SOME REMARKS ABOUT A POSSIBLE NEW STANDARD FOR TELEMETRY CHANNEL CODING WITH HIGH SPECTRAL EFFICIENCY

Gian Paolo Calzolari(1), Marco Chiani(2), Franco Chiaraluce(3), Roberto Garello(4)

(1)European Space Agency, D/TOS, ESOC
Darmstadt, Germany
E-mail: Gian.Paolo.Calzolari@esa.int

(2)Dipartimento di Elettronica, Informatica e Sistemistica
Università di Bologna, Italy
E-mail: mchiani@deis.unibo.it

(3) Dipartimento di Elettronica ed Automatica
Università di Ancona, Italy
E-mail: chifra@popcsi.unian.it

(4) Dipartimento di Elettronica
Politecnico di Torino, Italy
E-mail: roberto.garello@polito.it

ABSTRACT

The Consultative Committee for Space Data Systems (CCSDS) Telemetry Channel Coding standard must be updated for near-future missions characterized by high data rates and high spectral efficiency. New coding schemes satisfying these requirements are studied and compared in this paper. The attention is focused on turbo codes, product codes and low density parity check codes. Proper design is performed to match required code rates and data frame lengths. Thorough analysis proves these schemes represent highly credible candidates for standard updating. Coding scheme hierarchy strongly depends on complete applications scenario, including error rate requirements. The work here presented is one of the contributions to the current activities of CCSDS Sub Panel 1B “Channel Coding” where other Member Agencies are also very active.

I. INTRODUCTION

This paper deals with possible evolution (revision) of the Consultative Committee for Space Data Systems (CCSDS) recommendations on Telemetry Channel Coding [1] to address new high data rate space missions with high spectral efficiency. The activity has been started within CCSDS Panel 1B with the trigger of important investigations that have involved several cooperating space agencies. During a two-years of joint cooperation between the European Space Operation Center (ESOC) and the Universities of Ancona and Bologna (Italy), we have designed and analyzed a lot of “state-of-the-art” channel coding schemes, aiming to compare their performance both in terms of error correction capability and implementation complexity ([2], [3]).

The need for using bandwidth efficient coding schemes, particularly for near-Earth missions, arises from the observation that the data rates involved in the future missions will become increasingly high (up to some hundreds of Mbps) whilst the spectral crowding from many different satellites will oblige to consider a clever sharing of the available bandwidth.

In the former versions of the Blue Book on Telemetry Channel Coding, the opportunity to reach high spectral efficiency was limited to the adoption of the Reed Solomon (RS) scheme, that in the proposed RS(255,223) configuration is characterized by a code rate R = 0.875. This code rate, however, is halved when the RS code is concatenated with a 64-state rate-1/2 binary convolutional code (CC), that was probably the most popular proposal of the same recommendation.

The most recent version of the Blue Book (i.e., Issue 5) has introduced – as short term measure – another option for RS, by suggesting the adoption of the RS(255,239) code, which achieves R =

0.9375, and, most of all, has considered the possibility to puncture the convolutional code, in such a way as to increase its code rate to 2/3, 3/4, 5/6 or 7/8, instead of 1/2. The standard also contains a family of turbo codes, which are able to achieve very large coding gain, but only low code rates ranging from 1/2 to 1/6 have been standardized. On the other hand, the performance advantage achievable by using concatenated codes with iterative decoding is now well recognized and supported by their practical feasibility, so that times seem mature to extend their applicability also to rates higher than the more conventional ones. For this reason Agencies, like CNES, NASA, and ESA, have started research activities in this sense.

The starting point of ESA study was the pragmatic idea of puncturing the CCSDS turbo code [2]. After that, we have studied a lot of other options like partially systematic CCSDS turbo codes, serially concatenated convolutional codes, product codes, DVB-like turbo codes (i.e., based on the turbo code structure proposed for the DVB-S standard), low-density parity check codes, and others.

II. THE CCSDS STANDARD

The coding schemes currently standardized in [1] are:

- Two Reed-Solomon codes with 8-bit symbols: RS(255,223) and RS(255,239).
- A "mother" 64-state rate-1/2 binary convolutional code plus code rate 2/3, 3/4, 5/6, and 7/8 convolutional codes obtained by puncturing the mother code.
- All possible serial concatenation of a Reed-Solomon code (outer code) and a convolutional code (inner code) through an interleaver of length ($n \cdot I$) bytes (n is the Reed-Solomon codeword length and $I = 1, 2, 3, 4$, or 5).
- A family of turbo codes with nominal rates 1/2, 1/3, 1/4, or 1/6, obtained from two 16-state, rate-1/4 binary convolutional codes and an interleaver with size ranging from 1784 to 16384 bits.

III. REQUIREMENTS AND METHODS OF ANALYSIS

The codes studied for applications to high data rate future missions and possible inclusion in [1] should satisfy stringent requirements on:

- DATA RATES: Very high, ranging from a few Mbps to hundreds of Mbps.
- BANDWIDTH-EFFICIENCY: Large (due to high data rates and spectral crowding from many satellites). Target spectral efficiency should be at least 1.5 bit/s/Hz over 4-PSK (binary codes with code rate $R \geq 3/4$).
- POWER-EFFICIENCY: Very large coding gains needed (due to limited transmitted power from satellites of small dimensions).
- ERROR RATES: Some applications may require extremely low error probabilities (because of poor error resilience for video and image compression techniques), in some cases as low as Frame Error Rates = $10^{-7}/10^{-8}$.
- COMPLEXITY: Limited for both encoders (realized on board) and decoders (due to very high data rates involved).

The coding schemes considered for satisfying these requirements have been analyzed both by simulation and by analytical techniques. At high/medium error rates (e.g., Frame Error Rates higher or equal to 10^{-5}) simulation has been employed. At very low error rates (e.g., Frame Error Rates less or equal to 10^{-7}), reliable simulations are too long and practically unfeasible. In this region, analytical techniques shall therefore be employed. In fact, at high signal-to-noise ratio (SNR) the performance of any binary code is dominated by its minimum distance $d_{\min}$ and its multiplicity: in this region the code performance practically coincides with the union bound, truncated to the contribution of the first distance. For a $C(n,k)$ code of rate $R = k/n$, given $d_{\min}$, its multiplicity $A_{\min}$, and its information bit multiplicity $w_{\min}$ (defined as the sum of the Hamming weights of the $A_{\min}$ information sequences generating the codewords with weight $d_{\min}$), we can write, for very high SNR:

$$\text{BER} \approx \frac{1}{2}\frac{w_{\min}}{k}\operatorname{erfc}\sqrt{Rd_{\min}\frac{E_b}{N_0}}\,, \qquad \text{FER} \approx \frac{1}{2}A_{\min}\operatorname{erfc}\sqrt{Rd_{\min}\frac{E_b}{N_0}}\,. \tag{1}$$

Error Rate curves corresponding to Eq. (1) are often called “error floor” for concatenated codes with interleavers, like turbo codes. In fact, these codes are characterized by excellent performance at low SNR, and by a drastic slope change at high SNR. In this region the code performance are well approximated by Eq. (1), even if a small penalty must be also taken in account, due to the sub-optimality of the iterative decoding adopted.

This approach obviously requires the knowledge of the code minimum distance and its multiplicity, a non-trivial task. An algorithm to compute the minimum distance of turbo codes was presented in [4]. The exact computation of the minimum distance and the multiplicity of Hamming product codes has been provided in [5].

IV. CODES UNDER STUDY

A. *Punctured CCSDS turbo codes*

The CCSDS turbo code family (C_{TC}) of [1] is characterized by:

- Constituent codes: two equal 16-state, rate-1/4 convolutional codes with feedback polynomial ($1+D^3+D^4$), and feedforward polynomials ($1+D+D^3+D^4$), ($1+D^2+D^4$) and ($1+D+D^2+D^3+D^4$).
- Interleaver: Algorithmic, Berrou interleavers with length N = 1784, 3568, 7136, 8920, or 16384 bits (equal to data frame length F).
- Nominal rates: ranging from 1/6 to 1/2.

The first solution evaluated by ESA for updating [1] with higher values of code rate is the application of an external puncturer to this family of turbo codes instead of looking for a new “family” requiring different encoder, etc.. To do this, we have considered only the first feedforward polynomial. The basic encoder is depicted in Fig. 1. By varying the puncturing pattern, we have studied Punctured CCSDS Turbo Codes (PCTC) with:

- Nominal code rates: 3/4, 7/8, 8/9, 11/12, and 15/16.
- Data frame length: F = 1784, 7136, or 8920 bits.
- Puncturing patterns: systematic, designed following [6].

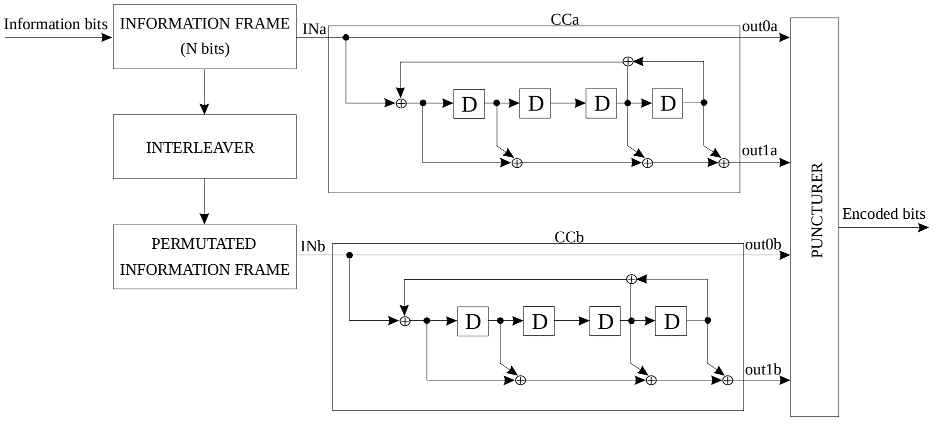

Fig. 1: Punctured CCSDS turbo encoder.

As explained in Section III, these schemes have been analyzed both by simulation and by evaluating their error floors. As an example, the performance of two designed PCTC with F = 8920 bits are depicted in Fig. 2 for code rates 3/4 and 7/8. Their simulation curves (corresponding to 15 iterations of the BCJR algorithm) are compared against their error floors. These are calculated by using the code minimum distance parameters which are $(d_{\min}/A_{\min}/w_{\min})$ = (10/29/169) for the rate-3/4 code and $(d_{\min}/A_{\min}/w_{\min})$ = (5/79/316) for the rate-7/8 code.

Punctured CCSDS turbo codes represent a pragmatic and efficient solution for high data rate bandwidth constrained applications. In fact, a single component could achieve a code rate ranging from 1/6 to 15/16, by only varying (in real-time if necessary) the external puncturing pattern. PCTC permutations (equal to those of C_{TC}) can be algorithmically generated, an important point for space missions because this avoids the use of very large memories on board. As for the maximum data rate achievable, currently PCTCs seem appropriate as a short/medium term solution for applications up to some tens of Mbps by a dedicated ASIC. However, advancements in technology and decoding algorithms development are in progress. In fact, some companies (Broadcom [7], IMEC [8], STMicroelectronics [9]) have recently announced turbo code chips able to work with data rates up to 155 Mbps. PCTC performance at low/medium SNR are excellent, and extremely difficult to improve. At high SNR (very low error rates) the error floor phenomenon partially reduces the coding gain, especially for high code rates. However, PCTCs remain competitive with other solutions, in particular for code rates 3/4 and 7/8, especially at not too low error rates.

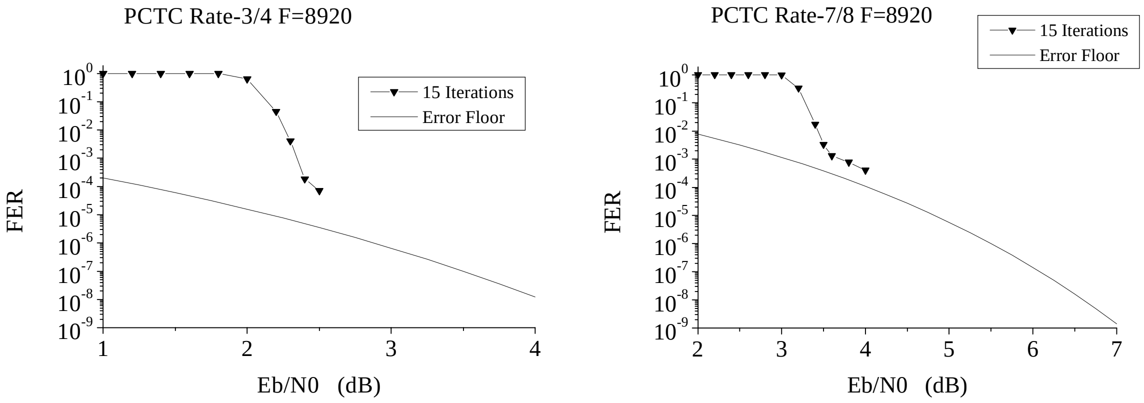

Fig. 2: Simulated performance vs. error floor for rate-3/4 and rate-7/8 PCTC.

B. Partially Systematic Turbo Codes

The CCSDS turbo codes are systematic, which means that the information bits are always transmitted. As an alternative, one can think to design Partially Systematic Turbo Codes (PSTCs), where a prefixed number of information bits are punctured (i.e., not transmitted) [10]. Thus the structure of the encoder remains that shown in Fig. 1, but the puncturer acts in a different manner by removing from the encoded sequence some information bits, too. Noting by ρ_u, ρ_{p1} and ρ_{p2} the proportion of the information bits, the parity bits out of the first constituent encoder and the second constituent encoder, respectively, that are not punctured, the designed code rate R results in:

$$\frac{1}{R} = \rho_u + \rho_{p1} + \rho_{p2} . \tag{2}$$

Usually, it is convenient to set $\rho_{p1} = \rho_{p2} = \rho_p$, and expression (2) simplifies accordingly. As an example, to obtain R = 3/4 we can choose ρ_u = 3/4 (which means that on average three out of four information bits are transmitted, and therefore one out of four information bits is punctured), and ρ_p = 7/24. Similarly, to have R = 7/8 we can choose ρ_u = 11/14 and ρ_p = 5/28. In both cases, the puncturing pattern is periodic, with period b given by the l.c.m. between the denominators of ρ_u and ρ_p.

Since most of the information bits are transmitted while most of the parity bits are punctured, it is convenient to specify the puncturing patterns in terms of locations of the punctured bits for the information bits (numbered from 1 to *b*) and in terms of the transmitted bits for the parity bits. To emphasize this difference, the puncturing pattern for the information bits will be denoted by P_u, while those for the parity bits will be denoted by $\overline{P}_{p1}$ and $\overline{P}_{p2}$, respectively.

The main advantage of using PSTCs instead of systematic turbo codes is the observation that "ad hoc" designs of partially systematic puncturing patterns allow to obtain punctured codes with larger minimum distances and better error floors. At the same time, the change to introduce in the recommended CCSDS turbo codes is minimal only involving, once again, the puncturer configuration. Examples of achieved larger minimum distances will be presented afterwards; they imply improved performance at very low error rates. As a counterpart, a PSTC generally exhibits worse performance at high/medium error rates. In the present implementation, PSTCs also pay a performance penalty for iterative decoding which is larger than for systematic PCTC (on the order of 0.5 dB against 0.1÷0.2 dB for systematic codes).

Two examples of performance evaluation for PSTCs with rates and frame lengths of interest for the considered applications are reported in Fig. 3 (*F* = 1784 bits, *R* = 3/4) and in Fig. 4 (*F* = 8920 bits, *R* = 7/8): the BER and FER curves are plotted, in terms of simulated results and error floors, and compared with those of the systematic code with the same parameters (systematic PCTC).

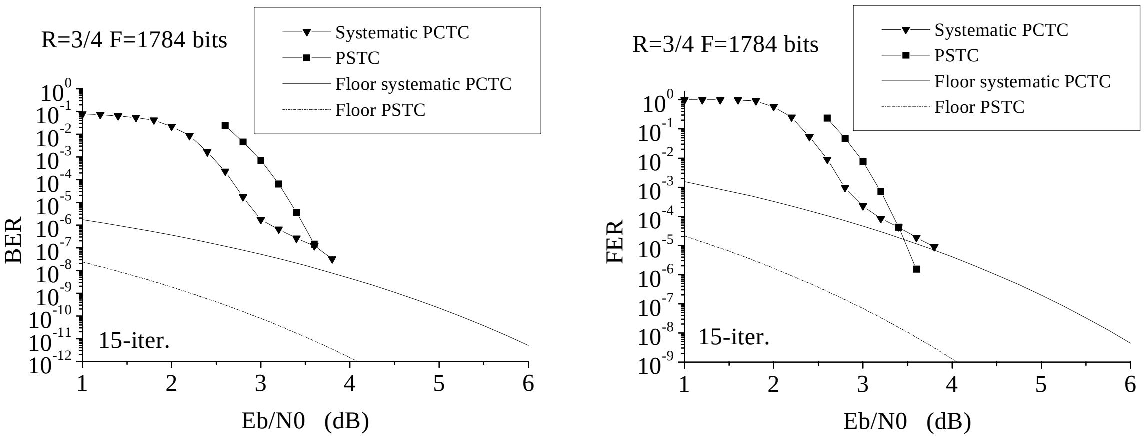

Fig. 3: BER and FER curves for a partially systematic CCSDS turbo code with F = 1784 bits and R = 3/4 against those of the systematic CCSDS turbo code with the same frame length and rate.

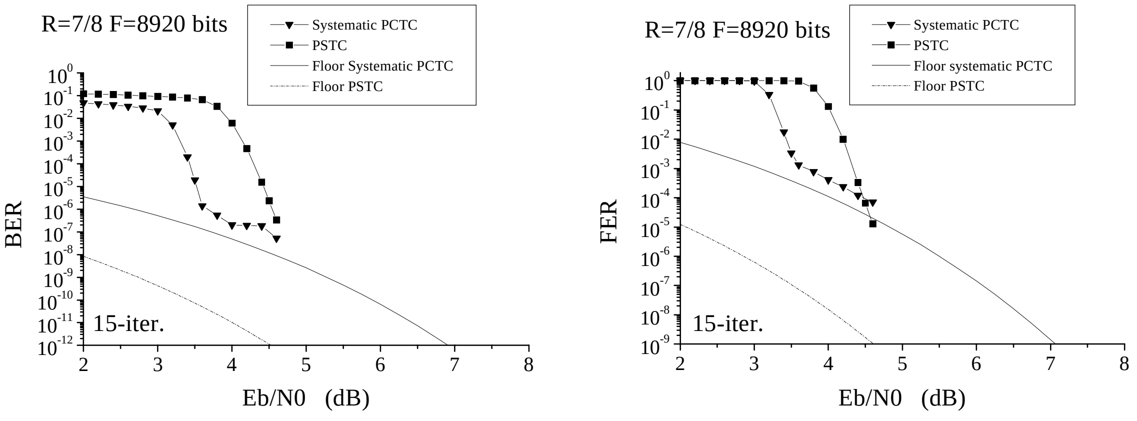

Fig. 4: BER and FER curves for a partially systematic CCSDS turbo code with F = 8920 bits and R = 7/8 against those of the systematic CCSDS turbo code with the same frame length and rate.

The PSTC of Fig. 3 has b = 24 with puncturing patterns P_u = (1, 4, 7, 13, 18, 23), $\overline{P}_{p1}$ = (1, 4, 7, 13, 16, 19, 23), $\overline{P}_{p2}$ = (2, 5, 9, 15, 18, 21, 24); the PSTC of Fig. 4, instead, is characterized by b = 28, P_u = (2, 7, 12, 17, 22, 27), $\overline{P}_{p1}$ = (2, 7, 12, 17, 22), $\overline{P}_{p2}$ = (7,12,17, 22, 27).

Moreover, the PSTC of Fig. 3 has estimated ($d_{\min}/A_{\min}/w_{\min}$) = (10/3/6) against ($d_{\min}/A_{\min}/w_{\min}$) = (6/4/8) of the systematic PCTC; that of Fig. 4 has estimated ($d_{\min}/A_{\min}/w_{\min}$) = (8/10/60) against ($d_{\min}/A_{\min}/w_{\min}$) = (5/79/316) of the systematic PCTC.

While the superior performance of the partially systematic solution as regards the error floor is evident from the triplets above, the figures demonstrate a potential advantage also as regards the simulated results: in fact, a cross-point has been found (for the FER) or can be foreseen (for the BER) below which the PSTC outperforms the systematic code. Unfortunately, in both the considered cases, we were not able to simulate extensively the performance of the PSTC, thus proving its convergence to the error floor. This is because the unacceptable elaboration time required. Also, the distances have been computed by limiting the input weight: the true minimum distances of the considered PSTCs might be lower than those reported above, even if the probability of this event should be low.

In spite of the incompleteness of the analysis, the preliminary results obtained clearly confirm the potential benefits of the partially systematic solution and the possibility to mitigate the limits of the more conventional systematic turbo codes in the region of low/very low error rates (when required).

The main difficulty of PSTCs is the design of the puncturing patterns: those reported before have been found through a series of iterated attempts. First of all, it is necessary to avoid fortuitous assumption of non-invertible encoders. In this case the iterative decoding algorithm might have bad performance or does not converge at all. Anyway, even being sure that the code is not degenerated, the design of an efficient partially systematic puncturing pattern remains an arduous task. Very recently, we have begun to develop some theoretical treatments that should guide the choice but, for the time present, we only rely on some simple rules of thumb.

In particular, we have seen that, in many cases, the signal-to-noise ratio where the water-fall region starts is much more affected by the value of ρ_u and ρ_p (permeability) rather than by the specific puncturing pattern. So, permeability must be chosen in such a way as to have a satisfactory behavior at low/medium signal-to-noise ratios. On the contrary, optimization of the puncturing patterns may have a great effect on the minimum distance, so determining the behavior for high signal-to-noise ratios.

The validity of these rules has not been proved in general terms; simply they worked for many of the cases we have considered, thus helping, in some way, the design. Actually, the huge number of degrees of freedom offered by the partially systematic solution seem to leave margins for further optimization.

C. Product codes

Similarly to PSTC, the main reason for proposing the adoption of turbo product codes (TPCs) as an alternative to punctured CCSDS turbo codes in the next telemetry channel coding standards, is the potential benefit they offer to improve performance at low/very low error rates. This is due to the large minimum distances they posses by construction, though partly attenuated by very large multiplicity which raise up their asymptotic bounds [5]. In addition, commercial chips produced by AHA implementing product codes co-decoders are already available, working up to 155 Mbit/s, and even useful for re-configurable applications [11].

The scheme of a product code is shown in Fig. 5: $k_1 \times k_2$ information bits are written into a matrix and encoded, first by rows and then by columns (or vice versa). The resulting product code, with size $n_1 \times n_2$, can be seen as a special case of serial concatenation. At the receiver, decoding is made by means of iterative procedures generally sub-optimal but simpler than BCJR [12].

Actually, one strength of product codes in the perspective of practical application seems to be their feasibility for parallel implementation, which justifies the high data rates mentioned before.

The constituent codes are usually extended Hamming (EH) codes or parity check (PC) codes, because of their simplicity. The combination of two EH codes ensures a minimum distance $d_{\min} = 16$, that of one EH code and one PC code a $d_{\min} = 8$, and that of two PC codes a $d_{\min} = 4$. In principle, the choice of two EH codes as constituent codes is preferable; anyway, wishing to have specified values for the frame length F and the code rate R, this is not always applicable. In Table I we have summarized the best codes we have selected for $F = 1784$ and 8920 bits and $R = 3/4$, 7/8 and 15/16; we can see that the resort to PC codes (that reduces the minimum distance value) becomes more and more necessary for increasing rates, most of all in the case of (relatively) short frame lengths.

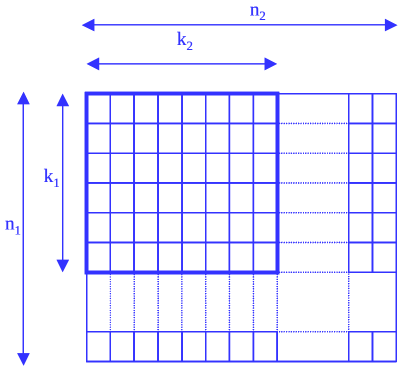

Fig. 5: Coded matrix of a product code.

Performance comparison with the PCTCs having, approximately, the same frame length and rate is shown in Fig. 6, limited to the FER curves for the sake of brevity. The error floor curves have been plotted by considering true multiplicity [5] while the simulated curves for the TPCs have been obtained by optimizing some feedback coefficients appearing in the decoding algorithm ([3], [12]). The number of iterations used (15) is the same for both kinds of codes.

The conclusions we can draw looking at the figures were expected: in most cases, the product codes are less powerful than punctured CCSDS turbo codes for high/medium error rates (usually down to FER $\approx 10^{-4}$) while become preferable for low/very low error rates thanks to their more favorable minimum distance properties.

The only (remarkable) exception is given by the PCTC with $F = 8920$ bits and $R = 3/4$: this code is able to outperform the corresponding product code, that indeed has a minimum distance of 16, for any reasonable error rate, included the case of a FER equal to 10^{-8} which can be seen as a formidable target even for the most exacting application.

Finally, it should be noted that instead of using PC codes as constituent codes, under constraints on the frame length and rate, one could think to adopt punctured product codes (i.e., obtained by an external puncturing), even if this may lead to a minimum distance decrease.

F **(bits)**	***R***	**row code**	**column code**	$d_{\min}$
1784	3/4	(45,38)	(54,47)	16
1784	7/8	(127,119)	(16,15)	8
1784	15/16	(106,105)	(18,17)	4
8920	3/4	(46,37)	(250,241)	16
8920	7/8	(99,91)	(106,98)	16
8920	15/16	(232,223)	(41,40)	8

Table I: List of designed product codes for the frame lengths and rates of interest (nominal values).

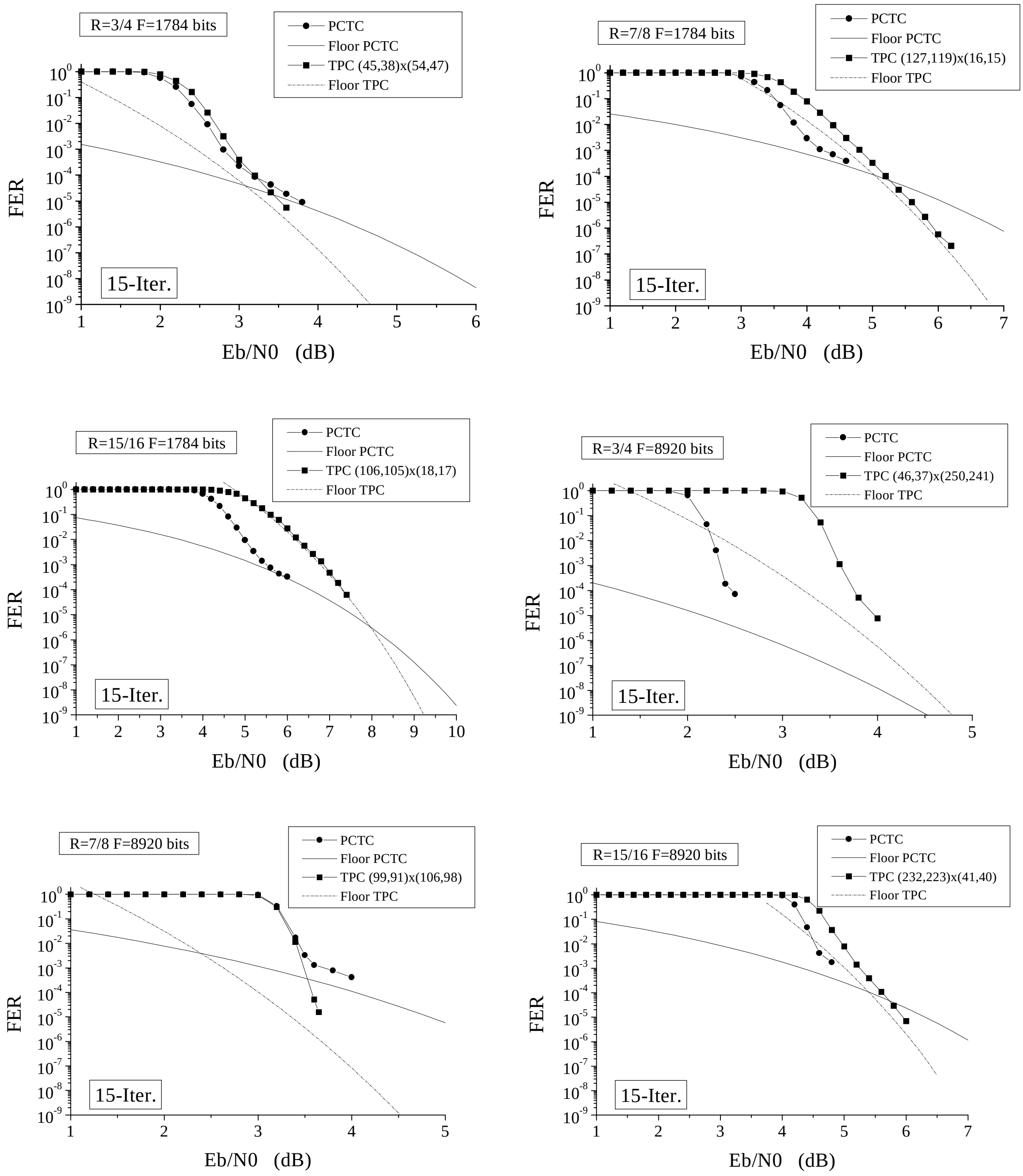

Fig. 6: Performance comparison between product codes (TPC) and punctured CCSDS turbo codes (PCTC) having approximately the same frame length and rate.

D. *Low Density Parity Check Codes*

A (n,k) binary Low Density Parity Check Code (LDPCC) is built from a very sparse (with a very small number of ones) parity check matrix H with $r = (n - k)$ rows and n columns. Every n-bit codeword $\underline{c}$ satisfies the equation $H\underline{c} = \underline{0}$. Invented by Gallager in 1962 [13], these codes have been disregarded until their recent re-discovery. Gallager's original codes were regular: the number of ones in any row (and in any column) was equal to a fixed, very small, number. A LDPCC can be represented by a bipartite graph, where a class of nodes represents the code symbols and the other class represents the parity check equations. All code symbols connected to a check node satisfy parity. In a regular LDPCC, each symbol participates in j equations, so that there are j branches leaving each variable node; also, each parity-check equation involves k symbols, so that there are k branches leaving each check node. Regular LDPCCs have good performance, but a little worse than turbo codes.

Irregular LDPCCs, proposed by Richardson, Shokrollahi, Urbanke in [14], outperform, on memory-less channels, not only regular LDPCCs but also, in some conditions, turbo codes. These excellent results have been obtained by allowing different degrees for each node (variable or check). Irregular LDPCC parity check matrix H is randomly generated by using proper, non-trivial, distributions.

A LDPCC can be decoded by applying the belief propagation algorithm [15] on the graph representing the code. This algorithm converges to the exact a posteriori probability when the graph has no loops. If this not the case (as for usual LDPCCs), the propagation algorithm can be iterated on the graph: the resulting penalty paid (with respect to ideal decoding) is very limited when a sufficient number of iterations is employed. A key point for implementation is that all computations can be executed in parallel on each different node. In line of principle, the algorithm is therefore available for an intensive parallelization. It is interesting to remark that Flarion [16] has already produced an IP core able to support extremely high data rates (up to some Gbps).

In [2], we have designed LDPCCs for matching CCSDS requirements in terms of code rates and data frame length. As an example, in Fig. 7, a LDPCC with F = 7136 bits and code rate 15/16 is compared against a PCTC with similar values.

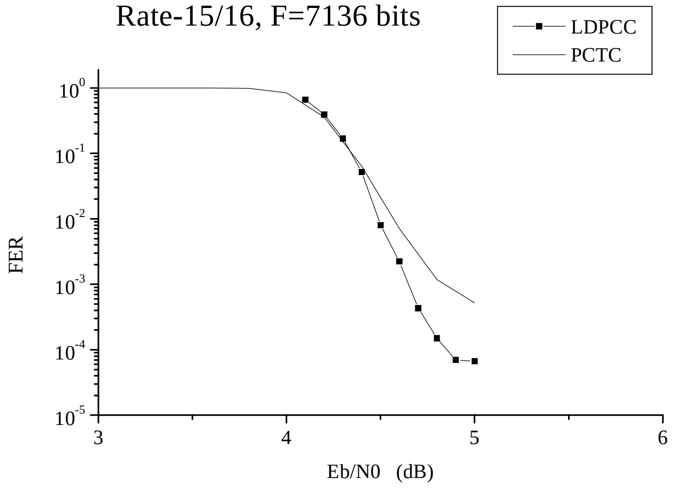

Fig. 7: Performance comparison between a LDPCC and a punctured CCSDS turbo code in the case of F = 7136 bits and R = 15/16.

In general, the performances of the designed LDPCCs were very good. For high code rates, near to the unity (e.g., $R > 7/8$) they clearly outperform PCTC at low error rates [2].

Recently, Low Density Parity Check Codes based on finite geometry have been introduced by Kou, Lin, and Fossorier in [17]. Exploiting useful properties of Euclidean and projective geometry over finite field, they are able to maintain very large minimum distance values. Moreover, a simple encoder structure based on the cyclic or quasi-cyclic code structure is always available. These codes are very promising, and some preliminary design for CCSDS applications was performed in [3]. The results were interesting, even if, in some cases, difficulties to match both code rate and data frame length constraint were encountered, leading to sub-optimal design.

By concluding, due to high data rates achievable by parallel implementation Low Density Parity Check codes look very promising for missions with very stringent requirements in terms of traffic, bandwidth and error rates.

V. DISCUSSION AND CONCLUSIONS

Concatenated codes, decoded by iterative algorithms, represent a realistic solution for improving the CCSDS channel coding standard and satisfy the need for new schemes with high code rate. Our analysis involved turbo codes, product codes and low density parity check codes. As expected, we have shown that an "all-powerful" code, able to outperform all the other schemes for any operation condition, does not exist, and besides the specific value of the frame length and code rate, the choice of the best code strictly depends on the value of the error rate required. One of the main issues is then the need for precise quality requirements on the error rates desired. These should be fixed by avoiding the "light" temptation to exaggerate in the requirements (so producing unrealistic previsions) whilst, when well thought, they could help to separate the appropriate solutions from the inappropriate ones.

REFERENCES

[1] CCSDS 101.0-B-5, "Telemetry Channel Coding", Blue Book, Issue 5, June 2001.

[2] University of Ancona, "Bandwidth-efficient coding schemes – Final Report", ESA/ESOC, Contract No. 14128/00/D/SW, Dec. 2000.

[3] University of Ancona, "Highly efficient channel codes for high data rate missions – Final Report", ESA/ESOC Contract No. 15048/01/D/HK (SC), Dec. 2001.

[4] R. Garello, P. Pierleoni, S. Benedetto, "Computing the free distance of turbo codes and serially concatenated codes with interleavers: algorithms and applications", Journal on Selected Areas in Communications, vol. 19, pp. 800-812, May 2001.

[5] F. Chiaraluce, R. Garello, "On the Asymptotic Performance of Hamming Product Codes", Proc. 6th International Symposium on Communication Theory and Applications, ISCTA '01, Ambleside, UK, pp. 329-334, 15-20 July, 2001.

[6] W. E. Ryan, O. E. Acikel, "Punctured Turbo-Codes for BPSK/QPSK Channels", IEEE Transactions on Communications, vol. 47, pp. 1315-1323, Sept. 1999.

[7] http://www.broadcom.com/

[8] http://www.imec.be/

[9] http://www.st.com/

[10] I. Land, P. Hoeher, "Partially Systematic Rate 1/2 Turbo Codes", Proceedings of International Symposium on Turbo Codes, Brest, France, pp. 287-290, Sept. 2000.

[11] http://www.aha.com/

[12] R. Pyndiah, "Near Optimum Decoding of Product Codes", IEEE Trans. on Communications, vol. 46, pp. 1003-1010, Aug. 1998.

[13] R. G. Gallager, "Low-Density Parity-Check Codes", IRE Transactions on Information Theory, vol. IT-8, pp. 21-28, Jan. 1962.

[14] T. Richardson, A. Shokrollahi, R. Urbanke, "Design of Capacity-Approaching Irregular Low-Density Parity Check Codes", IEEE Transactions on Information Theory, vol. 47, pp. 619-637, Feb. 2001.

[15] D. J. C. MacKay, R. J. McEliece, J. Cheng, "Turbo Decoding as an Instance of Pearl's Belief Propagation Algorithm", IEEE J. Selected Areas in Communication, vol. 16, pp. 140-152, Feb. 1998.

[16] http://www.flarion.com/

[17] Y. Kou, S. Lin and M.P.C. Fossorier. "Low Density Parity Check Codes Based on Finite Geometries: A Rediscovery and New Results". IEEE Transactions on Information Theory, vol. 47, pp.2711-2736, Nov. 2001.