ANALYSIS AND DESIGN OF COMBINED INTERLEAVER FOR TURBO CODES

ILIEV TEODOR

University of Rousse

Abstract: In this work are analyze a large number of random interleavers in order to draw conclusions about the characteristics of low – weight codewords. These insights are applied in our research to devise a novel interleaver design algorithm that proves particularly powerful for the case of component encoders with low memory. In this paper we identify the advantages and deficiencies of four design algorithms by analyzing their truncated distance spectra and comparing them each other.

Keywords: Turbo codes, interleaver, minimum distances, bit error rate

1. INTRODUCTION

The s- random and the backtracking design can easily combined, because the step – wise interleaver construction in both algorithms is very similar. A construction incorporating both algorithms must thus be given three parameters: the spreading parameter s, the target minimum distance $\delta_{min,targ}$ and the maximum weight $h_{trunc,des}$ of the error patterns (EP's) considered in the design. From [1] we see that the s – random and the backtracking design algorithms differ only in line 2. In this line, both algorithms try to identify "unfavourable" choices for the current element π_l. We define that a value is unfavourable, if it is unfavourable for either the spreading or the backtracking design algorithm. Hence, our new combined algorithm contains lines 0, 1 and 3 from the backtracking design algorithm, and the following new line 2:[1]

0. Initialize the set A_l of unfavourable values for the image π_l to $A_l = 0$;
1. Randomly choose π_l from $\{0,\ldots,K\text{-}1\}/(\pi(\{0,1,\ldots,l\text{-}1\})\cup A_l)$. If this set is empty and no choice is possible then terminate the program without having designed an interleaver;
2. If $|\pi_l - \pi_i| < s$ for any $i \in \{l - s + 1,\ldots,l - 1\}$, or if there exists a design – relevant woven error pattern (WEP) ϵ W_i, then put the unfavourable value π_l into A_l and return to line 1;
3. If $l=K\text{-}1$, then the interleaver construction is complete. Otherwise continue with the next design step $l \rightarrow l+1$;

2. ANALYSIS OF THE DESIGNED INTERVEAVER

The complexity of the combined algorithm presented in this work grows only linearly with K thanks to the utilization of the backtracking algorithm. [1,2]

We can expect that the combined algorithm does actually combine the advantages of the both algorithms, and that their deficiencies are eliminated. We suppose that the interleavers designed by the combined algorithm – the combined interleavers – have a larger δ_{min} than comparable s- random interleavers and lower multiplicities of low and medium – weight codewords than backtracking interleavers. A pleasant side effect is that the spreading introduced by the s – random part of the combined algorithm achieves a strong reduction of relevant WoC's with large h_{max}. In contrast to the pure backtracking design algorithm, choosing $h_{trunc,des}$ =4 should hence suffice to ensure $\delta_{min}= \delta_{min, targ}$ for combined interleavers. We have checked this assumption for a combined interleaver of length K=200, s=8, $\delta_{min, targ}$ =13 and $h_{trunc,des}$ =4. The distance spectrum measured with $h_{trunc,ana}$ =5 for weights $\delta \leq 14$ and $h_{trunc,ana}$ =4 for $\delta > 14$ is displayed in table 1. We find that all of our three assumptions are true: δ_{min} is significantly increased compared with the s – random interleaver, the multiplicities are lower than for the backtracking interleaver, and we have $\delta_{min}= \delta_{min, targ}$. Following the spectral thinning argumentation, for K>200 we have even more reason to believe that there are no combination of woven error patterns (WoC's) with h_{max} > 4 generating codewords of weight < $\delta_{min, targ}$. Thus, we will always use $h_{trunc,des}$ =4 in the combined algorithm.

Table 1. Measured coefficients A_δ of the truncated distance spectrum for combined interleaver

δ	h_{max}=2				h_{max}=3	h_{max}=4	h_{max}=5	A_δ
	I	II	III	IV				
13	0	0	0	0	47	0	3	50
14	50	111	0	0	7	37	0	205
15	0	0	0	0	260	11	-	271
16	37	286	0	0	120	205	-	648

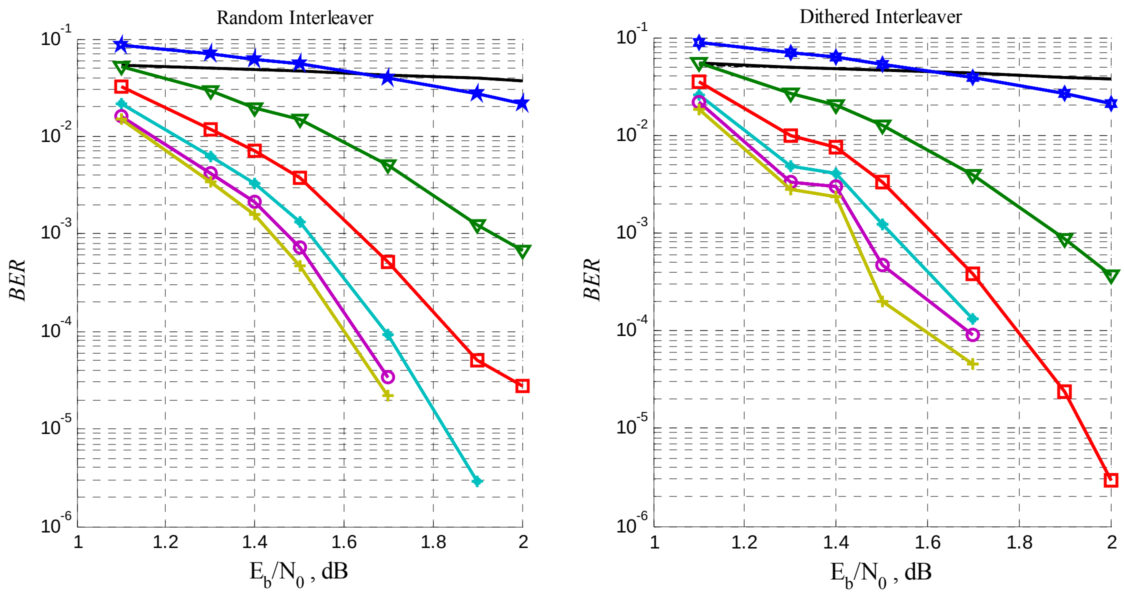

Fig.1
Simulated BER for turbo codes with code rate R=1/2 for random and dithered interleaver with six iteration

Of the large number of algorithms presented in vast literature on the interleaver design, we found the following four design algorithm to be of major interest: Dolinar's s – random design, Crozier's dithered golden interleavers, Khandani's algorithm and the proposed combined interleaver design algorithm [3]. In this work, we want to identify the advantages and deficiencies of these four algorithms by analyzing their truncated distance spectra and comparing them with each other.

Each of these algorithms has certain design goals, combating type I WEP's and/or increasing δ_{min} to a given value $\delta_{min, targ}$. Each algorithm is therefore best suited for specific interleaver lengths and component encoders, whereas it works less efficient for other lengths and encoders. In order to make a fair comparison, where each of the designs can show its strengths, we will therefore consider a large variety of Turbo code's parameters, i.e., interleaver length K, memory v of the component encoders and overall code rate R resulting from the puncturing. Since this works focuses primarily on low – complexity Turbo codes of v=2 and the interleaver design is of main importance for $K \in$ [200;1000]. We have chosen the following six configuration of parameters (K, v, R) for our testbed:

1. K≈200, v=2, R=1/2;
2. K≈500, v=2, R=1/2;
3. K≈1000, v=2, R=1/2;
4. K≈500, v=2, R=1/3;
5. K≈500, v=3, R=1/2;
6. K≈500, v=4, R=1/2;

Observe that Khandani's algorithm imposes restrictions on the applicable values of *K*. For all other algorithms, K can be chosen arbitrarily, such that 200, 500 и 1000 are valid values.

3. DESIGN PARAMETERS

The design parameters, which are required by the design algorithms, for the following four algorithms are:

***s*- random** – the spreading parameter *s*;

dithered golden – the constant *r*, as specified in [4], this value should be identical to the component scrambler's period length *p*. The remaining design parameters are identical for all designed interleavers: dither strength D_{dith} = 0.01 and constants m=1 and *j*=0. [4]

Khandani – the number L of classes (this value should be a multiple of p, note that [5] used the variable name K for this quantity, which collides with our variable for the interleaver size). All Khandani's interleavers were designed with a threshold *T*=14 for K≈200 and *T*=28 for K≈500 and K≈1000. The maximum number of iterations carried out in the optimization procedure was limited to 20. [5]

combined – the spreading parameter s and the target minimum distance $\delta_{min,\ targ}$. All combined interleavers were designed up to weight $h_{trunc,des}$ =4.

Most of the analyses were carried out with $h_{trunc,ana}$ =4 in order to limit the complexity. It is unlikely that WoC's with h_{max}=5 generate low – weight codewords for K>200. For K=200 (configuration A), we extended $h_{trunc,ana}$ to 5 for codewords of weight δ≤14, whereas for configuration E and F, we had to restrict $h_{trunc,ana}$ to 3 for higher weights δ.

Both, the dithered golden as well as the Khandani interleavers are based on spreading. However, as they emerge from a random construction and from an optimization process, respectively, their spreading parameter s is not determined before the interleaver's construction is complete. For these interleavers, we have measured the values displayed in table 2, where we also include the spreading parameters used in the construction of the s – random and the combined interleavers for comparison.

Table 2 Spreading parameters s measured in the dithered golden and the Khandani interleavers

Configuration	A	B, D	C	E	F
Dithered golden	5	10	18	11	15
Khandani	6	6	6	7	8
s – random	10	15	17	15	15
Combined	8	9, 11	10	11	11

Since all of the four considered design algorithms are at least partially based on spreading, and since the s – random interleaver is the first and the most simple proposition for a spreading interleaver, we will for each configuration use the s – random interleaver as a benchmark for the comparison.

A. The dithered golden interleaver is good at avoiding type I WEP's (up to δ=14), whereas it does not combat type II WEP's, whose multiplicity even grows for δ=12 compared with the s – random interleaver. The multiplicities grow also for WoC's with h_{max}=3 and 4, and we find that δ_{min} is caused by WoC with h_{max}=3. The details of the Khandani interleaver's distance spectrum resemble very much that of the s – random interleaver. The main difference is that the spreading of the former is optimized, such that all WoC's of type I generating codewords of weight 8 are avoided. As regards the combined interleaver, our expectations are met: the combined interleaver does indeed combine the advantages of the s – random and the backtracking design, i.e. δ_{min} is significantly increased – it is the highest of all four interleavers and the multiplicities of the low – weight codewords remain as for the s – random interleavers.

B. Comparing this s – random interleaver to that for configuration A, where the s – random interleaver for configuration C (K=1000) was compared with that for configuration A (K=200), because of the increased s, a higher δ_{min} is achieved, and thanks to the larger K and the spectral thinning effect, the importance of WoC's with , h_{max}>2 decreases. The results for the other three interleavers resemble those for configuration A. The dithered golden interleaver succeeds in combating WoC's of type I, but the number of WoC's of type II or with h_{max} =3 is much larger than for the s – random interleaver. For the Khandani interleaver, there is almost no improvement compared with the Khandani interleaver for configuration A. The multiplicities of all WoC's with h_{max} =2 are close to those of the s – random interleaver, but there is a major increase in the multiplicities of WoC's with h_{max} =3 and 4 compared with s – random interleaver. As for K=200, the combined interleaver obtains the highest δ_{min}, while the coefficients A_δ remain low. The A_δ are slightly increased in comparison with the s – random interleaver due to the lower value s used in the combined interleaver's design.

C. The comparison between the four interleavers yields similar results as for configuration B. Observe that there exists a very large number of type II WEP's with δ=12 for the dithered golden interleaver, and that it is these WEP's which determine δ_{min}.

D. Thanks to the lower code rate, the codeword weight associated with all kinds of WoC's is significantly increased compared with configuration B. Hence, all four interleavers obtain much higher minimum distances δ_{min}. Spreading becomes more efficient for low code rates, since most WoC's in spreading interleavers contain long EP's and generate hence large parity weights. Particularly when WoC's contains EP's of higher weights h=3,4,…, spreading causes a large associated codeword weight. This can be observed for all four interleavers, where the multiplicities of WoC's with h_{max} =3 and 4 are clearly lowered, such that these WoC's are only of minor importance. Observe that for the dithered golden, the Khandani and the combined interleaver, the minimum distance is determined by type II WEP's.

E. The dithered golden and the Khandani design demonstrate impressively their capability to combat type I WEP's. Yet, for both we see that the multiplicity of WoC's of type II and with h_{max} =3 and 4 is increased with respect to s – random interleaver. Indeed, it is a WoC with h_{max} =4, that determines δ_{min} for the Khandani interleaver, whereas it is WEP's of type II, that limit δ_{min} for the dithered golden interleaver. The combined interleaver obtains by far the highest δ_{min}.

F. By increasing the component encoder's memory to ν=4, we observe a similar effect as in configuration D, where the rate was lowered to R=1/3. The multiplicities of all low – weight codewords decrease strongly. The distance spectrum of the component encoder with ν=4 improves upon that for ν=2 as the free distance is higher and there are fewer EP's generating a low parity weight. Particularly, the minimum parity weight generated by weight – 2 EP's is clearly higher for ν=4 than for ν=2. One consequence is that all type IIWEP's generate codewords of weight $\delta \geq 20$, such that these WEP's play only a minor role in the truncated distance spectra for configuration F. For this configuration, the dithered golden interleaver exhibits a substantially improved distance spectrum with respect to the s – random interleaver: δ_{min} is clearly increased, even though the coefficients A_δ remain low. The distance spectrum of the Khandani interleaver agrees well with that of the s – random interleaver in most aspects, except that type I WoC's are effectively avoided. The Khandani interleaver represents a kind of "better" s – random interleaver.

4. ANALYSIS OF THE FOUR CONSIDERED INTERLEAVER DESIGN ALGORITHMS

Having analyzed the four considered interleaver design algorithms for six relevant configurations of the Turbo encoder, we can summarize the results:

4.1 S – random interleaver

These interleavers are very efficient in the thinning out the code's distance spectrum compared to random interleavers. Thus, the minimum distance is increased and the multiplicities of low and medium weight codewords are decreased. For growing K, s can be increased, such that more and more type I WEP's are avoided. The spreading also boots the spectral thinning effect. The number of relevant WoC's with h_{max} >2 decreases strongly for growing K. However s – random interleavers do not combat type II WEP's, such that the minimum distance is for almost all s – random interleavers not more than the minimum weight of the codewords associated with these WEP's (δ=12 for configurations A, B и C).

4.2 Dithered golden interleaver

WEP's of type I are even more efficiently avoided than in s – random interleavers, such that δ_{min} is slightly increased. On the other hand, the number of WoC's of type II or with h_{max} =3 is significantly increased for some configurations in comparison with s – random interleavers. The reason is the linear interleaver underlying the construction of dithered golden interleavers. Without dithering, the multiplicity of the type II WEP's would be in the order of K. Even the introduction of dithering does not represent a sufficient measure to completely avoid all type II WEP's. Therefore in most configurations type II WEP's generate the codewords at δ_{min} and these WEP's also limit the minimum distance, which is achievable with dithered golden interleavers (δ_{min} ≤12 for configuration A, B и C).

However, for powerful component encoders with larger memories ν>2, all type II WEP's generate relatively large codeword weights, weights δ≥20 in configuration F. For these Turbo codes, dithered golden interleavers represent a very powerful choice, since the codewords at δ_{min} are generated by WoC's other than WEP's of type II.

In our analysis we made a noteworthy observation: Owing to the underlying linear interleaver with its regular structure, for dithered golden interleavers, many low – weight codewords are generated by input words, which have a relatively large weight and which contain a large number of EP's.

4.3 Khandani's interleaver

Although the design algorithm differs strongly from the s – random design, the interleavers produced with Khandani's algorithm seem to have much in common with s – random interleavers. The achieved δ_{min} values differ only slightly. For component encoders with ν=2 (configurations A, B, C и D), the multiplicities of the four types of WoC's with h_{max}=2 are approximately the same as for the s – random interleaver. From this observation we recognize that Khandani's algorithm is mainly aimed at avoiding WoC's of type I and ignores those of Type II. For component encoders with ν=3 and 4, type I WEP's are reduced considerably compared with the s – random interleaver, whereas type II WEP's are not reduced. In all examples except configuration F, we find that the number of WoC's with h_{max} >2 is increased compared to s – random interleavers.

4.4 Combined interleaver

Of the four considered interleaver design algorithms, we found that the combined interleavers exhibit the best truncated distance spectrum. In all six configurations, δ_{min} is considerably increased with respect to the s – random interleaver. This is achieved by taking into account all kinds of WoC's in particular type II WEP's in the design algorithm. At the same time, this yields low multiplicities of the low and medium weight codewords due to the inherent spreading.

5. CONCLUSION

For comparison, in table 3 we have listed the free distances δ_{free}, which are achieved by by a Turbo code with the best of the designed interleavers and the free distance δ_{free} achieved by the best convolutional code, where the decoding complexity per information bit is similar for the convolutional and the Turbo code for each configuration. Here R and ν_{conv} are the rate and the memory of the convolutional code, respectively. The displayed free distances of the convolutional codes are taken from [6]. The complexity per informational bit for Viterbi decoding a convolutional code with binary input is defined as $C_{bit,\ conv}$=$2Z_{conv}$, where $Z_{conv} = 2^{\nu conv}$ is the number of trellis states of a convolutional code of memory ν_{conv}. [7]

Table 3. For each configuration, the table shows the minimum distance δ_{min}.

Configuration	A, B, C	D	E	F
R v_{conv}	1/2 7	1/3 7	1/2 8	1/2 9
δ_{free} convolutional code	10	16	12	12
δ_{min} Turbo code	13, 14, 16	24	18	20

For a fair comparison, we assume that the complexity of the Turbo decoder per decoded information bit is approximately $C_{bit} = 2 \cdot Z \cdot 2 \cdot 2 \cdot I$, where $Z = 2^{v}$ is the number of trellis states in the component descrambler and I is the number of iterations, which we assume to be 8 for comparison. The first factor 2 in the formula for C_{bit} is due to the existence of two trellis branches, that emerge from and merge into any trellis state. The second factor 2 stems from the fact that BCRJ algorithm, which is frequently used as the component descrambler, calculates the probabilities of the trellis states in a forward and backward recursion. The table shows that for all six considered configurations, the best Turbo code outperforms the best convolutional code with respect to the achieved minimum distance. This means that the asymptotic coding gain is higher for the Turbo code. [8] This is true already for short interleavers with K≈200, however the differences become even more explicit for larger K, even though the Turbo decoder's complexity per decoded information bit remains constant.

REFERENCES

[1] Breiling, M., Peeters, S., Huber, J., *Interleaver Design Using Backtracking and Spreading Methods*, Inter. Symp. on Inf. Theory, pp. 451, 2000

[2] Peeters, S., *Interleaver Design for Parallel Concatenated Convolutional Codes*. Master's thesis, Lehrstuhl für Nachrichtentechnik, Universität Erlangen – Nürberg, April 1999

[3] Hokfelt, J., Edfors, O., Maseng, T., *Interleaver design for turbo codes based on the performance of iterative decoding*. Intern. Conf. on Commun., pp. 93 – 97, 1999

[4] Crozier, S., Lodge, J., Hunt, A., *Performance of Turbo – Codes with Relative Prime and Golden Interleaving Strategies*, Intern. Mobile Satellite Conf., pp. 268 – 274, 1999

[5] Khandani, A., *Group Structure of Turbo – codes with Application to Interleaver Design*, Technical report, Dept. of Electr. and Comp. Eng., University of Waterloo, Canada, 1997

[6] Proakis, J., *Digital Communications*, McGraw Hill, 2nd edition, 1989

[7] Huber, J., *Grundlagen der Wahrscheinlichkeitsrechnung für iterative Decodierverfahren*. Submitted to Elektrotechnik und Informationstechnik, 2002

[8] Wicker, S., *Error control systems for digital communication and storage*, Prentice – Hall, 1st edition, 1995