Partially Systematic Rate 1/2 Turbo Codes

Ingmar Land and Peter Hoeher

Information and Coding Theory Lab
University of Kiel, Germany
{il,ph}@tf.uni-kiel.de

Abstract: *Partially systematic parallel concatenated (turbo) codes of rate 1/2 are considered. These are generated from a classical rate 1/3 turbo code by puncturing both the parity bits and the systematic bits. The so-called "error-floors" of the resulting codes are lower than those of their systematic counterparts. For some puncturing patterns, even the "waterfall region" is shifted to lower signal-to-noise ratios. Hence, the performance of rate 1/2 turbo codes can be enhanced by simply spreading the puncturing over the parity and the systematic bits without increasing encoding and decoding complexity.*

Keywords: Turbo codes, punctured codes, iterative decoding, weight distribution.

1. INTRODUCTION

The classical turbo encoder [1] consists of two recursive scramblers: the first one is directly fed with the info word, whereas the second one is fed with the interleaved info word. The turbo code word comprises the systematic part, which equals the info word, and the two parity parts, i.e., the parity words, which are the outputs of the two scramblers. In order to raise the code rate from 1/3 to 1/2, the two parity words are punctured.

To improve this type of rate 1/2 turbo code, research has been focused on two subjects: code design and interleaver design. A kind of boundary condition has always been that only the parity words are punctured, i.e., only systematic turbo codes have been considered.

When looking at optimal and suboptimal iterative decoding algorithms (IDA) [1] [2] [3], it becomes obvious that the code does not need to be systematic. In fact, the IDA can also be applied to a class of non-systematic concatenated codes, the so-called coupled codes [4]. Hence, there is a third, previously not regarded degree of freedom in turbo code design: the puncturing may be optimized with respect to the parity *and* the systematic bits.

Depending on the puncturing pattern, the resulting code may be systematic (none of the info bits are punctured), partially systematic (some of the info bits are punctured) or non-systematic (all of the info bits are punctured). The partially systematic turbo codes (PSTC) are discussed in this paper and they are compared to systematic (conventional) and non-systematic turbo codes.

The encoder and the decoder are described in the following two sections. Then, simulation results for the bit error rates and the word error rates are presented. The asymptotic behavior is analytically derived by means of the weight distribution in the last section.

2. ENCODER

The encoder of the PSTC is depicted in Figure 1. It consists of the interleaver (ILV), two scramblers (SCR1, SCR2) with the respective recursive generators $g_1(D)$ and $g_2(D)$, and the puncturer (PCT).

The code word $\boldsymbol{x} = (\boldsymbol{u}, \boldsymbol{p}_1, \boldsymbol{p}_2)$ comprises the info word $\boldsymbol{u}$, the parity word $\boldsymbol{p}_1$, which is the output of scrambler SCR1, and the parity word $\boldsymbol{p}_2$, which is the output of scrambler SCR2. The set of these words $\boldsymbol{x}$ build the "mother" turbo code of rate 1/3.

This code is punctured according to the $3 \times b$ puncturing matrix

$$B = \begin{bmatrix} \boldsymbol{b}_u \\ \boldsymbol{b}_{p1} \\ \boldsymbol{b}_{p2} \end{bmatrix}$$

with puncturing period b. The row vectors $\boldsymbol{b}_u$, $\boldsymbol{b}_{p1}$, and $\boldsymbol{b}_{p2}$ define the puncturing patterns of $\boldsymbol{u}$, $\boldsymbol{p}_1$, and $\boldsymbol{p}_2$, respectively.

The punctured code word $\boldsymbol{x}' = (\boldsymbol{u}', \boldsymbol{p}_1', \boldsymbol{p}_2')$ comprises the punctured info word $\boldsymbol{u}'$, the punctured parity word $\boldsymbol{p}_1'$ of scrambler SCR1, and the punctured parity word $\boldsymbol{p}_2'$ of scrambler SCR2. The set of these words $\boldsymbol{x}'$ build the PSTC of rate 1/2.

Let us define the permeability rate ρ of a word as the proportion of the bits that are *not* punctured. For example, $\rho = 1/4$ means that three of four bits are punctured and therefore only one out of four bits is transmitted. Then, a puncturing matrix B can be characterized by the permeability rate ρ_u of the info word and the permeability rate ρ_p of each of the parity words. The relation between the permeability rates and the PSTC rate R is given by

$$\frac{1}{R} = \rho_u + 2\rho_p$$

For the desired code rate $R = 1/2$ and a chosen info permeability rate ρ_u, the parity permeability rate ρ_p results in

$$\rho_p = 1 - \rho_u/2 \ .$$

Some valid combinations are listed in Table 1.

Using these definitions, a non-systematic turbo code is given by $\rho_u = 0$, a systematic turbo code by $\rho_u = 1$
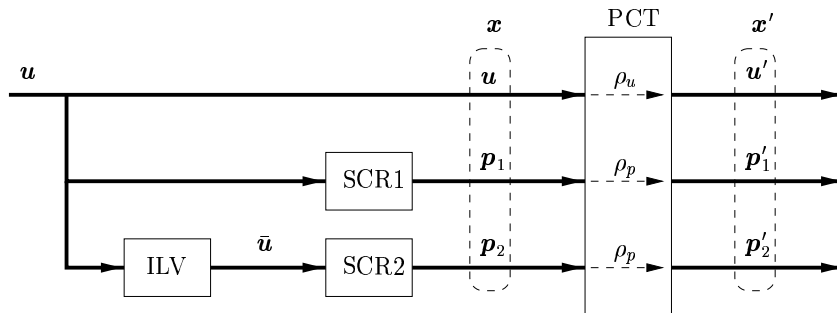
Figure 1: Encoder of the PSTC with the info permeability rate ρ_u and the parity permeability rate ρ_p.

(which is the conventional turbo code), and a PSTC by $0 < \rho_u < 1$.

3. DECODER

For decoding the PSTC, the usual method for decoding punctured codes is applied. Firstly, the channel measurements are de-punctured, i.e., the channel log-likelihood ratios of the punctured bits are set to zero, or equivalently, the channel probabilities of these bits are set to 1/2. Then, the "mother" code, i.e. the unpunctured rate 1/3 turbo code, is decoded by the well-known IDA [1] [3].

The intention of turbo decoding is to improve the reliability of the info bits with each iteration step. For a systematic turbo code, the initial reliabilities depend only on the channel quality. For a PSTC, however, they also depend on the info permeability rate ρ_u: the smaller ρ_u is, the lower the mean initial reliability will be. This might be an advantage of a systematic turbo code over a PSTC and will be analyzed later on.

4. SIMULATED ERROR RATES

To evaluate the performance of PSTCs, the coded transmission with binary phase shift keying over an additive white Gaussian noise channel was simulated.

The PSTCs are derived from a "mother" turbo code of info word length $K = 16384$ comprising two memory 2 scramblers with the generators

$$g(D) = \frac{1 + D^2}{1 + D + D^2} \quad .$$

To abstract from interleaver design, a random interleaver is used and a new interleaving pattern is generated for each info word, i.e., the so-called uniform-interleaving [5] is applied. The "mother" code is punctured in accordance with the puncturing patterns of Table 1. The resulting PSTCs are of rate $1/2$, what can easily be verified by the puncturing matrices. The PSTCs are decoded with 20 iterations.

In Figure 2 and Figure 3, the resulting bit error rates (BER) and word error rates (WER) are plotted versus the signal-to-noise ratio (SNR) E_b/N_0, where E_b denotes

B								ρ_u	ρ_p
$\begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$								1	1/2
$\begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$								3/4	5/8
$\begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$								1/2	3/4
$\begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$								1/4	7/8
$\begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$								0	1

Table 1: Puncturing matrices B for rate 1/2 PSTCs and the respective permeability rates ρ_u and ρ_p.

the energy per info bit and N_0 the one-sided power density of the noise.

Three facts can be observed:

1. The error-floors of both the BER and the WER decrease with decreasing ρ_u, except for $\rho_u = 0$.

2. For slight puncturing of the systematic word (high ρ_u), the waterfall region is shifted to lower SNR, whereas for stronger puncturing (low ρ_u), it is shifted to higher SNR with respect to the systematic code ($\rho_u = 1$).

3. The PSTC with $\rho_u = 0$ exhibits very poor performance.

Observation 1 can be explained as follows: For high SNR, the error-floor is due to the case where an input weight $w = 2$ causes output weights $h = d_{f,eff} = 4$ at both of the scramblers before puncturing, where $d_{f,eff}$ is the effective free distance of the given generator [5]. To achieve a high code word weight after puncturing, it is advantageous to put more puncturing to the systematic word than to the parity words, since then, it is likely that less of the overall output weight will be removed. The
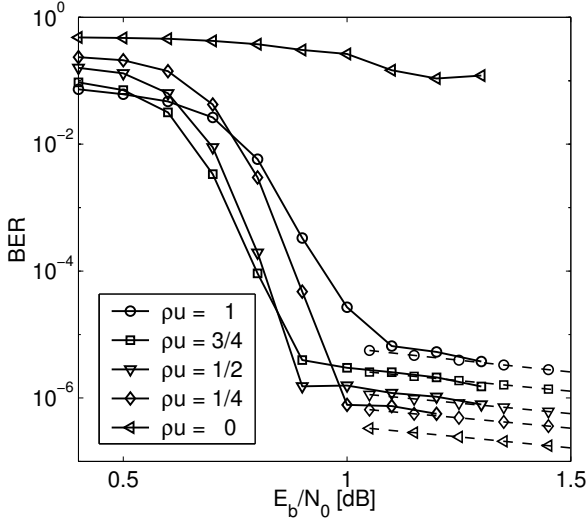
Figure 2: BER of rate $1/2$ PSTCs (info word length $K = 16384$, memory 2 scramblers) with various info permeability rates ρ_u, and approximated error-floors (dashed). The classical turbo code corresponds to $\rho_u = 1$.

Figure 3: WER of rate $1/2$ PSTC (info word length $K = 16384$, memory 2 scramblers) with various info permeability rates ρ_u, and approximated error-floors (dashed). The classical turbo code corresponds to $\rho_u = 1$.

higher code word weight will lead to a lower error-floor. This "practical" explanation will be proved analytically in the next section.

Observation 2 is due to two effects: (a) Spreading the puncturing over the systematic and the parity bits improves the code word distances (see next section). Hence, also the error rates in the waterfall region will be smaller. (b) The IDA needs a kind of seed to start, which is given by the reliabilities of the non-punctured systematic bits. The smaller the info permeability rate is, the worse works the IDA (compare to the last remark in the previous section). Due to these two contradictory effects, there is a trade-off between the (potential) performance of the code and the (effective) performance of the IDA.

Observation 3 can be explained as follows: For the case $\rho_u = 0$, the IDA starts with the decoding of parity bits produced by the first scrambler. Since this "code" contains only the trellis termination as redundancy, and since there is no pre-decoding information about the info bits available ($\rho_u = 0$), the decoding results after the first half-iteration will be almost random-like. In the second half-iteration, the situation is even worse: the "code" generated by the second scrambler also contains only the termination bits as redundancy. Additionally, the decoder is fed with pre-decoding information about the info bits computed in the first half-iteration, which is likely to be wrong. Therefore, the decoding result after the first (full) iteration step is likely to be wrong, too. Since the two component decoders will provide each other with distorted pre-decoding information during the IDA, the decoding of non-systematic turbo codes will fail most of the time.
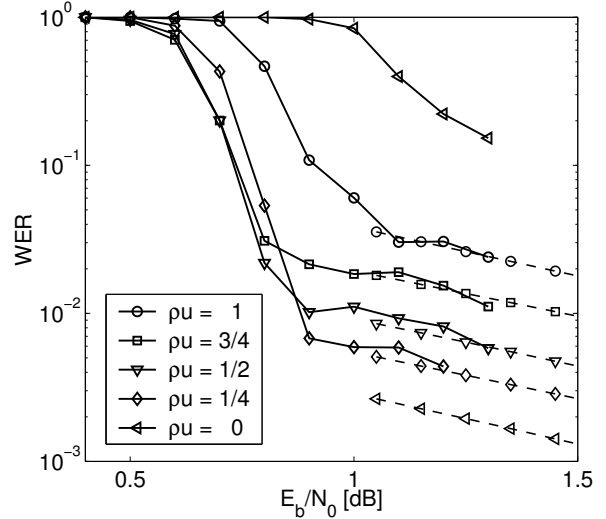
5. WEIGHT DISTRIBUTIONS AND APPROXIMATED ERROR-FLOORS

The weight distribution of a code allows to evaluate the code performance for ML decoding; especially for turbo codes, the error-floors can be approximated [6]. In the following, the weight distribution and the approximated error-floors of the PSTCs discussed in the previous section will be determined.

To calculate the coefficients $A_{w,h}$ of the input-output weight enumerating function (IOWEF) of a PSTC with uniform interleaver, a method similar to [5] was applied. Let denote w the weight of the info word $\boldsymbol{u}$, w' the weight of the punctured info word $\boldsymbol{u}'$, h_1 and h_2
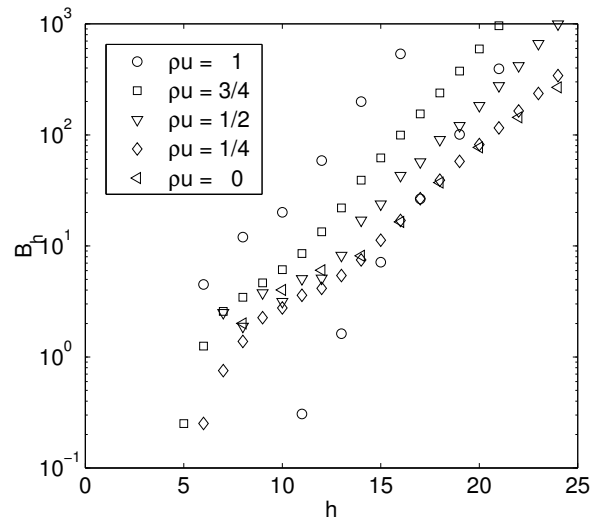

Figure 4: Coefficients B_h of rate $1/2$ PSTCs (info word length $K = 16384$, memory 2 scramblers) with various info permeability rates ρ_u.

the weight of the punctured parity words $\boldsymbol{p}_1'$ and $\boldsymbol{p}_2'$, and h the weight of the PSTC word $\boldsymbol{x}'$; let further denote $A^{\text{SCR1}}_{w,w'+h_1,n_1}$ the number of "compound" error events of scrambler 1 with input weight w and output weights w' and h_1, comprising n_1 concatenated "single" error events, and let denote $A^{\text{SCR2}}_{w,h_2,n_2}$ the number of "compound" error events of scrambler 2 with input weight w and output weight h_2, comprising n_2 concatenated "single" error events. Then [5],

$$A_{w,h} \approx \sum_{\substack{w'+h_1 \\ +h_2=h}} \sum_{n_1} \sum_{n_2} \frac{\binom{K}{n_1}\binom{K}{n_2}}{\binom{K}{w}} A^{\text{SCR1}}_{w,w'+h_1,n_1} A^{\text{SCR2}}_{w,h_2,n_2} \ .$$

With the coefficients

$$B_h = \sum_w A_{w,h} \ , \ \ D_h = \sum_w \frac{w}{K} A_{w,h} \ ,$$

the union-bound for the BER P_b and for the WER P_w is given by[1]

$$P_b \ \leq \ \sum_{h>0} D_h \cdot Q\left(\sqrt{2\,h\,R\,E_b/N_0}\right) \ ,$$

$$P_w \ \leq \ \sum_{h>0} B_h \cdot Q\left(\sqrt{2\,h\,R\,E_b/N_0}\right) \ ,$$

where code rate $R = 1/2$. If this bound is evaluated only for low code word weights, it represents a good approximation of the error-floor [6].

The coefficients B_h and D_h of the codes discussed in the previous section were computed for small code word weights h. The values of B_h, which represent the weight distribution, are depicted in Figure 4.

For $\rho_u = 1$, the coefficients B_h are very small for odd weights h; for even h they are much bigger than those for $\rho_u = 3/4$. Thus, the PSTC with $\rho_u = 3/4$ will probably outperform the systematic turbo code with $\rho_u = 1$, at least in the error-floor region. For $\rho_u = 3/4, 1/2, 1/4$, the coefficients B_h are decreasing with decreasing ρ_u for all weights h; therefore, puncturing of the systematic bits improves the PSTC. Note that this is valid for both the error-floor and the waterfall region. The B_h of the PSTC with $\rho_u = 1/4$ and those of the non-systematic turbo code ($\rho_u = 0$) are comparable for even weights h, but for odd h, $B_h = 0$ for the latter code. Thus, the non-systematic turbo code is the strongest one of these five codes.

These analytical results perfectly match to the simulations (see Figure 3) in the error-floor region (the poor performance of the non-systematic turbo code was already explained). In the waterfall region, however, the performance increases only for ρ_u up to about $1/2$. Since the codes have shown to become stronger even for $\rho_u < 1/2$, the observed performance degradation must be due to the suboptimal IDA.

An analogous analysis of the coefficients D_h leads to similar results for the BER.

The error-floors of the codes were approximated as explained above. They are depicted in Figure 2 and Figure 3. These approximated error-floors agree with the simulations and support the analytical results.

6. CONCLUSIONS

A new method was proposed to generate rate $1/2$ turbo codes by puncturing both the systematic and the parity bits of a rate $1/3$ "mother" turbo code, resulting in the partially systematic turbo codes (PSTC). For memory 2 component codes, the BER and the WER were simulated and the weight distribution and the approximated error-floors were computed. The PSTCs outperform the conventional systematic turbo codes in the error-floor region and, for an appropriately chosen info permeability rate, even in the waterfall region without increasing encoding and decoding complexity. It was proven by analysis of the weight distribution (a) that the PSTCs become the stronger, the more systematic bits are punctured; and (b) that the performance degradation in the waterfall region for strong puncturing of the systematic bits is due to the suboptimal iterative decoding algorithm.

REFERENCES

[1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon Limit Error-Correcting Coding and Decoding: Turbo-Codes," *Proc. IEEE ICC*, pp. 1064–1070, May 1993.

[2] J. Lodge, R. Young, P. Hoeher, and J. Hagenauer, "Separable 'MAP Filters' for the Decoding of Product and Concatenated Codes," *Proc. IEEE ICC*, pp. 1740–1745, May 1993.

[3] P. Robertson, P. Hoeher, and E. Villebrun, "Optimal and Sub-Optimal Maximum a Posteriori Algorithms Suitable for Turbo Decoding," *ETT*, pp. 119–125, Mar./Apr. 1997.

[4] S. Chaoui and U. Sorger, "Code Coupling," *Proc. Int. Symp. on Turbo Codes*, September 2000.

[5] S. Benedetto and G. Montorsi, "Unveiling Turbo Codes: Some Results on Parallel Concatenated Coding Schemes," *IEEE Trans. Inform. Theory*, vol. 42, no. 2, pp. 409–428, March 1996.

[6] O.Y. Takeshita, M.P.C. Fossorier, and D.J. Costello, "A New Technique for Computing the Weight Spectrum of Turbo-Codes," *IEEE Trans. Inform. Theory*, vol. 3, no. 8, pp. 251–253, August 1999.

---

[1] $Q(x) = 1/\sqrt{2\pi} \int_x^\infty \exp(-t^2/2)dt.$