Turbo–Code representation of RA–Codes and DRS–Codes for reduced decoding complexity

Simon Huettinger
Telecommunications Institute II,
University Erlangen-Nürnberg
Cauerstr. 7,
D-91058 Erlangen, Germany
Tel.: +49-9131-852-7116,
Fax: +49-9131-852-8919,
e-mail: simon.huettinger@LNT.de

Stephan ten Brink
Bell Laboratories,
Lucent Technologies
Room 4C-438,
101 Crawfords Corner Road,
Holmdel, NJ 07733, USA
Tel.: +1-732-949-9559,
Fax: +1-732-949-6010,
e-mail: stenbrink@bell-labs.com

Johannes Huber
Telecommunications Institute II,
University Erlangen-Nürnberg
Cauerstr. 7,
D-91058 Erlangen, Germany
Tel.: +49-9131-852-7116,
Fax: +49-9131-852-8919,
e-mail: huber@LNT.de

Abstract — **Recently very good iterative decoding performance close to Shannon's capacity limit [12] has been obtained by using a serially concatenated structure consisting of an outer repetition code and an inner rate one scrambler [5, 7]. In this paper we show that these codes can also be interpreted as parallel concatenated "turbo" codes, if we apply some restrictions on the interleaver design which, however, do not affect the decoding performance.**

In the parallel representation the message passing of the iterative decoder is improved. This reduces the computational complexity of decoding significantly as the required number of iterations is divided by c for a rate $R = 1/c$ code. In addition, the systematic bits of the code are exploited better.

Keywords: Turbo–Codes, Serial concatenation.

I. Introduction

Since 1993, when Berrou et al. [3] presented the first long concatenated codes — the Turbo–Codes — that could be decoded with reasonable complexity while performing close to Shannon's limit [12], there is a competition, how close to this limit nearly error free communication is possible with codes and decoders which are applicable in practice.

Recently, besides the LDPC–Codes [8], two new classes of codes [5, 7] have been presented, which are within 0.1 dB of Shannon's limit for very large block length ($N > 100000$). In this paper we show that these codes can also be represented as parallel concatenated "turbo" codes with a special interleaver design. The parallel representation is advantageous for iterative decoding and possibly more intuitive to understand as most people are more familiar with parallel concatenated codes.

In section II *Repeat Accumulate (RA) Codes* [7] are analyzed and it is shown by means of the EXIT chart technique [4] that the asymptotic behavior towards infinite block length for the parallel "Turbo–Like–Structure" is not different to the original RA–structure. Simulation results show that even for a fixed block length there is no loss in performance, while convergence of iterative decoding can be achieved with half the number of operations for a rate $R = 1/2$ code.

Section III presents the *Doped Repeat Scramble (DRS) Codes* [5] in the parallel representation. For this code the reduction of decoding complexity is even higher. The doped (systematic) bits can be used within both half–iterations to improve convergence.

II. Repeat Accumulate Codes

The building–blocks of *Repeat Accumulate Codes* [7] consist of an outer repetition code *rep* of rate $R = 1/2$, a very large interleaver $\boldsymbol{\pi} \hat{=} [\pi(1), \pi(2) \ldots]$ of length N i.e. $\pi(i) \in \{1, 2, \ldots, N\}$,$i \in \{1, 2, \ldots, N\}$ with $\pi(i) \neq \pi(j)$ for $i \neq j$ and an inner accumulator, which is a memory 1 recursive shift register, or equivalently a memory $m = 1$ scrambler *scr*, as the device usually is called in the context of Turbo–Codes. The trellis length of the scrambler is N. In this paper, we consider only RA–Codes with outer repetition codes of rate $R = 1/2$. The extension to lower rates is straightforward.

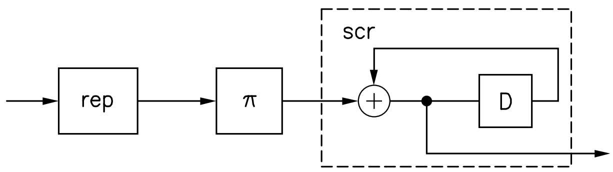

Figure 1: Original representation of RA–Codes.

In the following we show that the serially concatenated structure can be represented as a parallel concatenated code if we restrict ourselves to a subset of possible interleavers. The special interleavers we consider permute the first $K = N/2$ input bit positions $[1 \ldots K]$ to the first K output bits at the positions $\boldsymbol{\pi}_1 \hat{=} [\pi(1) \ldots \pi(K)]$ with $\pi(i) \leq K \quad \forall \quad i \leq K$. The input bits $[K+1 \ldots 2K]$, which are just a copy of the first K information bits, are permuted to the output bits at position $\boldsymbol{\pi}_2 \hat{=} [\pi(K+1) \ldots \pi(2K)]$ with $K < \pi(i) \leq 2K \quad \forall \quad K < i \leq 2K$. Although this subset of specific interleavers is very small compared to the set of all possible interleavers $\boldsymbol{\pi}$, results will show that the restriction does not affect the performance. It can be regarded as a simple interleaver design which avoids that original and the repeated bits are too close to each other.

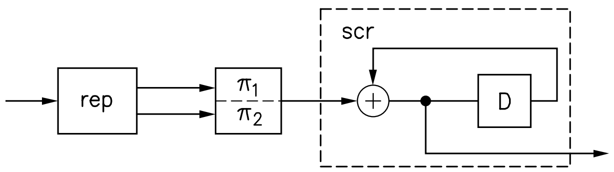

Figure 2: RA–Code with splitted interleaver.

Now we just split the inner code into two parts of trellis length $K = N/2$ by resetting the starting state of the accumulator to zero when the input crosses from interleaver $\boldsymbol{\pi}_1$ to $\boldsymbol{\pi}_2$. This separation into two independent trellises weakens the code only within a trellis fragment of about five times of the constraint length. For the accumulator this is only about ten bits which is negligible compared to the considered block length.

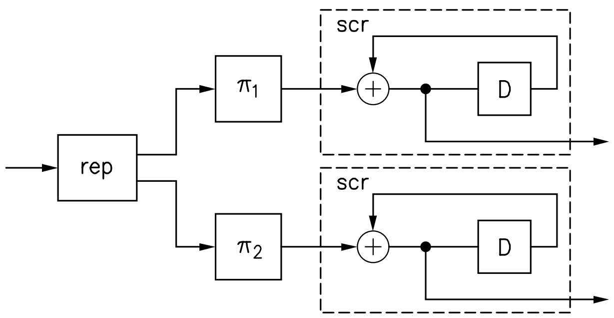

Figure 3: RA–Code with splitted interleaver and separated scramblers.

Obviously, it is possible to put an interleaver ${\boldsymbol{\pi}_1}^{-1}$ in front of the repeater, as permutation of the information bits before encoding does not change any properties of the coding scheme. Then this additional interleaver can be merged with $\boldsymbol{\pi}_1$ and $\boldsymbol{\pi}_2$. The resulting interleaver in the upper branch then is an identity interleaver, which can be omitted without changing the effective interleaving depth. In the lower branch the interleaver $\tilde{\boldsymbol{\pi}}_2$ now consists of a cascade of ${\boldsymbol{\pi}_1}^{-1}$ and $\boldsymbol{\pi}_2$. Furthermore, in the next step the repetition code is depicted as a node that passes on the information bits directly to an accumulator and also to the interleaver $\tilde{\boldsymbol{\pi}}_2$, which is the common representation for parallel concatenated "turbo" codes.

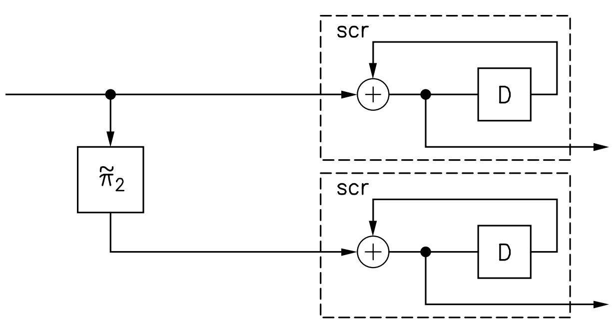

Figure 4: Turbo–Code representation of RA–Codes.

Now the serial concatenated encoder structure of RA–Codes has been transformed into the representation of a parallel concatenated code. To show that this representation is equivalent to the corresponding serially concatenated code we first analyze the asymptotic behavior for infinite block length. For Turbo–Codes it is well known that the interleaver gain is proportional to N^{-1}. In a serially concatenated coding scheme the interleaver gain can be calculated by the minimum distance of the outer code $d_{min,o}$. For a rate $R = 1/2$ repetition code $d_{min,o}$ is two. Hence, the interleaving gain [2] $N^{1-d_{min,o}} = N^{-1}$ is unchanged.

The EXIT charts [4] in Fig. 5 for the serial and in Fig. 6 for the parallel representation give a more detailed investigation of the asymptotic behavior (asymptotic with respect to interleaver size). In a simplified model for the iterative decoding process it is assumed that the a–priori and extrinsic information can be characterized by jointly independent Gaussian random variables. Then the decoder can be characterized by its transfer function, where at the input and output the mutual information between the Gaussian random variables and the transmitted sequence is measured. For large interleavers the extrinsic information is independent relative to decoding horizon over a large nuber of iterations, and hence, this model is in excellent agreement with measurements in concatenated coding schemes.

EXIT charts provide a visualization of the exchange of extrinsic information between the two component decoders. In addition to the prediction at which signal–to–noise ratio E_b/N_0 convergence of decoding can be expected, the trajectory of iterative decoding can also be estimated very precisely from the transfer characteristics of the component decoders. In Fig. 5 and Fig. 6 the transfer characteristics are plotted up to their first intersection. Further iterations do not improve performance as the trajectory gets stuck.

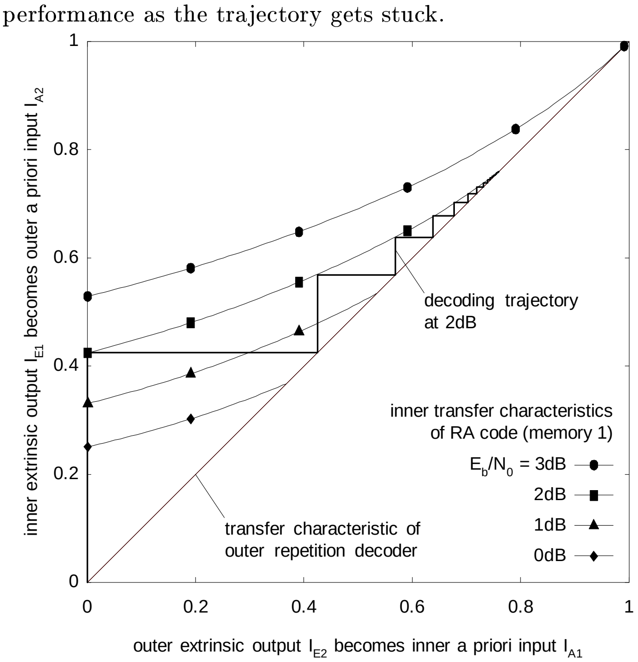

Figure 5: Extrinsic Information Transfer (EXIT) Chart for serial representation of RA–Codes.

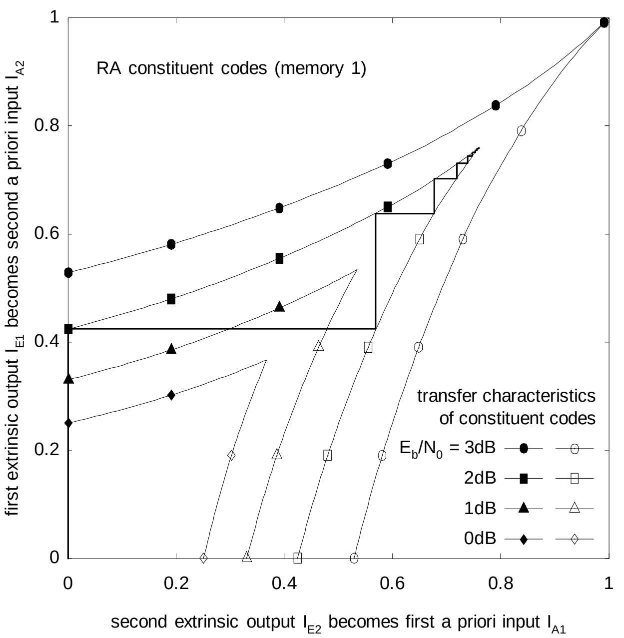

Figure 6: Extrinsic Information Transfer (EXIT) Chart for parallel representation of RA–Codes.

The comparison of Fig. 5 and Fig. 6 shows that only half the number of iterations are necessary, if the parallel representation of the code is used. This is due to the effect that the length of the vector of extrinsic information being exchanged is K in the parallel representation, whereas in the serial representation the vectors contain $N = 2K$ extrinsic information values. Hence, as the quality of the extrinsic information depends only on the scrambler which remains unchanged, the faster message passing leads to faster convergence of iterative decoding. To verify this result for finite block length both schemes have been simulated for $K = 10000$ and 20 iterations in the serial case and 10 iterations in the parallel case. The bit error results are shown in Fig. 7.

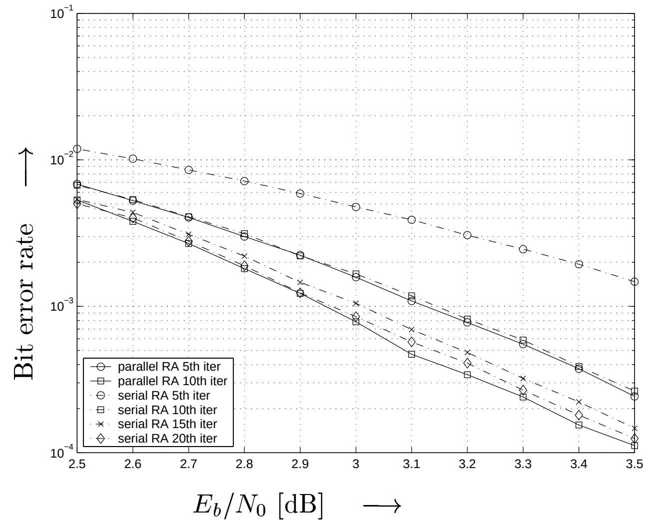

Figure 7: Performance of RA–Codes in serial and parallel representation in dependence of number of iteration (Blocklength $K = 10000$).

For the simulation of Fig. 7 a minimum of 1000 bit errors per E_b/N_0 value have been simulated. The ensemble performance is obtained by creating a new random interleaver for every transmitted block, which is an approximation of uniform interleaving [1]. The simulation result does not show significant differences between the parallel and the serial representation of the RA–Codes.

Although an RA–Code of rate $R = 1/2$ is not a very good code, as its convergence threshold is at $E_b/N_0 > 3.384$ dB [9], it is a member of a family of codes that provably approach the AWGN channel capacity for $R \to 0$. That is, the required E_b/N_0 for arbitrary small error probability approaches $\log 2$. The parallel representation can easily be extended to the whole family of RA–Codes of rate $R = 1/q$, as they simply correspond to parallel concatenated codes with q parallel branches. For lower rates the advantage of the optimized message passing in the parallel representation is emphasized even more. A–priori information for the K information bits is created after processing K code bits instead of $q \cdot K$ in the serial representation. Hence, for a rate $R = 1/q$ RA–Code the number of iterations is reduced by a factor of q.

III. Doped Repeat Scramble Codes

The DRS–Codes presented in [5] and [6] are generalized RA–Codes. The accumulator is replaced by a recursive scrambler with more memory.

For the construction of a rate $R = 1/2$ DRS–Code the recursive scrambler *scr* of memory $m = 3$ with feedback polynomial $G_r = 17$ and feedforward polynomial $G = 7$ (octal representation) which is shown in Fig. 8 has been found [5] as the one with the lowest possible convergence threshold. The corresponding inner transfer characteristic closely resembles the shape of a straight line which allows convergence of iterative decoding at low E_b/N_0 in combination with an outer repetition decoder.

But as most recursive scramblers are unsuitable for iterative decoding as long as no systematic information is available, additional systematic *doping* has to be applied to ensure convergence of iterative decoding. The term "doping" is used to express that the scrambled bits are *substituted* by their respective systematic counterparts. Doping of a ratio D_r of a number n_s of systematic bits to a number n_c of code bits of $D_r = n_s : n_c = 1 : 100$ is already enough to achieve a code of rate $R = 1/2$ with a convergence threshold ("pinch–off limit") as low as 0.27 dB.

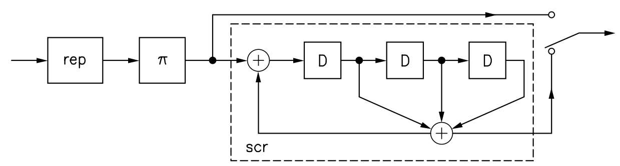

Figure 8: Original representation of rate $R = 1/2$ DRS–Codes.

After applying the procedure of interleaver splitting and trellis separation to this code in the same way as to the RA–Code of section II, the encoder consists of two parallel concatenated branches of doped scramblers. As in the case of RA–Codes the effective interleaving depth is unchanged, although $\tilde{\boldsymbol{\pi}}_2$ is only of length $K = N/2$. The separation into two interleavers doubles the effective interleaving length. But as one of the interleavers ($\boldsymbol{\pi}_1$) is obviously redundant, it is replaced by an identity interleaver.

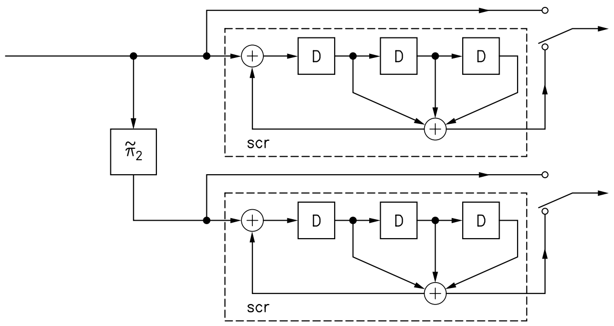

Figure 9: Rate $R = 1/2$ DRS–Code with splitted interleaver and separated scramblers.

As doping is transmission of systematic information it is also possible to depict the doped bits of *both* branches as *one* systematic branch. Then, the bit positions in the scrambled branches that correspond to doped systematic bits have to be punctured by a puncturer *pct*. But, as only a small number of systematic bits are used, there also has to be a puncturer $\overline{pct}$ in the systematic branch. The period length of $\overline{pct}$ is only half of *pct* as systematic bits for both scramblers have to be provided.

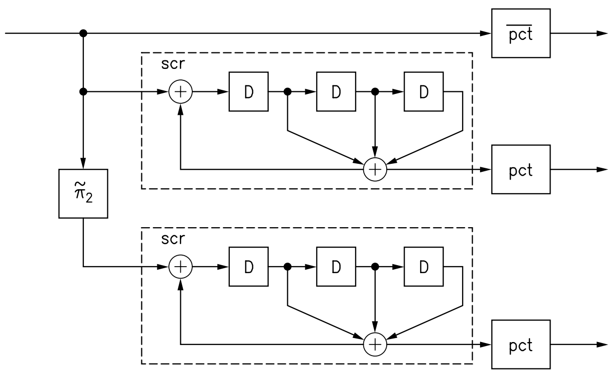

Figure 10: Turbo–Code representation of rate $R = 1/2$ DRS–Codes.

The conversion from serial to parallel representation of the encoder improves not only the message passing during iterative decoding, but also the use of systematic information. Hence, as in the case of RA–Codes the decoding complexity is halved, but also an additional gain is obtained by using the systematic symbols in both component decoders.

Analysis of the asymptotic behavior for infinite block length, as in the case of RA–Codes, is performed by means of EXIT charts in Fig. 11 and Fig. 12.

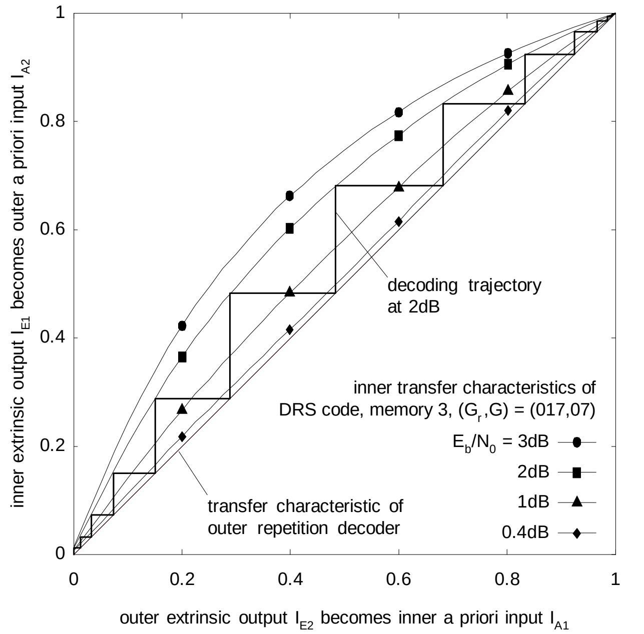

Figure 11: Extrinsic Information Transfer (EXIT) Chart for serial representation of the DRS–Code.

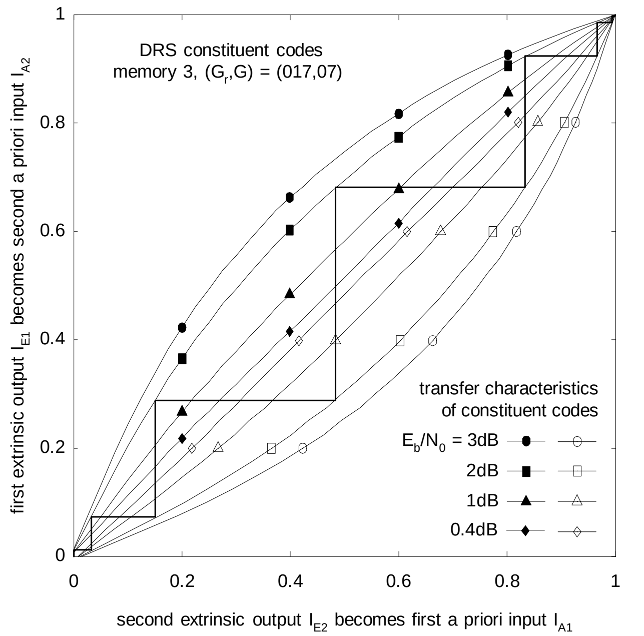

Figure 12: Extrinsic Information Transfer (EXIT) Chart for parallel representation of the DRS–Code.

The expected decoding trajectory for the parallel representation only needs half the number of lines compared to the simulation result of the trajectory of the serial DRS–Code. Hence, convergence is achieved by half the number of iterations. The gain of improved use of the systematic information is also covered in the graphical representation of the EXIT chart. As now all transmitted systematic bits from both component codes are available within every branch the rate of the component codes is lowered. Hence, the transfer characteristics of the component codes are slightly improved.

Simulation results of the DRS–Code verify the above results for finite block length. There is no drawback in restricting the interleaver to the proposed structure resulting in the equivalent parallel representation. But as the decoder structure is improved to make use of the extrinsic information as soon as possible, convergence of iterative decoding can be achieved with half the computational complexity. In the simulations of Fig. 13 a block length of $N = 20000$ and 50 iterations in the parallel and 100 iterations in the serial representation was used.

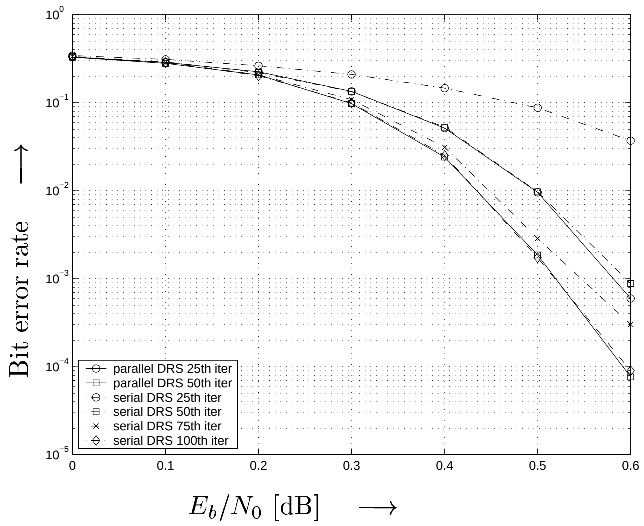

Figure 13: Performance of the rate $R = 1/2$ DRS–Code in serial and parallel representation in dependence of number of iteration (Blocklength $K = 10000$).

Fig. 13 shows that iterative decoding of the DRS–Code based on the parallel decoder structure is advantageous with respect to the number of iterations. In addition to that, the BER performance is slightly better for the parallel decoder owing to better exploiting the systematic (doped) bits in both half–iterations.

Similar results can be achieved in the conversion of the rate $R = 1/3$ DRS–code presented in [6] to the parallel representation. In this case we can expect a reduction of number of iteration by a factor of three. Furthermore the systematic (doped) bits should help to improve the BER performance even more than in the rate $R = 1/2$ case, as in the parallel representation three times the number of systematic bits compared to the serial representation are available during decoding of a constituent code.

In [6] a recursive scrambler of memory $m = 2$ with feedback polynomial $G_r = 6$ and feedforward polynomial $G = 7$ (octal representation) was found to be the matching counterpart to the outer rate $R = 1/3$ repetition code. After applying systematic doping with a doping ratio $D_r = n_s : n_c = 1 : 50$ convergence of iterative decoding is possible for the DRS–code depicted in Fig. 14 at $E_b/N_0 > -0.34$ dB, which is only 0.16 dB apart from the Shannon–Limit [12] for rate $R = 1/3$ codes.

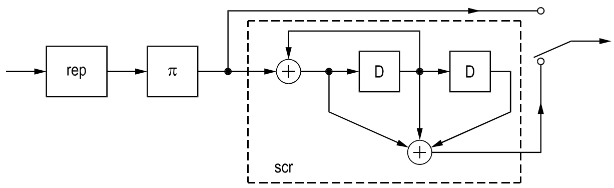

Figure 14: Original representation of rate $R = 1/3$ DRS–Codes.

Interleaver splitting and trellis separation leads to an encoder that consists of three parallel concatenated branches of doped scramblers. Adding a sytematic branch containing a puncturer $\overline{pct}$ and converting the doped scramblers to ordinary scramblers followed by a puncturer pct leads to the construction shown in Fig. 15.

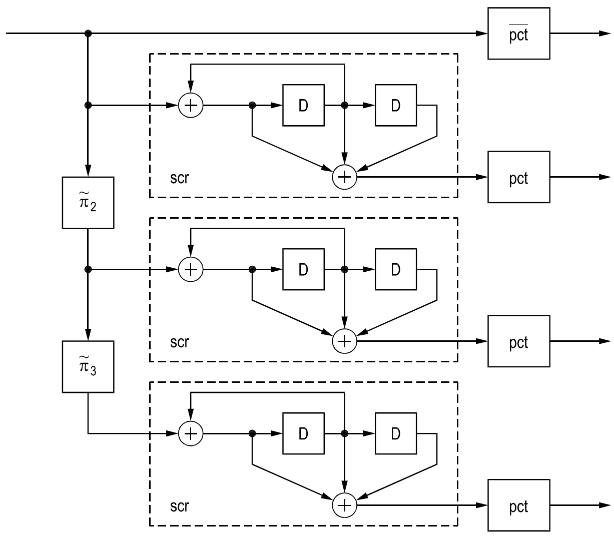

Figure 15: Turbo–Code representation of rate $R = 1/3$ DRS–Codes.

For this code an EXIT chart (not shown) can be plotted in three dimensions. It is symmetric with respect to the three axes, as all constituent codes are the same. As in the case of RA–Codes, the effective interleaving depth is unchanged, although $\tilde{\boldsymbol{\pi}}_2$ and $\tilde{\boldsymbol{\pi}}_3$ are each of length $K = N/3$. The advantage of improved message passing is even more emphasized in this example, as in the parallel representation a decoder only needs one third of the iterations needed to decode a DRS–Code in the serial representation.

Owing to the small memory of the constituent code the iterative decoder of this powerful code is of significantly lower complexity compared to the classical Turbo–Code of [3], which consists of a systematic branch and two branches with scramblers of memory $m = 4$ with feedback polynomial $G_r = 37$ and feedforward polynomial $G = 21$ (octal representation). Per decoded bit, an iterative decoder for the $R = 1/3$ DRS–Code visits 396 states, as it decodes three branches of constituent codes with 4 states within each of the 33 iterations. The iterative decoder for the classical Turbo–Code needs nearly

50% more operations, as it has to visit 576 states, if 18 iterations are assumed, which is required for a reasonable performance. Hence, the $R = 1/3$ DRS–Code outperforms the classical Turbo–Code by nearly 0.3 dB in power efficiency while having a significantly lower decoding complexity.

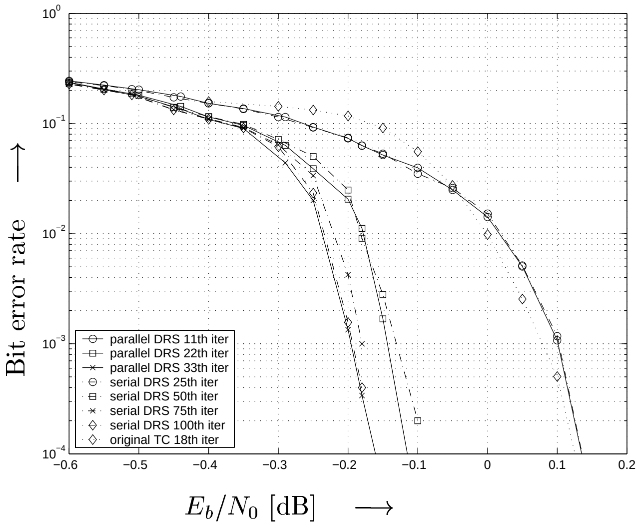

Figure 16: Performance of the rate $R = 1/3$ DRS–Code in serial and parallel representation in dependence of number of iteration (Blocklength $K = 10000$). For comparison the performance of the original $R = 1/3$ Turbo–Code [3] for $K = 10000$ and 18 Iterationen is also given.

Independent of our work, a code similar to the rate R=1/3 DRS-code in parallel representation has been found by Massey and Costello [10, 11], which is referred to as a "partially systematic code based on big-numerator/accumulator constituent code".

IV. Conclusion

The main result established in this paper is that the number of operations needed to iteratively decode a concatenated code strongly depends on the message passing. As soon as extrinsic information is available for all information bits it should be passed to the next component decoder.

In the construction of RA–Codes and the DRS–Code the inherent message passing suggested by the original encoder structure is suboptimal. Hence, with minor changes to the encoder, we were able to reduce the computational complexity of iterative decoding significantly.

References

[1] S. Benedetto and G. Montorsi, "Average performance of parallel concatenated block codes", *IEE Electronics Letters*, Vol.31, No.3, pp. 156–158, Feb. 1995.

[2] S. Benedetto and G. Montorsi, "Unveiling Turbo Codes: Some Results on Parallel Concatenated Coding Schemes", *IEEE Transactions on Information Theory*, Vol.32, No.2, pp. 409–428, Mar. 1996.

[3] C. Berrou, A. Glavieux, and P. Thitimashima, "Near Shannon Limit Error–Correcting Coding and Decoding: Turbo–Codes". *Proc. of ICC '93*, pp. 1064-1070, 1993.

[4] S. ten Brink, "Convergence of iterative decoding", *IEE Electronics Letters*, Vol.35, No.10, pp. 806–808, May 1999.

[5] S. ten Brink, "A rate one–half code for approaching the Shannon limit by 0.1dB", *IEE Electronics Letters*, Vol.36, No.15, pp. 1293–1294, July 2000.

[6] S. ten Brink, "Designing Iterative Decoding Schemes with the Extrinsic Information Transfer Chart", *AEÜ Int. J. Electron. Commun.*, No.6, pp. 389–398, Dec. 2000.

[7] "RA codes Achieve AWGN Channel Capacity", with R. J. McEliece. 13th AAECC Proceedings, Springer Publication 1719. p. 10-18,1999.

[8] R. G. Gallager, *Low Density Parity Check Codes*. PhD thesis, MIT, Cambridge, Mass., September 1960.

[9] H. Jin, and R. McEliece, "RA Codes Achieve AWGN Channel Capacity". *13th AAECC Proceedings*, Springer Publication 1719. pp. 10–18, 1999.

[10] P. C. Massey, "A 4-state Alternative Code to the Proposed 8-state Turbo Coding Standard", to be presented at the *IEEE Communication Theory Workshop*, Borrego Springs, CA, May 2001.

[11] P. C. Massey and D. J. Costello, Jr., "Turbo Codes with Recursive Nonsystematic Quick-Look-In Constituent Codes", to be presented at the *IEEE International Symposium on Information Theory*, Washington, DC, June 2001.

[12] C. Shannon, "A mathematical theory of communication". *Bell Systems Technical Journal*, pp. 379–423,623–656, 1948.