Computer and Information Science; Vol. 9, No. 1; 2016
ISSN 1913-8989 E-ISSN 1913-8997
Published by Canadian Center of Science and Education

Iterative Soft Permutation Decoding of Product Codes

Mohamed Askali[1], Fouad Ayoub[2], Idriss Chana[1] & Mostafa Belkasmi[1]

[1] MohammedV-Souisi University, SI2M Labo, ENSIAS, Rabat, Morroco

[2] CRMEF, Kenitra, Morroco

Correspondence: Mohamed Askali, MohammedV-Souisi University SI2M Labo, ENSIAS, Rabat, Morroco. E-mail: askali11@gmail.com/ayoubfouadn@gmail.com/idrisschana@gmail.com/belkasmi@ensias.ma

Received: August 18, 2015 Accepted: October 2, 2015 Online Published: January 31, 2016

doi:10.5539/cis.v9n1p128 URL: http://dx.doi.org/10.5539/cis.v9n1p128

Abstract

In this paper the performance of product codes based on quadratic residue codes is investigated. Our Proposed Iterative decoding SISO based on a soft permutation decoding algorithm (SPDA) as a component decoder. Numerical result for the proposed algorithm over Additive White Gaussian Noise (AWGN) channel is provided. Results show that the turbo effect of the proposed decoder algorithm is established for this family of quadratic residue codes.

Keywords: soft decoding, error correcting codes, turbo code, product codes, Iterative decoding, quadratic residue codes

1. Introduction

The turbo codes were invented in 1993 by Berrou, Glavieux and Thitimajshima, who had used the concatenation of convolutional codes, the elementary decoder BCJR (Bahl, Cocke, Jelinek and Raviv Algorithm) and SOVA (Soft output Viterbi Algorithm) as a soft output. Two years later, Pyndiah et al [4] presented an alternative for bloc codes using the product code based on BCH (Bose, Ray Chaudhurand Hocquenghem) codes, the Chase II algorithm, as a component decoder, and the Pyandiah's soft output generated according to the formula presented in. Since turbo codes have been subject of many publications for instance. The turbo codes have attracted the interest of the scientific community, especially for their high speed transmissions over the air. The word turbo is much more related to decoding than encoding. To make a turbo decoding we will need three basic elements where the first is the concatenation of several simple codes, the second is to develop or adopt an efficient elementary decoder conducted by a soft output and the third element is a decoding scheme in which the soft output is converted to extrinsic information exchanged between the components decoders in an iterative process. The implementation and the performance of an efficient turbo decoder depend on the complexity of these three elements. In addition to the computation of the extrinsic information which has to be done in a very short time, and the convergence based on the number of iterations, we will need an elementary decoder less complex. In this perspective comes our work in which we use our soft permutation decoding algorithm (SPDA) proposed in [5] as an elementary decoder given the importance of performance results obtained by our decoder SIHO, and we compute its soft output using extrinsic information according to Soleymani et al. The result obtained in terms of BER is very interesting. The rest of the paper is organized as follow: *section II* presents the quadratic residue (QR) codes, section III *describes* the construction of product codes, section IV describes the original version of Soft permutation decoding algorithm (SPDA) for QR codes, section V talks about the Soft output and the Schema of the iterative decoding of Soleymani, finally, Simulation results and analyses are given in Section VI.

2. Quadratic Residue Codes and Their Stabilizers

The n is a prime and $n \equiv \pm 1 (\mathrm{mod}\ 8)$, the quadratic residue code $\mathrm{QR}(n) = \mathrm{QR}\left(\mathrm{n}, (n+1)/2, d\right)$ is a cyclic code

with a generator polynomial $g(x) = \prod_{i \in Q} (x - \beta^i)$, where $Q = \{j^2 \bmod n : 1 \le j \le n-1\}$ is the set of all nonzero

quadratic residue integers modulo n and β is a primitive **n**th root of unity in GF(2^m), where m is the smallest positive integer such that n divides 2^m - 1. A $QR(n)$ code, where d is odd, can be extended to a $EQR(n)=EQR(n+1,(n+1)/2,d+1)$ code whose codewords are obtained by adjoining a parity-check bit to a fixed position ∞ of every codeword of the $QR(n)$ code. For all values of n, the binary $EQR(n)$ is invariant under the projective special linear group $PSL_2(n)$, which we define as follows:

For a prime $n \equiv \pm 1 (\mathrm{mod}\ 8)$, the set of permutations over {0,1,2,…,n-1,∞}, of the form $y \to (ay+b)/cy+d$ where a, b, c and d are elements of GF(n) verifying : ad-bc=1 form a group called the projective special linear group G=$PSL_2(n)$, of order $|G| = n.(n^2-1)/2$. $PSL_2(n)$ can be generated by the three following permutations [3]: $S: y \to y+1;\ V: y \to \rho^2 y;\ T: y \to -1/y.$ where ρ is a primitive element of GF(n). By a theorem of Gleason and Prange, the automorphism group of an extended quadratic residue code has a subgroup which is isomorphic to either $PSL_2(n)$.

3. Product Codes

The product codes are constructed by concatenation of two or more linear block codes. We consider two basic block codes C_1 and C_2 characterized by parameters (n_1, k_1, d_1,R_1) and (n_2, k_2, d_2,R_2) where n_i represent the code length, k_i the length of the message d_i the minimum distance Hamming and R_i the code rate . The product code $C_p = C_1 \times C_2$ is represented in the form of matrix with n_1 rows and n_2 columns. Where the information forms a sub-matrix M of k_1 lines and k_2 columns, each of the lines k_1 from M is coded by the code C_1 and each of the n_1 columns is coded by the code C_2. The resultant parameters of the product code C_p are (n_p, k_p, d_p, R_p) where $n_p = n_1 \times n_2$, $k_p = k_1 \times k_2$ and $d_p = d_1 \times d_2$, the code rate is $R_p = R_1 \times R_2$. An important Property of these codes is that if the n1 columns are by construction the C2‘s code words, and k2 lines are C1‘s code words, the n1-k1 remaining lines of the resultant code word are C1‘s code words. In other hand the major advantage of product codes is a gain in minimum distance, and their major disadvantage is the loss in codes rate, subsequently, we consider that codes C1 and C2 are identical.

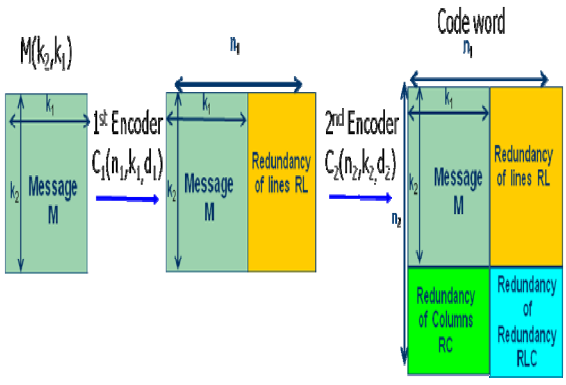
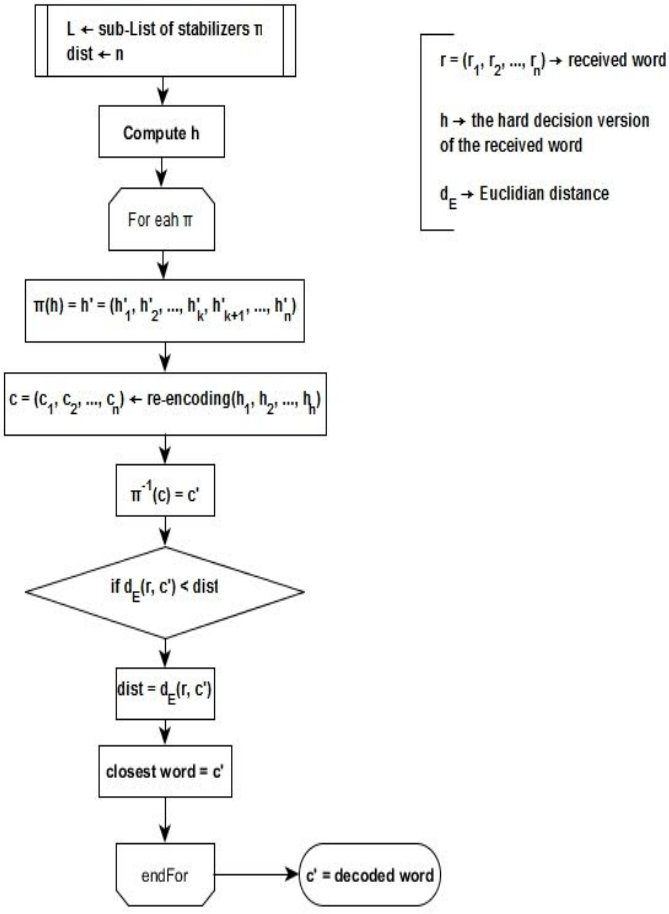


Figure 1. Construction of product code

4. Soft Permutation Decoding Algorithm (SPDA)

The SPDA algorithm proposed in [5], tries to find the closest codeword to the received word in terms of Euclidian distance. The algorithm works as shown below:



5. Soft Output and the Schema of the Iterative Decoding

5.1 Description of Confidence Value

The concept of confidence value designated by Φ is detailed in [4]. In this paragraph, we will simply give a brief description.

Let $X = \{x_0, x_1, \ldots, x_{n-1}\}$ be the transmitted code word, Φ = P{D = X/R} is the probability that the decoder takes a correct decision $D = \{d_0, d_1, \ldots, d_{n-1}\}$ given the received sequence $R = \{r_0, r_1, \ldots, r_{n-1}\}$; That is the assessment of the decision of the decoder. Computing Φ is impossible for a practical implementation, thus estimation has to be performed. To estimate Φ Soleymani et al are adopted a distance destructive denoted by $Dist_{dest}$ as a metric between R and D where only the positions increasing the Euclidian distance contribute, i.e. where the noise vector has a different polarity than the decision vector D, following.

$$Dist_{dest} = \sum_{j \in DES} \left(r_j - d_j\right)^2$$
$$\text{where } DES = \left\{ j \mid \left(r_j - d_j\right).d_j < 0 \right\} \quad (1)$$

There is a relationship between the confidence value Φ and the destructive Euclidean distance $Dist_{dest}$. Using software simulation according to [2] the influence of the variable **Eb/No** and the number of iterations may be omitted, and treat the confidence value **Φ** as a function of destructive Euclidean distance, written as:

$$\Phi = f\left(Dist_{dest}\right) \quad (2)$$

Table 1 below resumes the function between $Dist_{dest}$ and **Φ** for some used residue codes (RQ) codes.

Table 1. Confidence Value "Φ" Versus Distance destructive

QR(23,12,7)	DEST	<9	9	10	11	12	13	14	15	>15
	Φ	0.99	0,97	0,95	0,91	0,72	0,5	0,36	0,33	0
QR(41,21,9)	DEST	<15	16	17	18	19	20	21	22	>22
	Φ	0.99	0.98	0.95	0.89	0.76	0.55	0.29	0.14	0.00
QR(47,24,11)	DEST	<27	27	28	29	30	31	32	33	>33
	Φ	0.99	0,95	0,93	0,84	0,71	0,53	0,21	0,12	0

5.2 Computing Soleymani's Soft Output

In this subsection we give the computation of the soft output as it's described in [2] by Soleymani et al. Recall that $X = \{x_0, x_1, ..., x_{n-1}\}$ is the transmitted Codeword, the symbol x_j $j \in \{0, ..., n-1\}$ has certain confidence value **Φ**. The probability of x_j can be expressed as:

$$P(x_j = \pm 1 \mid R) = P(x_j = \pm 1, D = X \mid R) + P(x_j = \pm 1, D \neq X \mid R) \quad (3)$$

The first term represents the probability value when the decoder gives a correct codeword. applying Bayes' rule to this term will yield

$$P(x_j = \pm 1, D = X \mid R) = P(x_j = \pm 1, D = X, R).P(D = X, R) = P(x_j = \pm 1, D = X, R).\Phi \tag{4}$$

Since the decision bit d_j is known, then

$$P(x_j = \pm 1, D = X \mid R) = \begin{cases} \phi \ if \ \mathrm{d}_j = x_j \\ 0 \ \text{if} \ \mathrm{d}_j \neq x_j \end{cases} \tag{5}$$

The second term in (3) represents the probability value when the decoder decides in favor of a wrong codeword. In this case, we consider the transmitted symbol x_j is corrupted with Gaussian noise. Thus

$$P(x_j = 1, D \neq X) = \frac{\exp\left(\pm 2 r_j / \sigma^2\right)}{1 + \exp\left(\pm 2 r_j / \sigma^2\right)} \tag{6}$$

Again, we apply Bayes 'rule to the second term in (3) and get :

$$P(x_j = \pm 1, D \neq X \mid R) = P(x_j = \pm 1, D \neq X, R).P(D \neq X \mid R) = \frac{\exp\left(\pm 2 r_j / \sigma^2\right)}{1 + \exp\left(\pm 2 r_j / \sigma^2\right)}.(1 - \Phi) \tag{7}$$

Combining (3)-(7), the a posteriori probability of x_j is found as:

$$P(x_j = +1 \mid R) = \begin{cases} \phi + \dfrac{\exp\left(+2 r_j / \sigma^2\right)}{1 + \exp\left(+2 r_j / \sigma^2\right)}.(1 - \Phi) \ \text{if} \ \mathrm{d}_j = +1 \\ \dfrac{\exp\left(+2 r_j / \sigma^2\right)}{1 + \exp\left(+2 r_j / \sigma^2\right)}.(1 - \Phi) \ \text{if} \ \mathrm{d}_j = -1 \end{cases} \tag{8}$$

And

$$P(x_j = -1 \mid R) = \begin{cases} \dfrac{\exp\left(-2 r_j / \sigma^2\right)}{1 + \exp\left(-2 r_j / \sigma^2\right)}.(1 - \Phi) \ if \ \mathrm{d}_j = +1 \\ \phi + \dfrac{\exp\left(-2 r_j / \sigma^2\right)}{1 + \exp\left(-2 r_j / \sigma^2\right)}.(1 - \Phi) \ \text{if} \ \mathrm{d}_j = -1 \end{cases} \tag{9}$$

Similar to the traditional algorithm described in previous section, we can obtain the extrinsic information W_j by the following equation

$$\omega_j = \frac{\sigma^2}{2} \ln\left(\frac{P(x_j = +1 \mid R)}{P(x_j = -1 \mid R)}\right) - r_j \tag{10}$$

Substituting P(xj = +1|R) and P(xj = −1|R) , we get

$$\omega_j = d_j \left(\frac{\sigma^2}{2} \ln\left(\frac{\Phi + \exp\left(2 r_j d_j / \sigma^2\right)}{1 - \Phi} \right) - r_j d_j \right) \tag{11}$$

Unlike other list-based algorithms, soft outputs generated by (10) can be directly fed into the next decoding stage.

5.3 Iteratif Decoding Scheme

Soleymani's algorithm is intended for decoding turbo product codes; it may be considered an improvement of the chase algorithm / Pyndiah. Indeed, in their algorithm Soleymani et al. adopt the same elementary decoder that Pyndiah is that of Chase-II, by calculating the soft output based on the list of candidates provided by the elementary decoder "ie Chase-II" and the weighting factors α and β. While Soleymani base its calculation of Soft output decision on elementary decoder while rejecting the other candidates, and by evaluating the decision depending on its distance from the received word, to assign the value of trust **Φ** previously described. The latter is used to calculate the extrinsic information.

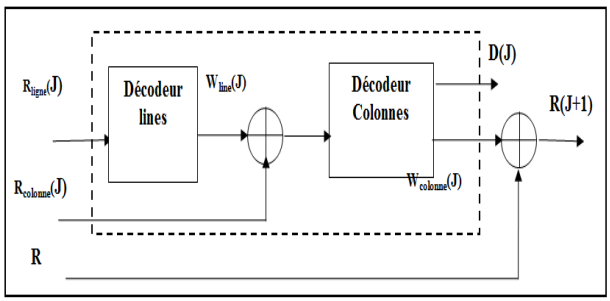


Figure 2. Iteration J[th] du Turbo decodeur

Figure.2 present the **J[th]** itération of Soleymani's Turbo decoder. The $\mathbf{R_{line}(J)}$ are the rows of the matrix **R(J)**, where **R(0)** is received at the matrix output channel. The $\mathbf{R_{line}(J)}$ are decoded by the first component decoder (rows decoder). And extrinsic information $\mathbf{W_{line}}$ **(J)** of the iteration J is calculated using the formula (11). $\mathbf{W_{line}}$ is added to the received word **R** to form the second component of the soft input decoder for decoding $\mathbf{R_{column}(J)}$ which represent the columns of the matrix **R (J)** (column decoder). This allows us to get a hard decision **D (J)** and the extrinsic information $\mathbf{W_{column}(J)}$**.** For the next iteration, $\mathbf{W_{column}}$ is added to the original received matrix R to form a matrix R (J + 1), which in turn must be injected into the first component decoder. This operation, which injects **R(J + 1),** to be repeated, and by default, it will become **R(J).** This iterative process stops when the maximum number of iterations is reached.

6. Simulations & Results

6.1 Permutations Effect on the Elementary Decoder

The Figure 3 shows that the increase of the number of the stabilizers improves the performances.

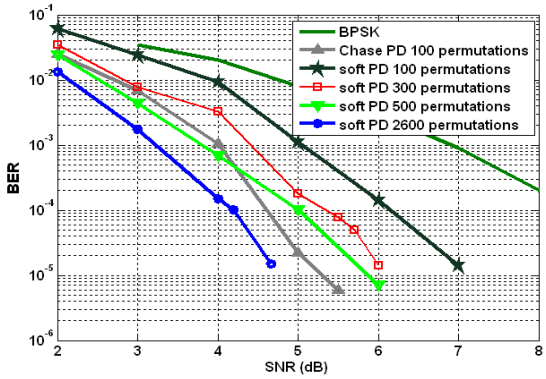


Figure 3. The performances of the soft decision algorithm for EQR(48,24,12) code

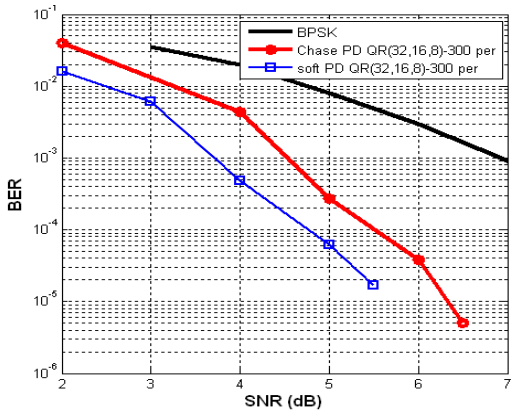


Figure 4. Comparaison between Chase PD and the soft decision algorithm for EQR (32,16,8) code

When the soft decision of permutation decoding algorithm works with 2600 stabilizers, for all values of the SNR, it is better than the Chase-2 decoding algorithm with 32 test sequences. The gain of coding is about 0.5 dB. As we see in the Figures 4, the performances of the soft decision of PD are equal or better than the Chase PD for the EQR(32,16,8).

6.2 Permutations Effect on the SISO Decoder

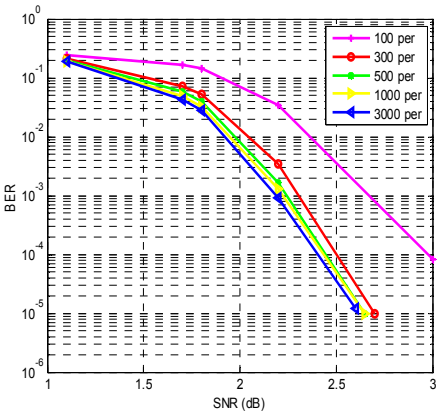


Figure 5. Effect of the number of permutations on the proposed SISO for QR (47,24,11)2 code

In Figure 5, we present the simulation results for the RQ (47, 24, 11)2 code for different numbers of stabilizers of the code. Its show that the gain depends on the number of the permutations and the improvement becomes negligible when the number of permutations is greater than 1000.

6.3 Turbo Effect

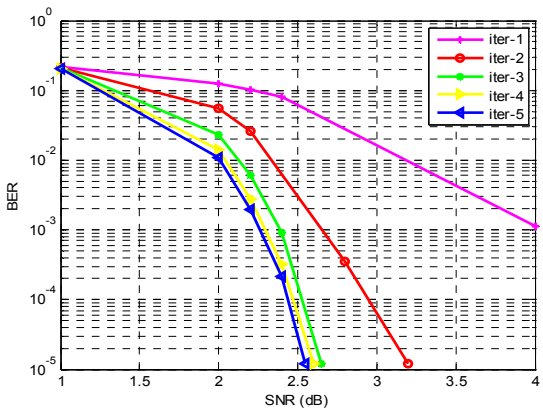


Figure 6. Turbo effect of the proposed SISO for QR$(47,24,11)^2$ code

The turbo effect: Figure 6 shows that the performances increase with number of iterations. According to these Figures we can see that the improvement is great after the first iterations. So, we note that the turbo effect of the proposed SISO is established for this family of codes.

7. Conclusion

In this paper, we have presented a new iterative decoding SISO of product codes based on quadratic residue codes. We have studied the effect of the number of permutations which stabilize QR codes, the number of iterations using simulations. As perspectives of this work, we will challenge other families of codes to be decoded by our iterative SISO decoder.

References

ASKALI, M., NOUH, S., & BELKASMI, M. (2012). A Soft decision version of the Permutation Decoding Algorithm", NTCCCS 12 workshop, 26-28, Oujda, Morocco.

Ayoub, F., Belkasmi, M., & Chana, I. (2010). Iterative Decoding of Generalized Parallel Concatenated OSMLD Codes. *Applied Mathematical Sciences Journal, 4*(41), 2021-2038.

Ayoub, F., Lahmer, M., Belkasmi, M., & Bouyakhf, El. H. (2010). Impact of the decoder connection schemes on iterative decoding of GPCB codes. *International Scholarly and Scientific Research & Innovation, 4*(1), 739-745.

Belkasmi, M., & Farchane, A. (2008). Iterative decoding of parallel concatenated block codes", Proceedings of ICCCE'08, Kuala Lumpur, Malaysia.

Belkasmi, M., Lahmer, M., & Benchrifa, M. (2006). Iterative Threshold Decoding of Parallel Concatenated Block Codes, Turbo Coding Conf, Munich.

C. Berrou, A., Glavieux, P., & Thitimajshima (1993). Near Shannon limit error correcting coding and decoding: Turbo Codes", IEEE Int. Conf. on Communications, ICC, 2, 1064-1070.

Chase, D. (1972). A class of algorithms for decoding block codes with channel measurement information. IEEE. Trans. *Inform. Theory, IT*(18), 170-182.

Elias, P. (1954). Error-free coding, IRE. Trans. *Inf. Theory, IT*(4), 29-37.

Fossorier, M. P. C., & Lin, S. (1995). Soft decision decoding of linear block codes based on ordered statistics, IEEE Trans. *Information Theory, 41*, 1379-1396.

Le, N., Soleymani, A. R., & Shayan, Y. R. (2005). Distance-based-decoding of block turbo codes. *IEEE Communications Letters, 9*(11).

Lucas, R., Bossert, M., & Breitbach, M. (1998). On Iterative Soft-Decision Decoding of Linear Binary Block Codes and Product Codes. *IEEE Journal on selected areas in communications, 16*(2), 276-296.

Pyndiah, R. (1998). Near optimum decoding of product codes: Block Turbo Codes. *IEEE Trans. on Communications, 46*, 1003–1010.

Copyrights