THE EFFECT OF PUNCTURING IN TURBO ENCODERS

Mats Öberg and Paul H. Siegel, Center for Wireless Communications, ECE
University of California, San Diego, La Jolla, California 92033-0407
E-mail: moberg@ucsd.edu
psiegel@ucsd.edu
Andrei Vityaev, Department of Mathematics
University of Washington, Seattle, WA 98195
E-mail: vityaev@math.washington.edu

Abstract—We study via simulation the effect of different combinations of three puncturing patterns and three classes of interleavers on the performance of turbo-coded systems. The interleaver types are conventional pseudo-random, odd-even pseudo-random and modified S-random. The impact of the unequal parity-bit distribution induced by puncturing is evaluated and shown to be reflected in individual bit-positional bit-error-rates.

I. INTRODUCTION

When turbo codes first were introduced by Berrou et al. [1] in 1993, puncturing was utilised to increase the rate of the code. Since then, puncturing has been used as a default procedure, except in analytical studies where puncturing has been avoided in order to simplify the analysis. Barbulescu [2] investigated the effect of puncturing by looking into the effect of using the conventional alternating-phase puncturing with an odd-even interleaver. However, a great deal remains to be learned about the interplay between puncturing techniques and interleavers and their combined effect on turbo code performance. Analysis of this interaction is difficult; however, simulations can provide useful insights and suggest fruitful directions for theoretical investigations. The purpose of this paper, then, is to present empirical results that shed light on the problem of selecting interleavers and puncturing patterns to optimise turbo-code performance.

II. INTERLEAVERS AND PUNCTURING RULES

The simulation results reported herein pertain to the effect of puncturing patterns on overall average bit-error-rate (BER) and average bit-positional BER.

In all of the simulations, the turbo code incorporated an interleaver \( \pi \) of length \( N = 10000 \), two recursive systematic convolutional (RSC) encoders with octal generator polynomials (37,21), and four-bit tail sequences to terminate the encoders. The decoding algorithm we used was the log-MAP algorithm as described by Pietrobon and Barbulescu [3]. The interleavers fall into the three categories below.

Conventional pseudo-random interleavers are generated by sequentially mapping bit-positions in the first dimension into pseudo-randomly selected bit-positions in the second dimension. If a selected bit-position in the second dimension has been previously selected, another target bit-position is pseudo-randomly selected.

Odd-even pseudo-random interleavers [2] are pseudo-random interleavers generated with the additional constraint that odd (resp. even) bit-positions are mapped into odd (resp. even) bit-positions.

Modified S-random interleavers incorporate a refined version of the S-random permutations defined by Divsalar, and Pollara [4]. In addition to prohibiting the mapping of a bit-position to another within a distance \( \pm S \) of a bit-position already chosen in any of the \( S \) previous selections, we have also prohibited the mapping of any pair of bit-positions separated by an integral number (\( \leq 5 \)) of constraint lengths of the RSC encoder to another such pair.

We remark that the odd-even interleaver allows us to apply alternating-phase puncturing without leaving any information bit without a corresponding parity bit. Also, the results in [5] show that the modified S-random interleaver will likely increase the minimum distance of the turbo code.

The unpunctured turbo encoder has rate \( R = 1/3 \). We denote the unpunctured output of the turbo encoder by \( s_0, p_0, q_0, s_1, p_1, q_1, s_2, p_2, q_2, \ldots \), where the superscript refers to the dimension to which the bit belongs, and the subscript refers to the time-index of the codewords. Here, \( s_i \) denotes the
<table>
<thead>
<tr>
<th>Puncturing pattern</th>
<th>Interleaver type</th>
<th>Odd-even</th>
<th>Pseudo-Random</th>
</tr>
</thead>
<tbody>
<tr>
<td>( V_A )</td>
<td></td>
<td>10000 P1</td>
<td>2500 P0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5000 P1</td>
<td>2500 P0</td>
</tr>
<tr>
<td>( V_g )</td>
<td></td>
<td>5000 P0</td>
<td>2500 P0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5000 P1</td>
<td>2500 P1</td>
</tr>
</tbody>
</table>

**TABLE I**

**Bit-class distribution: exact for odd-even, average for others**

ith information bit, \( p_l \) denotes the \( i \)th parity bit generated by the first encoder, and \( p_{l}^{'} \) denotes the \( i \)th parity bit generated by the second encoder, corresponding to information bit \( x_{n-l}(q) \).

We define two puncturing patterns with period 6 as follows. One punctures the two dimensions in alternating-phase, as described by the puncturing vector \( V_A = [110101] \). The other punctures the two dimensions in the same-phase, as described by the puncturing vector \( V_g = [111100] \). For a given interleaver and puncturing pattern, the information bits are classified into three bit-classes, P0, P1 and P2, where the numerical index indicates the number of unpunctured parity bits corresponding to each information bit. For the class of odd-even interleavers, Table I gives the number of information bits in each group, according to the puncturing pattern. For the class of conventional pseudo-random and modified S-random interleavers, the table gives the expected number of information bits in each group, again according to the puncturing pattern. In an effort to modify the bit-class distribution for the conventional pseudo-random and modified S-random interleavers, we defined a third, interleaver-dependent puncturing pattern, which we call a biasing pattern. The puncturing vector for the biasing pattern is time-dependent and defined as follows: \( V_g = [11k(2j)10k(2j)] \), where \( k(2j) = n^{-1}(2j)(\text{mod} 2) \)

\[
\frac{k(2j)}{N} = 1 - k(2j) \quad j = 0, 1, \ldots, N - 1
\]

This pattern increased the expected number of bits in class P1 to 7500, and reduced P0 and P2 to 1250 each.

### III. INTERLEAVER PERFORMANCE

In this section, we discuss the simulation results pertaining to the different combinations of interleavers and puncturing rules, at low SNR (0.7 dB) as well as moderately high SNR (1 dB and 2 dB).

Tables II and III give the simulated BER for SNR=2 dB and SNR=1 dB, respectively. The modified S-random interleaver with alternating-phase puncturing provides the lowest BER among the tested combinations for both SNR values. This phenomenon could be a manifestation of the increased free-distance achieved by the elimination of minimum-weight codewords [5,6].

The results also indicate that for the odd-even interleaver, the alternating-phase puncturing pattern \( V_A \) provides lower BER than the same-phase pattern \( V_g \). This follows from the fact, shown in [5], that minimum-distance events correspond to the existence of bit-position pairs separated by one or two constraint lengths in the first dimension that are mapped into bit-position pairs separated by one or two constraint lengths in the second dimension. We denote such mapping types by \( (1 \rightarrow 1) \), \( (1 \rightarrow 2) \), \( (2 \rightarrow 1) \), and \( (2 \rightarrow 2) \). If the constraint length \( L \) of the constituent encoder is odd, as in our case, an odd-even interleaver cannot cause a mapping of type \( (1 \rightarrow 2) \) or \( (2 \rightarrow 1) \). Depending on the puncturing pattern, the mapping \( (2 \rightarrow 2) \) can generate codewords of weight 6, 8 or 10. Specifically, with alternating-phase puncturing, the mapping \( (2 \rightarrow 2) \) will always generate a weight-8 codeword, with information-weight 2, parity-weight 2 in one dimension and parity-weight 4 in the other. On the other hand, with same-phase puncturing, the codeword will either have parity-weight 2 in each dimension or parity-weight 4 in each dimension, yielding weight 6 or 10. Thus, with same-phase puncturing, the multiplicity of minimum-weight codewords may be larger than with alternating-phase puncturing, causing an increase in the overall BER at moderate-to-high SNRs.

Fig. 1 shows the results of simulations at SNR=0.7 dB. The odd-even interleaver with the same-phase puncturing pattern has the best performance. We also evaluated a pure pseudo-random interleaver with the biasing pattern in an attempt to better understand the effects of the distribution of information bits among the parity-bit classes. This combination performed much worse than any of the other schemes at SNR=0.7 dB, with BER 4.6×10^-3, almost a factor of 2 higher than the measured BER for the alternating-phase pattern. A full explanation for the observed relative performances of the combinations of interleavers and puncturing patterns remains to be found.
TABLE II
Simulation results for SNR 2 dB

<table>
<thead>
<tr>
<th>Interleaver</th>
<th>Puncturing</th>
<th>BER / $10^{-7}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pseudo-rand.</td>
<td>Every oth.</td>
<td>16, 16, 17, 23</td>
</tr>
<tr>
<td>Pseudo-rand.</td>
<td>Pattern</td>
<td>13, 15</td>
</tr>
<tr>
<td>Constrained</td>
<td>Every oth.</td>
<td>3.5, 3.7, 4.1</td>
</tr>
<tr>
<td>Odd-even</td>
<td>Same</td>
<td>14, 14, 19, 25</td>
</tr>
<tr>
<td>Odd-even</td>
<td>Every oth.</td>
<td>11, 12, 17</td>
</tr>
</tbody>
</table>

TABLE III
Simulation results for SNR 1 dB

<table>
<thead>
<tr>
<th>Interleaver</th>
<th>Puncturing</th>
<th>BER / $10^{-7}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pseudo-rand.</td>
<td>Every oth.</td>
<td>61, 82, 99</td>
</tr>
<tr>
<td>Pseudo-rand.</td>
<td>Pattern</td>
<td>109 114</td>
</tr>
<tr>
<td>Constrained</td>
<td>Every oth.</td>
<td>49 55</td>
</tr>
<tr>
<td>Odd-even</td>
<td>Same</td>
<td>121 125 147</td>
</tr>
<tr>
<td>Odd-even</td>
<td>Every oth.</td>
<td>80 85 98</td>
</tr>
</tbody>
</table>

Fig. 1. Histograms of BER

Fig. 2. Bit error rate for different bit classes SNR=0.7 dB

Fig. 3. Bit error rate for different bit classes, SNR=0.7 dB

Fig. 4. Correlation of bit errors

IV. BIT-POSITIONAL ERROR-RATES

When the individual bit positions in a frame are considered, there are at least two factors that influence the probability of making an erroneous decoding decision. On one hand, the bit positions with a larger number of unpunctured parity bits may benefit from “extra protection”. On the other hand, due to the effects of the interleaver and the iterative decoding procedure, the decoding decision for any particular bit depends on all received samples.

Fig. 2 was derived from simulation results for a conventional pseudo-random interleaver with alternating-phase puncturing at SNR=0.7 dB. The plot shows the average BER for information bit positions in each of the three bit classes P0, P1, and P2. For odd-even interleavers with alternating-phase puncturing pattern and SNR=0.7 dB, the upper panel of Fig. 3 shows the average BER for the only bit-class P1. The bit-class P1 is in this case divided into two groups, according to whether the correspond-
ing parity bit belongs to the first or second dimension. Results for same-phase puncturing are shown in the lower panel of Fig. 3 for the only occurring bit-classes P0 and P2.

It is apparent from the figures that bit positions with more unpunctured parity bits achieve lower BER. This phenomenon has also been observed in ordinary punctured convolutional codes, where it can be explained by analyzing the effect of puncturing on the distance spectrum and the transfer function bound. This correlation between bit-class and BER has also been observed at the higher SNRs.

Regarding the second factor mentioned above, Fig. 4 illustrates the correlation among the bit-positional error-rates in the information frame for a conventional pseudo-random interleaver with alternating-phase puncturing at $\text{SNR} = 0.7 \text{ dB}$. The plot shows the conditional probability $P(\text{error at bit position } k + l | \text{error at position } k)$ versus $l$. The probability of making an error at bit position $k + l$, given that we have made an error at bit position $k$, decreases very slowly with $l$. Even for $l = 150$ the conditional probability is approximately 0.08 while the overall BER is 0.0017.

In order to produce the plot, 1221 frames of length 10000 were decoded in 18 iterations, yielding $2 \times 10^4$ bit errors distributed over 150 error frames. A conventional pseudo-random interleaver with alternating-phase puncturing was used, and the SNR was 0.7 dB.

Both factors mentioned above will influence the performance of the decoding scheme with a particular choice of puncturing and interleaving. For example, in the previous section it was observed that at $\text{SNR}=0.7 \text{ dB}$, the odd-even interleaver with same-phase puncturing performed better than odd-even interleavers with alternating-phase puncturing and the pseudo-random interleavers.

As our simulation results show in Figs. 2 and 3, the information bits with two corresponding parity bits transmitted enjoy lower probability of decoding error than bits with one or zero parity bits transmitted. In particular, when an odd-even interleaver is combined with same-phase puncturing pattern, every second bit has two parity bits transmitted, ensuring higher probability of correct decoding for that bit. Fig. 3 suggests that this increases the probability of correct decision for the adjacent bits having no corresponding parity bits.

V. Conclusions

In this paper we have simulated the performance of turbo coding schemes using combinations of three types of interleavers (conventional pseudo-random, odd-even pseudo-random and modified S-random) with three different puncturing patterns (alternating-phase, same-phase and biasing).

We have found that at moderate SNR (1 and 2 dB) the modified S-random interleaver with alternating-phase puncturing performed the best. At the same time, at low SNR (0.7 dB) the odd-even pseudo-random interleaver combined with same-phase puncturing performed the best, followed closely by the conventional pseudo-random interleaver with alternating-phase puncturing.

To provide more insight into the measured results, we investigated the probability of error for the individual bit positions in the frame. In particular, we studied the dependence of these probabilities on the extent of puncturing of the parity bits corresponding to the bit positions and on the decoding errors of nearby positions.

ACKNOWLEDGMENT

The authors would like to thank Dr. Jack K. Wolf for helpful discussions.

REFERENCES


