Selection of Input Frames in Designing Interleavers for TURBO Codes

K. Koora and H. Betzinger
Dresden University of Technology, Communications Laboratory, 01062 Dresden, Germany
e-mail: koora@ifn.et.tu-dresden.de

Abstract

In this paper we present two methods to use all possible inputs in designing interleavers for TURBO codes for large frames. The first method deals with the input frames of weight 2, where as the second method concentrates on all possible frames.

Introduction

The correction capacity of TURBO codes is dependent on different parameters like no. of iterations [1], selection of polynomials [2], selection of interleavers [3] etc.. In general, the convolutional codes with greater minimum distance '$d_{\min}$' leads to better correction. Since TURBO codes also belongs to this group, the increase in $d_{\min}$ leads to better correction of errors. One way to increase it is by selecting an optimum interleaver. In the next section we show a way to speed up the searching processes with the input frames of weight 2 and also discuss the achieved simulation results. The section following to it deals with a method of using all possible inputs of a given frame length for an interleaver search.

1 Selection of interleavers using input frames of weight 2

In [3] an algorithm is shown to select an optimum interleaver where the input frame weight 'd_{in}' is restricted to 2. If this method is applied to find an interleaver of size *N*=448, with *N* as frame length, then total no. of inputs is equal to 100128. On the other hand, it has to find an optimum from 448! possible interleavers. Here we describe an efficient method of search. For convenience, we represent 'd_i' as the weight of the redundancy part at the output of the *i*th encoder after puncturing. A desired final weight '$d'_{\min}$' at the output of the TURBO coder is given as a start parameter. At first, a weight table for all input frames is generated, where the first and second bit positions represent the *y*- and *x*-axis of the table respectively. The weights 'd_1' of the output blocks after coding once and puncturing are enlisted in their respective positions (see fig. 1 for an example). The starting state of the coder is set to a known state, e.g. '0' state as shown in [2]. Before tests are conducted to select an optimum random interleaver, all input frames of weight 2 which leads to $(d_1 + d_{in}) > d'_{\min}$ are discarded, as the interleaver is not having any influence on this d_1 and they however give rise to outputs with greater weights, i.e. $d_1 + d_2 + d_{in} > d'_{\min}$. This discarding leads to enormous time saving factor. It made easy to test 6049 inputs of frame length 448 instead of 100128 for a given polynomial. As the positions of the two bits of the frames are of interest, they are grouped together as one tupel. Now those tupels are put together to form a tupel table whose first bit position is same (refer to fig. 2). To start the search, either an arbitrary random interleaver or a given interleaver is considered, which is to be improved with respect to the final output weight $d_{\min}(= d_1 + d_2 + d_{in})$. Taking first tupel table, an exchange partner for the first bit position is chosen. Here one has to take into account whether this exchange has already been tested or it is allowed. With the help of the weight and tupel tables, the resulted distance of the so formed interleaver pattern is computed. Only those parts of the tupel table are effected whose bit position is grouped in the table and the rest are unchanged. If the result $d_{\min} > d'_{\min}$, the new change in the interleaver is made, otherwise this exchange position is discarded. This process is done until all tupel tables are checked with the slowly modified interleaver. Finally, the resulted interleaver is saved as a new random interleaver, if its total weight is greater than $d'_{\min}$. The given minimum weight is then changed to the newly computed weight, i.e. $d'_{\min} = d_{\min}$. To exit the interleaver search one can either use time or maximum $d_{\min}$.

To test this new algorithm, simulations were carried out with TURBO-Block-Codes using 2 iterations at SNR=3.75dB. The block length was fixed to 448, code rate to ½ and the memory of the RSC encoder to 3. A randomly selected interleaver was given as a starting parameter. With this systematic search for the interleavers, we were able to find 18 good random interleavers out of 10 million patterns which resulted in $d_{\min} = 23$. It was seen that the newly found optimum random interleaver resulted in BER of 4.37e-6 where as the given interleaver pattern only 1.26e-5. We observed a coding gain of 0.25dB at SNR=3.75dB. It is further notified that not all interleavers which resulted in a greater $d_{\min}$ have good correction capacity. We suspect that not only the consideration of all input frames of size N but also the distribution of the redundancy [4] are important factors in design of interleavers. Thus, the selection of interleavers using input frames of weight 2 not only leads to good but also to a bad one with respect to the BER.

2 Use of all possible input frames

Actually, minimum distance for block-codes is calculated using all possible input frames. For small *N* one can generate all possible inputs, e.g. if *N*=21 then there are $(2^{21}-1)$, i.e. 2097151 possible inputs excluding the all zero. By *N*=448 it is difficult to use all the frames. To solve this problem we made use of binary tree structure of the input frames as illustrated in figure 3. This selection algorithm of input frames depends only on the first time encoding and puncturing, i.e. d_1, where the interleaver doesn't have any influence. If and only if the resulted weight is smaller than the desired weight $(d_1 + d_{in} < d'_{\min})$, the path of the input frame is followed. The search of the input frames starts with *N*=1, i.e. with 2 possibilities '0' & '1'. After that it goes to the next node where there are 4 possible blocks '00', '01', '10' and '11'. As soon as the weight at a given node is greater than the desired weight it is discarded. In this way all the input frames which leads to an output weight less than the desired quality are taken into consideration to design an optimum interleaver.

3 Conclusions

In this paper we have shown two new methods to select the input frames for design of an optimum interleaver for TURBO codes. The first method is restricted only to the inputs of weight 2. Making use of pre computed weight tables and tupel tables one can fasten this search. The simulation results show that this method doesn't lead always to optimum interleavers because all the possible input blocks are not considered. The second method deals with the search of all possible input frames which can be used to design an optimum interleaver.

References

[1] C. Berrou, A. Glavieux, P. Thitimajshima, "*Near Shannon Limit Error Correcting Coding and Decoding: TURBO-Codes*", Pro. of IEEE Int. Conf. on Communications (ICC) 93, 23-24 May, Geneva, pp. 1064-1070

[2] K. Koora and A. Finger, "*A New Scheme to Terminate all Trellis of TURBO-Decoder for Variable Block Length*", in Proc. of the Int. Symposium on Turbo Codes and Related Topics, Ecole Nationale Supérieure des Télécommunications de Bretagne, France, 3-5 Sep. 1997, pp. 174 - 179

[3] P. Jung and M. Naßhan, "*Performance evaluation of turbo codes for short frame transmission systems*", Electronics Letters, Vol. 30, No. 2, 20th January 1994, pp. 111-112

[4] A. S. Barbulescu and S. S. Pietrobon, "*Interleaver design for turbo codes*", Electronics Letters, Vol. 30, No. 25, 8th December 1994, pp. 2107-2108

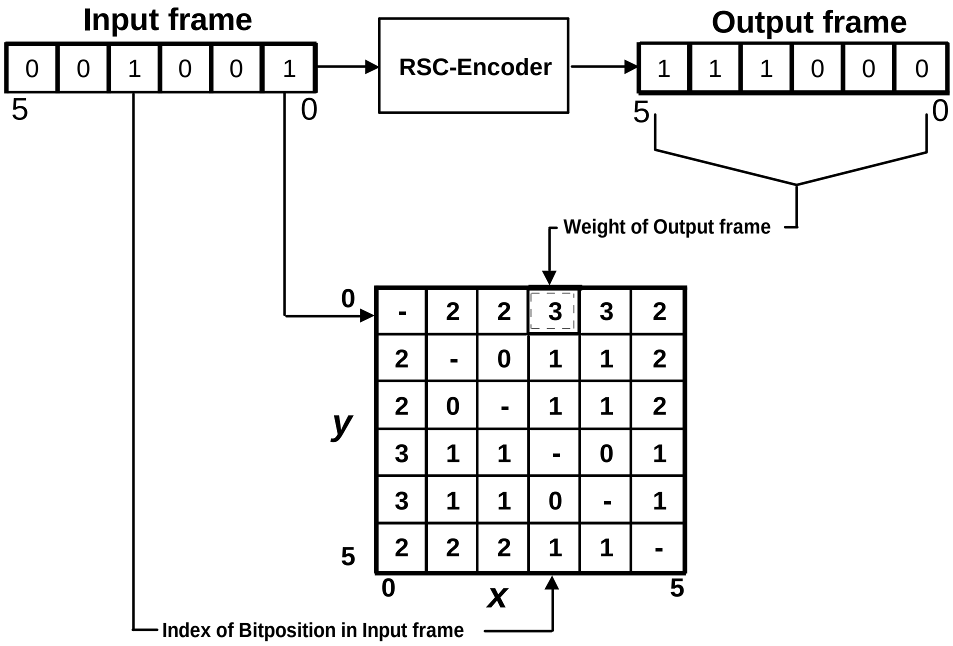

Figure 1: *Example for weight table preparation with frame length N=5.*

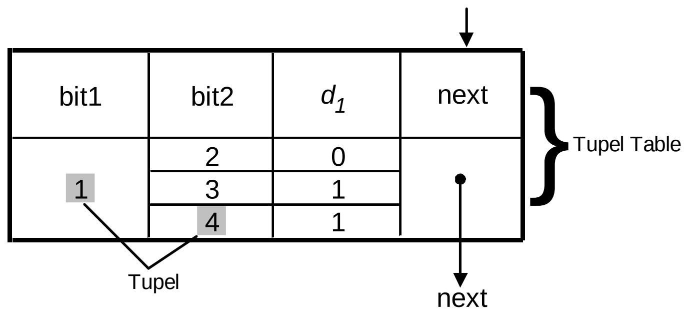

Figure 2: *Example of Tupel Table without the weight of the Input frame for* $d'_{min}=1$

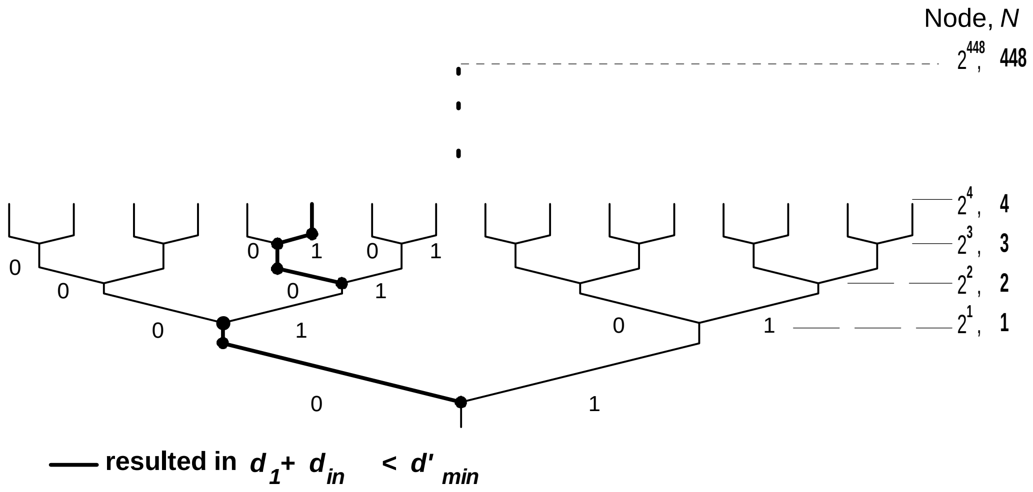

Figure 3: *Binary tree structure of search for input frames which leads to* $d_1 + d_{in} < d'_{\min}$.