Performance evaluation of parallel concatenated codes

Sergio Benedetto, Guido Montorsi
Dipartimento di Elettronica, Politecnico di Torino
C.so Duca degli Abruzzi 24, 10129 Torino, Italy
Ph. +39-11-5644031 - FAX +39-11-5644099

Abstract -- A parallel concatenated coding scheme consists of two simple systematic constituent encoders linked by an interleaver. The input bits to the first encoder are scrambled by the interleaver before entering the second encoder. The codeword of the parallel concatenated code consists of the input bits followed by the parity check bits of both encoders. In this paper we propose a method to evaluate the bit error probability of a parallel concatenated coding scheme in a way which is independent from the interleaver used. The two cases of parallel concatenated block codes and parallel concatenated convolutional codes are considered.

I. INTRODUCTION

The appearance of the so-called "turbo-codes" [1;2;3;4] has risen a large interest in the coding community toward coding schemes obtained by parallel concatenation of simple codes. The parallel codes so obtained promise to achieve considerable coding gains and yet admit iterative soft-decoding schemes whose complexity is not significantly higher than that of the decoder for the single constituent codes.

The main ingredients of a parallel concatenated code (PCC) are:

- two constituent codes (CC), normally very simple and easy to decode, which can be the same code. Actually any number of constituent codes could be considered: in this paper we will limit ourselves to two.
- An interleaver placed between the two encoders of the constituent codes.
- An efficient iterative algorithm performing the soft decoding of the parallel concatenated code.

Since the successful proposal of turbo-codes, neither serious attempts toward a theoretical explanation of the codes behavior/performance nor a sufficient comprehension of the role and relative importance of the aforementioned ingredients have appeared so far.

What is known in terms of performance is essentially due to measurement on a VLSI chip [5] and simulation [6] which, in itself, is not at all a simple task.

In this paper, we try to shed some light on the theoretical comprehension of PCC's. In particular, the main result of our analysis consists in a method to define and evaluate the average performance of a PCC, starting from the performance of the CC's.

Owing to its definition, the average performance, expressed in terms of bit error probability, turns out to be independent from the particular interleaver used, and helps in assessing what can be gained with given CC's and with an interleaver of a given length (or delay). Apart from its theoretical relevance, this result may be of some importance when designing the interleaver length in the presence of delay-constrained systems.

Nothing is said on how to find a good interleaver; simply, our results guarantee that there exists at least one interleaver yielding the average performance, and that, having found it, the search can be stopped.

We will describe the analytical tools in connection with parallel concatenated block code, and then extend the analysis to the case of parallel concatenated convolutional codes.

II. PARALLEL CONCATENATED BLOCK CODES

Given an \((n,k)\) systematic block code \(C\), its weight enumerating function (WEF) is defined as

\[
B^C(H) = \sum_{i=0}^{n} B_i H^i
\]

where \(B_i\) is the (integer) number of codewords with Hamming weight \((\text{number of ones})\) \(i\) and \(H\) is a dummy variable.

The WEF of a code can be used to compute the exact expression of the probability of undetected errors and upper bounds to the word error probability.

The input-redundancy weight enumerating function (IR-WEF) of the code is defined as:

\[
A^C(W,Z) = \sum_{i,j} A_{ij} W^i Z^j
\]

The IR-WEF splits each term of the WEF into the separate contributions of the information and of the parity-check bits to the total Hamming weight of each codeword, and thus represents a more detailed description of the weight profile of the code. It will prove crucial in the following, since the two input words to the constituent encoders, the second being obtained by interleaving the first, share the same Hamming weight. Notice that the IR-WEF is a characteristic of the encoder, joining input words and codewords, whereas the WEF only depends on codewords.

Consider now the conditional weight enumerating function \(A^C_w(Z)\) of the parity check bits generated by the code \(C\) in...
second encoders, respectively, the weight of the corresponding codeword of $C_P$ will be $w + z_1 + z_2$.

We want now to obtain the IRWEF $A_{CP}^c(W, Z)$ of $C_P$ starting from the knowledge of those of the constituent codes. Clearly, the result will depend on the interleaver used in the concatenated scheme, and this prevents from obtaining general results.

To overcome this difficulty, we introduce an abstract interleaver called uniform interleaver, defined as follows.

**Definition 1**

A uniform interleaver of length $k$ is a probabilistic device which maps a given input word of weight $w$ into all distinct $(W/Z)$ permutations of it with equal probability $1/k$.

From the definition, it is apparent that the conditional weight enumerating function $A_w^C(Z)$ of the second code is independent from that of the first code thanks to the uniform randomization produced by the interleaver.

As a nice consequence of this, we can easily evaluate the conditional weight enumerating function of the PCBC which uses the uniform interleaver as the product, suitably normalized, of the two conditional weight enumerating functions of the constituent codes:

$$A_{CP}^c(Z) = A_1^c(Z) \cdot A_2^c(Z).$$

Also, from (1) we obtain the IRWEF of the code $C_p$ as:

$$A_{CP}^c(W, Z) = \sum_{w=1}^{k} W^w A_{CP}^c(Z).$$

So far, we have shown that introduction of the uniform interleaver permits an easy derivation of the weight enumerating functions of the PCBC. However, in practice, one is confronted with deterministic interleavers, which give rise to one particular permutation of the input bits. So, what is the significance of previous definitions and equations?

To answer this question, we state without proof (for a proof, see [7]) the main property of a PCBC which uses the uniform interleaver.

Let $A_{CP}^c(W, Z)$ be the IRWEF of the code $C_{P_k}$ obtained using the particular interleaver $I_k$. We first prove that:

$$E_k[A_{CP}^c(W, Z)] = A_{CP}^c(W, Z),$$

where $E_k$ means expectation with respect to the whole class of interleavers.

This, in turn, guarantees that the performance obtained with the uniform interleaver are achievable by at least one deterministic interleaver.

So far, we have used an interleaver with length $k$ equal to that of the input word of the block code. It is straightforward to extend all previous results to the more general case...
where we use as basic codeword for a PCBC $l$ consecutive codewords of the component code code $C_1$. In particular, the IRWEF of the new component $(l n_1, l k)$ code is given by:

$$A^{C_1}(W, Z) = [A^{C_1}(W, Z)]^l.$$  

(8)

The conditional weight enumerating function of the new component code can still be obtained from the IRWEF through:

$$A^{C_1}_w(Z) = \left. \frac{1}{w!} \frac{\partial A^{C_1}(W, Z)}{\partial W} \right|_{W=0}$$  

(9)

From the conditional weight enumerating functions of the two new constituent codes, owing to the property of the uniform interleaver of length $l k$, we obtain the conditional weight enumerating function of the $(l(n_1 + n_2 - k), l k)$ PCBC as:

$$A^{C_1}_w(Z) = \left. \frac{A^{C_1}_w(Z)}{l k} \right|_{W=0}.$$  

(10)

From this point on, the performance of the new PCBC can be obtained as before through (3).

As an example, taking as component code the Hamming code $(7, 4)$, we can build different PCBC of increasing complexity by simply grouping $l$ consecutive codewords.

Applying the upper bound (3) in its widely known tighter version [8], we obtain the results reported in Fig. 2, where the bit error probabilities for the considered code and various interleaver lengths $l = 1, 2, 10, 20, 100$ are plotted versus the signal-to-noise ratio $E_b/N_0$. For comparison, also the curve of the constituent code is reported.

The figure shows that a gain of 2 dB can be achieved increasing $l$, and this does not require a significant additional computational effort if iterative soft-decoding procedures are applied. The only price which has to be paid is an increased delay. Also, we notice from the results that the beneficial effect of increasing the interleaver length tend to reduce as long as $l$ becomes very large.

III. PARALLEL CONCATENATED CONVOLUTIONAL CODES

The first applications of parallel concatenated coding schemes used as constituent codes convolutional codes. They are known in the coding community by the nickname of turbo codes, and the main reason for their successful implementation resides in the availability of efficient algorithms for soft iterative decoding [1;2;3]. In our context, we will call them parallel concatenated convolutional codes (PCCC). A block diagram showing how they work is presented in Fig. 3.

The behavior is similar to those of a PCBC, the main difference being the fact that now the interleaver of length $N$ does not contain an integer number of input words, since the input sequences are infinite in length.

In this section, we will extend to PCCC the analysis already applied to PCBC, and approximate their average performance by suitably defining an equivalent block code and applying to it the previous procedure. This approach is justified by the fact that, when the length $N$ of the interleaver that links the two convolutional encoders is much greater than their constraint lengths, the performance of the PCCC becomes very close to that of the PCBC obtained by truncating the trellises of the convolutional codes after $N$ steps.

We will consider an equivalent block code whose trellis representation is the truncation at step $N$ of the trellis of the convolutional code, and whose codewords lead the trellis into the identity state at step $N$. Our goal is to derive the IRWEF of such a block code, starting from the knowledge of a suitably defined error events enumerating function of the convolutional code. In the following, we will describe an algorithmic approach that allows the evaluation of the most significant terms of the IRWEF.

Let us consider Fig. 4. By our previous hypotheses on the
block code which approximates the constituent convolutional code, any codeword belonging to the block code is obtainable by combining set of error events of the convolutional code with suitable sequences of "0" so that the total length equals \( N \).

As an example, a single error event of length \( l \) smaller than \( N \) produces all codewords with \( N-l \) zeroes positioned before and after the error event. All these codewords share the same input and redundancy weight, so that they can be grouped together. The multiplicity \( K[l,1] \) of the block codewords originated from this single error event in the IRWEF of the block code equals the number of partitions of \( N-1 \) into two numbers:

\[
K[l,1] = \binom{N-l+1}{1} = N-l+1.
\]

Proceeding this way, one can obtain the general expression for the multiplicity of codewords originated by a single combination of \( n \) error events with total length \( l \):

\[
K[l,n] = \binom{N-l+n}{n}.
\]

Let \( T^C(W,Z,L,\Omega) \) be the transfer function of the convolutional code which enumerates all paths in the trellis leaving the zero-state at step 1, and remerging into the zero-state at or before step \( N \), with possible remergences into the zero-state at other steps in between, with the condition that, after remerging, they leave the zero-state immediately at the successive step:

\[
T^C(W,Z,L,\Omega) = \sum_{i,j,m,n} T_{i,j,m,n} W^i Z^j L^m \Omega^n
\]

where \( T_{i,j,m,n} \) is the number of paths in the trellis originated by an input sequence of weight \( i \), with weight of the redundant bits equal to \( j \), length \( m \) and \( n \) remergences with the zero-state (and hence concatenating \( n \) error events). Examples of these concatenations of single error events were shown in Fig. 4.

As for the case of block codes, we define the conditional transfer function \( T^C_w(Z,L,\Omega) \) as

\[
T^C_w(Z,L,\Omega) = \sum_{j,m,n} T_{w,j,m,n} Z^j L^m \Omega^n
\]

Passing now to the codewords of the equivalent block code, we notice that each path of length \( m \) and number of remergences \( n \) belonging to \( T^C_w(Z,L,\Omega) \) gives rise to \( K[m,n] \) codewords with the same input and redundancy weights, so that the conditional IRWEF \( A^C_w(Z) \) of the equivalent block code can be obtained as

\[
A^C_w(Z) = \sum_j A_{w,j} Z^j
\]

with

\[
A_{w,j} = \sum_{m,n} K[m,n] T_{w,j,m,n}.
\]

An efficient algorithm able to compute the most significant terms, i.e. those sequences with Hamming weights smaller than a given threshold, of the transfer function \( T^C(W,Z,L,\Omega) \) has been implemented, elaborating on the algorithm described in [8] to evaluate the transfer function of a convolutional code, and then yielding as output the conditional IRWEF of the equivalent block code.

Having obtained the two conditional IRWEF's of the constituent convolutional codes, we can proceed as before for concatenated block codes under the hypothesis of using a uniform interleaver by plugging their expressions into (5) and (1) to obtain an approximation to the IRWEF of the PCCC, and, from it, the error probability performance.

As an example, consider a PCCC employing as constituent codes the same recursive systematic convolutional code with constraint length 3 and free distance 5 whose encoder structure is shown in Fig. 5. We have constructed different PCCC through interleavers of various lengths, and passed through the steps aforedescribed to evaluate their performance.

In Table 1 the coefficient \( D_m \) for the evaluation of the bit error probability of or the resulting PCCC for \( N = 100,1000,10000 \) are reported. The effect of longer interleavers is clearly apparent from the table. Namely, the multiplicity of the terms which dominate the performance (those with lower Hamming weight) decrease when \( N \) increases, approximately as \( 1/N \). As to the free distance, it has increased from 5 to 8.

In Fig. 6 we present the bit error probabilities of different PCCC employing the same CC with interleavers of different lengths and, for comparison, the normalized curve of the CC. As anticipated when commenting Table 1, there is a decrease by a factor of 10 in the bit error probability for an increasing by the same factor of the interleaver length. Gains up to 4 dB's are achievable.

In the previous derivation, an approximation consisted in neglecting the sequences of the constituent convolutional
Table 1. Coefficients $D_m$ for the evaluation of the bit error probability of the PCCC obtained from the 4-state recursive convolutional codes and interleavers with lengths 100,1000,10000.

<table>
<thead>
<tr>
<th>Hamming distance</th>
<th>100</th>
<th>1000</th>
<th>10000</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>3.8900E-02</td>
<td>3.9881E-03</td>
<td>3.9988E-04</td>
</tr>
<tr>
<td>9</td>
<td>7.6590E-02</td>
<td>7.9965E-03</td>
<td>7.9960E-04</td>
</tr>
<tr>
<td>10</td>
<td>0.1136</td>
<td>1.1918E-02</td>
<td>1.1991E-03</td>
</tr>
<tr>
<td>11</td>
<td>0.1508</td>
<td>1.5861E-02</td>
<td>1.5985E-03</td>
</tr>
<tr>
<td>12</td>
<td>0.1986</td>
<td>1.9887E-02</td>
<td>1.9987E-03</td>
</tr>
<tr>
<td>13</td>
<td>0.2756</td>
<td>2.4188E-02</td>
<td>2.4017E-03</td>
</tr>
<tr>
<td>14</td>
<td>0.4079</td>
<td>2.9048E-02</td>
<td>2.8102E-03</td>
</tr>
<tr>
<td>15</td>
<td>0.6292</td>
<td>3.4846E-02</td>
<td>3.2281E-03</td>
</tr>
<tr>
<td>16</td>
<td>1.197</td>
<td>6.5708E-02</td>
<td>6.0575E-03</td>
</tr>
<tr>
<td>17</td>
<td>2.359</td>
<td>1.1457</td>
<td>1.3697E-02</td>
</tr>
<tr>
<td>18</td>
<td>4.383</td>
<td>0.2984</td>
<td>2.8543E-02</td>
</tr>
<tr>
<td>19</td>
<td>7.599</td>
<td>0.5472</td>
<td>5.2989E-02</td>
</tr>
<tr>
<td>20</td>
<td>12.58</td>
<td>0.9171</td>
<td>8.9441E-02</td>
</tr>
<tr>
<td>21</td>
<td>20.46</td>
<td>1.437</td>
<td>0.1403</td>
</tr>
<tr>
<td>22</td>
<td>33.31</td>
<td>2.144</td>
<td>0.2082</td>
</tr>
<tr>
<td>23</td>
<td>54.65</td>
<td>3.090</td>
<td>0.2957</td>
</tr>
<tr>
<td>24</td>
<td>91.23</td>
<td>4.465</td>
<td>0.4177</td>
</tr>
<tr>
<td>25</td>
<td>154.9</td>
<td>6.716</td>
<td>0.6133</td>
</tr>
<tr>
<td>26</td>
<td>265.5</td>
<td>10.67</td>
<td>0.9577</td>
</tr>
<tr>
<td>27</td>
<td>455.6</td>
<td>17.65</td>
<td>1.574</td>
</tr>
<tr>
<td>28</td>
<td>779.0</td>
<td>29.61</td>
<td>2.646</td>
</tr>
<tr>
<td>29</td>
<td>1327</td>
<td>49.31</td>
<td>4.450</td>
</tr>
<tr>
<td>30</td>
<td>2257</td>
<td>80.57</td>
<td>7.267</td>
</tr>
<tr>
<td>31</td>
<td>3842</td>
<td>128.6</td>
<td>11.60</td>
</tr>
<tr>
<td>32</td>
<td>6556</td>
<td>201.3</td>
<td>18.64</td>
</tr>
<tr>
<td>33</td>
<td>11221</td>
<td>311.5</td>
<td>27.57</td>
</tr>
<tr>
<td>34</td>
<td>19261</td>
<td>481.2</td>
<td>41.88</td>
</tr>
<tr>
<td>35</td>
<td>33143</td>
<td>748.8</td>
<td>63.94</td>
</tr>
</tbody>
</table>

Fig. 6. Average upper bounds to the bit error probability for a PCCC obtained linking two identical 4-state recursive convolutional encoder with interleaving lengths 100,1000,10000.

IV. Conclusions

We have proposed a method to evaluate the bit error probability of a parallel concatenated coding scheme independently from the interleaver used. Crucial was the introduction of a probabilistic interleaver called uniform interleaver which permits an easy derivation of the weight enumerating function of the parallel concatenated code starting from the weight enumerating function of the constituent codes. The two cases of parallel concatenated block codes and parallel concatenated convolutional codes were considered.

REFERENCES