# PERFORMANCE OF REED-SOLOMON BLOCK TURBO CODE

Omar AITSAB and Ramesh PYNDIAH

Télécom Bretagne, Technopôle Brest Iroise, BP-832, 29285 BREST, FRANCE. (Tel : (33) 98 00 10 70, Fax : (33) 98 00 10 98) (Email : Omar.Aitsab@enst-bretagne.fr, Ramesh.Pyndiah@enst-bretagne.fr)

**ABSTRACT** - Thanks to recent progress in the iterative decoding of concatenated codes, several new fields of investigation have appeared. In this paper, we present a first approach of the iterative decoding of Reed-Solomon (RS) product codes: "Turbo codes RS". Two methods to construct RS product codes are given. The iterative decoding of the RS product codes is based on the soft decoding and the soft decision of the component codes. The performance of RS turbo codes have been evaluated on the Gaussian and Rayleigh channels using Monte Carlo simulation. Coding gains up to 5.5dB for a BER (Bit Error Rate) of  $10^{-5}$  have been obtained on the Gaussian channel. This new coding scheme is very attractive for data storage applications where the RS product codes are often used.

### I. INTRODUCTION

Error correcting codes (ECC) are tody widely used for digital transmissions, notably in the field of TV broadcasting and mobile radiocommunications. This interest for the ECC results mainly from the use of powerful source coding techniques which are more and more sensitive to transmission errors. Furthermore, the progress in VLSI technology opens the way for the implementation of more complex algorithms. So we can use more and more powerful codes.

The product codes are powerful codes which present large minimum Hamming distance for a given code rate. In spite of those good characteristics, one must be careful while decoding in order to take advantage of all the information contained in the codes.

In 1993, Berrou [1] obtained exceptional performances with the convolutional "turbo codes". Those "turbo codes" are made from two concatened recursive convolutional codes, associated with an iterative decoding algorithm based on soft decoding of the component codes and soft decision of decoded bits [2]. The succes of convolutional turbo codes encouraged R. Pyndiah to investagte an equivalent block turbo code. R. Pyndiah [3] showed that it was possible to obtain the same performance with block turbo codes. Those bloc turbo codes are built according to the concept of product codes proposed by Elias in 1954 [4] using BCH codes. The block turbo codes are decoded by an iterative algorithm using soft decoding of the component codes and soft decision of decoded bits [3]. Furtheremore, this work has shown that the BCH block turbo codes present the best performance for the applications with high code rates  $(R \ge 0.7)$ .

Reed-Solomon codes (RS) are block codes with non binary elements (q-ary symbols). They present the best code rate for a given minimum Hamming distance. Thus, it was interesting to evaluate the performance of block turbo codes using RS code.

In section II, we present two methods for the constuction of RS product codes and in section III, we describe the soft decoding algorithm and give some results of RS soft decoding. In section IV, we give the soft decision algorithm derived from the theoretical Log-Likelihood-Ratio and in section V we describe the iterative algorithm for decoding RS product codes. Section VI and VII are dedicated to simulation results of BER function of signal to noise ratio for different RS turbo codes over Gaussian and Rayleigh channels while in section VIII we discuss the results of RS turbo codes.

# II. CONSTRUCTION OF PRODUCT CODE II.1. Reed-Solomon codes

RS codes are BCH codes with non binary elements belonging to  $GF(q=2^m)$ . Each *q*-ary symbol of the Galois field can be mapped to *m* binary elements. The main parametres of a RS code are  $(n, k, \delta)$ , where *n* is the code word lenght, *k* is the number of information symbols and  $\delta$  its minimal Hamming distance.

## **II.2.** Product codes

Let us consider two linear block codes  $\mathcal{C}^1$  having parametres  $(n_1, k_1, \delta_1)$  and  $\mathcal{C}^2$  having parametres  $(n_2, k_2, \delta_2)$ . The product code  $\mathcal{P} = \mathcal{C}^1 \otimes \mathcal{C}^2$  is obtained by :

- 1) placing  $(k_1 \times k_2)$  information symbols in an array of  $k_1$  rows and  $k_2$  columns,
- 2) coding the  $k_1$  rows using code  $\mathcal{C}^2$ ,
- 3) coding the  $n_2$  columns using code  $\mathcal{C}^2$ ,

as illustrated in figure 1.



**Figure 1** : Construction of product code  $\boldsymbol{\mathcal{P}} = \boldsymbol{\mathcal{C}}^1 \otimes \boldsymbol{\mathcal{C}}^2$ 

It is shown [4] that the  $(n_1 - k_1)$  last rows of the matrix are code words of  $\mathfrak{C}^2$  exactly as the  $(n_2 - k_2)$  last columns are code words of  $\mathfrak{C}^1$  by construction. The parameters of the resulting product code  $\mathfrak{P}$  are given by  $n = n_1 \times n_2$ ,  $k = k_1 \times k_2$ ,  $\delta = \delta_1 \times \delta_2$  and the code rate R is given by  $R = R_1 \times R_2$  where  $R_i$  is the code rate of code  $\mathfrak{C}^i$ . This classical method has been used for the construction of RS product codes where the matrix M contains  $k_1 \times k_2$  *q-ary* information symbols. The two codes  $\mathfrak{C}^1$  and  $\mathfrak{C}^2$  must have the same code word length  $n_1 = n_2$  so that they have the same *qary* symbols.

A second method has been studied to build a RS product codes based on the binary elements of the *q*-ary symbols. We have used the same code  $\mathcal{C}^{1} = \mathcal{C}^{2}$ . The matrix **M** contains  $k_1 \times m$  rows and  $k_2 \times m$  columns of binary elements. Each one of  $k_1 \times m$  row is converted into *q*-ary symbols and coded by code  $\mathcal{C}^{2}$ . The code words obtained are converted back into binary elements and each  $n_2 \times m$  column is converted into *q*ary symbols and coded by code  $\mathcal{C}^{1}$ .

With this method each bit is coded twice but not with the same neighbouring bits and simulations showed that the  $(n_1 - k_1) \times m$  last rows are code words of  $\mathcal{C}^2$  and the  $(n_2 - k_2) \times m$  last columns are code words of  $\mathcal{C}^1$  too.

We shall consider in this paper the results of RS product codes built with the 1<sup>st</sup> method, which gave the best results.

## **III SOFT DECODING OF RS CODE**

If we consider the transmission of block coded binary symbols  $\{-1,+1\}$  using QPSK signaling over a Gaussian channel, the sequence **R** at the input of the RS decoder can be modelised by:

where 
$$\mathbf{R} = \mathbf{E} + \mathbf{N}$$
 (1)  
 $\begin{pmatrix} \eta_1 & \cdots & \eta_j & \cdots & \eta_n \\ \vdots & \vdots & \eta_j & \vdots & \vdots \\ r_{m1} & \cdots & r_{mj} & \cdots & r_{mn} \end{pmatrix}$  is the received data

$$\mathbf{E} = \begin{pmatrix} e_{11} & \cdots & e_{1j} & \cdots & e_{1n} \\ \vdots & \vdots & e_{ij} & \vdots & \vdots \\ e_{m1} & \cdots & e_{mj} & \cdots & e_{mn} \end{pmatrix}$$
 is the transmitted code word  
$$(e_{ij} = \pm 1), \quad \mathbf{N} = \begin{pmatrix} n_{11} & \cdots & n_{1j} & \cdots & n_{1n} \\ \vdots & \vdots & n_{ij} & \vdots & \vdots \end{pmatrix}$$
 are Additive

 $\binom{n_{m1} \cdots n_{mj} \cdots n_{mn}}{m}$ White Gaussian Noise (AWGN) samples of stantard deviation  $\sigma$ . Decoding the received sequence **R** according to the maximum likelihood criteria is given by :

$$\mathbf{D} = \mathbf{C}^{i} \text{ if } \Pr\left(\mathbf{E} = \mathbf{C}^{i} / \mathbf{R}\right) > \Pr\left(\mathbf{E} = \mathbf{C}^{l} / \mathbf{R}\right) \quad \forall l \neq i (2)$$
where  $\mathbf{C}^{i} = \begin{pmatrix} c_{11}^{i} & \cdots & c_{1j}^{i} & \cdots & c_{1n}^{i} \\ \vdots & \vdots & c_{ij}^{i} & \vdots & \vdots \\ c_{m1}^{i} & \cdots & c_{mj}^{i} & \cdots & c_{mn}^{i} \end{pmatrix}$  is the *i*<sup>th</sup> code word

of code  $\boldsymbol{\mathcal{C}}$  with parameters  $(n, k, \delta)$ 

and 
$$\mathbf{D} = \begin{pmatrix} d_{11} & \cdots & d_{1j} & \cdots & d_{1n} \\ \vdots & \vdots & d_{ij} & \vdots & \vdots \\ d_{m1} & \cdots & d_{mj} & \cdots & d_{mn} \end{pmatrix}$$
 the decision correponding

to maximum likelihood transmitted sequence conditionally to

**R**. For received samples corrupted by AWGN, decoding rule (2) is simplified into :

$$\mathbf{D} = \mathbf{C}^{i} \text{ if } \left| \mathbf{R} - \mathbf{C}^{i} \right|^{2} < \left| \mathbf{R} - \mathbf{C}^{l} \right|^{2} \quad \forall l \neq i$$
(3)

where : 
$$\left| \mathbf{R} - \mathbf{C}^{i} \right|^{2} = \sum_{j=1}^{n} \sum_{f=1}^{m} \left( r_{jf} - c_{jf}^{i} \right)^{2}$$
 (4)

For RS codes, the number of code words  $q^k$  is relatively large and optimum sequence decoding is too complex for implementation. In 1972, Chase proposed an algorithm [5] which approximates optimum sequence decoding of block codes with low computation complexity and a small performance degradation. Instead of reviewing all the code words, the Chase algorithm selects the most probable code words using channel information **R** to apply decision rule (3). For this, we take a hard decision on **R** which gives a binary word  $\mathbf{Y}^0$  ( $y_{jf} = \pm 1$ ). Modifying the sign of the elements of  $\mathbf{Y}^0$  corresponding to the

pleast reliable components of **R**, we obtain  $2^p - 1$  new words  $\mathbf{Y}^l$ ,  $l = 1....(2^p - 1)$ . The algebric decoding of  $2^p$  words  $\mathbf{Y}^l$ 

gives the subset of code words  $C^{l}$  which is used for soft decoding (3).

## Simulation results:

We have simulated different RS codes on a Gaussian channel using the Chase algorithm. We noticed that the greater the number of test sequences  $(2^p)$ , the more the subset of decoded code words is likely to contain the transmitted code word, and the better is the coding gain. We have choose seventeen test sequences and the coding gain amelioration obtained is up to 2dB at BER of  $10^{-5}$ . In figure 2, we compare the performance of the RS(255,223,33) with hard decoding and the RS(31,27,5) with soft decoding.



**Figure 2**: Comparison of the RS code (255,223,33) using hard decoding with the RS code (31,27,5) using soft decoding

Both codes have the same coding rate R = 0.87 and give close results at a BER of 10<sup>-5</sup>. We note that the correction power of RS(255,223,33) code is t = 16, whereas that of RS(31,27,5) is t = 2. We know that for an algebric decoder, the number of computations increases exponentially with t. Thus, the 2<sup>nd</sup> code will be easier to implement even if we use

16 test sequences for the soft decoding. This comparaison is quite interesting since it shows that for applications requiring a maximum BER above  $10^{-5}$ , a small RS code with soft decoding gives better results than a more powerful code with hard decoding.

### **IV. SOFT DECISION OF DECODED RS CODES**

To compute the reliability of the decoded bits, we have used the algorithm propesed by Pyndiah [3][6]. This reliability function is given by the Log Likelihood Ratio (*LLR*) of the decision  $d_{if}$  which is defined by:

$$LLR_{jf} = \ln \frac{\Pr\{e_{jf} = +1 / \mathbf{R}\}}{\Pr\{e_{jf} = -1 / \mathbf{R}\}}$$
(5)

where  $e_{jf}$  is the binary element in position (j, f) of the transmitted code word  $\mathbf{E}$  (1 < j < n and 1 < f < m). It is shown [6] that the expression of the  $LLR_{jf}$  can be approximated, in the case of the AWGN, by :

$$LLR_{jf} \approx \frac{1}{2\sigma^2} \left\{ \left| \mathbf{R} - \mathbf{C}^{\min(-1)} \right|^2 - \left| \mathbf{R} - \mathbf{C}^{\min(+1)} \right|^2 \right\}$$
(6)

where  $\mathbf{C}^{\min(+1)}$  and  $\mathbf{C}^{\min(-1)}$  are two code words at minimal Euclidean distance from **R** with  $c_{jf}^{\min(+1)} = +1$  and  $c_{jf}^{\min(-1)} = -1$  ( $\mathbf{C}^{\min(+1)}$  and  $\mathbf{C}^{\min(-1)}$  are choosen among the subset of code word given by Chase algorithm). By expanding relation (6) using (4) we obtain :

$$LLR_{jf} \approx \frac{2}{\sigma^2} \left( r_{jf} + \sum_{\substack{x=1\\x\neq j}}^{n} \sum_{\substack{z=1\\z\neq f}}^{m} r_{xz} c_{xz}^{\min(+1)} \rho_{xz} \right)$$
(7)

where:

$$((x,z) \neq (j,f)), \ \rho_{xz} = \begin{cases} 0 \ if \ c_{xz}^{\min(+1)} = c_{xz}^{\min(-1)} \\ 1 \ if \ c_{xz}^{\min(+1)} \neq c_{xz}^{\min(-1)} \end{cases}$$

If we normalise the approximated *LLR* of  $d_{jf}$  with respect to  $2/\sigma^2$  we obtain :

$$r'_{jf} = \frac{\sigma^2}{2} LLR_{jf} = r_{jf} + w_{jf}$$
(8)

where :

$$w_{jf} = \sum_{\substack{x=1 \ x \neq j}}^{n} \sum_{\substack{z=1 \ x \neq f}}^{m} r_{xz} c_{xz}^{\min(+1)} \rho_{xz}$$
(9)

The estimated normalized *LLR* of decision  $d_{jf}$ ,  $r'_{jf}$  (relation 8), is given by input samples  $r_{jf}$  plus  $w_{jf}$  which is independent of  $r_{jf}$ . The *LLR* of  $r'_{jf}$  is an estimation of the soft decision of the RS decoder.

To compute the normalized  $LLR_{jf}$  of binary elements at the output RS decoder, we must first select the code word at minimal Euclidean distance from **R**. Let  $\mathbf{C}^{\min(+i)}$  be this code word,  $\mathbf{C}^{\min(+i)}$  has a binary element *i* at position (j, f)  $(i = \pm 1)$ . Then we look for code word  $\mathbf{C}^{\min(-i)}$  at minimal Euclidean distance from **R** among the code word subset obtained by Chase algorithme.  $\mathbf{C}^{\min(-i)}$  must have -i as binary element at position (j, f).

If the  $C^{\min(-i)}$  code word is founded, the soft decision  $r'_{if}$  of  $d_{if}$  can be computed using the relation given below :

$$r'_{jf} = \left(\frac{\left(M^{\min(-i)} - M^{\min(i)}\right)}{4}\right) c_{jf}^{\min(i)}$$
(10)

where  $M^{\min(-i)}$  and  $M^{\min(i)}$  represent respectively the  $C^{\min(-i)}$  Euclidean distance from **R** and  $C^{\min(+i)}$  Euclidean distance from **R**; else we use the relation:

$$r'_{jf} = \beta. c_{jf}^{\min(i)} \tag{11}$$

where  $\beta$  is a constant which is a function of the BER and is optimized by simulation.

# V ITERATED DECODING OF RS PRODUCT CODES

For an RS product code  $\mathfrak{P} = \mathfrak{C} \otimes \mathfrak{C}$  ( $\mathfrak{C} = (n,k,\delta)$ ), the received sample matrice  $[\mathfrak{R}]$  has  $n \times m$  rows and n columns. This matrix is diveded into n sub-matrices  $\mathbb{R}$ . Each one of these n matrices  $\mathbb{R}$  represents an elementary RS code word. The decoder decodes the lines (or columns) of the matrix, estimates the normalized *LLR*  $[\mathfrak{R}']$  as describe in section IV and gives as output  $[\mathbb{W}(1)]$  ( $[\mathbb{W}(1)]=[\mathfrak{R}']-[\mathfrak{R}]$ ). Then the decoder performs the same operations on the columns (or rows) using as input :

$$[\Re(1)] = [\Re] + \alpha(1) [\mathbf{W}(1)]$$
(12)

This decoding procedure is then iterated by cascading the elementary decoders as illustrated in Figure 3. The coefficient  $\alpha(k)$  is used to reduce the influence of  $[\mathbf{W}(k)]$  in the first iterations where the BER is relatively high and thus  $[\mathbf{W}(k)]$  is not absolutely reliable.



Figure 3 : Block diagram of elementary RS decoder

# VI. PERFORMANCE OF RS TURBO CODES OVER GAUSSIAN CHANNEL

The upper bound of the asymptotic coding gain  $G_a$  in the case of maximum likelihood sequence decoding depends on the minimal code distance  $[(G_a \leq 10 \log(R.\delta)]]$ . We have used expurged RS codes as elementary code  $\mathcal{C}$  because they increase the minimal distance of the product code for a slight discrease in the coding rate. It is clear that the performance of the turbo decoder depends on the number of test sequences. In our investigation, we used seventeen test sequences based on the

five least reliable bits. This is a compromise between performance and complexity. In order to have the parameters  $\alpha(k)$  and  $\beta(k)$  independent of the product code, we normalized the average value of the computed  $w_{jf}$  to +1 for samples  $w_{jf}>0$  and -1 for samples  $w_{jf}<0$ . The performances of RS turbo codes have been evaluated over a Gaussian channel using Monte Carlo simulation. The RS product codes has the same elementary code  $\Re^{1}=\Re^{2}=\Re$ . In figure 4, we present the BER of RS turbo code RS(63,56,8)xRS(63,56,8).



Figure 4 : Bit Error Rate versus signal to noise ratio after each of the first four iterations using the iterative algorithm with  $\mathfrak{e}^{1}=\mathfrak{e}^{2}=RS(63,56,8)$ 

We can observe that the coding gain improves after each iteration. For a BER of  $10^{-5}$ , we gain respectively 1.6dB, 0.3dB and 0.1dB at each additional iteration. The total coding gain is about 5dB compared with an uncoded transmission, for a BER of  $10^{-5}$ . However, increasing the number of iterations above four doesn't bring any significant coding gain. This results from the fact that the slope of the BER curve is too steep after iteration 4. In figure 5, we have plotted the BER versus signal to noise ratio of different RS turbo codes at the fourth iteration.



Figure 5 : Bit Error Rate versus signal to noise ratio after at fourth iteration using the iterative algorithm of different RS turbo codes.

We observe that the slope of BER curves increases with the code word length and the minimum Hamming distance elementary code. However, for a fixed code word length, increasing the minimum Hamming distance above 8 doesn't bring any ameliorations because the number of test sequences is fixed to seventeen for all simulated RS turbo codes. This is shown by the BER curve of RS(31,26,6)xRS(31,26,6) and RS(31,24,8)xRS(31,24,8) (see figure 5). To solve this problem we must increase the number of test sequences when we increase the minimum Hamming distance of product code.

To evaluate the performance of RS turbo codes, we have compared the coding gain at the fourth iteration with an upper bound of the asymptotic coding gain  $G_a$  in the case of maximum likelihood sequence decoding given by :

$$G_a \le 10\log(R.\delta) = (G_a)_{\max} \tag{13}$$

Since RS product codes are powerful codes, it is also interesting to compare the signal to noise ratio required for a given BER ( $10^{-5}$  for example) with Shannon's theoretical minimum signal to noise ratio to achieve an error free transmission over a Gaussian channel. The results are summarized in the table below (see figure 6).

$\begin{array}{c} \text{RS} \\ \text{product code} \\ (\mathbf{\mathscr{C}}^1 = \mathbf{\mathscr{C}}^2) \end{array}$	R	δ	(G <sub>a</sub> ) max (dB)	Gain at 10 <sup>-5</sup> (dB)	Sh's limit (dB)	SNR at 10 <sup>-5</sup> (dB)
(15,12,4)	0.64	16	10.1	5.48	0.47	4.10
(31,28,4)	0.81	16	11.1	5.05	1.07	4.42
(31,26,6)	0.70	36	14.0	5.48	0.68	4.10
(31,24,8)	0.60	64	15.8	5.53	0.33	4.06
(63,60,4)	0.91	16	11.6	4.53	1.43	5.06
(63,58,6)	0.84	36	14.8	4.86	1.20	4.63
(63,56,8)	0.79	64	17.0	5.05	1.00	4.40

**Figure 6** : Table comparing the performance of the RS turbo codes at the fourth iteration and theoretical performance.

The coding gain at the fourth iteration for a BER of  $10^{-5}$ , is winthin  $\pm 0.5$  dB for the seven RS turbo codes. The maximum coding gain obtained is about 5.5 dB. The difference between the obtained gain for a BER of  $10^{-5}$  and  $(G_a)_{\text{max}}$  is higher the higher is the minimum Hamming distance of product code. This result is due to the fixed number of test sequence used for all RS turbo codes. If we compare the performance at iteration 4 of the RS turbo codes with the theoretical limit of Shannon, we find a difference  $\Delta S$  of the order of 3.5 dB for a BER of  $10^{-5}$ .

#### VII. PERFORMANCE OF RS TURBO CODES OVER RAYLEIGH CHANNEL

In the case of Rayleigh channel, the received samples are given by :

$$\mathbf{R} = \lambda \mathbf{E} + \mathbf{N} \tag{14}$$

where N is an AWGN and  $\lambda$  is an Rayleigh attenuation. As for as Gaussian channel, the components of [W(k)] are normalized and we have used the same values for  $\alpha(k)$  and  $\beta(k)$ . In figure 7 and 8 we compare the RS turbo code performance over a Gaussian and a Rayleigh channels.



Figure 7 : performance of RS(31,26,6)xRS(31,26,6)turbo code over a Gaussian and a Rayleigh channels.

Product code ( $\mathcal{C}^1 = \mathcal{C}^2$ )	SNR at 10 <sup>-5</sup> (dB) Gauss	SNR at 10 <sup>-5</sup> (dB) Rayleigh	Δ SNR (dB)
RS(15,12,4)	4.10	10.0	5.90
RS(31,28,4)	4.42	12,1	7.68
RS(31,26,6)	4.10	10,4	6.30
RS(63,60,4)	5,06	15,1	10.04

Figure 8 : Table comparing the performance of RS turbo codes at the fourth iteration over a Gaussian and a Rayleigh channels.

We observe that the BER curves have the same slope at iteration 4 for a Gaussian and Rayleigh channels. The BER curves are simply shifted by about 6dB to the right in the case of a Rayleigh channel. We note that the smaller the code rate the smaller is the degradation.

These results have been obtained without any knowledge of the attenuation  $\lambda$ . If we assume that  $\lambda$  is known the BER will be divided by about 2 only.

## VIII. CONCLUSION

In this paper we have presented the first results of RS product codes decoded using the new iterative decoding algorithm [3][6]. The simulations have shown very good results, better than those obtained so far with other algorithms [7][8]. We have obtained BER curves with identical slopes for the Gaussian and the Rayleigh channels. For a BER of  $10^{-5}$ , the coding gain obtained is up to 5.5 dB and the difference with theoritical Shannon's limit is about 3.5 dB after four iteration on a Gaussian channel. These good results are obtained thanks to an iterative decoding based on :

1] soft decoding of each rows (or columns),

2] soft decision at the output of the soft decoder,

On the other hand, the iterative decoding algorithm seems less efficient in the case of RS codes than in the case of BCH codes[6]. This is probably due to the fact that the RS code is a q-ary code.

Nevertheless, the RS codes are more often used than the BCH codes, above all in the field of satellite transmission and data storage [9] which gives these codes very good perspectives in the future.

### ACKNOWLEDGEMENT

This work has been financially supported by the Centre National d'Etudes des Télécomunications (CNET) and the Centre Commun d'Etudes de Télédifussion et Télécommunications (CCETT). The authors would like to thank P. combelles for his helpful comments and discussions.

## REFERENCES

- C. Berrou, A. Glavieux and P. Thitimajshima, "Near Shannon limit error-correcting coding and décoding : Turbo-codes (1)", *IEEE Int. Conf. on Comm. ICC'93*, vol 2/3, May 93, pp. 1064-1071.
- [2] C.Berrou, P. Adde, E. Angui and S. Faudeil, "A low complexity Soft-output Viterbi decoder", *IEEE Int. Conf.* on Comm. ICC'93, vol 2/3, May 93, pp.737-740.
- [3] R. Pyndiah, A. Glavieux, A. Picart and S. Jacq, "Near optimum decoding of product codes", *IEEE Globecom'94*, vol 1/3, pp 339-343.
- [4] F.J. Macwilliams and N.J.A. Slone, "The theory of error correcting codes", North-Holland publishing company, 1978, pp. 567-580.
- [5] D. Chase, "A class of algorithms for decoding block codes with measurement information", *IEEE Trans. Inform. Theory*, vol IT-18, Jan. 72, pp. 170-182.
- [6] S. Jacq, R. Pyndiah, A. Picart "Algorithme Turbo : un nouveau procédé de décodage pour les codes produits" *GRETSI'95, Juan-les-Pins*, France, 18-21 Sept., pp 529-532.
- [7] S.M. Reddy, "On decoding iterated codes," *IEEE Trans. Inform. Theory*, vol IT-16, Sept. 1970, pp. 624-627.
- [8] S.M. Reddy and J.P. Robinson, "Rondom error and burst correction by iterated codes," *IEEE Trans. Inform. Theory*, vol IT-18, Jan. 1972, pp. 182-185.
- [9] Seung Ho Kim, Sang Wu Kim and Chee Sun Won, "Decoding strategies for Reed-Solomon product codes: application to digital recording systems," *IEEE Trans. on Consumer Electronics*, vol. 38, No. 3, August 92.