Upper Bound on 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.: +49–9131–852–7668, Fax: +49–9131–852–8919
Email: breiling@lnt.de
http://www.lnt.de/∼breiling

Accepted for publication in the IEEE Transactions on Communications
(submitted on November 19, 1999)
Keywords: turbo code, minimum distance bounds, interleaver design

Abstract

An upper bound on the minimum distance of turbo codes is derived, which depends only on the interleaver length and the component scramblers employed. The derivation of this bound considers exclusively turbo encoder input words of weight 2. The bound does not only hold for a particular interleaver but for all possible interleavers including the best. It is shown that in contrast to general linear binary codes the minimum distance of turbo codes cannot grow stronger than the square root of the block length. This implies that turbo codes are asymptotically bad. A rigorous proof for the bound is provided, which is based on a geometric approach.

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

I. Introduction

Since their discovery by Berrou, Glavieux and Thitimajshima in 1993 [1], turbo codes (also termed *Parallel Concatenated Convolutional Codes*) have attracted much attention from coding theorists. Turbo codes are still the most power efficient binary channel codes known so far, but only if high and medium Bit Error Rates (BER) are required. The power efficiency of the code can in general be improved by increasing the codeword length, if the interleaver, forming a vital part of the encoder, is randomly chosen. For random interleavers, the BER curve has an *error floor*, i.e. the power efficiency decreases strongly if very low BERs are required. Several contributions, e.g. [2],[3], have shown that the reason for this behaviour can be found in the turbo code's distance spectrum. Averaged over all possible interleavers, a turbo code possesses codewords at relatively low Hamming weights even for large interleavers. In order to reduce the number of low Hamming weight codewords and also to increase the minimum distance of turbo codes (these are the two characteristic features of the distance spectrum determining the error floor), the interleaver has to be carefully designed. Several algorithms have been proposed to design turbo code interleavers, e.g.[4],[5],[6]. It is observed that these methods can improve the minimum distance $\delta_{\min}$ of turbo codes and that $\delta_{\min}$ grows with the interleaver length K, but the achievable $\delta_{\min}$ remains much lower than what can be expected from existing bounds on $\delta_{\min}$ for general linear binary codes. The paper [7] presents a method, how the minimum distance $\delta_{\min}$ of a turbo code can be upper bounded for any *given* interleaver. In contrast to that paper, in this contribution, we will provide an upper bound, which is valid for *all*, i.e. also for the *best* possible designed interleavers. The bound quantifies hence, which $\delta_{\min}$ values are unattainable for turbo codes, and it can be used to assess the performance of existing interleaver design algorithms. The bound depends only on the interleaver length K, the component scramblers, and the puncturing pattern employed, and it is valid for all values of K greater than a certain value which depends on the period length of the component scramblers. In several examples we see that for large values of K, our new bound is tighter than the tightest possible upper bound, which is valid for an arbitrary linear binary code.

The paper is organized as follows: In Section II we give a brief overview of our system model, the notation and the terminology used in the paper. Section III presents our upper bound derived and proves it rigorously. In Section IV, we discuss the implications of the bound. Finally, we conclude our discussion in Section V.

II. System model

Our model of the turbo encoder is displayed in Fig. 1. It consists of three parallel branches. As the upper branch shows, the turbo encoder input word $\boldsymbol{u} = (u_0; ..; u_{K-1})$ of length K forms the systematic part of the codeword $\boldsymbol{c}$. In the paper we will use the two equivalent notations $\boldsymbol{u} = (u_0; ..; u_{K-1})$ and $u(D) = \bigoplus_{i=0}^{K-1} u_i \cdot D^i$ simultanously, where both $\bigoplus$ and $\oplus$, denote the addition in GF(2). The middle and lower branches in Fig. 1 generate the parity symbols of the codeword. We refer to the codeword parts generated in these two branches as the *component* parity words $\boldsymbol{c}^{(1)}$ and $\boldsymbol{c}^{(2)}$. The middle and lower branches will analogously be referred to as the first and the second component of the turbo code, and all quantities $\bullet$ belonging to these branches will be denoted by $\bullet^{(1)}$ and $\bullet^{(2)}$, respectively. Both components contain scramblers. These are binary filters with an infinite impulse response (IIR), whose structure consists of a shift–register with a feedback–branch and a feedforward–branch and where addition and multiplication is binary (i.e. from GF(2)). Similar to $u(D)$ above, the feedback– and the feedforward–branches can be described by polynomials in D (a dummy variable denoting the delay operation).

In the first component, the input word $\boldsymbol{u}^{(1)} = \boldsymbol{u}$ is scrambled in its natural order. Additionally, the input word $\boldsymbol{u}$ is permuted to the second component input word $\boldsymbol{u}^{(2)}$ and then scrambled. After the scrambling, the scrambler outputs are possibly punctured appropriately in order to reduce the number of parity symbols in $\boldsymbol{c}$. In this paper, we restrict ourselves to the case of two identical component scramblers and puncturers. The permutation from $\boldsymbol{u}$ to $\boldsymbol{u}^{(2)}$ is performed by an interleaver, which can be described by a transposition vector $\boldsymbol{\pi}$ of length K. The permutation executes the transpositions according to the following rule: $u^{(2)}_{\pi_i} = u^{(1)}_i$, where $i = 0; ..; K-1$. The turbo encoder is represented in this paper by a function 'enc', which maps an input word $\boldsymbol{u}$ to the associated codeword $\boldsymbol{c} = \mathrm{enc}_{\boldsymbol{\pi}}(\boldsymbol{u})$ employing a given interleaver $\boldsymbol{\pi}$. Note that we use the term *input* word for $\boldsymbol{u}$ instead of *information* word, since we will not rule out that $\boldsymbol{u}$ contains redundancy symbols in order to terminate one or both component scrambler trellises in the zero–state (cf. [8]). However, the upper bound established in this paper is valid with or without the termination of one or both of the component scramblers.

III. New Upper Bound and Proof

In this Section, we derive an upper bound on the minimum distance $\delta_{\min}$ of a turbo code, which depends only on the interleaver length and the component scramblers employed in the turbo encoder. The bound is presented in Theorem 1, and it is proved using a geometric approach, considering only input words $\boldsymbol{u}$ of weight 2. The reason for restricting ourselves to exclusively weight 2–input words is that for a subset of these input sequences, the associated response of the component scrambler is analytically tractable. Moreover, considering these particular input sequences takes into account the "spectral thinning" effect reported in [3]. This effect represents the observation that for a random interleaver of a larger length K, the distance spectrum's low weight terms are mainly caused by the considered weight 2–input sequences. In our derivation, we proceed by first upper–bounding the component parity word weights associated with the considered weight 2–

input words in the following Lemma.

Let p denote the period of a scrambler of memory ν with a feedback branch, which can be described by a monic feedback–polynomial of degree ν, cf. [9], and a feedforward branch. It is commonly known that an input sequence of two "1"s separated by a multiple of p positions from each other leads to a finite–length error event. The first "1" in this sequence starts the error event by forcing the scrambler out of the zero–state, and the second "1" terminates the error event by leading the scrambler back to the zero–state. These weight 2–input sequences can hence be expressed as $u(D) = D^j \oplus D^{j+p\cdot i}, \quad \forall i \in \mathbb{N}; j \in \mathbb{N} \cup \{0\}$, where i denotes the length in number of scrambler periods of the error event. Considering the associated scrambler output and taking into account a subsequent output puncturing, we obtain the following Lemma.

Lemma 1: Let the function 'scrp' denote the mapping from input to output of a given device, which consists of a <u>scr</u>ambler as described above with a subsequent <u>p</u>uncturer. Then for any combination of a scrambler and a puncturer, there exist two non–negative integer numbers $\alpha; \beta \in \mathbb{N} \cup \{0\}$ such that the following inequality holds for all non–negative integer values $i; j \in \mathbb{N} \cup \{0\}$:

$$\text{wght}(\text{scrp}(u(D))) \leq \alpha \cdot i + \beta \qquad \text{where } u(D) = D^j \oplus D^{j+p\cdot i}. \tag{1}$$

wght($\bullet$) denotes the Hamming weight of a binary vector $\bullet$, and p represents the scrambler's period. The quantity α represents the linear and β denotes the constant term of this upper bound, when the length of the error–event grows with the number i of periods.

The proof for this Lemma is straightforward: The error event associated with $u(D)$ has length $p \cdot i + 1$, hence the scrambler output weight cannot exceed $p \cdot i + 1$ and it obviously cannot be increased by the puncturer. Thus, the above inequality holds, if we set $\alpha = p$ and $\beta = 1$. □

In many cases, the above inequality can be replaced by an equality. This is always the case for scramblers with an unpunctured output. A special case is that the scrambler uses a primitive feedback polynomial of degree ν, which results in a period $p = 2^\nu - 1$, and a monic feedforward polynomial of the same degree ν but different from the feedback polynomial. One can show that Eq. (1) is valid with equality for $\alpha = 2^{\nu-1}$ and $\beta = 2$ in this case, i.e. if the output is not punctured.

Having found an upper bound on the weight of the component parity words for particular component input words, we can now state an upper bound on the minimum distance of the turbo codeword.

Theorem 1: The minimum distance $\delta_{\text{min,TC}}(K; \alpha; \beta; p)$ of a parallel concatenated convolutional code with interleaver length K is upper bounded by

$$\delta_{\text{min,TC}}(K; \alpha; \beta; p) \leq \alpha \cdot \frac{(\lceil K/p \rceil - 1) \cdot \left(2 + \sqrt{2\lceil K/p^2 \rceil}\right)}{\lceil K/p^2 \rceil - 2} + 2\beta + 2,$$

for every possible interleaver (immaterial of a termination of one or both of the two component scramblers), if $K > 2p^2$. The brackets $\lceil r \rceil$ denote the smallest integer number $\geq r$ (upwards rounding function), and $\alpha; \beta$ are suitable non–negative numbers (ideally the smallest), which are determined by the bound of Lemma 1, based on the component scramblers of the turbo encoder. p denotes the period of these scramblers.

Proof:

For a linear code, the minimum distance is identical to the minimum of the Hamming weights of all codewords $\boldsymbol{c} = \text{enc}(\boldsymbol{u})$ apart from the zero–codeword. We want to upper bound the minimum distance of a turbo code for *all* possible interleavers $\boldsymbol{\pi}$, and we define hence:

$$\delta_{\min,\text{TC}} = \max_{\boldsymbol{\pi}} \left\{ \min_{\boldsymbol{u} \in \mathcal{U} \setminus \{\boldsymbol{0}\}} \{\text{wght}(\text{enc}_{\boldsymbol{\pi}}(\boldsymbol{u}))\} \right\},$$

where $\mathcal{U} \setminus \{\boldsymbol{0}\}$ is the set of all possible input words except the zero–word $\boldsymbol{0}$. We can upperbound $\delta_{\min,\text{TC}}$ by considering only a subset $\tilde{\mathcal{U}}(\boldsymbol{\pi}) \subset \mathcal{U} \setminus \{\boldsymbol{0}\}$ of input words:

$$\delta_{\min,\text{TC}} \leq \max_{\boldsymbol{\pi}} \left\{ \min_{\boldsymbol{u} \in \tilde{\mathcal{U}}(\boldsymbol{\pi})} \{\text{wght}(\text{enc}_{\boldsymbol{\pi}}(\boldsymbol{u}))\} \right\}, \tag{2}$$

since the minimum weight of the codewords associated with the considered input words remains the same or becomes greater for *every possible* interleaver $\boldsymbol{\pi}$, if we consider only a subset $\tilde{\mathcal{U}}(\boldsymbol{\pi})$ of the original arguments in the minimization. The subset $\tilde{\mathcal{U}}(\boldsymbol{\pi})$, which we will exclusively consider in this proof, is composed of input words of weight 2, which result in an output weight in the first and second component, that can be upper bounded as in Lemma 1. That is, we consider a subset of weight 2–input words, where the two "1"s are separated by a multiple of p positions in $\boldsymbol{u}$ as well as $\boldsymbol{u}^{(2)}$.

In order to identify this subset $\tilde{\mathcal{U}}(\boldsymbol{\pi})$, let us make the following definitions. The set of indices $\{0; ..; K-1\}$ can be partitioned into p disjoint subsets of indices:

$$\mathcal{I}_l^{(1)} = \{i \quad | \quad i = 0; ..; K-1 \ \wedge \ i \bmod p = l\},$$

with $l = 0; ..; p-1$. Thus, each of the p subsets $\mathcal{I}_l^{(1)}$ belongs to a distinct equivalence class with respect to the $\bmod p$–operation. Any input word of weight 2, for which the positions of its two "1" elements are in the same subset $\mathcal{I}_l^{(1)}$, results in a finite–length error–event in the first component scrambler. The two "1"s are separated by a multiple of p positions in this case, such that the first "1" starts the error–event and the second "1" terminates it. We have hence an input word in the first component, for which Lemma 1 upper bounds the weight of the associated parity word. Note that the subset $\mathcal{I}_0^{(1)}$ has cardinality $\|\mathcal{I}_0^{(1)}\| = \lceil K/p \rceil$.

Regarding the second component, the permutation $\boldsymbol{\pi}$ further partitions the set of indices $\{0; ..; K-1\}$ analogously into the following p disjoint subsets:

$$\mathcal{I}_l^{(2)}(\boldsymbol{\pi}) = \{i \quad | \quad i = 0; ..; K-1 \ \wedge \ \pi_i \bmod p = l\},$$

where $l = 0; 1; ..; p-1$. Let $\mathcal{I}_{i;j}(\boldsymbol{\pi}) = \mathcal{I}_i^{(1)} \cap \mathcal{I}_j^{(2)}(\boldsymbol{\pi})$ denote the intersection of indices for $i; j \in \{0; ..; p-1\}$. These subsets $\mathcal{I}_{i;j}(\boldsymbol{\pi})$ have the following property. Any input word of weight 2, having its two "1"s in positions which belong to the same subset $\mathcal{I}_{i;j}(\boldsymbol{\pi})$, results in a finite–length error–event in *both* component scramblers, and the parity word weights can be upper bounded as in Lemma 1. A similar partitioning of all input symbol positions was suggested in [4]. For every possible permutation $\boldsymbol{\pi}$, there exists *at least* one subset $\mathcal{I}_{0;m_0}(\boldsymbol{\pi})$ among the p disjoint subsets $\mathcal{I}_{0;j}(\boldsymbol{\pi}),\ j = 0; ..; p-1$ with cardinality $\|\mathcal{I}_{0;m_0}(\boldsymbol{\pi})\| \geq \left\lceil \frac{\lceil K/p \rceil}{p} \right\rceil$. To show this, let us assume the contrary: $\|\mathcal{I}_{0;j}(\boldsymbol{\pi})\| < \left\lceil \frac{\lceil K/p \rceil}{p} \right\rceil \quad \forall j = 0; ..; p-1$. The cardinality of $\mathcal{I}_0^{(1)}$ keeps the equation:

$$\begin{aligned} \|\mathcal{I}_0^{(1)}\| &= \left\lceil \frac{K}{p} \right\rceil \\ &= \| \bigcup_{j=0}^{p-1} \mathcal{I}_{0;j}(\boldsymbol{\pi}) \| \\ &= \sum_{j=0}^{p-1} \|\mathcal{I}_{0;j}(\boldsymbol{\pi})\|, \end{aligned}$$

as the p subsets $\mathcal{I}_{0;j}(\boldsymbol{\pi})$ are disjoint. Moreover, our assumption $\|\mathcal{I}_{0;j}(\boldsymbol{\pi})\| < \left\lceil \frac{\lceil K/p \rceil}{p} \right\rceil \quad \forall j = 0; ..; p-1$ implies that $\|\mathcal{I}_{0;j}(\boldsymbol{\pi})\| \leq \left\lceil \frac{\lceil K/p \rceil}{p} \right\rceil - 1 \quad \forall j = 0; ..; p-1$ (as the cardinality must be an integer). Therefore, we get:

$$\begin{aligned} \|\mathcal{I}_0^{(1)}\| &\leq \sum_{j=0}^{p-1} \left\lceil \frac{\lceil K/p \rceil}{p} \right\rceil - 1 \\ &= p \cdot \left\lceil \frac{\lceil K/p \rceil}{p} \right\rceil - p \\ &< p \cdot \left(\frac{\lceil K/p \rceil}{p} + 1 \right) - p \\ &= \left\lceil \frac{K}{p} \right\rceil, \end{aligned}$$

which contradicts the equality $\|\mathcal{I}_0^{(1)}\| = \lceil K/p \rceil$. Hence our assumption $\|\mathcal{I}_{0;j}(\boldsymbol{\pi})\| < \left\lceil \frac{\lceil K/p \rceil}{p} \right\rceil \quad \forall j = 0; ..; p-1$ must be false, and consequently there exists at least one subset of cardinality $\|\mathcal{I}_{0;m_0}(\boldsymbol{\pi})\| \geq \left\lceil \frac{\lceil K/p \rceil}{p} \right\rceil$. Furthermore, one can show that $\left\lceil \frac{\lceil K/p \rceil}{p} \right\rceil = \left\lceil \frac{K}{p^2} \right\rceil$. The value of the index m_0 obviously depends on the particular interleaver $\boldsymbol{\pi}$, and the exact notation would be $m_0(\boldsymbol{\pi})$, but for the sake of readability, we keep this dependence only in mind and omit it in the index m_0.

For evaluating Eq. (2), we consider only the subset $\tilde{\mathcal{U}}(\boldsymbol{\pi})$ of input words of weight 2, whose two "1"s are elements of $\boldsymbol{u}$ with indices $i; j \in \mathcal{I}_{0;m_0}(\boldsymbol{\pi})$, i.e. $i, j \bmod p = 0$ and the associated (through permutation) elements of $\boldsymbol{u}^{(2)}$ have indices $\pi_i \bmod p = \pi_j \bmod p = m_0$:

$$\tilde{\mathcal{U}}(\boldsymbol{\pi}) = \{D^i \oplus D^j \quad | \quad i; j \in \mathcal{I}_{0;m_0}(\boldsymbol{\pi}) \ \wedge \ i \neq j\}.$$

Note that the sets $\tilde{\mathcal{U}}(\boldsymbol{\pi})$ and $\mathcal{I}_{0;m_0}(\boldsymbol{\pi})$ depend on the specific interleaver $\boldsymbol{\pi}$. Since the turbo codeword consists of three parts, its Hamming weight can be split into three terms

$$\mathrm{wght}(\boldsymbol{c}) = \mathrm{wght}(\boldsymbol{u}) + \mathrm{wght}(\boldsymbol{c}^{(1)}) + \mathrm{wght}(\boldsymbol{c}^{(2)}).$$

For every $\boldsymbol{u} \in \tilde{\mathcal{U}}(\boldsymbol{\pi})$, we can then use the upper bound of Lemma 1 (observe that this upper bound becomes equality in many cases as stated above):

$$\begin{aligned} \mathrm{wght}(\mathrm{enc}_{\boldsymbol{\pi}}(\boldsymbol{u})) &= \mathrm{wght}(\boldsymbol{u}) + \mathrm{wght}(\mathrm{scrp}(\boldsymbol{u})) + \mathrm{wght}(\mathrm{scrp}(\boldsymbol{u}^{(2)})) \\ &\leq 2 + \alpha \cdot (|i-j|/p + |\pi_i - \pi_j|/p) + 2\beta, \end{aligned}$$

where $u(D) = D^i \oplus D^j \in \tilde{\mathcal{U}}(\boldsymbol{\pi})$ or equivalently $i; j \in \mathcal{I}_{0;m_0}(\boldsymbol{\pi});\ i \neq j$. The function 'scrp' represents the combination of scrambler and puncturer, which we assume identical for both components.

For a specific interleaver $\boldsymbol{\pi}$, we have hence

$$\min_{\boldsymbol{u} \in \tilde{\mathcal{U}}(\boldsymbol{\pi})} \{\mathrm{wght}(\mathrm{enc}_{\boldsymbol{\pi}}(\boldsymbol{u}))\} \quad \leq \quad \alpha \cdot \min_{\substack{i;j \in \mathcal{I}_{0;m_0}(\boldsymbol{\pi}) \\ i \neq j}} \{|i-j|/p + |\pi_i - \pi_j|/p\} + 2\beta + 2,$$

where i and j are the indices of the two "1" elements in $\boldsymbol{u}$. Thus, Eq. (2) can be rewritten as follows

$$\delta_{\mathrm{min,TC}} \leq \alpha \cdot \max_{\boldsymbol{\pi}} \{ \min_{\substack{i;j \in \mathcal{I}_{0;m_0}(\boldsymbol{\pi}) \\ i \neq j}} \{|i-j|/p + |\pi_i - \pi_j|/p\}\} + 2\beta + 2. \tag{3}$$

To perform this maximization with respect to all possible interleavers $\boldsymbol{\pi}$, we need the following

Lemma 2: Let $\mathcal{R} \subset \mathbb{R}^2$ be a rectangular surface with edges parallel to the co–ordinate axes and edge lengths w_1 and w_2 (let the edges be included in $\mathcal{R}$). Let the set $\mathcal{P}$ be composed of $M > 2$ points picked from $\mathcal{R}$, such that the points of $\mathcal{P}$ satisfy the distance constraint $\|\boldsymbol{r} - \boldsymbol{q}\|_1 \geq d \quad \forall \boldsymbol{r}; \boldsymbol{q} \in \mathcal{P};\ \boldsymbol{r} \neq \boldsymbol{q}$ for a given positive constant $d \in \mathbb{R}^+$, where we use the ℓ_1–norm $\|\boldsymbol{r}\|_1 \stackrel{\triangle}{=} |r_1| + |r_2| \quad \forall \boldsymbol{r} = (r_1; r_2) \in \mathbb{R}^2$.

Then, for every such set $\mathcal{P}$ the corresponding distance d is upper–bounded by the inequality:

$$d \leq \frac{w_1 + w_2 + \sqrt{w_1^2 + w_2^2 + 2w_1w_2(M-1)}}{M-2}.$$

Proof:

We can assign a *neighbourhood* $\mathcal{N}_d(\boldsymbol{r}) \subset \mathbb{R}^2$ to every point $\boldsymbol{r} \in \mathbb{R}^2$ by defining

$$\mathcal{N}_d(\boldsymbol{r}) = \{\boldsymbol{z} \quad | \quad \|\boldsymbol{z} - \boldsymbol{r}\|_1 < \frac{d}{2}\}.$$

Fig. 2a displays such a neighbourhood. For any two distinct points $\boldsymbol{r};\boldsymbol{q} \in \mathcal{P};\boldsymbol{r} \neq \boldsymbol{q}$, their associated neighbourhoods must be disjoint. This can easily be shown by assuming the converse: There exists a point $\boldsymbol{z} \in \mathcal{N}_d(\boldsymbol{r}) \cap \mathcal{N}_d(\boldsymbol{q})$. Then by virtue of the triangle inequality, which holds for any norm, we obtain $\|\boldsymbol{r}-\boldsymbol{q}\|_1 \leq \|\boldsymbol{r}-\boldsymbol{z}\|_1 + \|\boldsymbol{z}-\boldsymbol{q}\|_1 < 2 \cdot \frac{d}{2}$, which is a contradiction to the fact that $\boldsymbol{r};\boldsymbol{q} \in \mathcal{P}$.

Without loss of generality, let us assume that $\mathcal{R} = [0;w_1] \times [0;w_2]$. All elements of $\mathcal{P}$ are points from this rectangular surface. Hence all the M neighbourhoods associated with the elements of $\mathcal{P}$ are *disjoint* subsets of the rectangular surface $(-d/2;w_1+d/2) \times (-d/2;w_2+d/2)$. The area of this surface is $(w_1+d)\cdot(w_2+d)$. The surface area of each of the M neighbourhoods $\mathcal{N}_d$ is $d^2/2$ (see Fig. 2a). We have hence

$$M \cdot \frac{d^2}{2} \leq (w_1+d)\cdot(w_2+d),$$

which results in

$$\left(\frac{M}{2}-1\right)d^2 - (w_1+w_2)d - w_1 \cdot w_2 \leq 0.$$

If $M > 2$, we obtain for positive d the upper bound

$$d \leq \frac{w_1+w_2}{M-2} + \sqrt{\left(\frac{w_1+w_2}{M-2}\right)^2 + \frac{2w_1w_2}{M-2}},$$

which after some simple manipulations gives the upper bound of Lemma 2. □

Observe that the bound given above is fairly tight when $(w_1+w_2)^2 \ll 2w_1w_2\cdot(M-2)$. This can be illustrated by an example. Let us consider the case of a lattice $\mathcal{Q}$ like in Fig. 2b

$$\mathcal{Q} = \{(i;j)\cdot G \quad | \quad i,j \in \mathbb{Z}\} \tag{4}$$

with the generator matrix

$$G = \begin{bmatrix} d & 0 \\ d/2 & d/2 \end{bmatrix},$$

which represents the densest packing of infinitely many points according to the above distance constraint [10]. Let us consider a rectangular surface $\mathcal{R}$, whose edges are parallel to the co–ordinate axes and have lengths $w_1 = i\cdot d$ and $w_2 = j \cdot d$, $i;j \in \mathbb{N}^+$. Let $\mathcal{P}$ be the set of lattice points contained in the rectangle $\mathcal{R}$ (the edges belong to the rectangle). Then the maximum number of points in $\mathcal{P}$ is $M = 2i\cdot j + i + j + 1$, which is obtained by placing the rectangle $\mathcal{R}$, such that there is a lattice point in each of the four corners, see Fig. 2b. For this example, let us examine the case of a square, i.e. $w_1 = w_2 = i \cdot d$. Let the <u>u</u>pper <u>b</u>ound on d according to Lemma 2 be denoted by d_{UB}. This quantity normalized by d is shown in Fig. 3 for $M = 2i^2+2i+1$ over growing i. We see that the bound becomes fairly tight already for $i \geq 4$ corresponding to $M \geq 41$.

If $(w_1 + w_2)^2$ is not much smaller than $2w_1w_2 \cdot (M-2)$, then the bound can probably be tightened. However, Lemma 2 provides an upper bound for arbitrary many points not confined to positions in a regular lattice and for a rectangle of arbitrary edge lengths.

Now we are able to continue with the proof of Theorem 1. From what is stated above, we know that there exists at least one $\mathcal{I}_{0;m_0}(\boldsymbol{\pi})$ of a cardinality, which is $\geq \left\lceil \frac{K}{p^2} \right\rceil$. From Eq. (3) we see that our task is now to find the specific interleaver $\boldsymbol{\pi}_0$ maximizing

$$\min_{\substack{i;j \in \mathcal{I}_{0;m_0}(\boldsymbol{\pi}) \\ i \neq j}} \{|i-j|/p + |\pi_i - \pi_j|/p\}, \tag{5}$$

where $\mathcal{I}_{0;m_0}(\boldsymbol{\pi})$ depends on the considered interleaver $\boldsymbol{\pi}$. Let us define the following mapping:

$$\boldsymbol{f}_{\boldsymbol{\pi},m_0}(i) \triangleq \Big(\; i/p; \quad (\pi_i - m_0)/p \; \Big) \quad \forall i \in \mathcal{I}_{0;m_0}(\boldsymbol{\pi}).$$

Thus, $\boldsymbol{f}_{\boldsymbol{\pi},m_0}$ maps indices to two–dimensional points, and we can rewrite Eq. (5) as follows:

$$\min_{\substack{i;j \in \mathcal{I}_{0;m_0}(\boldsymbol{\pi}) \\ i \neq j}} \{|i-j|/p + |\pi_i - \pi_j|/p\} = \min_{\substack{i;j \in \mathcal{I}_{0;m_0}(\boldsymbol{\pi}) \\ i \neq j}} \{\|\boldsymbol{f}_{\boldsymbol{\pi},m_0}(i) - \boldsymbol{f}_{\boldsymbol{\pi},m_0}(j)\|_1\}.$$

The range $\boldsymbol{f}_{\boldsymbol{\pi},m_0}(\mathcal{I}_{0;m_0}(\boldsymbol{\pi}))$ of the mapping is a subset of $[0; \lceil K/p \rceil - 1]^2$. We can obtain a geometrical representation of the indices belonging to $\mathcal{I}_{0;m_0}(\boldsymbol{\pi})$ by graphically depicting their images from the mapping $\boldsymbol{f}_{\boldsymbol{\pi},m_0}$, i.e. the indices correspond to at least $\left\lceil \frac{K}{p^2} \right\rceil$ points inside the square $[0; \lceil K/p \rceil - 1]^2$. Any input word from $\tilde{\mathcal{U}}(\boldsymbol{\pi})$ can then be represented by a pair of these points, and the associated codeword weight grows linearly with the distance between the two points (with respect to the $\| \bullet \|_1$ norm). For a given interleaver $\boldsymbol{\pi}$, the minimum weight of all codewords associated with $\tilde{\mathcal{U}}(\boldsymbol{\pi})$ grows therefore linearly with the minimum distance between the points of the set $\boldsymbol{f}_{\boldsymbol{\pi},m_0}(\mathcal{I}_{0;m_0}(\boldsymbol{\pi}))$. The maximization of Eq. (3) can now be reformulized as follows: Find the specific interleaver $\boldsymbol{\pi}_0$ with an associated index set $\mathcal{I}_{0;m_0}(\boldsymbol{\pi}_0)$, for which the minimum distance between the $\geq \left\lceil \frac{K}{p^2} \right\rceil$ points of $\boldsymbol{f}_{\boldsymbol{\pi},m_0}(\mathcal{I}_{0;m_0}(\boldsymbol{\pi}_0))$ becomes maximal. A more general problem can be stated as follows: How can we arrange $\geq \left\lceil \frac{K}{p^2} \right\rceil$ points in the square $[0; \lceil K/p \rceil - 1]^2$, such that their minimum distance is maximized? From Lemma 2 we know, that for any set $\mathcal{P}$ composed of $M = \left\lceil \frac{K}{p^2} \right\rceil > 2$ points out of the set $[0; \lceil K/p \rceil - 1]^2$, the maximum of the minimum distance between points of $\mathcal{P}$ is upper–bounded by

$$\begin{aligned} \max_{\mathcal{P}} \{ \min_{\substack{\boldsymbol{r};\boldsymbol{q} \in \mathcal{P} \\ \boldsymbol{r} \neq \boldsymbol{q}}} \|\boldsymbol{r} - \boldsymbol{q}\|_1 \} \quad &\leq \quad \frac{2\left(\lceil K/p \rceil - 1\right) + \sqrt{2\left(\lceil K/p \rceil - 1\right)^2 + 2\left(\lceil K/p \rceil - 1\right)^2 \cdot (M-1)}}{M-2} \\ &= \quad \frac{\left(\lceil K/p \rceil - 1\right) \cdot \left(2 + \sqrt{2M}\right)}{M-2}. \end{aligned}$$

Since increasing the number of points in $\mathcal{P}$ cannot increase the maximum minimum distance, the above upper bound is also valid for $\geq \left\lceil \frac{K}{p^2} \right\rceil$ points in the set $\mathcal{P}$.

Putting all pieces together, we obtain for $\left\lceil \frac{K}{p^2} \right\rceil > 2$:

$$\max_{\boldsymbol{\pi}}\{ \min_{\substack{i;j \in \mathcal{I}_{0;m_0}(\boldsymbol{\pi}) \\ i \neq j}} \{|i-j|/p + |\pi_i - \pi_j|/p\} \leq \frac{(\lceil K/p \rceil - 1) \cdot \left(2 + \sqrt{2\lceil K/p^2 \rceil}\right)}{\lceil K/p^2 \rceil - 2},$$

where the set $\mathcal{I}_{0;m_0}(\boldsymbol{\pi})$ is composed of $\geq \left\lceil \frac{K}{p^2} \right\rceil$ indices depending on the considered interleaver $\boldsymbol{\pi}$ as described above. Finally, the upper bound for the minimum distance of the turbo code is

$$\delta_{\text{min,TC}} \leq \alpha \cdot \frac{(\lceil K/p \rceil - 1) \cdot \left(2 + \sqrt{2\lceil K/p^2 \rceil}\right)}{\lceil K/p^2 \rceil - 2} + 2\beta + 2,$$

where $(\alpha; \beta)$ is a pair of parameters, for which the upper bound of Lemma 1 holds for the component scramblers employed. The condition $\left\lceil \frac{K}{p^2} \right\rceil > 2$ translates into $\frac{K}{p^2} > 2$ and finally $K > 2p^2$. □

IV. Discussion of the Upper Bound

Figs. 4 and 5 show the proposed new upper bound $\delta_{\text{min,UB}}$ on the minimum distance normalized by the codeword length N for turbo codes using different component scramblers ("$\nu = 2; 3; 4$" in the Figure). The smallest values α and β in the bound of Lemma 1 are given in Table I for component scramblers of memory $\nu = 2; 3; 4$, which have maximum period $p = 2^\nu - 1$. As already stated in Section III, we can compute the values α and β analytically for unpunctured scrambler output (resulting in an overall turbo code of rate $R = 1/3$), and Lemma 1 holds with equality in this case. On the other hand, for regular puncturing of every other output symbol (overall code rate $R = 1/2$) the values given in Table I were determined manually. The transfer function associated with the scramblers is displayed as FF/FB, where FF and FB are numbers (in octal), whose binary representations correspond to the feedforward and feedback (recursion) polynomials, respectively.

Also in the Figures, the Hamming (upper) bound on the normalized minimum distance as well as the *asymptotic* upper bound by McEliece et al. is displayed, cf. [11]. The values of the McEliece et al.–bound are $\delta_{\text{min}}/N < 0.26$ for every linear binary code of rate 1/3 and $\delta_{\text{min}}/N < 0.18$ for rate 1/2. Furthermore, the Gilbert–Varshamov–bound is shown. Observe that this is not an *upper* bound but an existence bound on the achievable minimum distance of a linear binary code of a given rate, cf. [11]. Since for large K the Gilbert–Varshamov–bound is above the new upper bounds, there cannot exist an upper bound, which is valid for every linear binary code, and which is tighter than our new bounds for the special case of binary turbo codes. For large interleaver lengths K, the new upper bounds for binary turbo codes are consideringly lower than the existing bounds for general linear binary codes. In both Figures we see that the new upper bound is much lower for component scramblers of a short period ($p = 3$ for $\nu = 2$) than for a long period ($p = 15$ for $\nu = 4$). The new bounds indicate therefore that the achievable increase in $\delta_{\min}$ by designing interleavers is necessarily smaller for component scramblers of small memory than for larger memory. Achieving a required minimum distance $\delta_{\min}$ by designing the turbo code interleaver is hence a more challenging task, when the turbo decoder has to have low complexity and consequently the component scramblers have a small memory.

TABLE I

$(\alpha; \beta)$ BOUNDING VALUES FOR DIFFERENT COMPONENT SCRAMBLERS

ν	Transf.fct.	p	$(\alpha;\beta)$ f. $R = 1/3$	$(\alpha;\beta)$ f. $R = 1/2$
2	$5_8/7_8$	3	(2;2)	(1;2)
3	$17_8/13_8$	7	(4;2)	(2;2)
4	$35_8/23_8$	15	(8;2)	(4;2)

The upper bound of Theorem 1 can be simplified by using the relationship $r \leq \lceil r \rceil < r + 1 \quad \forall r \in \mathbb{R}$. By applying these inequalities, we obtain the following simpler upper bound, which is less tight than the upper bound of Theorem 1.

Theorem 2: The minimum distance $\delta_{\min,\mathrm{TC}}(K;\alpha;\beta;p)$ of a parallel concatenated convolutional code with interleaver length K and component scramblers, which can be described by the parameters α; β and the period p, is upper bounded by

$$\delta_{\min,\mathrm{TC}}(K;\alpha;\beta;p) < \alpha \cdot \frac{K \cdot \left(2p + \sqrt{2K + 2p^2}\right)}{K - 2p^2} + 2\beta + 2.$$

Proof: Straightforward.

We see immediately that the upper bound of Theorem 2 grows asymptotically for large K like $\alpha \cdot \sqrt{2K}$, which means that *turbo codes are always and even with the best possible interleaver asymptotically bad*, since $\delta_{\min,\mathrm{TC}}/N \to 0$ for $N \to \infty$.

Comparing our upper bounds with the results of [12], we find that the bound of [12] on the minimum distance does not grow asymptotically with K, i.e. it grows in the order $O(1)$, whereas our bound grows as $O(\sqrt{K})$. The reason is that Kahale and Urbanke's bound is a *statistical* bound on the minimum distance, when the interleaver is randomly drawn. This bound is valid with probability → 1, when the interleaver length K grows to infinity. By contrast, our bound is valid for *finite* K, and it is not statistical, i.e. it is valid for *all* — including the *best* — possible interleavers. The following example will make clearer that both bounds can be valid simultanously: Imagine a case, where there are $(K-1)!$ among the $K!$ possible interleavers of length K, for which the minimum distance grows as $O(\sqrt{K})$, whereas for the remaining $K! - (K-1)! = (K-1) \cdot (K-1)!$ interleavers the minimum distance grows as $O(1)$. Clearly, for a randomly drawn interleaver, its associated minimum distance is $O(1)$ with probability $(K-1) \cdot (K-1)!/K! = (K-1)/K \to 1$ for $K \to \infty$, although there are numerous interleavers with the higher minimum distance $O(\sqrt{K})$.

The paper [12] also states an analog statistical bound on the minimum distance of *Serially Concatenated Convolutional Codes* (SCCC) with a random interleaver. Here, the minimum distance grows as $O(K^{(\delta_{\text{free,o}}-2)/\delta_{\text{free,o}}})$, where $\delta_{\text{free,o}}$ is the free distance of the *outer* convolutional code in the concatenation. We see that for large K, an SCCC with a *random* interleaver will on average match or even outperform (as regards to the minimum distance) a turbo code, i.e. a parallel concatenated convolutional code, with the *best* possible interleaver, if $\delta_{\text{free,o}} \geq 4$. This comparison once again highlights the superiority of SCCCs to turbo codes, when minimum distances are considered. Similar results were found in [13], where the distance spectra of turbo codes and SCCCs were determined for the case of *uniform interleaving* [2] and time–varying component scramblers. In [14], the same authors derived upper bounds on the BER for SCCCs and turbo codes applying the tangential sphere bound on the distance spectrum of the uniform interleaver. They showed that the error floor is indeed lower for SCCCs than for turbo codes, which is due to an increased minimum distance of the considered SCCC ensemble.

Note that an immediate relationship between the minimum distance and the level of the error floor exists only for sufficiently high signal–to–noise ratios (SNRs), for which the channel's cutoff rate is above the code rate. However, their high power–efficiency makes turbo codes (and SCCCs) particularly appealing at lower SNRs, where the run of the BER curve is determined not only by the minimum distance term. The paper [14] states that the error floor at these low SNRs is caused by a whole range of subsequent terms in the distance spectrum up to twice the Gilbert–Varshamov–distance.

Recall that we have derived an *upper* bound, which is valid for *all* interleavers of a given length K. This means on the other hand that there exist interleavers with a *low* corresponding minimum distance, for which the presented bound is *loose*. Take as an example randomly chosen interleavers: [2] shows that averaged over all those interleavers there exist codewords at low distances. As shown in [12] and as was stated above, for almost every interleaver the minimum distance is lower than a constant independent of the interleaver length K, such that for large K our bound becomes loose for almost every (randomly chosen) interleaver.

However, the quality of the new bound should be evaluated by comparing it with minimum distances achieved with the *best* known interleavers, i.e. *designed* interleavers instead of *random* interleavers. Several good interleavers — as long as only input words $\boldsymbol{u}$ of weight 2 are considered — have been presented in the literature: For weight 2–input words, the designed "non–random" interleavers of [5] and the similar interleavers of [6] both achieve minimum codeword weights in the same order $O(\sqrt{K})$ as the upper bound of Theorem 1. These interleavers have been explicitly designed to make the codeword weights associated with all *weight 2*–input words as large as possible. However, the *actual* minimum distances corresponding to these interleavers are caused by input words $\boldsymbol{u}$ of a *higher* weight than 2, such that the actual minimum distances are much lower than $O(\sqrt{K})$. In fact, [5] and [15] show that for every interleaver based on linear congruences, like the two mentioned interleavers of [5] and [6], there exist *weight 4*–input words generating a low codeword weight.

Based on this observation and making use of the "spectral thinning" effect pointed out in [3], Dolinar and Divsalar introduce *s–random* interleavers in [5], which are among the most power–efficient interleavers known for the time being. These interleavers satisfy the following *spreading condition*: The two "1" positions i and j of every weight 2–input word $\boldsymbol{u}$ must be separated by at least s positions in either the first or the second component input word, i.e. $|i-j| \geq s$ or $|\pi_i - \pi_j| \geq s$, where s is a threshold chosen as large as possible. This reminds of the examinations in Section III, where we found that the considered minimum separation $|i-j| + |\pi_i - \pi_j|$ is upper–bounded by $O(\sqrt{K})$. Indeed, for s–random interleavers, s can be chosen in the same order as this upper bound, such that the minimum codeword weight for weight 2–input words grows as $O(\sqrt{K})$. Except for the above separation condition, the permutations of s–random interleavers are chosen *randomly*, such that the "spectral thinning" argumentation [3] holds. Thus, input weights greater than 2 do in general produce high codeword weights for s–random interleavers, which results in a clearly increased minimum distance compared to pure random interleavers and a higher power–efficiency. Nonetheless it is shown in [16] that for most s–random interleavers, the minimum distance $\delta_{\min}$ is caused by an input word of weight 4, and that this $\delta_{\min}$ value does not grow with the interleaver length K. Thus the minimum distance of s–random interleavers remains lower than the upper bound derived in the present paper. Recently a new interleaver design approach was proposed in [17], which uses a slightly modified version of the spreading condition given above. The spreading goal introduced in [17] is to maximize the minimum of $|i-j| + |\pi_i - \pi_j|$ for all pairs $i; j \in \{0; ..; K-1\}$. This goal is closely related to the one given in Eq. (5), and we can immediately see that this interleaver design goal is related to the bound of Eq. (3). The "diagonal interleavers" of [17] are based on a lattice construction as described by Eq. (4) and as displayed in Fig. 2b, and the constructed interleavers yield quite remarkable values in Eq. (5). For these interleavers, the achieved minimum of $|i-j| + |\pi_i - \pi_j|$ is in fact approximately $\sqrt{2K}$, which is the asymptotic upper bound of Lemma 2, if we set $w_1 = w_2 = K-1$ and $M = K$. Hence the minimum codeword weight is large for input weight 2 (cf. Eq. (3)). Crozier states in [17] that the achieved minimum distances are very large even for higher input weights, however they are much lower than what our new bound suggests.

Hence we conclude that our upper bound is tight, if only input words of weight 2 are considered, but it can probably be tightened, when higher input weights are taken into consideration, too. This remains an object of further study.

V. Conclusion

We have derived a new upper bound on the achievable minimum distance of turbo codes, which is for turbo codes asymptotically tighter than the tightest possible upper bound for general linear binary codes. The bound was proved by considering only a subset of encoder input words of weight 2 and by using a geometric approach. The new bound implies that the minimum distance of turbo codes cannot grow stronger than the square root of the codeword length. Thus, turbo codes are asymptotically bad, even when the best of all possible interleavers is employed.

Acknowledgements

The authors would like to thank the anonymous reviewers for their numerous constructive comments, which helped to improve the quality of this paper.

References

[1] Claude Berrou, Alain Glavieux, and Punya Thitimajshima. Near shannon limit error-correcting coding and decoding: Turbo codes. *International Conference on Communications*, pages 1064–1070, 1993.

[2] Sergio Benedetto and Guido Montorsi. Unveiling turbo codes: Some results on parallel concatenated coding schemes. *IEEE Transactions on Information Theory*, 42(2):409–428, 1996.

[3] Lance Perez, Jan Seghers, and Daniel Costello. A distance spectrum interpretation of turbo codes. *IEEE Transactions on Information Theory*, 42(6):1698–1709, 1996.

[4] A. Khandani. Design of turbo-code interleaver using Hungarian method. *Electronics Letters*, 34(1):63–65, 1998.

[5] S. Dolinar and D. Divsalar. Weight distributions for turbo codes using random and nonrandom permutations. *TDA Progress Report*, 42-122:56–65, 1995.

[6] Kenneth Andrews, Chris Heegard, and Dexter Kozen. Interleaver design methods for turbo codes. *Intern. Symposium on Inf. Th.*, page 420, 1998.

[7] W. Blackert, E. Hall, and S. Wilson. An upper bound on turbo code free distance. *International Conference on Communications*, pages 957–961, 1996.

[8] O. Jörssen and H. Meyr. Terminating the trellis of turbo-codes. *Electronics Letters*, 30(16):1285–1286, 1994.

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

[10] J. Conway and N. Sloane. *Sphere Packings, Lattices and Groups.* Springer Verlag, 1988.

[11] F. MacWilliams and N. Sloane. *The Theory of Error-Correcting Codes.* North-Holland, 1977.

[12] Nabil Kahale and Rüdiger Urbanke. On the minimum distance of parallel and serially concatenated codes. *available at* `lcavwww.epfl.ch/~ruediger/publications.html`, 1997.

[13] Igal Sason and Shlomo Shamai (Shitz). On union bounds for random serially concatenated turbo codes with maximum likelihood decoding. *Europ. Trans. on Telecomm.*, 11(3):271–282, 2000.

[14] Igal Sason and Shlomo Shamai (Shitz). Improved upper bounds on the ml decoding error probability of parallel and serial concatenated turbo codes via their ensemble distance spectrum. *IEEE Transactions on Information Theory*, 46(1):24–47, 2000.

[15] Oscar Takeshita and Daniel Costello. On deterministic linear interleavers for turbo-codes. *Allerton Conf. on Comm.*, pages 711–712, 1997.

[16] Marco Breiling, Stein Peeters, and Johannes Huber. Interleaver design using backtracking and spreading methods. *Intern. Symposium on Inf. Th.*, page 451, 2000.

[17] Stewart Crozier. New high–spread high–distance interleavers for turbo–codes. *20–th Symp. on Communications, Kingston, Canada, available at* `www.crc.ca/fec/publications.htm`, pages 3–7, May 2000.

List of Figures

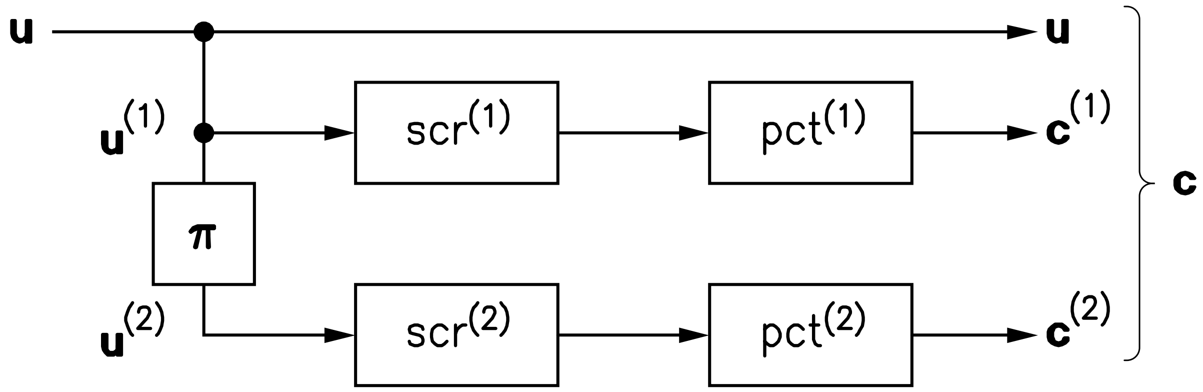

Fig. 1. System model of the turbo encoder

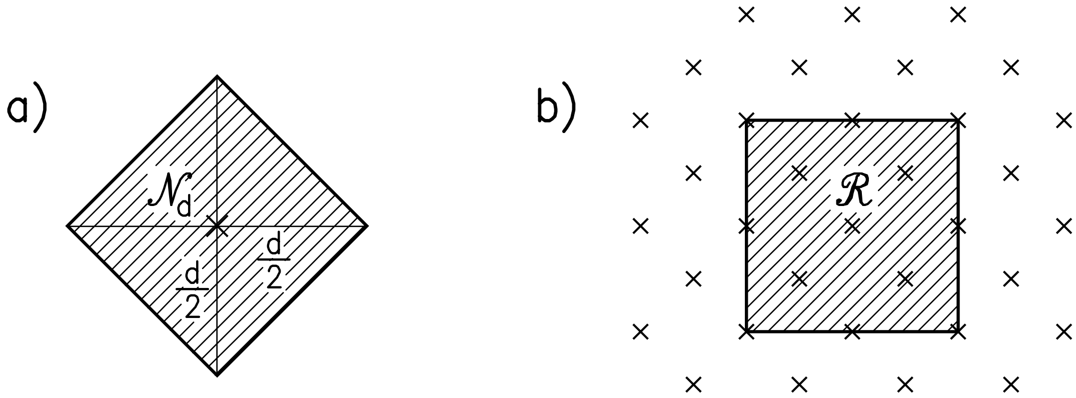

Fig. 2. a) A neighbourhood, b) a rectangle placed in a lattice

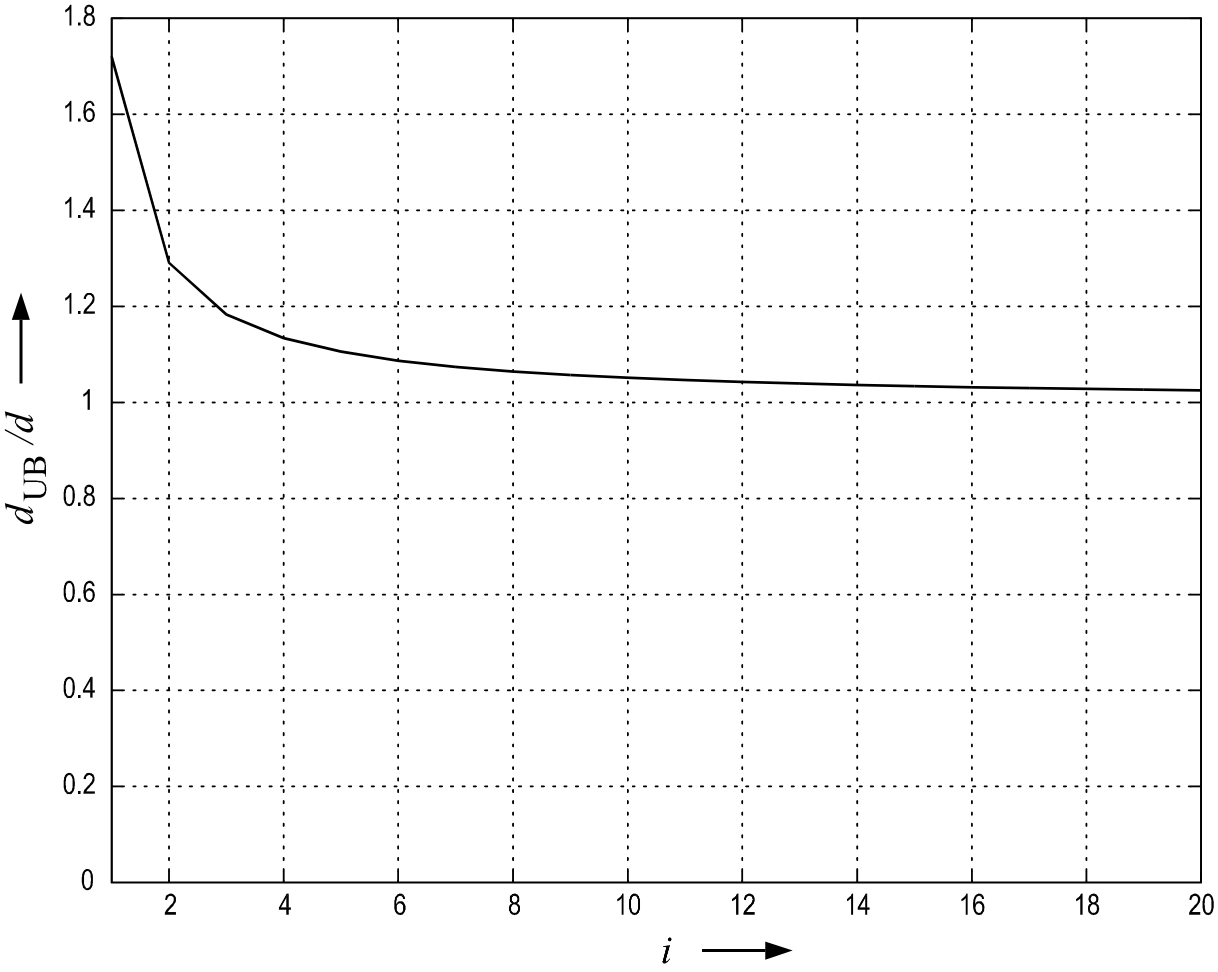

Fig. 3. Evaluation of the tightness of the upper bound of Lemma 2

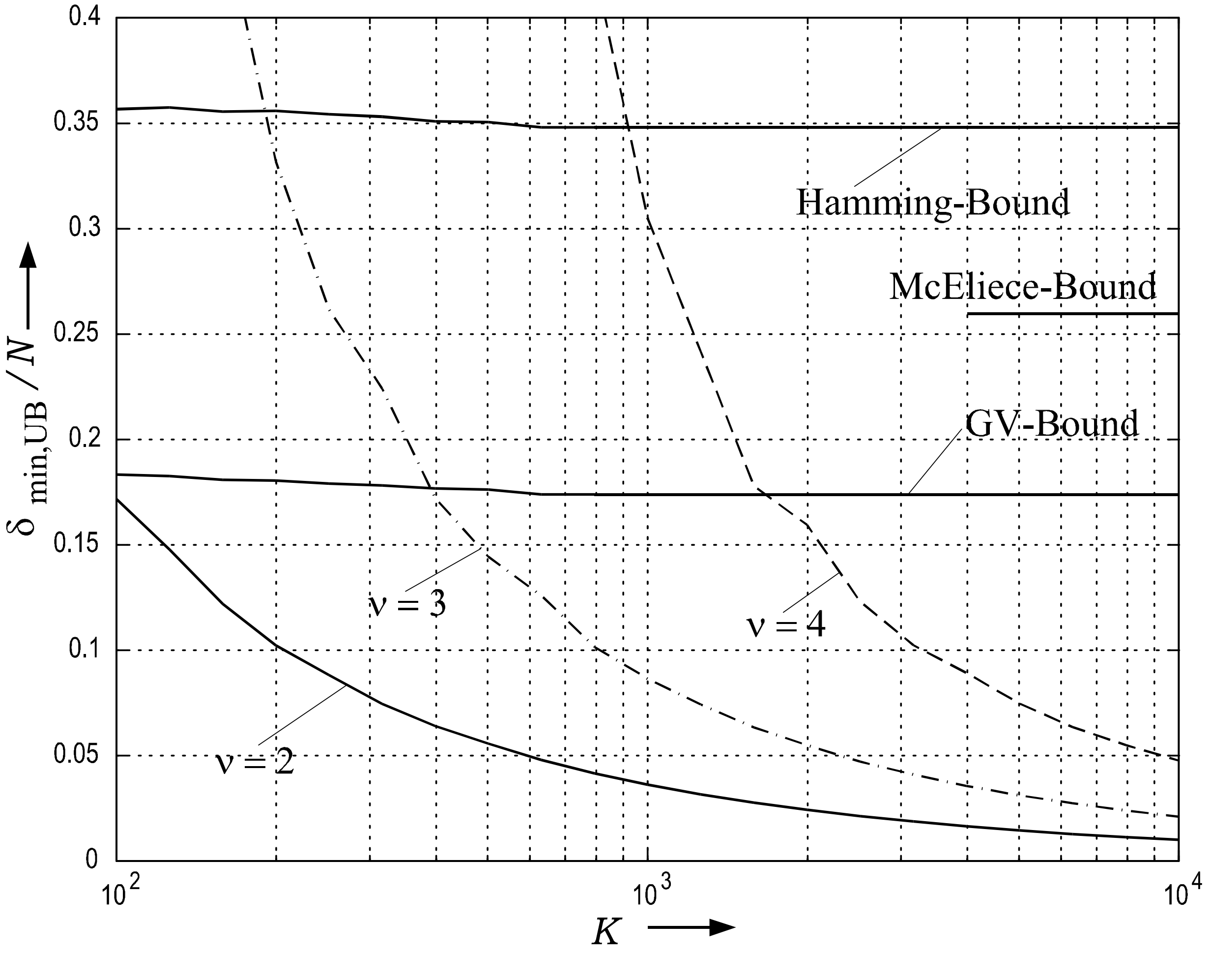

Fig. 4. (Upper) bounds on the normalized minimum distance for rate 1/3

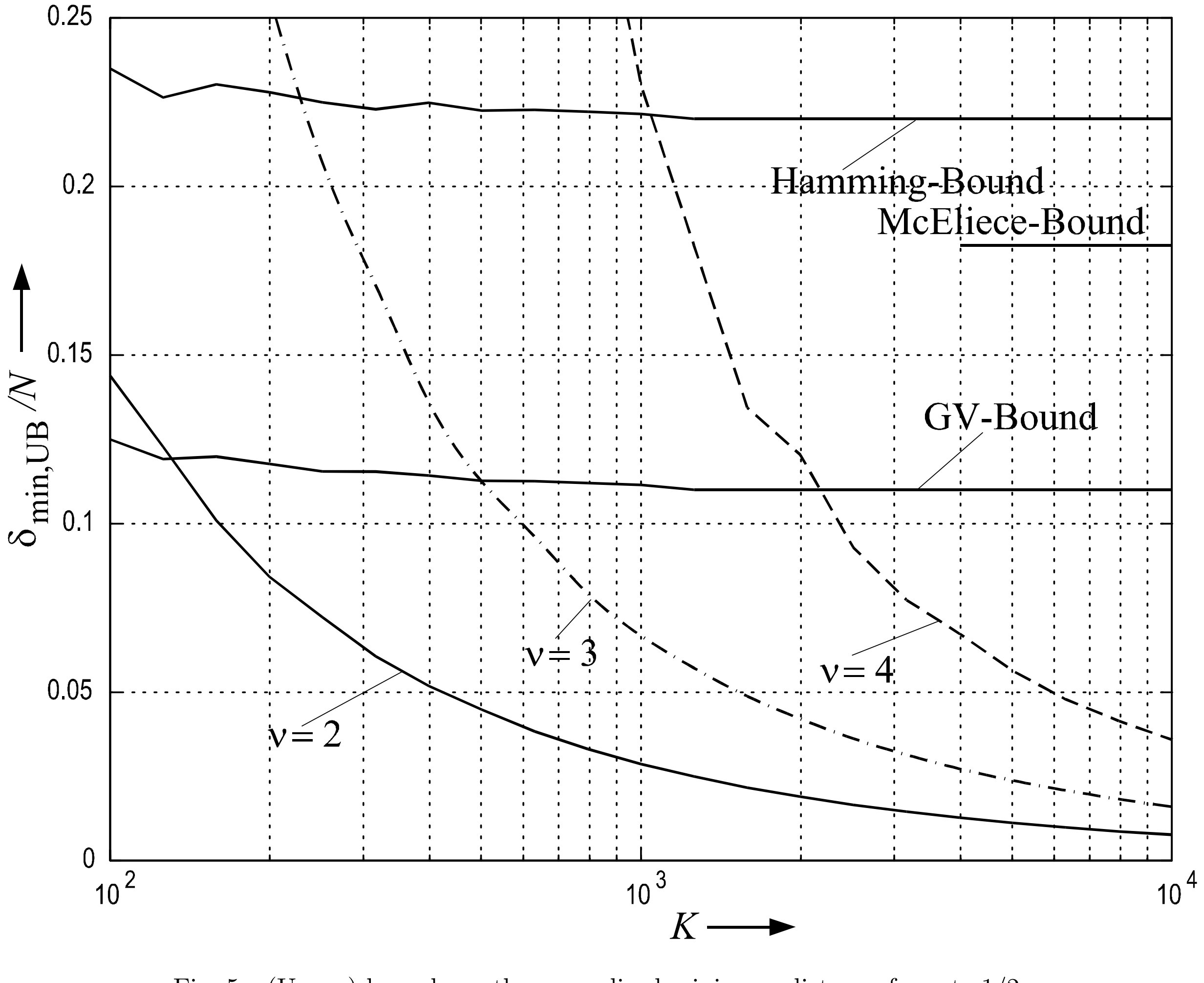

Fig. 5. (Upper) bounds on the normalized minimum distance for rate 1/2