8th International Conference on DEVELOPMENT AND APPLICATION SYSTEMS
Suceava, Romania, May 25 – 27, 2006

PROPOSAL OF A COST FUNCTION FOR THE ITERATIVE GROWTH OF INTERLEAVER'S LENGTH

Lucian TRIFINA[1], Horia BALTA[2]
"Gheorghe Asachi" Technical University of Iasi
Bd. Carol I, no. 11, Iasi, Romania
[1]luciant@zeta.etc.tuiasi.ro
Technical University of Timisoara
Bd. V. Parvan, 1900 Timisoara, Romania

Abstract. *In this paper a new cost function for iterative growth algorithm of interleaver's length for turbo codes is proposed, that approximates better the bit error rate. With this cost function the performances of turbo codes which use the resulted interleaver are improved. The paid price for this is the bigger complexity in obtaining the interleaver, although when computing resources are sufficient this problem is of less importance.*
Keywords: *interleaver, turbo codes, iterative algorithms, optimization.*

Introduction

The turbo codes represent a very good error correcting codes class in the small signal to noise ratios domain. In the initial variant these codes result by parallel concatenation of two systematic recursive convolutional codes with an interleaver.
The interleaver is an essential component of the turbo codes. With a good interleaver design the performance can approaches the Shannon limit capacity [1].
The bit error rate corresponding to turbo coding-decoding system decreases inverse proportionally with the length of interleaver [2]. On the other side, the bigger is the length of interleaver, the greater is the decoding delay. There are applications that need small decoding delays, as in the case of real time speech transmission [3].
For big lengths of interleavers a good behavior is achieved with the random ones [4]. For small lengths, however, a proper design of the interleaver must be done.
In [5] a search algorithm for an optimal interleaver matched to a certain code, which is efficient for small lengths, is presented.
The interleaver delay is computed based on the fact from the theory of groups that any permutation on a set of elements can be written as a product of disjoint cycles [6]. Hence, if the permutation π is written as product of cycles:

$$\pi=\prod_j c_j(\pi), \quad (1)$$

then the permutation delay is:

$$d(\pi)=\max_j (d(c_j(\pi))). \quad (2)$$

If the interleaver is considered as a finite-state permuter (a queue where transpositions are effectuated), and its representation is made by the transposition vector, then the permutation delay is the maximum element of the corresponding transposition vector, minus one [5].
The iterative growth of the interleaver length is performed using the transposition vector and is based on the representative error patterns set for the component code of the turbo code and on the minimization of a certain cost function related to the bit error rate. The used algorithm supposes next steps for a single error pattern and its error phases [5]:

Set the index which corresponds to the size of the optimized interleaver to L1. Initialize the recursion by exhaustively solving for:

$$T_{L_1}^* = \arg\min_{T_{L_1}} C\left(T_{L_1}\right), \quad (3)$$

for manageable L_1. In (3) C(T_{L_1}) represents the cost corresponding to the transposition vector T_{L_1}, of length L_1.

2) Find k_{L_1} from:

$$k_{L_1} = \arg\min_k \left\{ C_e\left(k, T^*_{L_1}\right) + C_{ne}\left(k, T^*_{L_1}\right) \right\} \qquad (4)$$

and form $T^*_{L_1+1} = \left(k_{L_1}, T^*_{L_1}\right)$. In (4) $C_e(k, T^*_{L_1})$ represents the cost corresponding to the previous transposition vector at which the value k is added for the error phases affected by the first transposition and $C_{ne}(k, T^*_{L_1})$ represents the cost of the same transposition vector at which the value k is added for the error phases not affected by the first transposition.

3) Set $L_1=L_1+1$ and iterate Step 2) to grow the transposition vector to the desired size.

The extension to the multiple error phases' case is obvious, consisting in summing of the cost function over all error patterns and all their phases. Considering the error patterns set E^1, E^2, ..., E^M, of lengths $L_E{}^1$, $L_E{}^2$, ..., $L_E{}^M$ and all their phases, then the next cost function associated to transposition vector T_{L+1}, for an interleaver of length (L+1), as below:

$$\hat{C}\left(T_{L+1}\right) = \sum_{j=1}^{M} \left(\sum_{n=1}^{L-L_{E^j}+2} f\left(T_{L+1}, E^j_{n,(L+1)}\right) + \sum_{n=1}^{L-L_{E^j}+2} f\left(T^{-1}_{L+1}, E^j_{n,(L+1)}\right) \right), \qquad (5)$$

because there are L-$L_E{}^j$+2 error phases for the error pattern E^j.

Suppose we wish to grow an interleaver of length L. Then we assume the data is framed, and the frame length is a multiple of the interleaver length, for example pL. All error patterns phases of length pL will be rows of a matrix Q. At iteration j, when the length of interleaver is L_1+j, we consider splitting the Q matrix into an integral number of subblocks of length L_1+j, and then generating the cost function for the overall matrix Q.

The cost function for interleaver growth includes the effect of both forward and inverse permutation since in its formulation we must take into account the both permutations due to the turbo code symmetry.

The design procedure is quite sensitive to this cost function, especially in the initial phases when the interleaver length is short.

In [5], to simplify the evaluation of the total cost, we consider truncating its expression by keeping only the most significant terms in the expression. A first cost function with 2I terms is:

$$\hat{C}_1 = \hat{C}_{1f} + \hat{C}_{1r} = \frac{1}{L}\left\{ \sum_{i=1}^{I} w_i^f e^{-d_i^{tf} R_c E_b/N_0} + \sum_{j=1}^{I} w_j^r e^{-d_j^{tr} R_c E_b/N_0} \right\}, \qquad (6)$$

where $\hat{C}_{1f}$ and $\hat{C}_{1r}$ are the functions associated with the forward and inverse permutation, respectively. For forward permutation w_i^f is the input weight of the ith least distant codeword in the search set from the all zero codeword, and d_i^{tf} is the total parity distance (associated with both constituent recursive convolutional codes of the turbo code). Similarly, for the cost function of the inverse permutation, w_j^r and d_j^{tr}, are defined respectively.

The second parametric cost function with 2I terms is:

$$\hat{C}_2 = \hat{C}_{2f} + \hat{C}_{2r} = \frac{1}{L}\left\{ \sum_{i=1}^{I} m_i^f e^{-d_i^{f} R_c E_b/N_0} + \sum_{j=1}^{I} m_j^r e^{-d_j^{r} R_c E_b/N_0} \right\}, \qquad (7)$$

where m_i^f, m_j^r are the multiplicities of the ith least distant codeword in the search set from the all zero codeword, at the output of only one of the component code (the one receiving the forward permuted sequence), and d_i^f, d_j^r are the corresponding distances of this codeword for the cost function $\hat{C}_{2f}$, associated with the

forward permutation and the cost function $\hat{C}_{2r}$, associated with inverse permutation, respectively.

The proposal of a new cost function

The proposed cost function takes into account both the input sequence weight and the multiplicity of the overall codeword resulted at the turbo code output, corresponding to the search set (i.e. the representative error patterns set for the constituent convolutional code). This cost function has the expression:

$$\hat{C}_3 = \hat{C}_{3f} + \hat{C}_{3r} = \\ = \frac{1}{L}\left\{ \sum_{w^f=1}^{W_{th}} w^f \cdot \sum_{d^f=d^f_{\min}}^{3L} B_{w^f,d^f} \cdot e^{-d^f R_c E_b/N_0} + \right. \\ \left. + \sum_{w^r=1}^{W_{th}} w^r \cdot \sum_{d^r=d^r_{\min}}^{3L} B_{w^r,d^r} \cdot e^{-d^r R_c E_b/N_0} \right\}, \quad (8)$$

where W_{th} is the maximum weight of the error patterns from the search set, and B_{w^f,d^f} is the number of codewords with weight d^f, resulted from the information sequences of weight w^f, associated with the forward permutation and, similarly, B_{w^r,d^r} is the number of codewords with weight d^r resulted from the information sequences of weight w^r, associated with the inverse permutation, both resulted only from the search set of the error patterns. The cost function $\hat{C}_3$ can be written in the equivalent form:

$$\hat{C}_3 = \hat{C}_{3f} + \hat{C}_{3r} = \frac{1}{L}\left\{ \sum_{d^f=d^f_{\min}}^{3L} B_{d^f} \cdot e^{-d^f R_c E_b/N_0} + \right. \\ \left. + \sum_{d^r=d^r_{\min}}^{3L} B_{d^r} \cdot e^{-d^r R_c E_b/N_0} \right\}, \quad (9)$$

where B_d^f and B_d^r, respectively, represent the error coefficients restrained to the search set and are given by the next relations:

$$B_{d^f} = \sum_{w^f=1}^{W_{th}} w^f \cdot B_{w^f,d^f}, \quad (10)$$

$$B_{d^r} = \sum_{w^r=1}^{W_{th}} w^r \cdot B_{w^r,d^r}. \quad (11)$$

With this cost function the results are improved since the function better approximates the bit error rate expression, and the interleaver is formed by the minimization of this cost function. These improvements are reflected in a smaller bit and frame error rate at big signal to noise ratios, where the interleaver makes the difference between the performances of various turbo codes. Obviously this interleaver cannot be used for „on the fly" applications that necessitate a fast generation of the interleaver at each transmitted data frame.

Case study

In order to compare the experimental results obtained with the cost function proposed in this paper with the results from the field literature we consider the tested code from [5]. The turbo code used in simulations is the one with *G*=[1, 5/7] generator matrix of constituent convolutional code. The trellis of the superior component code is closed, and the trellis of the inferior component code (the one that takes the interleaved sequence) is open.

The error patterns set used in the interleaver iterative growth algorithm is the one from [5], and the length of transposition vector determined for each cost function was grown up to 100. For $\hat{C}_1$ and $\hat{C}_2$ cost functions, all the terms resulted from the error patterns set were considered for a valid comparison with the proposed $\hat{C}_3$ cost function, although this leads to an increase of the simulation time. In computing the cost functions, the number of columns from *Q* matrix was considered, in the initial phase, equal with the least integer multiple of the length of the interleaver so that the number of partitions of the matrix to result as an integer for each error pattern from the search set. Afterwards the number of columns was considered equal with the length of the interleaver at current iteration.

For the above mentioned error pattern set W_{th}=5, and M=16.

For the simulations emphasizing the bit error rate (BER) and the frame error rate (FER) performances, obtained with the proposed cost function in comparison with $\hat{C}_1$ and $\hat{C}_2$ cost functions, the frame length is equal with the length of interleaver and they are made for

The modulation used in simulation is BPSK and the turbo decoding algorithm is MAP [7], in BCJR log-MAP variant [8], with twelve iterations. The BER and FER curves in SNR$\in$[2.5 dB; 3.2dB] domain are given in the figure 1 and 2, respectively. BER and FER values from the above mentioned domain are given in the tables 1 and 2, respectively.

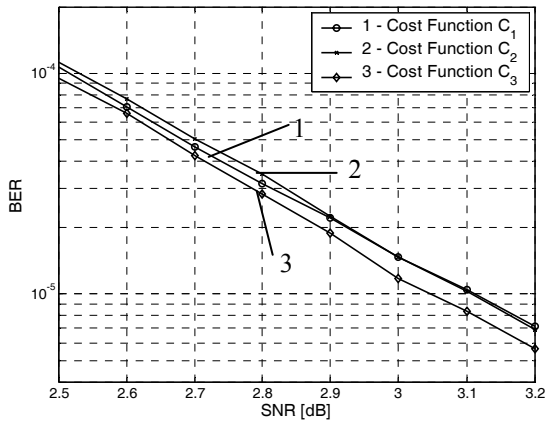

Figure 1. BER curves for the three cost functions at high SNR

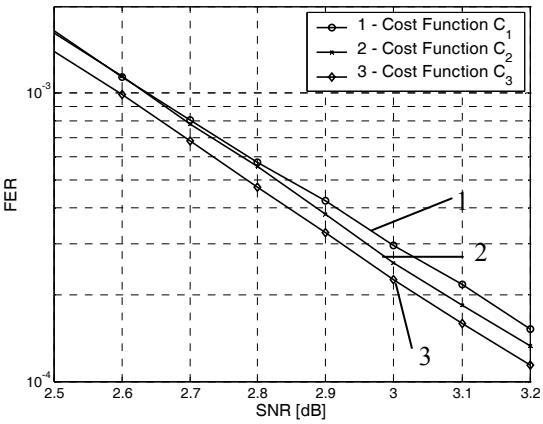

Figure 2. FER curves for the three cost functions at high SNR

AWGN channel at 0.1 dB resolution.

Table 1. BER×10^5 values for the three cost functions at high SNR

SNR [dB]	**2.5**	**2.6**	**2.7**	**2.8**	**2.9**	**3.0**	**3.1**	**3.2**
$\hat{C}_1$	10,648	7,009	4,649	3,165	2,210	1,471	1,048	0,713
$\hat{C}_2$	11,193	7,618	5,044	3,492	2,258	1,467	1,030	0,692
$\hat{C}_3$	9,488	6,559	4,237	2,837	1,886	1,176	0,837	0,565

Table 2. FER×10^4 values for the three cost functions at high SNR

SNR [dB]	**2.5**	**2.6**	**2.7**	**2.8**	**2.9**	**3.0**	**3.1**	**3.2**
$\hat{C}_1$	16,45	11,37	8,05	5,75	4,23	2,97	2,17	1,52
$\hat{C}_2$	16,16	11,45	7,81	5,57	3,80	2,57	1,84	1,33
$\hat{C}_3$	13,98	9,87	6,84	4,73	3,29	2,35	1,59	1,14

Conclusions

In this paper a new cost function for the design of an interleaver with the iterative growth of its length was proposed.
The proposed cost function takes into account both the input sequence weight and the multiplicity of the codeword resulted at the output of the turbo encoder, corresponding to the search set. The interleaver results by minimization of the proposed cost function. In this way the proposed cost function approximates better the bit error rate, leading to superior performances in comparison to the cost functions known in literature. The superior performances are paid by a high complexity which, obviously, is not a problem if sufficient computing resources are at disposal. In the effectuated simulations the code tested in [5] was considered, in order to perform a comparison between the performances obtained with the proposed cost function and the known cost functions. We observe both from the figures 1 and 2 and the tables 1 and 2 that by using the proposed cost function, smaller bit and frame error rates are obtained.

References

[1] Berrou, C., Glavieux, A., and Thitimajshima, P. (1993) *Near Shannon Limit Error-Correcting Coding and Decoding: Turbo-Codes*, Proceedings of ICC 1993, Geneva, Switzerland, pp. 1064-1070

[2] Woodard, J.P., Keller, T. and Hanzo, L. (1997) *Turbo Coded Orthogonal Frequency Division Multiplex Transmission of 8 kbps Encoded Speech*, Proceedings of ACTS'97, Aalborg, Denmark, pp 894-899

[3] Benedetto, S., and Montorsi, G., (1996) *Unveiling Turbo Codes: Some Results on Parallel Concatenated Coding Schemes*, IEEE Transactions on Information Theory, Vol. 42, No. 2, pp. 409-428

[4] Dolinar, S. and Divsalar, D., (1995) *Weight Distributions for Turbo Codes Using Random and Nonrandom Permutations*, JPL TDA Progress Report 42-122

[5] Daneshgaran, Fred and Mondin, Marina, (1999) *Design of Interleavers for Turbo Codes: Iterative Interleaver Growth Algorithms of Polynomial Complexity*, IEEE Transactions on Information Theory, Vol. 45, No. 6

[6] M. Hall, Jr., (1976) *The Theory of Groups*, 2nd ed. New York: Chelsea

[7] Bahl, L. R., Cocke, J., Jelinek, F., and Raviv, J. (1974) *Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate*, IEEE Transactions on Information Theory

[8] Robertson, P., Hoeher, P. and Villlebrun, E. (1997) *Optimal and Sub-Optimal Maximum A Posteriori Algorithms Suitable for Turbo Decoding*, European Transactions on Telecommunications, Vol. 2, No. 8