloor 1/2\cdot(1+\sqrt{4p-3})\rfloor+1)$*.*

The condition $\Lambda_2 \in (p; K/2]$ ensures that there actually exist $\Lambda^{(2)}$ values, such that $(\Lambda^{(1)}, \Lambda^{(2)}) \in \mathcal{L}_4$ for the given $\Lambda^{(1)}$. This is hence a necessary and sufficient condition for using a value $\Lambda^{(1)} \in \{p, 2p, 3p, \ldots\}$ in the minimization of Ineq. (34).

Another simplified upper bound can be derived from Theorem 4 as Theorem 3 was derived from Theorem 1 in Section III-E. Once again, we relax the tightness of the bound by first fixing $\Lambda^{(1)} = \Lambda^{(2)} = \Lambda_0$, then using the lower and upper bounds of Ineq. (15) for the ceiling function, and finally finding the optimum value Λ_0. Doing this, we can derive the following Theorem:

Theorem 6 *The minimum distance of a parallel concatenated convolutional code with parameters* (K, α, β, p) *with* $K \geq \left(1/2+\sqrt{p-3/4}\right)^3+1$ *is upper-bounded by*

$$\begin{aligned}
\delta_{\min,\max} &\leq 4+4\alpha\cdot\left(\left\lceil\frac{K}{p\cdot\lfloor((K-1)^{2/3}-(K-1)^{1/3}+1)/p\rfloor}\right\rceil-1\right)+4\beta & (35)\\
&\leq 4+4\alpha\cdot\left(\left\lceil\frac{K}{(K-1)^{2/3}-(K-1)^{1/3}+1-p}\right\rceil-1\right)+4\beta & (36)\\
&< 4+4\alpha\cdot\frac{K}{(K-1)^{2/3}-(K-1)^{1/3}+1-p}+4\beta. & (37)
\end{aligned}$$

Proof: See Appendix.

Having developed upper bounds on $\delta_{\min,\max}$ based on two different combinatorial approaches in this section, we will discuss the results obtained here in the next section.

IV. Discussion of the upper bounds

In this section, we will compare the upper bounds that were derived in this paper with existing bounds on the minimum Hamming distance. We will examine, which bound is

tightest for different values of the interleaver length K. Moreover, we will discuss the implications of the bounds derived here on the distance spectrum of Turbo codes.

Figs. 5 and 6 illustrate the three different upper bounds $\delta_{\min,\mathrm{UB}}$ on the minimum distance $\delta_{\min,\max}$ of Turbo codes of overall rate $R = 1/3$ (i.e. no puncturing in the encoder) using component scramblers of memory $\nu = 2$ and $\nu = 4$ respectively. The (α, β, p)-parameters of the considered component scramblers are given in Table I. The bounds shown in the figures are the upper bound of [17] based on geometric considerations (labeled by "Geom" in the figures), the upper bound of Theorems 1 and 2 ("Combi2"), which is based on combinatorics and considers only Turbo encoder input words $\boldsymbol{u}$ of weight 2, and thirdly the upper bound of Theorems 4 and 5 ("Combi4"), which is based on combinatorial considerations on input words $\boldsymbol{u}$ of weight 2 and 4. The figures show the behaviour of the bounds for growing input word length (= interleaver length) K (note that the codeword length is three times as large). The rapid fluctuations of the bounds in the figures are due to the quantization effects of the discrete minimization carried out in the calculation of the upper bounds. For a comparison, let us also consider three well-known bounds on the minimum Hamming distance of general linear binary block codes of rate $R = 1/3$, which are also displayed in the figures. The Hamming upper bound [23] grows approximately as $1.04 \cdot K$, the Gilbert-Varshamov existence bound [23] ('GV-Bound' in the figures) grows approximately as $0.52 \cdot K$ and the (asymptotic) McEliece/Rodemich/Rumsey/Welch upper bound [23] is $\delta_{\min} \leq 0.78 \cdot K$.

We see in the figures that the upper bound of Theorems 1,2 can be lower than that of Theorems 4,5 for very small K, whereas the bound of Theorems 4,5 becomes the lowest of all three already for rather small K and remains the lowest bound for growing K. In none of our examples, the bound of [17] is the tightest upper bound. Observe that the upper bound of Theorems 4,5 is valid already for $K \geq K_0 = p \cdot (\lfloor 1/2 \cdot (1 + \sqrt{4p-3}) \rfloor + 1)$ (see Section III-F), which is less than the minimum K required for the other two bounds. This is understandable, since $\mathcal{L}_2 \subseteq \mathcal{L}_4$ and hence there always exists a pair $(\Lambda^{(1)}, \Lambda^{(2)}) \in \mathcal{L}_4$ (i.e. we can calculate the bound of Theorems 4,5), if there exists a pair $(\Lambda^{(1)}, \Lambda^{(2)}) \in \mathcal{L}_2$ (i.e. we can calculate the bound of Theorems 1,2). The bound of Theorems 1,2 can be calculated for $K > p^2$, whereas the bound of [17] can only be calculated for $K > 2p^2$.

In Fig. 7, we compare the upper bounds of Theorems 1,2 with the simplified upper bound of Theorem 3, and the upper bounds of Theorems 4,5 with the simplified upper bound of Theorem 6 in a logarithmic scale. Note that both simplified bounds were derived such that they are upper bounds on the corresponding original upper bounds. The comparison of Fig. 7 was performed for Turbo codes of rate $R = 1/3$ (i.e. without any puncturing) again, which employ the component scramblers of memory $\nu = 3$ given in Table I. We see that the simplified bounds are very loose for small lengths K, and become tighter for $K \to \infty$. The simplified bound of Theorem 3 seems to converge faster to the original

upper bound of Theorems 1,2 than the simplified bound of Theorem 6 to its corresponding original upper bound. This is comprehensible since we had to apply more simplifications and bounding techniques in the derivation of Theorem 6 than for Theorem 3. In [17] it was derived, that the minimum distance of a Turbo code can asymptotically (for growing K) be upper-bounded by $\alpha \cdot \sqrt{2K}$. The asymptotic upper bound implied by Theorem 3 is $2\alpha \cdot \sqrt{K}$, which is larger by a factor of $\sqrt{2}$ than the asymptotic bound of [17]. However, we see from Theorem 6 that the minimum distance of Turbo codes cannot asymptotically grow stronger than $4\alpha \cdot \sqrt[3]{K}$ with the input word length K. This is the reason why the upper bound of Theorem 6 is asymptotically always tighter than that of Theorem 3 for large K.

As in [17], we recognize that *all* Turbo codes are asymptotically bad, i.e. even if the best possible interleavers are employed. In contrast to bounds on the minimum distance of general linear binary block codes, i.e. the Hamming or McEliece et al. upper bounds and the Gilbert-Varshamov existence bound, where the minimum distance grows asymptotically linearly with the codeword length, we have shown here that in the case of Turbo codes, the minimum distance can grow at most like the third root of the codeword length. This means that our new bounds for the special case of Turbo codes rapidly diverge from the existing bounds for general linear binary block codes. Turbo codes are hence not exactly the "Shannon codes" as what they are sometimes refered to, although their *bit error rate* (BER) performance is very close to what we can expect from "Shannon codes". However, as regards to the *word error rate* (WER), simulations show that a Turbo code employing a long randomly chosen interleaver performs badly. We have shown here that even a Turbo code using the *best* possible interleaver suffers from a relatively low minimum Hamming distance, which limits its WER performance. Battail introduced the term *weakly random-like* [10] for this type of codes exhibiting a bad WER performance but a good performance as regards to the BER. In contrast, codes with distance spectrum approximating that of random codes are refered to as *strongly* random like, and they exhibit a high power-efficiency in terms of BER *and* WER. Battail identified Turbo codes with *random* interleavers as weakly random-like. Since the new upper bounds established in the present paper asymptotically diverge from the Gilbert-Varshamov bound, which gives the natural minimum distance for Shannon's random codes, Turbo codes cannot not be strongly random-like codes for *any* interleaver. At best, Turbo codes can be weakly random-like, unless a very bad interleaver is employed (if, e.g., the identity interleaver of Section III-F is used, the Turbo code degenerates to a conventional convolutional code).

Figs. 5, 6 and 7 indicate that already from relatively small input word lengths K, the minimum distance of a Turbo code is limited by input words $\boldsymbol{u}$ of weight 4 rather than those of weight 2. The fact that weight 4 input words are more important for bounding the minimum distance than weight 2 words indicates that weight 4 input words should always

be taken into account, when interleavers are to be designed by sophisticated algorithms. These input words of weight 4 generating Turbo codewords of low weight are the vector sum in GF(2) of two weight-2 error words (see Fig. 3b and Section III-F). This suggests that input words of weight 6, which are the sum of *three* weight-2 error words, might produce codewords of even lower weight than weight-4 error words for larger K. Setting up an upper bound on the minimum distance of Turbo codes considering input words of weight 6, 8, 10 etc. remains a topic for future examination, but the authors conjecture that the upper bounds on the minimum distance of Turbo codes derived in this paper can be considerably lowered for large K. What will be the resulting lower limiting curve on the ensemble of these upper bounds?

It might appear strange that all upper bounds derived in this paper do hardly depend immediately on the scrambler period p. For example, the magnitude of the optimum quantities $\Lambda^{(1)}, \Lambda^{(2)}$ in Theorem 1 is mainly determined by condition *(2)*, where p does not even appear. By contrast, in [17], a direct dependence on the period p becomes obvious in the upper bounds derived. The difference is that for the derivations in [17] a partitioning into the p equivalence classes with respect to the $\bmod p$ operation was performed. The proceeding in the present paper is quite different, since we partition the indices into Λ subsets, which are subsets of the p equivalence classes themselves. So the only dependence of the bounds on p is the restriction of the number Λ to multiples of p. The influence of this restriction is negligible for large K. However, there is an indirect dependence of the new upper bounds on p, which comes from the fact that the differences between the indices belonging to the same index subset are multiples of p. This fact expresses itself in the presence of the factor α (scrambler output weight per period) in all bounds, which as a rule grows with the period length p. From this argumentation and from the figures, it becomes obvious that the upper bounds on the maximum attainable minimum Hamming distance $\delta_{\min,\max}$ becomes the higher, the larger the value α and hence the scrambler period p is. The bounds reflect therefore the fact that component scramblers of larger memory and with a primitive feedback-polynomial (resulting in the maximum possible period) should be chosen in order to increase the minium distance and lower the error-floor, cf. e.g. [12].

V. Conclusion

In this paper, new upper bounds on the minimum Hamming distance of Turbo codes with arbitrary interleavers are presented. These bounds are based on combinatorial considerations; firstly, bounds are derived by exclusively taking into consideration Turbo encoder input words of weight 2, secondly, further bounds are developed, which take into account input words of weights 2 as well as 4. The new bounds seem to be tighter than the existing bound of [17], particularly for medium and larger K. One of the new bounds proves that the minimum distance of Turbo codes can asymptotically grow only less than the

third root of the codeword length. Rigorous proofs for all bounds are given, examples of the derived bounds for given Turbo codes are displayed, and the implications of the new bounds are discussed.

Acknowledgement

The authors are grateful to the two anonymous reviewers for their careful reading of the paper and their large efforts, which helped to improve the presentation of the material significantly.

Appendix

Proof of Theorem 6

The upper bound of Theorem 4 is based on the minimization

$$\min_{(\Lambda^{(1)},\Lambda^{(2)})\in\mathcal{L}_4} \left\{ \left\lceil \frac{K}{\Lambda^{(1)}} \right\rceil + \left\lceil \frac{K}{\Lambda^{(2)}} \right\rceil \right\}. \tag{38}$$

In order to derive a simplified upper bound, we restrict ourselves to pairs $(\Lambda^{(1)}, \Lambda^{(2)}) = (\Lambda_0, \Lambda_0)$. We then have to find values Λ_0, such that (a) the sum

$$\left\lceil \frac{K}{\Lambda^{(1)}} \right\rceil + \left\lceil \frac{K}{\Lambda^{(2)}} \right\rceil = 2 \left\lceil \frac{K}{\Lambda_0} \right\rceil \tag{39}$$

becomes close to the above minimum, which implies that Λ_0 is as large as possible, and (b) that this choice of Λ_0 is valid, i.e. $(\Lambda_0, \Lambda_0) \in \mathcal{L}_4$.

For the forthcoming derivations, we will need the following inequalities for the floor function:

$$\begin{array}{rccccc} & \lfloor x \rfloor & \leq & x & < & \lfloor x \rfloor + 1 \\ \Leftrightarrow & x - 1 & < & \lfloor x \rfloor & \leq & x \end{array} \qquad \forall x \in \mathbb{R}. \tag{40}$$

Let us first focus on the condition *(2)* on $\mathcal{L}_4$ in Theorem 4. The following inequality is true for the left-hand side of this condition:

$$\Lambda_0 \binom{\Omega_0}{2} + \Omega_0 \cdot \Psi_0 > \Lambda_0 \cdot \frac{1}{2} \left(\frac{K}{\Lambda_0} - 1 \right) \cdot \left(\frac{K}{\Lambda_0} - 2 \right), \tag{41}$$

since $\Omega_0 = \lfloor K/\Lambda_0 \rfloor > K/\Lambda_0 - 1$ (cf. Ineq. (40)), and since $\Omega_0 \cdot \Psi_0 \geq 0$, where $\Psi_0 = K \bmod \Lambda_0$. If we find a value $\Lambda_0 \in \mathbb{N}$, which satisfies

$$\Lambda_0 \cdot \frac{1}{2} \left(\frac{K}{\Lambda_0} - 1 \right) \cdot \left(\frac{K}{\Lambda_0} - 2 \right) \geq \binom{\Lambda_0}{2}, \tag{42}$$

then we know that the pair $(\Lambda^{(1)}, \Lambda^{(2)}) = (\Lambda_0, \Lambda_0)$ satisfies condition *(2)* of Theorem 4. Rewriting Ineq. (42) provides the following sufficient condition on Λ_0:

$$\Lambda_0^3 - 3\Lambda_0^2 + 3\Lambda_0 K - K^2 \leq 0. \tag{43}$$

To find the set of values Λ_0, for which the inequality holds, we have to determine the roots of this cubic polynomial of Λ_0. Following the guidelines of Cardano's formula (cf. [24, page 120]), we perform the substitution $\Lambda_0 = y + 1$ and obtain the reduced form

$$y^3 + 3(K-1)y - (K-1)\cdot(K-2) \leq 0. \tag{44}$$

The discriminant of a cubic is $\Delta^2 = K^2/4(K-1)^2$, which is > 0 for $K > 1$, i.e. all relevant values K. There is hence a unique real root y_0 of the cubic, which can be computed as

$$y_0 = (K-1)^{2/3} - (K-1)^{1/3}. \tag{45}$$

Thus, the condition of Ineq. (42) is satisfied, if and only if

$$\Lambda_0 \leq y_0 + 1 = (K-1)^{2/3} - (K-1)^{1/3} + 1. \tag{46}$$

This last condition is hence sufficient, so that the condition *(2)* of Theorem 4 is satisfied for the pair $(\Lambda^{(1)}, \Lambda^{(2)}) = (\Lambda_0, \Lambda_0)$.

Let us choose

$$\begin{aligned} \Lambda_0 &= p \cdot \left\lfloor \frac{y_0 + 1}{p} \right\rfloor & (47)\\ &= p \cdot \left\lfloor \frac{(K-1)^{2/3} - (K-1)^{1/3} + 1}{p} \right\rfloor. & (48) \end{aligned}$$

We can easily verify using Ineq. (40) that $\Lambda_0 \leq y_0 + 1$, and thus, condition *(2)* of Theorem 4 is satisfied. Moreover, Λ_0 is a multiple of p, and hence this Λ_0 value also satisfies condition *(1)* of Theorem 4.

We still have to find a condition on K such that $\Lambda_0 \in \{p, \ldots, K/2\}$. By simple arithmetic manipulations, we can show that $(K-1)^{2/3} - (K-1)^{1/3} + 1 \geq p$ if $K \geq \left(1/2 + \sqrt{p - 3/4}\right)^3 + 1$. Additionally, we find that $(K-1)^{2/3} - (K-1)^{1/3} + 1 < (K+1)/2$ for $K > 1$.

Since we have found that $(\Lambda_0, \Lambda_0) \in \mathcal{L}_4$, Theorem 4 immediately implies the following upper bound on the minimum distance (cf. Eq. (39)):

$$\delta_{\min,\max} \leq 4 + 4\alpha \cdot \left(\left\lceil \frac{K}{\Lambda_0} \right\rceil - 1 \right) + 4\beta, \tag{49}$$

where Λ_0 is defined in Eq. (48). From Ineq. (40) it follows that

$$\begin{aligned} \Lambda_0 &= p \cdot \left\lfloor ((K-1)^{2/3} - (K-1)^{1/3} + 1)/p \right\rfloor & (50)\\ &> (K-1)^{2/3} - (K-1)^{1/3} + 1 - p. & (51) \end{aligned}$$

Therefore, we can state an upper bound on the upper bound of Ineq. (49):

$$\delta_{\min,\max} \leq 4 + 4\alpha \cdot \left(\left\lceil \frac{K}{(K-1)^{2/3} - (K-1)^{1/3} + 1 - p} \right\rceil - 1 \right) + 4\beta. \tag{52}$$

Finally, we can remove the ceiling function by employing Ineq. (15):

$$\delta_{\min,\max} < 4 + 4\alpha \cdot \frac{K}{(K-1)^{2/3} - (K-1)^{1/3} + 1 - p} + 4\beta. \tag{53}$$

■

References

[1] Claude Berrou and Alain Glavieux, "Reflections on the Prize Paper: "Near optimum error-correcting coding and decoding: turbo codes"," *IEEE Inf. Th. Society Newsletter*, vol. 48, no. 2, pp. 1,24–31, 1998.

[2] Claude Berrou, Alain Glavieux, and Punya Thitimajshima, "Near Shannon Limit Error-Correcting Coding and Decoding: Turbo Codes," *Intern. Conf. on Comm.*, pp. 1064–1070, 1993.

[3] Thomas J. Richardson, M. Amin Shokrollahi, and Rüdiger L. Urbanke, "Design of Capacity-Approaching Irregular Low-Density Parity Check Codes," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 619–637, 2001.

[4] Stephan ten Brink, "A rate one–half code for approaching the Shannon limit by 0.1 dB," *IEE Electronics Letters*, vol. 36, no. 15, pp. 1293–1294, 2000.

[5] Patrick Robertson, "Illuminating the Structure of Code and Decoder of Parallel Concatenated Recursive Systematic (Turbo) Codes," *Global Conf. on Comm. (GlobeCom)*, pp. 1298–1303, 1994.

[6] S. Dolinar and D. Divsalar, "Weight Distributions for Turbo Codes Using Random and Nonrandom Permutations," *JPL-TDA Progress Report*, vol. 42-122, pp. 56–65, 1995.

[7] A. Khandani, "Design of turbo-code interleaver using Hungarian method," *IEE Electronics Letters*, vol. 34, no. 1, pp. 63–65, 1998.

[8] Fred Daneshgaran and Marina Mondin, "Design of Interleavers for Turbo Codes: Iterative Interleaver Growth Algorithms of Polynomial Complexity," *IEEE Transactions on Information Theory*, vol. 45, no. 6, pp. 1845–1859, 1999.

[9] Marco Breiling, Stein Peeters, and Johannes Huber, "Interleaver Design Using Backtracking and Spreading Methods," *Intern. Symp. on Inf. Th.*, p. 451, 2000.

[10] Gérard Battail, "On random-like codes," *Canadian Workshop on Inf. Th.*, 1995.

[11] Sergio Benedetto and Guido Montorsi, "Unveiling Turbo Codes: Some Results on Parallel Concatenated Coding Schemes," *IEEE Transactions on Information Theory*, vol. 42, no. 2, pp. 409–428, 1996.

[12] Lance Perez, Jan Seghers, and Daniel Costello, "A Distance Spectrum Interpretation of Turbo Codes," *IEEE Transactions on Information Theory*, vol. 42, no. 6, pp. 1698–1709, 1996.

[13] Atousa Mohammadi and Weihua Zhuang, "Variance of the Turbo-Code Performance Bound over the Interleavers," *Conf. on Vehicular Technology Comm. (VTC) (Houston)*, pp. 2368–2372, 1999.

[14] Yuri Svirid, "Weight Distributions and Bounds for Turbo-Codes," *Europ. Trans. on Telecomm.*, vol. 6, no. 5, pp. 543–555, 1995.

[15] Nabil Kahale and Rüdiger Urbanke, "On the Minimum Distance of Parallel and Serially Concatenated Codes," *available at* `lcavwww.epfl.ch/~ruediger/publications.html`, 1997.

[16] W. Blackert, E. Hall, and S. Wilson, "An Upper Bound on Turbo Code Free Distance," *Intern. Conf. on Comm.*, pp. 957–961, 1996.

[17] Marco Breiling and Johannes Huber, "Upper Bound on the Minimum Distance of Turbo Codes," *accepted for publication in the IEEE Transactions on Communications*, 1999.

[18] O. Jörssen and H. Meyr, "Terminating the trellis of turbo-codes," *IEE Electronics Letters*, vol. 30, no. 16, pp. 1285–1286, 1994.

[19] Solomon Golomb, *Shift Register Sequences*, Aegean Park Press, revised edition, 1982.

[20] Stefan Höst, *On Woven Convolutional Codes*, Ph.D. thesis, Universitetet i Lund, 1999.

[21] Richard Brualdi, *Introductory Combinatorics*, North-Holland, 1977.

[22] Marco Breiling, *Analysis and Design of Turbo Code Interleavers*, Ph.D. thesis, Universität Erlangen-Nürnberg [in preparation], 2001.

[23] Stephen Wicker, *Error control systems for digital communication and storage*, Prentice-Hall, 1st edition, 1995.

[24] I. Bronshtein and K. Semendyayev, *Handbook of Mathematics*, Springer Verlag, 3rd edition, 1997.

List of Figures

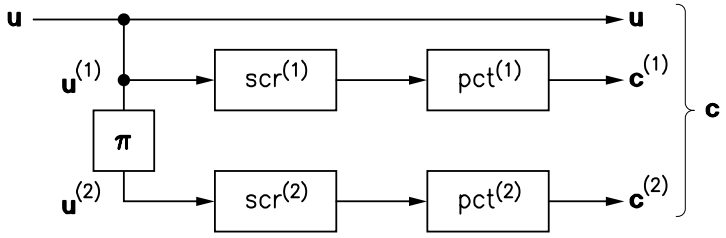

Fig. 1. System model of the Turbo encoder.

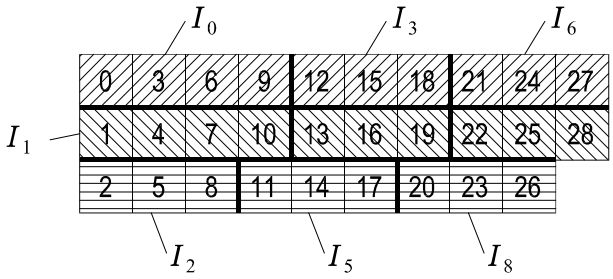

Fig. 2. Example for the array representation of the employed index partition strategy for interleaver length $K = 29$, scrambler period $p = 3$ and $\Lambda = 9$ index subsets.

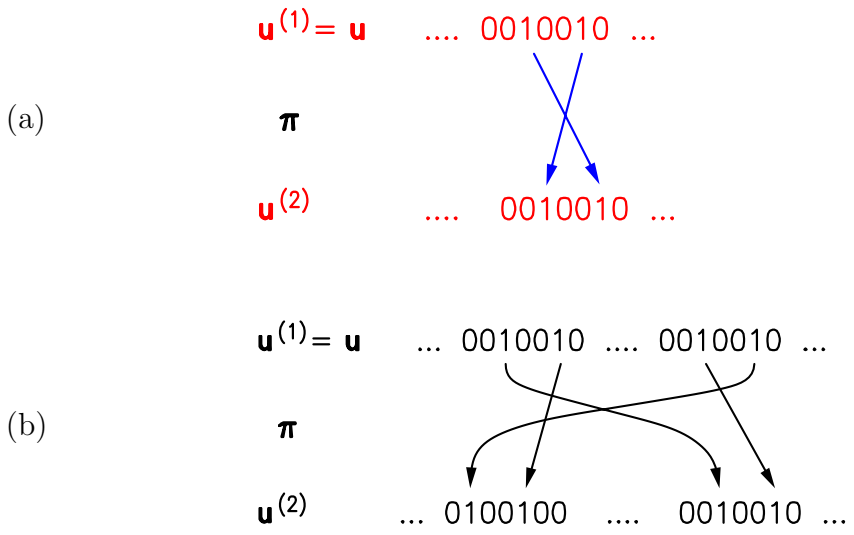

Fig. 3. (a) A weight-2 error word in $\boldsymbol{u}$ is permuted to a weight-2 error word in $\boldsymbol{u}^{(2)}$, (b) a weight-4 error word in $\boldsymbol{u}$ is permuted to a weight-4 error word in $\boldsymbol{u}^{(2)}$.

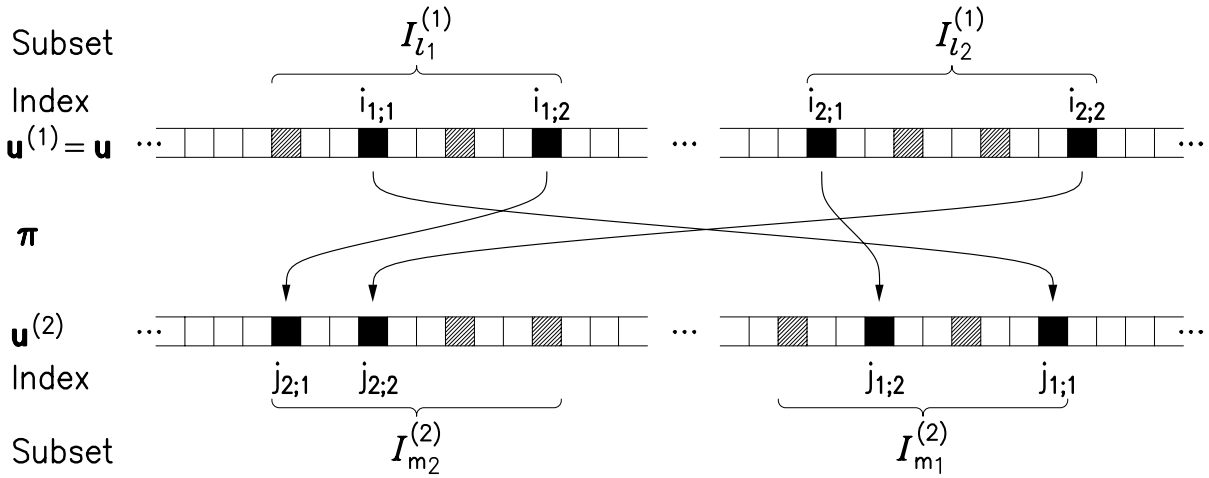

Fig. 4. A graphical representation of the situation considered in Case 2.

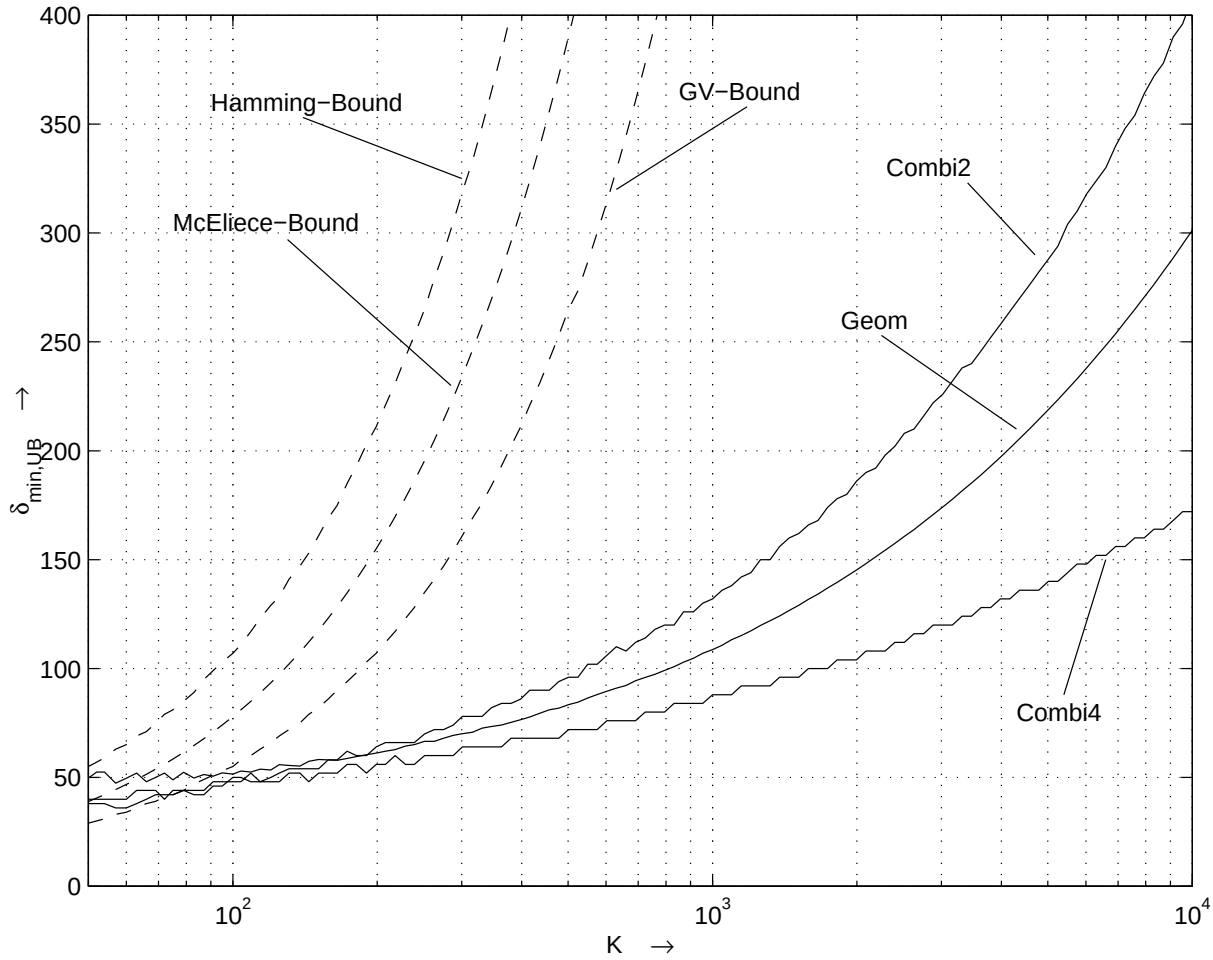

Fig. 5. Three upper bounds on the minimum Hamming distance $\delta_{\min,\max}$ for Turbo codes of rate $R = 1/3$ with component scramblers of memory $\nu = 2$: "Geom" geometric bound of [17]; "Combi2" combinatorial bound for weight 2 input words according to Theorems 1 and 2; "Combi4" combinatorial bound for weight 4 input words according to Theorems 4 and 5. Also shown are three bounds for general linear binary block codes.

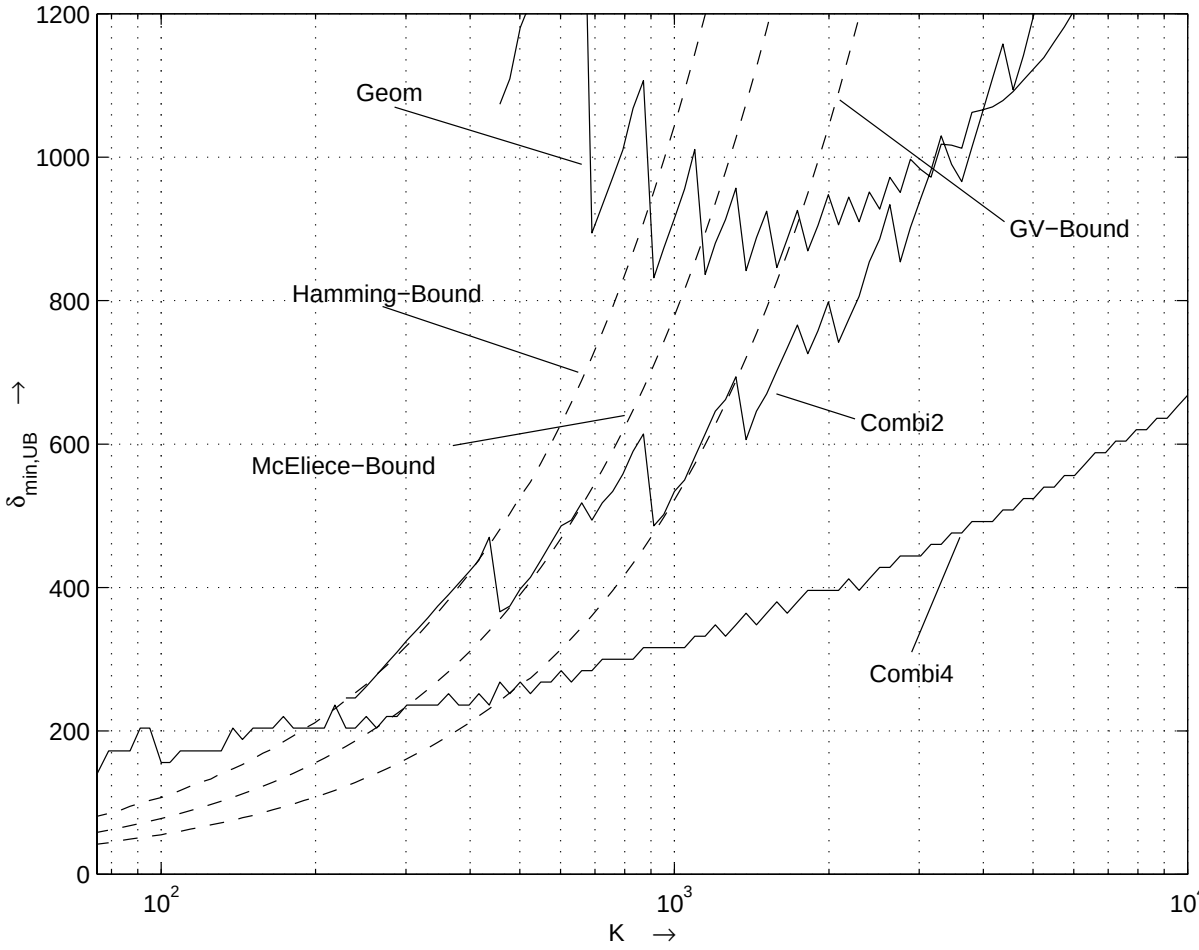

Fig. 6. Three upper bounds on the minimum Hamming distance $\delta_{\min,\max}$ for Turbo codes of rate $R = 1/3$ with component scramblers of memory $\nu = 4$: "Geom" geometric bound of [17]; "Combi2" combinatorial bound for weight 2 input words according to Theorems 1 and 2; "Combi4" combinatorial bound for weight 4 input words according to Theorems 4 and 5. Also shown are three bounds for general linear binary block codes.

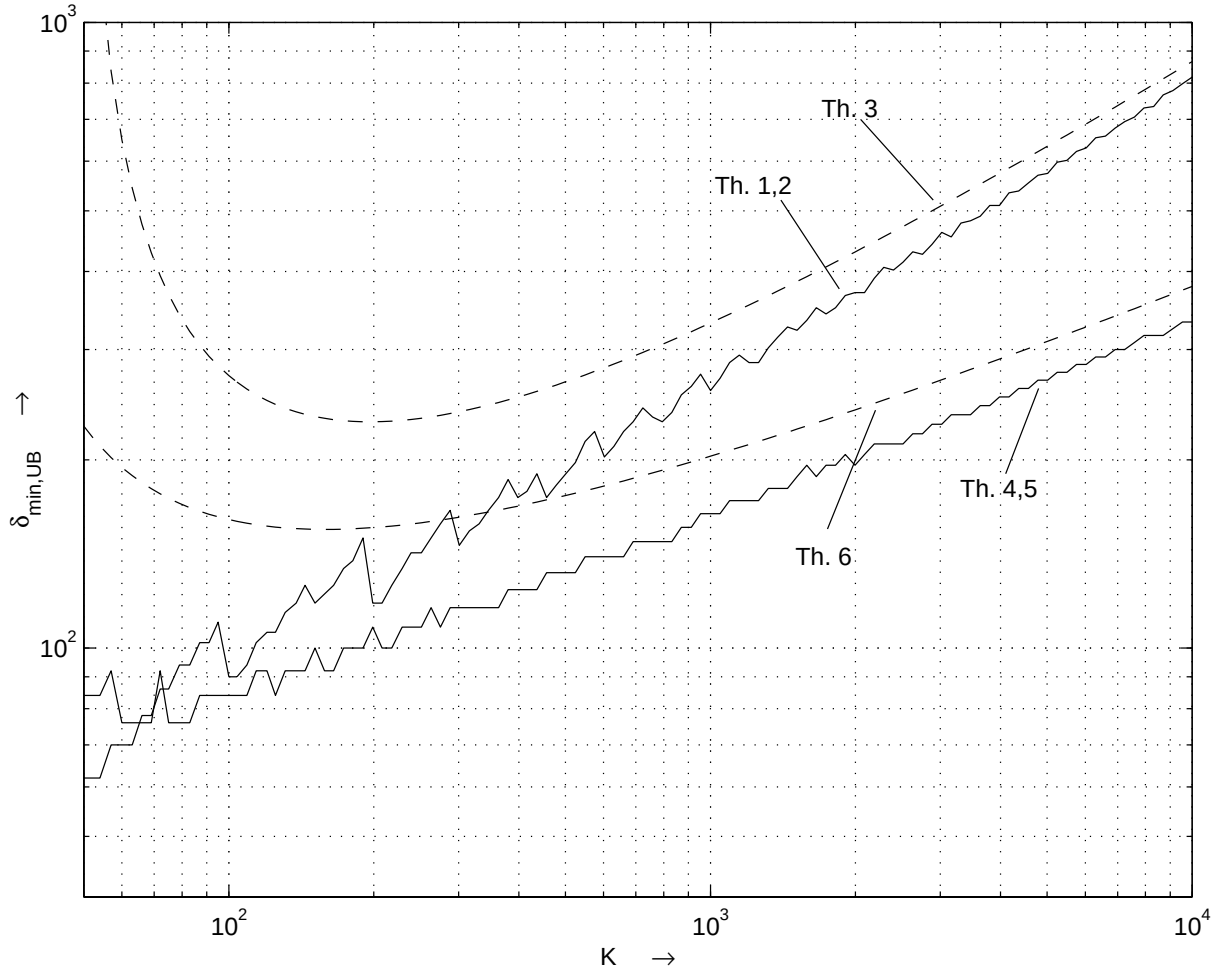

Fig. 7. Comparison of the derived upper bounds on the minimum Hamming distance of Theorems 1,2 and 4,5 with their simplified versions of Theorems 3 and 6 for Turbo codes of rate $R = 1/3$ employing memory $\nu = 3$ component scramblers.