



HAL
open science

Low-complexity high-rate Reed-Solomon block turbo Codes

Rong Zhou, Raphaël Le Bidan, Ramesh Pyndiah, André Goalic

► **To cite this version:**

Rong Zhou, Raphaël Le Bidan, Ramesh Pyndiah, André Goalic. Low-complexity high-rate Reed-Solomon block turbo Codes. IEEE Transactions on Communications, 2007, 55 (9), pp.1656 - 1660. 10.1109/TCOMM.2007.904365 . hal-01876249

HAL Id: hal-01876249

<https://hal.science/hal-01876249>

Submitted on 2 May 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Low-Complexity High-Rate Reed–Solomon Block Turbo Codes

Rong Zhou, *Member, IEEE*, Raphaël Le Bidan, *Member, IEEE*, Ramesh Pyndiah, *Senior Member, IEEE*,
and André Goalic, *Member, IEEE*

This letter considers high-rate block turbo codes (BTC) obtained by concatenation of two single-error-correcting Reed–Solomon (RS) constituent codes. Simulation results show that these codes perform within 1 dB of the theoretical limit for binary transmission over additive white Gaussian noise with a low-complexity decoder. A comparison with Bose–Chaudhuri–Hocquenghem BTCs of similar code rate reveals that RS BTCs have interesting advantages in terms of memory size and decoder complexity for very-high-data-rate decoding architectures.

Index Terms—Block turbo codes (BTC), Reed–Solomon codes, product codes.

I. INTRODUCTION

Since their introduction in 1993 [1], turbo codes have received widespread interest within the digital communications community. The original concept of iterative soft-input soft-output (SISO) decoding of concatenated convolutional codes has been extended to block turbo codes (BTC) [2] and to low-density parity-check (LDPC) codes [3]. Today, turbo codes have become a crucial industrial technology for error correction in digital transmission systems since they offer an excellent tradeoff between complexity and performance.

This letter focuses on BTCs constructed from Reed–Solomon (RS) constituent codes. BTCs provide an interesting alternative to convolutional turbo codes and LDPC codes for applications requiring high code rates ($R > 0.8$) and high-data-rate decoders. BTCs were originally introduced using binary Bose–Chaudhuri–Hocquenghem (BCH) constituent codes. BCH BTC is now an efficient and mature technology already in use in several proprietary satellite transmission systems, and has been recently adopted as an option in the IEEE 802.16 Standard for wireless Metropolitan Area Networks (MAN). Innovative architectures have been proposed that can achieve decoding speeds of several gigabits per second [4], [5]. An experimental demonstration of a forward error correction (FEC) for 10-Gb/s optical communication systems based on a BCH BTC has been recently reported in [6]. However, an important limitation of BCH BTCs is that very large code lengths ($>60\,000$ bits) are required to achieve high code rates ($R > 0.9$). This limitation may conflict with practical systems constraints. In particular, a large block length increases the decoding latency and also the memory complexity of a decoder architecture.

The first investigations on RS BTCs were reported in [7], considering the expurgated RS constituent codes decoded with the Chase–Pyndiah SISO algorithm. However, the performance obtained were quite far (>1.6 dB) from the theoretical limit. Similar results were observed with extended RS constituent codes. In retrospect, this was due to a limitation in the decoder complexity. For example, Sweeney showed in 2000 that this gap could be reduced to less than 1 dB by using the Dorsch algorithm [8] instead of the Chase algorithm. The Dorsch algorithm (and other variants such as ordered statistics decoding [9]) generates candidate codewords for soft-output computation by successive reencoding of the most reliable independent bits in the received word [10]. In this way, each algebraic decoding attempt is replaced by a low-complexity reencoding operation. Hence, more candidate codewords can be considered (50 in [8] versus 17 in [7]), and better performance can be achieved. However, even though reencoding has little cost, the Dorsch algorithm still involves a sorting step and a Gaussian elimination. These two functions remain very complex in terms of number of operations. In addition, they do not lend themselves readily to parallel processing. Nevertheless, these results confirm the potential of RS BTCs.

In this letter, we consider the RS BTCs obtained by concatenation of two nonextended nonexpurgated single-error-correcting (SEC) RS constituent codes. Two constructions for the product code, i.e., symbol and binary concatenation, are described and compared. We show that the two constructions can provide near-optimum performance with a low-complexity Chase–Pyndiah decoder using only 16 test patterns. A comparison with binary BCH BTCs of similar code rate reveals that RS BTCs based on symbol concatenation have interesting practical advantages in terms of memory size and decoder complexity. Performance over the binary symmetric channel (BSC) and the binary erasure channel (BEC) are also investigated.

This letter is organized as follows. The construction of the RS product code is described in Section II. The corresponding iterative decoding algorithm is developed in Section III. Simulation results for binary transmission over an additive white Gaussian noise (AWGN) channel and a comparison with BCH BTCs are presented in Section IV. Performance of RS BTCs over the BSC and the BEC are analyzed in Section V. Conclusions follow in Section VI.

The authors are with the Signal and Communication Department, CNRS TAMCIC (UMR 2872) Research Laboratory, Groupe des Ecoles Télécommunications/Ecole Nationale Supérieure des Télécommunications (GET/ENST), Bretagne Technopôle Brest-Iroise, Brest Cedex 3, France (e-mail: rong.zhou@enst-bretagne.fr; raphael.lebidan@enst-bretagne.fr; ramesh.pyndiah@enst-bretagne.fr; andre.galic@enst-bretagne.fr).

II. RS PRODUCT CODES

The concept of product codes introduced by Elias in 1954 offers a simple and efficient method to construct powerful codes from simple constituent codes [11]. Classical product codes are obtained by serial concatenation of two linear block codes $\mathcal{C}_1(n_1, k_1, d_1)$ and $\mathcal{C}_2(n_2, k_2, d_2)$ over $\text{GF}(2^q)$, separated by a symbol-by-symbol row-column interleaver. Parameters n_i, k_i , and d_i ($i = 1, 2$) stand for the length, dimension, and minimum distance of each component code, respectively. The resulting product code $\mathcal{P}(n_p, k_p, d_p)$ has length $n_p = n_1 n_2$, dimension $k_p = k_1 k_2$, rate $R_p = R_1 R_2$, and minimum distance $d_p = d_1 d_2$. The n_1 rows of the product code are codewords of \mathcal{C}_1 , and the n_2 columns are codewords of \mathcal{C}_2 [12].

We focus here on RS product codes using nonextended and nonexpurgated $(2^q - 1, 2^q - 3, 3)$ SEC RS constituent codes over $\text{GF}(2^q)$. The codes have generator polynomial $g(x) = (x - \alpha)(x - \alpha^2)$, where α is a primitive element of $\text{GF}(2^q)$. Each element in $\text{GF}(2^q)$ can be represented by q bits using some basis of $\text{GF}(2^q)$ over $\text{GF}(2)$. The polynomial basis $\{1, \alpha, \dots, \alpha^{q-1}\}$ is the usual choice, although other bases exist [12]. Hence, two different constructions arise for the RS product code, depending on whether the concatenation is performed at the symbol level or at the bit level.

As shown in Fig. 1(a), symbol concatenation is obtained by encoding a $k_1 \times k_2$ 2^q -ary information matrix along the first k_1 rows, and then along the n_2 columns. This yields a $n_1 \times n_2$ coded matrix with elements in $\text{GF}(2^q)$. The binary image of the symbol product code has length $n_1 n_2 q$ bits and dimension $k_1 k_2 q$ bits. The second construction, referred to as binary concatenation, is depicted in Fig. 1(b). Starting from a $k_1 q \times k_2 q$ binary information matrix, a $n_1 q \times n_2 q$ binary coded matrix is obtained by encoding along the first $k_1 q$ rows, and then along the $n_2 q$ columns using the binary image of the RS constituent codes. The resulting binary product code has length $n_1 n_2 q^2$ bits and dimension $k_1 k_2 q^2$ bits. It is to be noted that these parameters are q times larger than for symbol concatenation. Both constructions will be considered and compared in the following.

III. TURBO DECODING OF RS PRODUCT CODES

Given a bit-level observation matrix \mathbf{R} in log-likelihood ratio (LLR) form, iterative “turbo” decoding of product codes is realized by decoding successively the rows and columns of \mathbf{R} using a SISO decoder, and exchanging extrinsic information between the two decoding processes.

A. SISO Decoding of RS Codes

The RS SISO decoder consists of a soft-input hard-output (SIHO) decoder combined with a soft-output computation unit. In this letter, the SISO decoding was performed at the bit level, independent of the construction (binary or symbol concatenation) of the product code.

The SIHO decoding of a row (or column) of the RS product code was realized using a modified version of the Chase-2 algorithm [13]. Given a soft-input sequence $\mathbf{r} = (r_{1,1}, \dots, r_{n,q})$ corresponding to a row or column of \mathbf{R} , the Chase-2 decoder first

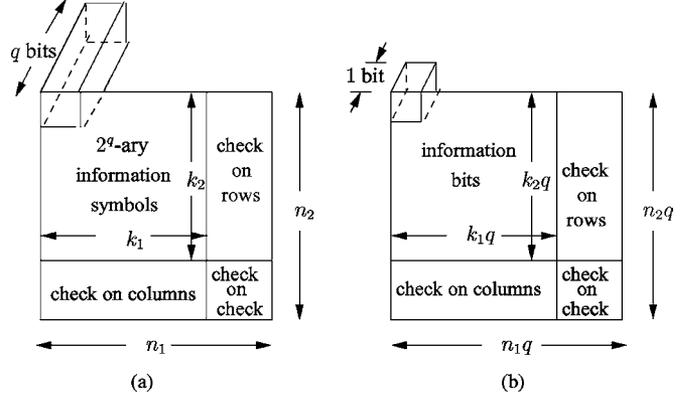


Fig. 1. RS product codes. (a) Based on symbol concatenation over $\text{GF}(2^q)$. (b) Based on binary concatenation.

forms the binary hard-decision sequence $\mathbf{y} = (y_{1,1}, \dots, y_{n,q})$ from \mathbf{r} . The reliability of the decision on the j th bit in the i th RS symbol is measured by the magnitude $|r_{i,j}|$ of the corresponding soft input. 2^s error patterns are generated by considering all possible combinations of 0 and 1 in the s least reliable bit positions. These error patterns are added to the hard-decision sequence \mathbf{y} to form candidate sequences. Successive algebraic decodings of the candidate sequences returns a list containing at the most 2^s distinct candidate codewords. Among them, the codeword \mathbf{d} at minimum Euclidean distance from the observation \mathbf{r} is selected as the final decision. The direct method of Peterson–Gorenstein–Zierler (PGZ) was used for algebraic decoding of the SEC constituent RS codes, since it has very low complexity in this context compared to other decoding procedures, such as the Berlekamp–Massey shift-register synthesis approach or the Euclidean algorithm. Specifically, apart from the syndromes computation, the PGZ algorithm involves only two divisions in $\text{GF}(2^q)$ to obtain the position and the value of an error [14, Ch. 6]. Also note that the syndromes need not be fully recomputed at each decoding attempt but can be simply updated based on the bits that have been flipped by the current test pattern [15].

Soft-output computation was performed according to the method described in [2]. For a given bit j in the i th RS symbol, the list of candidate codewords is searched for a *competing* codeword \mathbf{c} at minimum Euclidean distance from \mathbf{r} and such that $c_{i,j} \neq d_{i,j}$. If such a codeword exists, the soft output $r'_{i,j}$ on this bit is given by

$$r'_{i,j} = \left(\frac{\|\mathbf{r} - \mathbf{c}\|^2 - \|\mathbf{r} - \mathbf{d}\|^2}{4} \right) d_{i,j} \quad (1)$$

where $\|\cdot\|^2$ denotes the squared norm of a sequence. Otherwise, the soft output is computed as

$$r'_{i,j} = r_{i,j} + \beta d_{i,j} \quad (2)$$

where β is a predefined positive constant that increases with the iterations. We note that a slightly better performance can be obtained by using a variable β , as suggested in [16]. The extrinsic information $w_{i,j}$ relative to bit j in symbol i is finally given by the difference between the soft output $r'_{i,j}$ computed

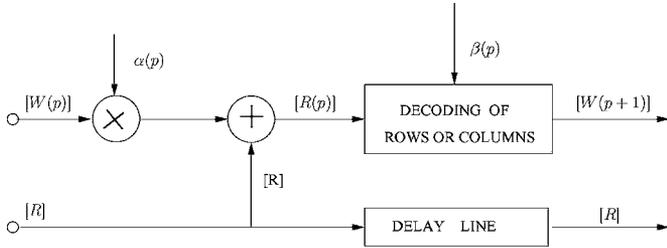


Fig. 2. Block diagram of the turbo decoder.

by the SISO decoder and the soft-input sample $r_{i,j}$

$$w_{i,j} = r'_{i,j} - r_{i,j}. \quad (3)$$

B. Iterative Decoding of the Product Code

The block diagram of the turbo decoder at the p th half iteration is depicted in Fig. 2. A half iteration stands for a row or column decoding step and one iteration comprises of two half iterations. The SISO decoder's input matrix at half iteration p is given by

$$\mathbf{R}(p) = \mathbf{R} + \alpha(p)\mathbf{W}(p) \quad (4)$$

where $\alpha(p)$ is a scaling factor used to reduce the influence of extrinsic information, and $\mathbf{W}(p)$ is the extrinsic information matrix delivered by the SISO decoder at the previous half iteration. Decoding stops when a given maximum number of iterations has been performed, or when an early-termination condition (stop criterion) is met.

C. Stop Criterion

A stop criterion improves the convergence of the iterative decoding process, and reduces the average power consumption of the decoder by decreasing the average number of iterations required to decode a block. An efficient stop criterion is readily derived based on the particular structure of the product code. If all the rows (resp. columns) after column (resp. row) decoding at a given half iteration have zero syndromes, then the decoding algorithm has converged and the decoding process is stopped.

IV. PERFORMANCE OVER AWGN

A. Simulation Results

We have simulated the performance of three RS product codes based on nonextended nonexpurgated SEC RS codes over GF(16), GF(32), and GF(64), respectively, using quadrature phase-shift keying (QPSK) signaling over AWGN. Both the binary and symbol concatenation have been considered. The parameters of the codes are summarized in Table I. For each code, we have computed the corresponding theoretical limit (TL) given by the sphere-packing bound (SPB) at a target frame error rate (FER) of 10^{-3} . The SPB provides a lower bound on the minimum E_b/N_0 required to achieve a given FER with a finite length code [17]. We note that the computation of the bound includes here a correction term to account for the capacity penalty resulting from the use of finite binary inputs.

TABLE I
RS PRODUCT CODES PARAMETERS FOR SYMBOL AND BINARY
CONCATENATION, AND CORRESPONDING TL AT A TARGET FER OF 10^{-3}

RS product code	N (bits)	K (bits)	R	$(E_b/N_0)_{TL}$
$(15, 13)^2$ symbol	900	676	0.7511	2.75 dB
$(15, 13)^2$ binary	3600	2704	0.7511	2.22 dB
$(31, 29)^2$ symbol	4805	4205	0.8751	3.38 dB
$(31, 29)^2$ binary	24025	21025	0.8751	3.10 dB
$(63, 61)^2$ symbol	23814	22326	0.9375	4.15 dB
$(63, 61)^2$ binary	142884	133956	0.9375	4.05 dB

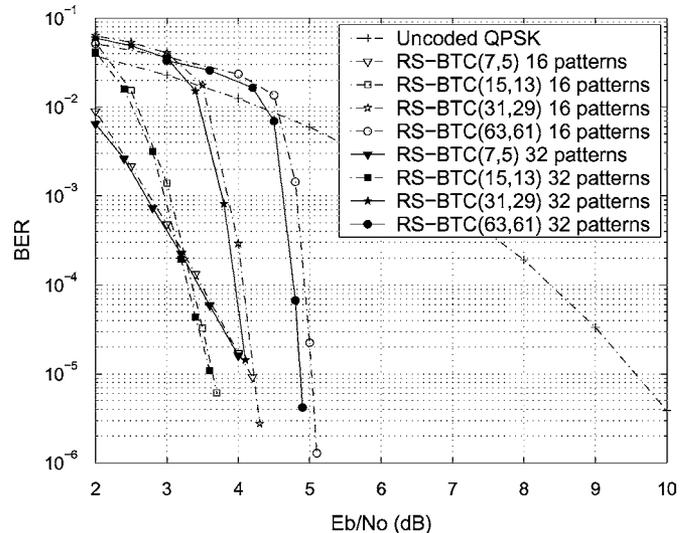


Fig. 3. Performance of RS BTCs with symbol concatenation and QPSK modulation over AWGN after eight turbo decoding iterations.

The two concatenation schemes have been simulated for each code, using 16 and 32 test patterns for the Chase-Pyndiah algorithm, respectively. A maximum of eight iterations with stop criterion was considered in the turbo decoding process. The extrinsic information was normalized as described in [7]. The following values were used for the coefficients α and β :

$$\begin{aligned} \alpha &= (0.00, 0.10, 0.20, 0.25, 0.30, 0.35, 0.40, 0.45, \\ &\quad 0.50, 0.55, 0.60, 0.65, 0.70, 0.90, 1.00, 1.00) \\ \beta &= (0.20, 0.30, 0.40, 0.50, 0.55, 0.60, 0.65, 0.70, \\ &\quad 0.75, 0.80, 0.85, 0.90, 1.00, 1.00, 1.00, 1.00). \end{aligned}$$

The simulation results are shown in Figs. 3 and 4. We first note that although the considered product codes have a small binary minimum distance (only nine, as verified by computer search), no error floor is observed down to a bit-error rate (BER) of 10^{-6} . It is worth mentioning that eight iterations are indeed required to obtain near-optimum performance, whereas six iterations are usually sufficient for BCH BTCs. We also observe that doubling the number of test patterns yields a minor improvement (≤ 0.15 dB) for both concatenation schemes. We found that a higher number of test patterns do not bring additional coding gain. This suggests that 16 test patterns provide an ideal tradeoff between complexity and performance in this context.

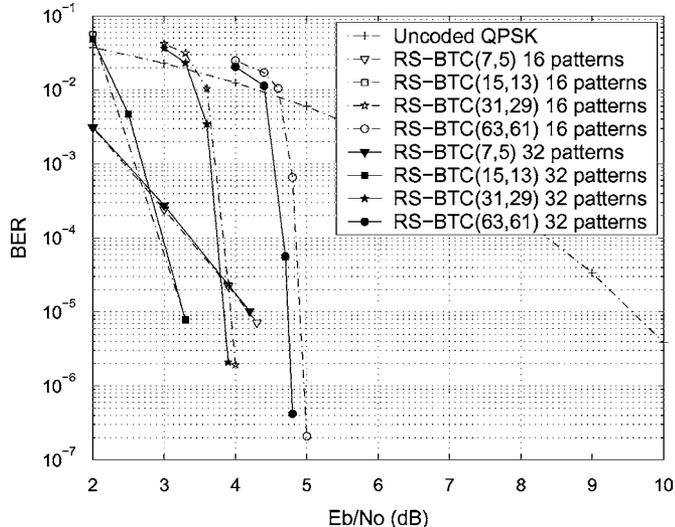


Fig. 4. Performance of RS BTCs with binary concatenation and QPSK modulation over AWGN after eight turbo decoding iterations.

TABLE II
 E_b/N_0 GAP BETWEEN THE SIMULATED PERFORMANCE OF RS BTCs AND THE CORRESPONDING TL AT AN FER OF 10^{-3}

Symbol concatenation	16 test patterns	32 test patterns
RS(15, 13) ²	0.80 dB	0.80 dB
RS(31, 29) ²	0.80 dB	0.75 dB
RS(63, 61) ²	0.80 dB	0.70 dB
Binary concatenation	16 test patterns	32 test patterns
RS(15, 13) ²	1.00 dB	1.00 dB
RS(31, 29) ²	0.80 dB	0.70 dB
RS(63, 61) ²	0.80 dB	0.65 dB

We have measured the gap between the simulated performance of the codes and the corresponding TL (given in Table I) at a target FER of 10^{-3} . The results are given in Table II. We notice that the two concatenation schemes perform within 1 dB of the TL. However, although they exhibit similar performance, the two concatenations are not equivalent in terms of complexity. Binary concatenation has a code length q times larger than symbol concatenation, and also involves q times more SISO decoding operations per iteration. Hence, symbol concatenation seems preferable to binary concatenation from the complexity point of view.

B. Comparison With BCH BTCs

A comparison between RS BTCs based on symbol concatenation and binary BCH BTCs of similar code rate has been conducted for QPSK transmission over AWGN. We have measured the gap Δ between the simulated performance of the codes and the corresponding TL at a target FER of 10^{-3} . This is a fair comparison since the TL takes the block size into account. Simulation results for BCH BTCs have been obtained from [2] and [18]. The gaps are given in Table III. We observe that although the two solutions provide near-optimum performance within 1 dB of the TL, RS BTCs have an information block size K that is about three times smaller than the corresponding BCH BTCs. A detailed comparison between FPGA

TABLE III
 E_b/N_0 GAP Δ BETWEEN SIMULATED PERFORMANCE AND THE CORRESPONDING TL FOR BCH BTCs AND RS BTCs BASED ON SYMBOL CONCATENATION AND OF SIMILAR CODE RATE AT AN FER OF 10^{-3}

Product code	K (bits)	R	$(E_b/N_0)_{TL}$	Δ
RS(15, 13) ²	676	0.7511	2.75 dB	0.80 dB
BCH(64, 57) ²	3249	0.7932	2.53 dB	0.88 dB
RS(31, 29) ²	4205	0.8751	3.38 dB	0.80 dB
BCH(128, 120) ²	14400	0.8789	3.18 dB	0.83 dB
BCH(256, 239) ²	57121	0.8716	3.01 dB	0.90 dB
RS(63, 61) ²	22326	0.9375	4.15 dB	0.80 dB
BCH(256, 247) ²	61009	0.9309	3.94 dB	0.70 dB

turbo decoding architectures for the BCH(128, 120)² and the RS(31, 29)² BTCs has been conducted in [19]. This analysis shows that the smaller block size of RS BTCs yields a considerable reduction of 67% in memory complexity. In addition, the decoding latency is also reduced. The complexity (in terms of number of logic elements) of the SISO decoder for the RS code is only 13% larger in comparison with the BCH code. On the other hand, RS BTCs based on symbol concatenation involve less SISO decoding operations since data processing is performed at the symbol level. For example, the BCH(128, 120)² requires a total of $6 \times 2 \times 128 = 1536$ SISO decodings for a maximum of six iterations. In contrast, the RS(31, 29)² needs a total of $8 \times 2 \times 31 = 496$ SISO decodings for a maximum of eight iterations. Very-high-data-rate turbo decoding architectures require simultaneous decoding of the rows and columns of the received matrix [4], [5]. Therefore, RS BTCs compare favorably in terms of complexity with BCH BTCs in this context since fewer elementary decodings have to be realized in parallel.

V. PERFORMANCE OVER THE BSC AND THE BEC

We finally consider the performance of high-code-rate RS BTCs over the BSC and the BEC. These channel models naturally arise in many applications where soft channel outputs may not be available at the turbo decoder input for economical (data storage systems) or technological (gigabits per second optical transmissions) reasons.

The BSC is a discrete channel model with binary inputs and binary outputs. Transmission over the BSC can be modeled by a BPSK transmission over an AWGN channel followed by a binary threshold detector. Depending on the sign of the received samples, the threshold detector delivers binary decisions $+A$ and $-A$ to the block turbo decoder. As a result, all soft inputs are equally reliable at the beginning of the iterative decoding process. If we now apply two symmetric thresholds at the output of the AWGN channel, we obtain a binary erasure equivalent channel model with three outputs: $-A$, 0, and $+A$, where 0 denotes a bit erasure. It is to be noted that in contrast to the classical definition of the BEC, our model takes both errors and erasures into account.

The performance of RS BTCs based on symbol concatenation with high-rate SEC component codes have been evaluated by simulation over these two channel models. Performance curves are available in [20]. We have measured the E_b/N_0 gap between

TABLE IV

E_b/N_0 GAPS Δ BETWEEN SOFT INPUTS AND QUANTIZED INPUTS FOR RS BTCS BASED ON SYMBOL CONCATENATION AT A BER OF 10^{-5}

Product code	$(E_b/N_0)_{\text{AWGN}}$	Δ_{BSC}	Δ_{BEC}
RS(15, 13) ²	3.64 dB	2.02 dB	0.74 dB
RS(31, 29) ²	4.23 dB	1.67 dB	0.49 dB
RS(63, 61) ²	5.03 dB	1.33 dB	0.34 dB

the performance obtained with unquantized soft inputs (AWGN channel) and the performance obtained with binary (BSC) and ternary inputs (BEC) at a target BER of 10^{-5} . The corresponding gaps denoted by Δ_{BSC} and Δ_{BEC} , respectively, are given in Table IV.

As expected, we observe that for a given code, the performance degradation is smaller for ternary inputs than for binary inputs. We also note that the performance gap decreases as the code length increases. In particular, the gap is only 1.33 dB for the RS(63, 61)² BTC over the BSC. This is much smaller than the 2–3 dB gap usually observed with classical binary codes. In addition, the gap reduces to only 0.34 dB over the BEC channel for the same code. This result is particularly promising and suggests that long high-rate RS BTCs may represent an attractive alternative to concatenated RS codes or BCH BTCs for the next-generation FEC in optical communication systems [21].

VI. CONCLUSION

We have shown that high-rate RS product codes based on nonextended nonexpurgated SEC constituent codes can achieve near-optimum performance over the AWGN channel with a low-complexity SISO decoder. Two concatenations have been described and compared. In particular, we have shown that RS BTCs based on symbol concatenation have interesting advantages in terms of decoder complexity and decoding latency for very-high-data-rate turbo decoding architectures, compared to binary BCH BTCs of similar code rate. We have also considered that multiple-error-correcting RS component codes may yield even better performance with the advantage of additional versatility in the product code design. However, this will, in turn, increase the overall complexity of the turbo decoder. Consequently, the search for efficient low-complexity SISO decoders for RS codes currently remains an important issue for future work.

ACKNOWLEDGMENT

The authors wish to thank A. Picart, E. Piriou, C. Jégo, and P. Adde for their contributions to this paper and for many stimulating discussions.

REFERENCES

- [1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near shannon limit error-correcting coding and decoding: Turbo-Codes," in *Proc. IEEE Int. Conf. Commun.*, vol. 2, Geneva, Switzerland, May 1993, pp. 1064–1070.
- [2] R. Pyndiah, "Near-optimum decoding of product codes: Block turbo codes," *IEEE Trans. Commun.*, vol. 46, no. 8, pp. 1003–1010, Aug. 1998.
- [3] R. G. Gallager, "Low-density parity-check codes," *IEEE Trans. Inf. Theory*, vol. 8, no. 1, pp. 21–28, Jan. 1962.
- [4] J. Cuevas, P. Adde, S. Kerouedan, and R. Pyndiah, "New architecture for high data rate turbo decoding of product codes," in *Proc. IEEE Global Telecommun. Conf. 2002*, vol. 2, Taipei, Taiwan, Nov., pp. 1363–1367.
- [5] C. Jego and P. Adde, "Row-column parallel turbo decoding of product codes," *Electron. Lett.*, vol. 42, no. 5, pp. 296–297, Mar. 2006.
- [6] T. Mizuochi *et al.*, "Forward error correction based on block turbo code with 3-bit soft decision for 10-Gb/s optical communication systems," *IEEE J. Sel. Topics Quantum Electron.*, vol. 10, no. 2, pp. 376–386, Mar. 2004.
- [7] O. Aitsab and R. Pyndiah, "Performance of Reed-Solomon block turbo codes," in *Proc. IEEE Global Telecommun. Conf. 1996*, vol. 1–3, London, U.K., Nov., pp. 121–125.
- [8] P. Sweeney and S. Wesemeyer, "Iterative soft-decision decoding of linear codes," *Inst. Electr. Eng. Proc. Commun.*, vol. 147, no. 3, pp. 133–136, Jun. 2000.
- [9] M. Fossorier and S. Lin, "Soft-decision decoding of linear block codes based on ordered statistics," *IEEE Trans. Inf. Theory*, vol. 41, no. 5, pp. 1379–1396, Sep. 1995.
- [10] B. G. Dorsch, "A decoding algorithm for binary block codes J -ary output channels," *IEEE Trans. Inf. Theory*, vol. IT-20, no. 3, pp. 391–394, May 1974.
- [11] P. Elias, "Error-free coding," *IEEE Trans. Inf. Theory*, vol. IT-4, no. 4, pp. 29–37, Sep. 1954.
- [12] F. J. Macwilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*. Amsterdam, The Netherlands: North-Holland, 1978.
- [13] D. Chase, "A class of algorithms for decoding block codes with channel measurement information," *IEEE Trans. Inf. Theory*, vol. IT-18, no. 1, pp. 170–182, Jan. 1972.
- [14] A. M. Michelson and A. H. Levesque, *Error-Control Techniques for Digital Communications*. New-York: Wiley, 1985.
- [15] S. A. Hirst, B. Honary, and G. Markarian, "Fast Chase algorithm with an application in turbo decoding," *IEEE Trans. Commun.*, vol. 49, no. 10, pp. 1693–1699, Oct. 2001.
- [16] P. Adde and R. Pyndiah, "Recent simplifications and improvements in block turbo codes," in *Proc. 2nd Int. Symp. Turbo Codes Related Topics*, Brest, France, Sep. 2000, pp. 133–136.
- [17] S. Dolinar, D. Divsalar, and F. Pollara, "Code performance as a function of block size," *JPL Telecommun. Mission Oper. Prog. Rep.*, vol. 42–133, pp. 21–28, May 1998.
- [18] R. Pyndiah and P. Adde, "Performance of high code rate BTC for non-traditional applications," in *Proc. 3rd Int. Symp. Turbo Codes Related Topics*, Brest, France, Sep. 2003, pp. 157–160.
- [19] E. Piriou, C. Jego, P. Adde, R. Le Bidan, and M. Jezequel, "Efficient architecture for Reed Solomon block turbo code," presented at the IEEE Int. Symp. Circuits Syst., Kos, Greece, May 2006.
- [20] R. Zhou, A. Picart, R. Pyndiah, and A. Goalic, "Potential applications of low complexity non-binary high code rate block turbo codes," in *Proc. IEEE Mili. Commun. Conf. 2004*, Monterey, CA, Oct. 2004.
- [21] T. Mizuochi, "Recent progress in forward error correction for optical communication systems," *IEICE Trans. Commun.*, vol. E88-B, no. 5, pp. 1934–1946, May 2005.