International Journal of Electronics Engineering, 3 (2), 2011, pp. 251–256

Performance Comparison of RS Code and Turbo Codes for Optical Communication

Reena Tyagi[1], Puneet Chandra Srivastava[1], Mahendra Kumar[2] & R. K. Singh[3]

[1]ECE Dept., Raj Kumar Goel Engg.College,Pilakhua (Ghaziabad), Mahamaya Technical University, Noida ,(UP), India

[2]Associate Professor,ECE Dept.,BIET, Jhansi (UP)

[3]Professor, ECE Dept., Kumaon Engg. College, Dwarhat (Almora), Uttarakhand

Email: reenatyagi94@gmail.com

Abstract: RS codes can be considered as serious competitors to turbo codes in terms of performance and complexity. They are based on a similar philosophy i.e. constrained random code ensembles and iterative decoding algorithms. In this paper, we present the performance comparison of RS code and block turbo codes. The RS code coupled with receive diversity techniques are employed as the error correction scheme over optical fiber communication Channels by employing binary pulse position modulation (BPPM) modulation scheme for optical fiber communication. The performance of codes is evaluated in term of bit error rate (BER) for a given value of Eb/No. Simulation results demonstrate that the bit error rate (BER) performance of the Turbo codes (concatenated RS codes) provide better random as well as burst error correction capability as compared to RS codes.

Keywords: Channel coding, RS Code, Turbo Code, Optical Communication Channel, Receiver Diversity, BER, BPPM, NECG.

1. INTRODUCTION

FEC is widely used in wired and mobile communication, deep space communication as well as data storage systems. In the recent past, it has begun to find applications of FEC in optical links [10]. Error correcting codes are broadly classified in two categories, viz., Block Codes and Convolutional Codes. We use block codes in optical communication systems since they operate at very high data rate and by using block codes we can find low overhead codes that are capable of correcting random errors due to noise, and burst errors due to dispersion and inter-channel cross talk with special emphasis on complexity and cost. It is difficult to implement convolutional codes that operate at high code rate required for fiber-optic systems [9]. Algebraic block codes, such as Bose-Chaudhuri-Hocqueaghem (BCH) and Reed-Solomon (RS) codes are capable of correcting multiple bit-errors with the low overhead constraint.

In case of fiber-optic communication systems operating at very high data rate (Rc>0.8). While selecting an error correcting code one should take into account the practical limitation imposed by the hardware to make it feasible to introduce an overhead of (n-k) symbols. Thus, low overhead constraint becomes an important parameter while selecting FEC for optical communication application. In this paper, we have done a comparative analysis of the performance of different concatenated code using RS code considering the low overhead requirement as a prime design criterion.

2. ALGEBRIC DECODING ALGORITHMS FOR RS CODES

2.1. Berlekamp–Massey Algorithm for RS Codes

Decoding of the nonbinary RS codes involves not only determining the location of errors, but also their magnitudes. The error polynomial is

$$e(X) = e_{j_1} X^{j1} + e_{j_2} X^{j2} + \dots\dots\dots + e_{jl} X^{jl} \qquad (1)$$

For the RS case e_{j1}, e_{j2}, ... e_{ji} are field elements of GF (2^m). The 2t syndrome equations are

$$s_1 = e_{j1}(\beta_1) + e_{j2}(\beta_2) + \dots\dots\dots + e_{jl}(\beta_l)^3$$

$$s_2 = e_{j1}(\beta_1)^2 + e_{j2}(\beta_2)^2 + \dots\dots\dots + e_{jl}(\beta_l)^{2t}$$

$$s_3 = e_{j1}(\beta_1)^3 + e_{j2}(\beta_2)^3 + \dots\dots\dots + e_{jl}(\beta_l)^3$$

$$s_{2t} = e_{j1}(\beta_1)^{2t} + e_{j2}(\beta_2)^{2t} + \dots\dots\dots + e_{jl}(\beta_l)^{2t}$$

The σ_i's and s_j's are related by the Newton Identities and shown in [14] that the syndrome *Sj* can be expressed in recursive form as a function of σ_i's and the earlier syndromes s_{j-1}......., s_{j-i} such that

$$s_j = -(\sigma_l s_{j-l} + \sigma_{l-1} s_{j-l+1} + \dots\dots\dots + \sigma_1 s_{j-1} = -\sum_{i=1}^{l} \sigma_i s_{j-i} \qquad (2)$$

for $j = l + 1, l + 2, \dots\dots\dots\dots, 2t$

The error locator polynomial $\sigma(X)$ will be determined after $2t$ iterations instead of t iterations. As the roots of the error locator polynomial $\sigma(X)$ are determined, the task of the decoder is to find the error magnitude at the error location numbers. Now, we will utilize the following error evaluator polynomial $\Omega(X)$ having the degree equal to the degree of $\sigma(X)$.

$$\Omega(X) = [1 + s(X)]\sigma(X)$$

$$= (1 + s_1 X + s_2 X^2 + \ldots\ldots\ldots\ldots)(1 + \sigma_1 X + \sigma_2 X^2 + \ldots\ldots\ldots\ldots)$$

$$= 1 + (s_1 + \sigma_1)X + (s_2 + s_1\sigma_1 + \sigma_2)X^2 + (s_3 + s_2\sigma_1 + s_1\sigma_2 + \sigma_3)X^3 + \ldots\ldots\ldots\ldots \quad (3)$$

$$= 1 + \Omega_1 X + \Omega_2 X^2 + \ldots\ldots\ldots\ldots$$

Using $2t$ syndromes and coefficients of $\sigma(X)$, the error magnitudes are computed. The Forney algorithm [7] used to derive the error magnitudes. Equation 3 can be rewritten as

$$[1 + s(X)]\sigma(X) = (1 + \Omega_1 X + \Omega_2 X^2 + \ldots\ldots\ldots\ldots + \Omega_{2t} X^{2t}) \bmod X^{2t+1} \quad (4)$$

The equation can be rearranged in terms of the known polynomial $\sigma(X)$, the error locator polynomial $\Omega(X)$ and $s(X)$ the syndrome polynomial

$$\Omega(X) = [1 + s(X)]\sigma(X) \bmod X^{2t+1} \quad (5)$$

$\Omega(X)$ is naturally related to the error locations and error values by the t relations

$$\Omega(\beta_l^{-1}) = e_{jl} \prod_i (1 - \beta_i \beta_l^{-1}) : \textit{for } l = 1, 2, \ldots\ldots\ldots, t \quad (6)$$

The error magnitudes are computed using the expression

$$e_{jl} = \frac{-\beta_l \Omega(\beta_l^{-1})}{\sigma'(\beta_l^{-1})} \; \textit{for } l = 1, 2, \ldots\ldots\ldots, t \quad (7)$$

where $\sigma'(X)$ denotes the formal derivative of $\sigma(X)$ with respect to X.

The formal derivative is similar to the usual derivative, but does not have the same interpretation. If $f(X) = f_0 + f_1 X + f_2 X^2 + \ldots\ldots\ldots + f_n X^n + \ldots\ldots\ldots$ is a polynomial over GF(q), then the formal derivative $f'(X)$ is defined as $f'(X) = f_1 + 2f_2 X + \ldots\ldots\ldots\ldots + nf_n X^{n-1} + \ldots\ldots\ldots\ldots$ .The and quotient rules are applied to formal derivatives. Since $f(X) \in GF(2^m)$, then $f'(X)$ has no odd power term.

Applying the definition of formal derivative to $\sigma(X)$ equation 7 can be written as

$$e_{jl} = \frac{\Omega(\beta_l^{-1})}{\prod_{i=1, i\neq l}^{t} (1 + \beta_i \beta_l^{-1})} \quad (8)$$

where e_{ji} is the error value at the coordinate in r specified by β_l

The iterative procedure based on which the Berlekamp-Massey algorithm work is described in [3, 8, 13, 15]. After the $2t$ iterations we have $\sigma(X)$ whose roots are determined by the Chien search. The inverse of the roots are the error location numbers β_l's. Now we know where the errors are in r but not their values. The decoder has to find the error magnitudes at these locations, which is accomplished by the Forney algorithm. The decoding is accomplished by adding the error pattern to r over GF (q).

2.2. Euclid's Algorithm for BCH and RS Codes

The Euclid's algorithm is a recursive technique to find the greatest common divisor (GCD) of two polynomials [16]. If $f(X)$ and $g(X)$ are two polynomials where $\deg(f(X)) \geq \deg(g(X))$ then the GCD is computed dividing $f(X)$ by $g(X)$ recursively so that the algorithm always converges to a remainder polynomial $d(X) = 0$ and the last nonzero polynomial $d(X)$ is the GCD. The recursive relation between $f(X)$ and $g(X)$ is obtained by writing the initialization equation [6] such that

$$m(X) f(X) + n(X) g(X) = d(X), \quad (9)$$

where $m(X)$ and $n(X)$ are intermediate polynomials obtained during the division process.

The division process is explained in short

- At the i[th] iteration $m(X) f^{(i)}(X) + n(X) g^{(i)}(X) = d^{(i)}(X)$
- For the i+1[th] iteration $f^{(i+1)}(X) = g^{(i)}(x)$ and $g^{(i+1)}(X) = d^{(i)}(X)$
- The algorithm terminates when $d^{(i)}(X) = 0$ and $g^{(i)}(X)$ is the GCD of $f(X)$ and $g(X)$

In the case of decoding BCH and RS codes we are not interested in the GCD of $f(X)$ and $g(X)$ but the intermediate polynomials $m(X)$ and $n(X)$ at each iteration. We define a quotient polynomial [6].

$$q_i(X) = \left[\frac{d_{i-2}(X)}{d_{i-1}(X)}\right]_0^\infty \text{ where } [\ ]_0^\infty \text{ denotes non negative powers of } X \quad (10)$$

then $d_i(X) = d_{i-2}(X) - q_i(X) d_{i-1}(X)$

$$m_i(X) = m_{i-2}(X) - q_i(X) m_{i-1}(X) \text{ and} \quad (11)$$

$$n_i(X) = n_{i-2}(X) - q_i(X) m_{i-1}(X)$$

The initial conditions for the algorithm are

$$m_{-1}(X) = n_0(X) = 1; m_0(X) = n_{-1}(X) = 0$$

$$d_{-1}(X) = f(X) \textit{ and } d_0(X) = g(X)$$

At the i[th] iteration $m_i(X) f(X) + n_i(X) g(X) = d_i(X)$.

Coming back to the decoding of BCH/RS codes we re-write the key equation (5) for our analysis

$$\sigma(X)\left[1+s(X)\right]=\Omega(X)\left|\mathrm{mod}\ X\right|^{2t+1}$$

From equation (9), we can write

$$n_i(X)g(X)=d_i(X)\ \mathrm{mod}\ f(X) \qquad (12)$$

Compare equation (9) with the key equation

$$n_i(X)=\sigma(X), d_i(X)=\Omega(X)\ and\ f(X)=X^{2t+1}$$

To observe that the approach produces the desired solution to the key equation, we need to utilize the property of Euclid's algorithm [11] that states

$$\deg\left[n_i(X)\right]+\deg\left[d_{i-1}(X)\right]=\deg\left[f(X)\right]=2t+1 \text{ and}$$

$$\deg\left[n_i(X)\right]+\deg\left[d_i(X)\right]<2t+1$$

For $l \le t$ errors the solution of interest has

$$\deg\left[\Omega(X)\right]=\deg\left[\sigma_i(X)\right]\le t$$

There exists only one polynomial $\sigma(X)$ with degree not greater than t, which satisfies the key equation (5). This means that the intermediate result at the ith iteration provides the solution of our interest to the key equation. Thus, simply applying the Euclid's algorithm until deg $[d_i(X)] \le t$ gives solution to the key equation. Rest of the decoding involves finding the roots of $\sigma(X)$ by the Chien search. The inverses of the roots are the error location numbers in a usual manner.

3. BLOCK TURBO CODE

Concatenated coding schemes were first proposed by Forney [5] as a method for achieving large coding gains by combining two or more relatively simple building block or component codes (sometimes called constituent codes). Turbo codes were first introduced in 1993 by Berrou, Glavieux, and Thitimajshima [1,2], where a scheme is described to achieves a bit-error probability of $10^{-5,}$ using a rate 1/2 code over an additive white Gaussian noise (AWGN) channel and BPSK modulation at an *Eb/N*0 of 0.7 dB. The codes are constructed by using two or more component codes on different interleaved versions of the same information sequence. Whereas, for conventional codes, the final step at the decoder yields hard-decision decoded bits (or, more generally, decoded symbols), for a concatenated scheme such as a turbo code to work properly, the decoding algorithm should not limit itself to passing hard decisions among the decoders.

3.1. Principles of Iterative (Turbo) Decoding

In a typical communications receiver, a demodulator is often designed to produce soft decisions, which are then transferred to a decoder. The error-performance improvements of systems utilizing such soft decisions compared to hard decisions are typically approximated as 2 dB in optical AWGN. Such a decoder could be called a soft input/hard output decoder, because the final decoding process out of the decoder must terminate in bits (hard decisions) [1]. With turbo codes, where two or more component codes are used, and decoding involves feeding outputs from one decoder to the inputs of other decoders in an iterative fashion, a hard-output decoder would not be suitable. Because hard decisions into a decoder degrade system performance (compared to soft decisions). Hence, decoding of turbo codes in optical channels needs a soft input/soft output decoder. For the first decoding iteration of such a soft input/soft output decoder in optical communication, illustrated in Figure 1, we generally assume the binary data to be equally likely, yielding an initial a priori LLR value of L(d) = 0.

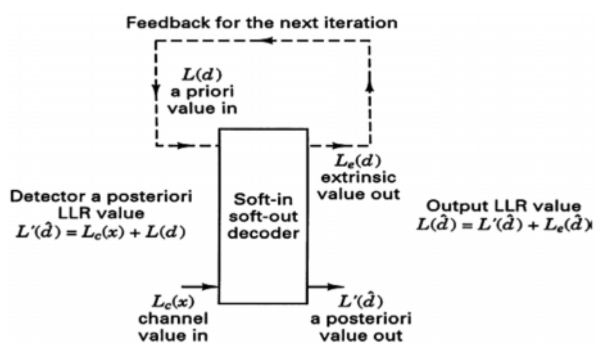

Figure 1: Soft Input/Soft Output Decoder (for a Systematic Code)

The channel LLR value, Lc(x), is measured by forming the logarithm of the ratio of the values of –ℓ1 and –ℓ2 for a particular observation of x as shown in Figure 2. The output L(dˆ) of the decoder in Fig. 1 is made up of the LLR from the detector, L2 (dˆ),and the extrinsic LLR output, Le(dˆ), representing knowledge gleaned from the decoding process. As illustrated in Figure 1, for iterative decoding, the extrinsic likelihood is fed back to the decoder input, to serve as a refinement of the a priori probability of the data for the next iteration.

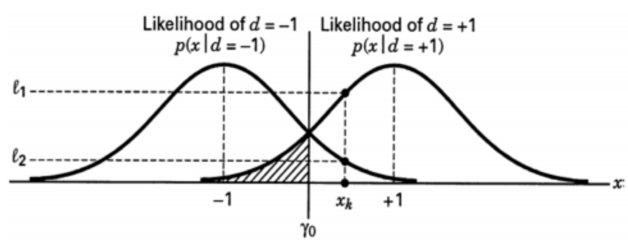

Figure 2: Likelihood Functions

4. PRODUCT CODE

Product codes (or iterated codes) [4, 12] are serially concatenated codes using two or more short block codes to form long block codes. If C^1 $(n_1, k_1, d_{min}1)$ and C^2 $(n_2, k_2, d_{min}2)$

are two systematic linear block codes, then the product code $P = C^1 \otimes C^2$ is obtained by

- placing ($k_1 * k_2$) information symbols in a matrix of k_1 rows and k_2 columns
- coding the k_1 rows using code C^2
- coding the n_2 columns using code C^1

The resultant product code [13] P (n, k, d_{min}) has $n = n_1 * n_2$, $k = k_1 * k_2$, $d_{min} = d_{min1} * d_{min2}$ and code rate is given by $R = R_1 * R_2$.

4.1. RS Product Code

In our work we have used the classical method for the construction of RS product codes, where the information symbol matrix contains $k_i * k_o$ q-ary information symbols. The codes C^1 and C^2 have the same code length $n_i = n_o$. The resultant code design scheme is easy to understand by Figure 3 where the span N of the interleaver is equal to the code length n_o of the outer code and the depth D is equal to the information length k_i of the inner code.

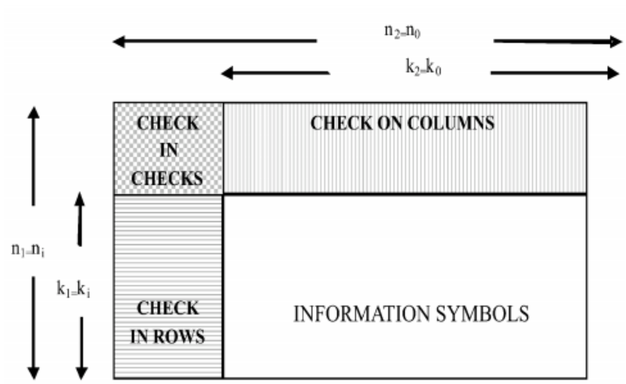

Figure 3 : Construction of Product Code

4.2. Decoding of RS Product Code

In the present paper, the product codes based on RS component codes are decoded by sequentially decoding the rows and columns of P by hard-decision algebraic bounded distance decoder for optical communication, which is sub-optimal but has significant reduced decoding complexity compared to the optimum ML decoder. We have simulated the different RS product codes on a Gaussian channel with sequential row by column hard-input/hard-output (HIHO) component decoders using the Berlekamp-Massey algorithm. Soft-decision decoding of the component RS codes [11] with SISO decoders will definitely provide additional coding gain but the decoding complexity will be extremely high. Another limiting factor is the high data rate at which optical communication systems operate. Implementing the SISO decoders at such a high data rate is impracticable. Further iterated decoding of RS product codes is performed using the component HIHO decoders.

5. SIMULATION RESULTS

Figure 4 depicts the serially concatenated RS (255, 239) and RS (255, 239) code with a row-column interleaver of length 255 bytes and depth 239 bytes on a image. The serially concatenated RS (255, 239) and RS (255, 223) with a row-column interleaver of length 255 bytes and depth 223 bytes on same image are shown in Figure 5. The output of the inner encoder is transposed to show that SCBC with row-column interleaver is apparently the product code.

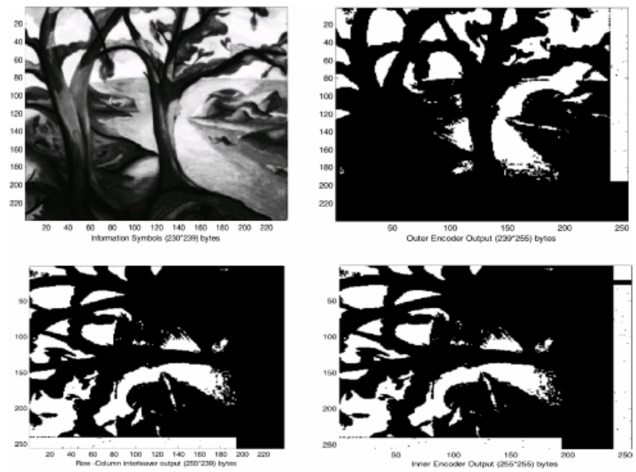

Figure 4: Serial Concatenation of RS Codes

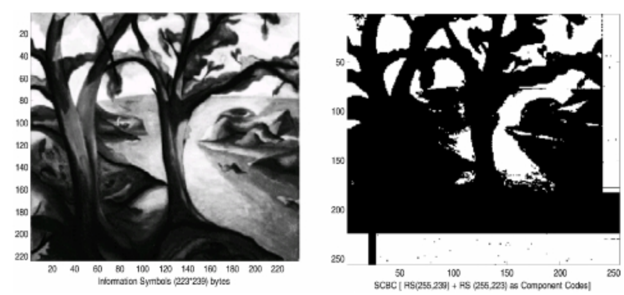

Figure 5: RS Product Code

The different RS code parameters are presented in Table 1. From Table 1 we observe as the code rate decreases the redundancy in the code, the minimum distance of the code and the asymptotic coding gain increases. The code rate and redundancy are inversely proportional to each other.

The analytical results for the candidate RS codes for BER are presented in Figure 6. All simulations are performed on Matlab piece-wise for an output BER of $\leq 10^{-8}$ for the candidate RS codes.

Table 1
RS Code Parameters

Candidate RS(n, k) Codes	*Code Rate (k/n)*	*Redundancy (n-k)/k (%)*	d_{min} *(2*t+1)*	$G_a(dB)$ *Asymptotic*
RS(255, 247)	0.9686	3.24	9	9.4
RS(255, 239)	0.9372	6.69	17	12.02
RS(255, 223)	0.8745	14.35	33	14.6

The simulated results presented in Figure 7 are confirmed with the analytical results in Figure 6.

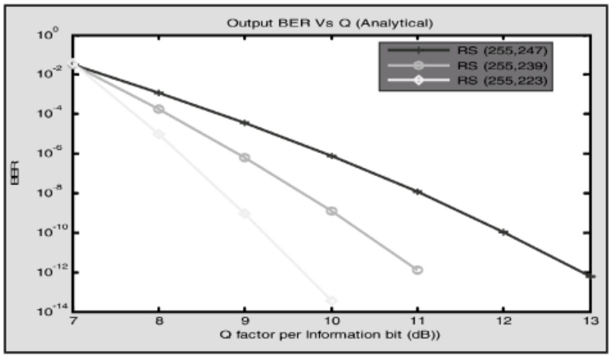

Figure 6: Theoretical Performance of the RS Codes

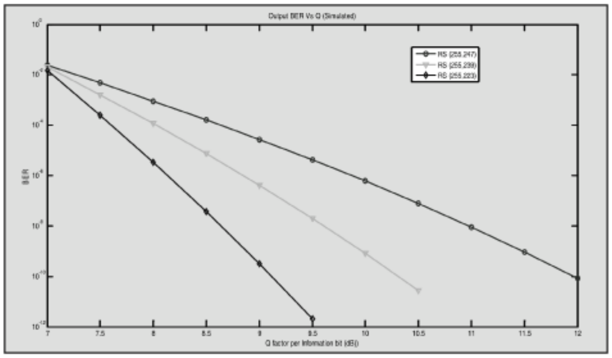

Figure 7: Simulated Performance of the RS Codes

Net electrical coding gain (NECG) is commonly used to quantify FEC performance and indicates an improvement in the SNR or Q factor at the receiver due to FEC. The comparison of the performance of the candidate RS codes in terms of coding gain is presented in Table 2.

Table 2
Coding Gain Comparison of the RS Codes at Output BER of 10^{-8}

Candidate RS (n,k) Codes	*Redundancy (n-k)/k (%)*	*NECG (dB) @ (Theoretical)*	*BER ≤ 10^{-8} (Simulated)*
RS (255,247)	3.24	3.5	3.37
RS (255,239)	6.69	4.42	4.28
RS (255,233)	14.35	5.25	5.15

It can be noticed from the Table 2, the increased redundancy in the code pays in terms of NECG. The RS (255, 239) code with almost twice the redundancy compared to the RS (255, 247) code offers roughly more than 1 dB additional coding gain. Approximately 2 dB additional coding gain is offered by the RS (255, 223) code with almost 4.5 times more redundancy than the RS (255, 247) code.

Concatenated codes provide an increased coding gain but with only a linear increase in hardware cost. Good design of the concatenated code lies in proper selection of the inner and the outer code to meet the low overhead requirement. We extend the assumptions used to compute the output BER of RS codes to evaluate the approximate analytical output BER of the concatenated RS codes. The RS product code parameters are presented in Table 3.

Table 3
RS Product Code Parameters

Candidate RS Product Codes	*Code Rate $R=R_1*R_2$*	*Redundancy (%) $(n_1*n_2-k_1*k_2)/ k_1*k_2$*	*$d_{min} = d_{min1} * d_{min2}$*
RS (255, 239) + RS (255, 239)	0.8784	13.83	289
RS (255, 239) + RS (255,223)mn	0.8196	22.00	561

The approximate analytical output BER for RS product codes are presented in Figure 8. The simulations were performed piece-wise for an output BER of ≤10^{-8} on Matlab. The simulated results for same RS product codes are presented in Figure 9. The simulated performance will be definitely poor than the approximate analytical performance. Since the decoding is a two-step decoding procedure based on HIHO component decoders, which is sub-optimal. Hence, the approximated analytical performance in Figure 8 can be considered as a lower bound on the output BER performance for the candidate RS product codes.

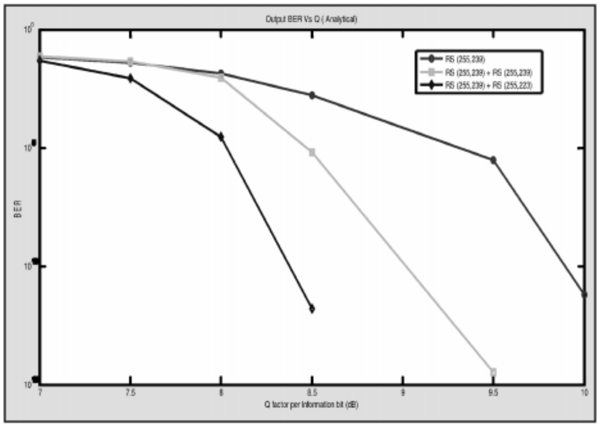

Figure 8: Approximate Output BER Performance of RS Product Codes

Table 4
Coding Gain Comparison of Codes with 14% Redundancy

Candidate Code	*Redundancy (%)*	*Code Rate*	*NECG (dB)@ 10^{-8}*
RS (255, 223)	14.35	0.8745	5.15
RS (255, 239) + RS (255, 239)	13.83	0.8784	5.3

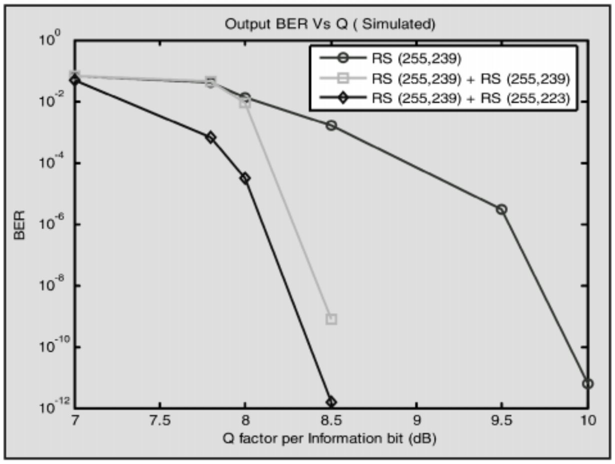

Figure 9: Simulated Output BER Performance of RS Product Codes

Further, we present the comparison in terms of net coding gain performance offered by the different coding schemes with comparable redundancy and code rate in Table 4.

6. CONCLUSION

It can be observed from Table 4, even though the candidate codes offer comparable performance for the same amount of redundancy, the computational complexity of the encoder and decoder for these candidate codes are not the same. The RS (255, 223) encoder has slightly less computational complexity compared to RS (255, 239) + RS (255, 239) encoder. Although, the RS (255, 223) decoder has higher computational complexity compared to the RS (255, 239) decoder, the decoder for RS (255, 239) + RS (255, 239) code involves two component RS (255, 239) decoders and a memory element for the deinterleaver. However, the overall encoder/decoder complexity for these two codes is comparable. From the results presented in Figure 6,7,8 and 9, we conclude that while designing an error correcting scheme for reliable communication we have to tradeoff not only redundancy against NECG but also complexity against NECG. So, Simulation results demonstrate that the bit error rate (BER) performance of the Turbo codes (concatenated RS codes) provide better random as well as burst error correction capability as compared to RS codes.

REFERENCES

[1] Berrou, C. and Glavieux, A., "Near Optimum Error Correcting Coding and Decoding: Turbo-Codes," *IEEE Trans. on Communications*, Vol. 44, No. 10, October 1996, pp. 1261-1271.

[2] Berrou, C., Glavieux, A. and Thitimajshima, P., "Near Shannon Limit Error-Correcting Coding and Decoding: Turbo Codes," *IEEE Proceedings of the Int. Conf. on Communications*, Geneva, Switzerland, May 1993 (ICC '93), pp. 1064-1070.

[3] E. R. Berlekamp, Algebraic Coding Theory, New York, McGraw-Hill, 1968.

[4] F. J. MacWilliams and N. J. A. Sloane, The Theory of Error Correcting Codes, Amsterdam: North Holland, 1977.

[5] Forney, G. D. Jr., Concatenated Codes (Cambridge, MA: MIT Press, 1966).

[6] G. Clark, J.Cain, Error-Correcting Coding for Digital Communications, New York, Plenum Press, 1981.

[7] G. D. Forney, On Decoding BCH Codes, IEEE Transactions on Information Theory, Vol. IT-11, pp. 549-557, October 1965.

[8] I. S. Reed, X. Chen, Error Control Coding For Data Networks, Kluwer Academic, Cop. 1999.

[9] ITU-T G.975, Forward Error Correction for Submarine Applications, October 2000

[10] N.Ramanujam, A.B. Puc, G.Lenner, H.D. Kidorf, C. R. Davidson, I. Hayee, J-X, Cai, M. Nissov, A. Pilipetskii, C.Rivers, N. S. Bergano, Forward Error Correction (FEC) Techniques in Long-haul Optical Transmission Systems, Lasers and Electro-Optics Society 2000 Annual Meeting. LEOS 2000. 13th Annual Meeting. *IEEE*, Vol. 2, pp. 405-406 2000.

[11] O. A. Sab, R. Pyndiah, Performance of Reed-Solomon Block Turbo Code, Proceeding of IEEE GLOBECOM Conference, Vol. 1/3, pp. 121-125, Nov. 1996.

[12] P. Elias, Error-Free Coding, IRE Transaction on Information Theory, Vol. IT-4, pp. 29-37, Sept. 1954.

[13] S.B.Wicker, Error Control Systems for Digital Communication and Storage, Upper Saddle River, NJ, Prentice Hall Inc., 1995.

[14] S.B.Wicker, Error Control Systems for Digital Communication and Storage, Upper Saddle River, NJ, Prentice Hall Inc., 1995.

[15] S. Lin, D. J. Costello, Jr., Error Control Coding, Englewood Cliffs, NJ, Prentice Hall Inc., 1983.

[16] Y. Sugiyama, M. Kasahara, S.Hirasawa, T.Namekawa, A Method for Solving Key Equations for Decoding Goppa Codes, Information and Control, Vol. 27, pp. 87-89, January 1975.