Iterative Threshold Decoding Of High Rates Quasi-Cyclic OSMLD Codes

Karim Rkizat
Mohammed V University in Rabat, ENSIAS
Labo SIME, Team TSE
Rabat, Morocco

Anouar Yatribi
Mohammed V University in Rabat, ENSIAS
Labo SIME, Team TSE
Rabat, Morocco

Mohammed Lahmer
Moulay Ismail University
High School of Technologie
Meknes, Morocco

Mostafa Belkasmi
Mohammed V University in Rabat, ENSIAS
Labo SIME, Team TSE
Rabat, Morocco

***Abstract*—Majority logic decoding (MLD) codes are very powerful thanks to the simplicity of the decoder. Nevertheless, to find constructive families of these codes has been recognized to be a hard job. Also, the majority of known MLD codes are cyclic which are limited in the range of the rates. In this paper a new adaptation of the Iterative threshold decoding algorithm is considered, for decoding Quasi-Cyclic One Step Majority logic codes (QC-OSMLD) codes of high rates. We present the construction of QC-OSMLD codes based on Singer difference sets of rate 1/2, and codes of high rates based on Steiner triple system which allows to have a large choice of codes with different lengths and rates. The performances of this algorithm for decoding these codes on both Additive White Gaussian Noise (AWGN) channel and Rayleigh fading channel, to check its applicability in wireless environment, is investigated.**

***Keywords*—Iterative threshold decoding; Quasi-Cyclic codes; OSMLD codes; Majority logic decoding; Steiner Triple System; BIBD**

I. Introduction

Today LDPC codes [1] are present in most Telecom standards like DVB-S2 and WiMAX [2]. However, the decoding of these codes remain algorithmically complex and in situations such as the DVB-S2 [3] are often concatenated with codes such as Reed Solomon to improve performances. In our point of view, the MLD codes are better competitors for LDPC codes and this for several reasons. In fact, the hardware implementation of these codes is very simple and only requires AND gates. The cyclic OSMLD codes can be decoded iteratively by an extension of the Massey algorithm [4] which is less complex than the believe propagation algorithm but almost with the same performances.

In this article, the studied subject is QC-OSMLD codes which, unlike the cyclic OSMLD codes, offer a wide range of rates equivalent to that used in the standards. The first QC-OSMLD codes were constructed by L. Townsend and E. Weldone [5], but most of these codes are constructed by either computer search or hand through trial-and-error, except the construction based on Singer Difference set, which is a geometry projective method. Later, Chen Zhi and al[6] had given a mathematical formulation for the construction of QC-OSMLD codes with high rates, these codes are based on Steiner Triple system (STS) .

Iterative threshold decoding QC-OSMLD codes of rate 1/2 has proven to perform remarkably well on Additive White Gaussian Noise (AWGN) channel [7]. the purpose of this paper is to investigate the performance of iterative threshold decoding of QC-OSMLD codes of rate $\frac{n_0-1}{n_0}$ constructed from Singer Difference Set, and STS on both Rayleigh fading channel and AWGN channel, is investigated .

The organization of the paper is as follows. The first section provides the reader with a concise description of not only the OSMLD codes but also the majority logic decoding algorithm and the Quasi-Cyclic Codes. Afterwards, the second section defines the Singer Difference Set, and it presents the constructed codes based on this algorithm. Section 3 is about the construction of QC-OSMLD codes of rate of the form $\frac{n_0-1}{n_0}$ based on STS, starting with a description of Balanced Incomplete Block Design (BIBD), then presenting the STS construction, and eventually, presenting the different constructed codes. Section 4 introduces the encoding method after describing the iterative threshold decoding algorithm and explaining the modification made for the Rayleigh fading channel. Finally, the last part presents the simulations results and analyses the ITD algorithm for decoding the constructed codes on both Rayleigh fading channel and AWGN channel.

II. Quasi-cyclic OSMLD Codes

A. OSMLD Codes

Consider an (n, k) linear code C with parity-check matrix H. The row space of H is an (n, n-k) code, denoted by $C^\perp$, which is the dual code of C or the null space of C. For any vector v in C and any vector w in $C^\perp$, the inner product of v and w is zero [8]. Now let consider that a codeword vector in C is transmitted over a binary symmetric channel. Taking into consideration that $e(e_1, e_2,..., e_n)$ and $r(r_1, r_2,..., r_n)$ are the error vector and the received vector, respectively. Then r = v + e. The construction of the below linear sum of the received vector for any vector w in the dual code C$^{\perp}$:

$$A = \sum_{p=1}^{n} r_p w_p \tag{1}$$

Which is called a parity-check sum. Using the fact that <w,v>=0, the following relationship between the parity-check sum A and error digits in e is obtained:

$$A = \sum_{p=1}^{n} e_p w_p \tag{2}$$

Suppose that there exist J vectors in the dual code C$^{\perp}$, which have the following properties:

1) The jth component of each vector w$_i$ is a 1.
2) For i≠j there is at most one vector whose ith component is a 1.

These J vectors are said to be orthogonal on the jth digit position. They are called orthogonal vectors. Now, let us form J parity-check sums from these J orthogonal vectors,
For each i in 1,.., J $A_i = \sum_{p \neq 1} e_p + e_j$ the error digit e$_j$ is checked by all the check sums above. Because of the second property of the orthogonal vectors, any error digit other than e$_j$ is checked by at most one check sum. These J check sums are said to be orthogonal on the error digit e$_j$. If all the error digits in the sum A$_i$ are zero for i≠j, the value of A$_i$ is equal to e$_j$. Based on this fact, the parity-check sums orthogonal on e$_i$ can be used to estimate e$_i$, or to decode the received digit r$_i$.

B. Majority logic decoding principle

The error digit e$_j$ is decoded as 1 if at least one-half of the check sums orthogonal on e$_j$, are equal to 1; otherwise, e$_j$ is decoded as 0 like majority rule [8]. When C is a cyclic code, each e$_i$ can be decoded simply by cyclically permuting the received word r into the buffer store.

Example 1:
Let us consider the (7,3) code, which is the short code in difference set codes class. This code is specified by the perfect difference set P=0, 2, 3 of order 21. From this perfect set, the following three check sums orthogonal on e$_7$ could be formed:
A$_1$ = e$_4$ + e$_5$ + e$_7$
A$_2$ = e$_2$ + e$_6$ + e$_7$
A$_3$ = e$_1$ + e$_3$ + e$_7$
If a simple error e=(000001) occurs, then A$_1$ = A$_2$ = A$_3$ = 1. If a double error occurs; for example, e$_7$=1 and one value of e$_1$, ..., e$_6$ is equal to 1, then two values of A$_i$ are 1. So we can say that :
- e$_7$=1 if only and if at least 2 values of A$_i$ are 1
- e$_7$=0, otherwise

C. Quasi-cyclic Codes

A code is said to be quasi-cyclic if every cyclic shift of a codeword by p positions results in another codeword [9]. Therefore, a QC codes are a generalization of cyclic codes with p = 1. A QC code (mn$_0$, mk$_0$) with a minimum distance d based on difference set can be specified with k$_0$ disjoints difference sets {D$_1$,D$_2$,...,D$_{k_0}$} such that D$_i$(d_{i0},d_{i1},d_{i2},...,$d_{i(S-1)}$) of order S, chosen from the set {0,1,2,..., mk$_0$} [5]. The parity check matrix H in the systematic form of such code is completely defined as follows:

$$H = [P_1 P_2 ... P_{k_0} I_{n-k}] \tag{3}$$

The circulant matrix P$_i$ is deducted from the difference set D$_i$; the elements of D$_i$ can specify the position in the matrix header P$_i$ with one, while d$_{ij}$ represents one in the position j, the others rows are obtained by a cyclic shift of the header. Where I represents the identity matrix.

The majority logic decoding algorithm for QC codes is the same as cyclic codes. However, there is a little bit difference between them. Hence, in cyclic codes each error digit e$_i$ can be decoded by cyclically permuting the received word r, but in QC codes in systematic form, shift is done cyclically by one position of each (n-k) bits simultaneously.

Example 2:
Let consider the QC code C(6,3,3). This code is of the rate 1/2 and based on the Singer difference set DS{0,1} of order 2.

The parity check matrix H in systematic form is [P I$_3$]

$$\begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

The parity-check sum orthogonal on e$_3$ is obtained from the parity check matrix H :
A$_1$ = e$_2$ + e$_3$ + e$_5$
A$_2$ = e$_1$ + e$_3$ + e$_6$

III. Construction based on Singer Difference Set

A. Signer Difference Set

A difference set [10] of order S and modulo m ≥ S(S - 1)+1 is defined as a collection of S integer specified from the set {0,1,...,m-1} such that no two of the S(S - 1) ordered differences modulo m are identical. If m=S(S - 1)+1, then for any non-zero integer n < m, there is exactly one pair of elements in the difference set such that their difference is congruent to n modulo m. Such a set is called a perfect difference set.

Singer [11] has demonstrated how to construct such sets when $S = p^u + 1$ and p is prime. Points and lines in the projective geometry PG(m - 1, q) form a difference set of parameters [q,m]. The construction of Singer difference sets with parameters [q,m] is straightforward if a primitive polynomial of degree m in F_q is known. The following is the basis of Singer algorithm:

1) Choose a primitive polynomial of degree m in F_q

$$f(x) = x^m + \sum_{i=1}^{m} a_i x^{m-i} \tag{4}$$

2) Choose the start value λ_0 = 0, λ_1, λ_2,..., λ_{m-1}

3) Calculate the recurrence relation

$$\lambda_n = -\sum_{i=1}^{m} a_i \lambda_{n-i} \tag{5}$$

4) The set of integers { $0 \leq$ i $< \frac{q^m - 1}{q-1}$: λ_i = 0 } is a Pefect Difference Set

Example 3:

1) Taking primitive polynomial $x^4 + x^3 + 2$ of degree 4 on F_3
2) Choosing the start sequence λ_0 = 0, λ_1 = 0, λ_2 = 1
3) Then, calculating the recurrence relation as follows: λ_n = 2λ_{n-1} + λ_{n-4} yields the sequence 10001212201112222020211201021002212022002000 . . .
4) The Positions where λ_i = 0 are {1,2,3,9,17,19,24,26,29,30,35,38,39} that form the Perfect Difference Set.

B. *Construction based on Singer Difference Set*

The Singer construction allows to have codes with rate $\frac{1}{2}$. It's possible to construct a single perfect difference set of order S = p^u + 1 and modulo m = p^{2u} + p^u + 1 where p is a prime number, and u is a positive integer. As a results, it's always possible to construct codes with minimum distance p^u + 2 and length 2(p^{2u} + p^u + 1).

All constructed codes are OSMLD and completely orthogonalizable since the orthogonal parity-check equations number J is always equal to d$_{min}$-1, where d is the minimum distance of the code.

In [5], Townsend and Weldon has constructed a rate $\frac{1}{2}$ codes of small length up to n = 366. With the help of Magma [12], many perfect difference sets have been constructed, which allows to construct a large number of QC-OSMLD codes of large lengths up to n = 2.10^9.

Table 1 shows a part of constructed codes. The parameters in this table are :

P^u : P is prime, and u an integer

m : is the modulo, m = p^{2u} + p^u + 1

(n , k) : n is the length and k is the dimension of the code

d : is the minimum distance of the code

Difference Sets : Represent the constructed Singer difference set

Density : is the density of the parity check matrix H

LDPC : specifies if the code is Low Density Parity Check (LDPC) code, for that it's obligatory to have density $\leq$ log$_2$(n)

From this table, it's clear that the majority of QC-OSMLD codes are LDPC codes which allow us to decode them with LDPC decoder like Sum-Product, Belief Propagation ...

IV. Construction based on Steiner Triple System

Historically, Smith [13] presented in 1968 an application of incomplete block designs to the construction of several families of error-correcting codes which may be decoded using a relatively simple majority logic decoding procedure. However, he didn't give any explicit construction for such designs. Special cases of these codes are equivalent to the Self-orthogonal Quasi-cyclic codes based on Perfect Difference Sets discussed by Townsend and Weldon [5] (1967).

Chen Zhi and al stated an [6] explicit constructions of many classes of difference families considered as base blocks for Steiner designs. He presented a construction of infinite optimal self-orthogonal quasi-cyclic codes with high rates.

This section describes briefly the different construction methods of QC-OSMLD codes based on block design. Also, a part of constructed codes generated automatically by Matlab programmes is represented.

A. *Balanced Incomplete Block Design*

A Balanced Incomplete Block design (BIBD) [10] is a pair (V, B) where V is a set and B is a collection of b k-subsets of V (blocks) such that each element of V is contained in exactly r blocks, and any 2-subset of V is contained in exactly λ blocks. The numbers v, b, r, k, and λ are said parameters of the BIBD.

Trivial necessary conditions for the existence of a BIBD(v, b, r, k, λ) are :

1) $vr = bk$
2) $r(k-1) = \lambda(v-1)$

The incidence matrix of a BIBD (V, B) with parameters v, b, r, k, λ is a $v \times b$ matrix $A = (a_{ij})$, in which $a_{ij} = 1$ when the i^{th} element of V occurs in the j^{th} block of B, and $a_{ij} = 0$ otherwise.

B. *Construction methods*

1) QC-OSMLD codes based on STS ($v = 6t + 1$): This construction [6][14] is applicable for $k = 3$ and v is a power of a prime of the form $v \equiv 1 (mod\, 6)$. Considering for each v, $GF(v = p^e)$, the Galois field of order v. Let ω be a primitive root. Then the STS difference family with parameters $v = 6t + 1$, $b = 6t^2 + t$, $r = 3t$, $k = 3$, $\lambda = 1$, is described by base blocks typically given by the form :

$$(\omega^0, \omega^{2t}, \omega^{4t}),\ (\omega^i, \omega^{2t+i}, \omega^{4t+i}),\ (\omega^{t-1}, \omega, \omega^{5t-1})$$

Another construction proposed by Rosa [15][10], for which the knowledge of a primitive root is not required, may be applicable for these designs using Skolem Sequences.

From [14] the block designs given above, an infinite class of optimal QC-OSMLD $[(t+1)(6t+1), t(6t+1), 4]$ codes with the basic block length $n_0 = (t+1)p^{e-1}$ is specified.

TABLE I: Rate $\frac{1}{2}$ QC-OSMLD codes based on Perfect Difference sets

P^u	m	(n,k)	d	Difference Sets	Density	$\log_2$(n)	LDPC
1	3	(6,3)	3	0 1	66,66	2.5849	No
2	7	(14,7)	4	0 1 3	28 .57	3,8073	No
3	13	(26,13)	5	0 1 3 9	19.23	4,7004	No
2^2	21	(42,21)	6	0 1 4 14 16	14.28	5,3923	No
5	31	(62,31)	7	0 1 3 10 14 26	11.29	5,9541	No
7	57	(114,57)	9	0 1 6 15 22 26 45 55	8.33	6,5849	No
2^3	73	(146,73)	10	0 1 12 20 26 30 33 35 57	7.89	6,8328	No
3^2	91	(182,91)	11	0 1 37 39 51 58 66 69 82 86	6.04	7,5077	Yes
11	133	(266,133)	13	0 1 3 17 21 58 65 73 100 105 111 124	4.88	8,0552	Yes
13	183	366,183)	15	0 1 3 24 41 52 57 66 70 96 102 149 164 176	4.09	8,5156	Yes
2^4	273	(546,273)	18	0 1 22 33 83 122 135 141 145 159 175 200 226 229 231 238 246	3,09	9,09	Yes
17	307	(614,307)	19	0 1 3 30 37 50 55 76 98 117 129 133 157 189 199 222 293 299	2,75	9,26	Yes
19	381	(762,381)	21	0 1 3 13 28 51 65 82 86 104 112 145 201 212 217 241 261 307 339 375	2,26	9,57	Yes
23	553	(1106,553)	25	0 1 3 14 31 60 64 109 146 151 185 265 286 313 321 337 357 375 454 460 479 486 501 544	1,77	10,11	Yes
29	871	(1742,871)	31	0 1 3 23 30 41 88 97 132 165 169 186 201 211 235 306 319 345 425 431 542 547 561 592 604 620 668 719 811 819	1,66	10,76	Yes
31	993	(1986,993)	33	0 1 3 13 101 127 154 169 204 210 226 235 259 289 297 317 356 434 474 478 495 538 570 584 589 607 618 654 749 756 801 920	1,37	10,95	Yes
37	1407	(2841,1407)	39	0 1 3 25 32 82 99 208 313 410 453 479 487 557 621 649 709 736 742 782 827 837 848 890 895 899 913 951 1040 1088 1123 1142 1172 1213 1252 1272 1288 1395	1,24	11,47	Yes
47	2257	(4514,2257)	49	0 1 3 131 138 143 296 377 381 457 566 590 690 712 773 802 891 905 973 979 996 1030 1039 1050 1065 1075 1083 1102 1123 1238 1270 1337 1387 1434 1528 1541 1590 1606 1636 1757 1788 1816 1858 1914 1978 2033 2144 2219	1,06	12,14	Yes
97	9507	(19014,9507)	99	0 1 3 37 52 191 308 332 433 914 919 984 1093 1155 1231 1238 1600 1678 1723 1732 1755 1773 1826 1930 1938 2099 2116 2141 2457 2712 2859 3058 3187 3466 3524 3655 3675 3748 4139 4145 4183 4297 4301 4518 4528 4600 4720 4777 4964 5043 5054 5176 5268 5329 5356 5496 5526 5601 5617 5851 6151 6173 6491 6539 6759 6778 6792 6878 7021 7163 7226 7290 7490 7650 7747 7860 7941 8028 8056 8154 8304 8339 8370 8438 8450 8505 8534 8574 8797 9005 9048 9094 9107 9133 9154 9270 9326 9400	0,49	14,21	Yes
181	32943	(65886,32943)	183	0 1 129 145 211 306 460 514 547 748 771 800 894 1044 1101 1152 1277 1553 1798 1833 1840 1888 1924 2118 2381 2431 2564 2601 2613 3054 3308 3669 4369 4507 4620 4807 4839 5136 5342 5452 5623 5798 5808 5914 6488 6577 6798 6816 7063 7590 7745 7894 7935 7993 7995 8365 9166 9234 9572 9836 10220 10263 10355 10692 10764 10895 11081 11272 11376 11598 11645 12078 12215 12453 12498 12536 12807 12973 13250 13296 13384 13423 13858 13935 14408 14494 14603 14818 14892 15318 15397 15478 15625 15797 16219 16454 16607 17068 17141 17200 17211 17330 17696 17722 18264 18291 18433 18659 18715 18795 18958 19607 19714 19879 20145 20324 20523 20585 21192 21349 21370 21373 21728 22555 22586 22815 22929 23208 23376 23535 23550 23894 24074 24326 24490 24518 24802 24808 24904 24926 25681 25822 25839 26204 26421 26440 26474 26518 26538 26543 26658 26966 27006 27071 27363 28337 28404 28504 28697 28895 28971 29246 29883 29897 29958 30097 30106 30110 30322 30352 30473 30771 31030 31192 31380 31582 32046 32445 32676 32739 32747 32832	0,27	16,00	Yes

2) QC-OSMLD codes based on STS ($v = 6t + 3$): The construction of such designs using the Extended Skolem Sequences is proposed. A Skolem sequence of order n is a sequence $S = (s_1, s_2, ..., s_{2n})$ of $2n$ integers satisfying the conditions :

1) for every $k \in \{1, 2, ..., n\}$ there exists exactly two elements $s_i, s_j \in S$ such that $s_i = s_j = k$
2) if $s_i = s_j = k$ with $i < j$, then $j - i = k$.

Skolem sequences are also written as collections of ordered pairs $\{(a_i, b_i) : 1 \leq i \leq n,\ b_i - a_i = i\}$ with $\cup_{i=1}^{n}\{a_i, b_i\} = \{1, 2, ..., 2n\}$

Example 4:
A Skolem sequence of order 5 : $S = (1, 1, 3, 4, 5, 3, 2, 4, 2, 5)$ or, equivalently, the collection $\{(1, 2), (7, 9), (3, 6), (4, 8), (5, 10)\}$.

An extended Skolem sequence of order n is a sequence $ES = (s_1, s_2, ..., s_{2n})$ of $2n+1$ integers satisfying conditions 1 and 2 of the previous definition and :

3) there is exactly one $s_i \in ES$ such that $s_i = 0$.

The element $s_i = 0$ is called the hook or zero of the sequence.

3) Construction By A. Rosa: Suppose $\{1, ..., 3n+1\}\backslash\{2n+1\}$ can be partitioned into m triples $\{a, b, c\}$ such that $a+b = c$ or $a + b + c \equiv 0(mod\, 6n + 3)$. (This problem is called the second Heffter difference problem). Then the set of all triples $\{0, a, a+b\}$, together with "short block" $\{0, 2n+1, 4n+2\}$, is a $(6n + 3, 3, 1)$ cyclic partial difference family; the base blocks for a $STS(6n + 3)$. Heffter's second difference problem is solved using extended Skolem sequences of order n with a hook in the nth position. From such a sequence, the pairs (b_r, a_r) is constructed such that $b_r - a_r = r$, for $1 \leq r \leq n$. Then the set of all triples $(r, a_r + n, b_r + n)$ is taken, for $1 \leq r \leq n$.

Below an explicit construction for the required Skolem sequences (as ordered pairs) :

$$n = 4s\ :\ \begin{cases} (r, 4s - r + 1) & r = 1, ..., s - 1 \\ (s + r - 1, 3s - r) & r = 1, ..., s - 1 \\ (4s + r + 1, 8s - r + 1) & r = 1, ..., s - 1 \\ (5s + r + 1, 7s - r + 1) & r = 1, ..., s - 1 \\ ((2s - 1, 2s), (3s, 5s + 1)), & (3s + 1, 7s + 1), (6s + 1, 8s + 1) \end{cases}$$

$$n = 4s+1,\ (n>5)\ :\ \begin{cases} (r, 4s-r+2) & r=1,...,2s \\ (5s+r, 7s-r+3) & r=1,...,s \\ (4s+r+2, 8s-r+3) & r=1,...,s-2 \\ (2s+1, 6s+2), (6s+1, 8s+4), & (7s+3, 7s+4) \end{cases}$$

$$n = 4s+2,\ (n>2)\ :\ \begin{cases} (r, 4s-r+3) & r=1,...,2s \\ (4s+r+4, 8s-r+4) & r=1,...,s-1 \\ (5s+r+3, 7s-r+3) & r=1,...,s-2 \\ (2s+1, 6s+3), (2s+2, 6s+2), & (4s+4, 6s+4) \\ (7s+3, 7s+4), & (8s+4, 8s+6) \end{cases}$$

$$n = 4s-1\ :\ \begin{cases} (r, 4s-r) & r=1,...,2s-4 \\ (4s+r+1, 8s-r) & r=1,...,s-2 \\ (5s+r, 7s-r-1) & r=1,...,s-2 \\ (2s, 6s-1), (5s, 7s+1), & (4s+1, 6s), (7s-1, 7s) \end{cases}$$

When $n = 2$, take the sequence :

$$\{(1,2), (4,6)\}$$

When $n = 5$, take :

$$(1,5),\ (2,7),\ (3,4),\ (8,10),\ (9,12)$$

When $n = 1$, the sequence does not exist. The construction above gives $STS(6n+3)$ with parameters $(v, b, r, k, \lambda) = (6t+3, (3t+1)(2t+1), 3t+1, 3, 1)$.

4) QC-OSMLD codes based on STS ($v = 12t+7$): This construction [10] is available for v a prime power in the form $v = 12t+7$. Let ω be a primitive root of the Galois field $GF(12t+7 = p^e)$. Then, the base blocks of a design with parameters $v = 12t+7$, $b = (2t+1)(12t+7)$, $r = 3(2t+1)$, $k = 3$, $\lambda = 1$ are given in the form $(\omega^{2i}, \omega^{2i+2t}, \omega^{4t+i})$

5) QC-OSMLD codes based on STS ($v = 12t+1$): This construction [6] is applicable for v a prime power in the form $v = p^e = 12t+1$. Let ω be the primitive root of $GF(p^e)$ such that $\omega^{4t} - 1 = \omega^q$ where q is odd. Then, the base blocks of a design with parameters $(v = 12t+1, b = t(12t+1), r = 3(12t+1), k = 4, \lambda = 1)$ are given in the form : $(0, \omega^0, \omega^{4t}, \omega^{8t})$, $(0, \omega^{2i}, \omega^{2i+4t}, \omega)$ $(0, \omega^{2t-2}, \omega^{6t-2}, \omega^{10t-2})$ such that $i = 0, ..., t-1$. Block designs [6] given above specify an infinite class of optimal QC-OSMLD $(n, k, d_{min}) = ((t+1)(12t+1), t(12t+1), 5)$ codes with basic block length $n_0 = (t+1)p^{e-1}$.

6) QC-OSMLD codes based on STS ($v = 20t+1$): This construction [6] is applicable for v a prime power in the form $v = p^e = 20t+1$. Let ω be the primitive root of $GF(p^e)$ such that $\omega^{4t} + 1 = \omega^q$ where q is odd. Then the base blocks of a design with parameters $(v = 20t+1, b = t(20t+1), r = 5t, k = 5, \lambda = 1)$ are given in the form : $(\omega^{2i}, \omega^{4t+2i}, \omega^{8t+2i}, \omega^{12t+2i}, \omega^{16t+2i})$ such that $i = 0, ..., t-1$. Block designs [6] given above specify an infinite class of optimal QC-OSMLD $(n, k, d_{min}) = ((t+1)(20t+1), t(20t+1), 6)$ codes with basic block length $n_0 = (t+1)p^{e-1}$.

C. New constructed codes

The following tables represent a small part of many constructed QC-OSMLD codes based on the methods described above. The length n, the dimension k, the minimum distance d, and the rate, and also the base blocks which represent the headers of the circulant matrix of the parity-check matrix H, are represented. Due to the significant number of base blocks in high rate, aren't represented in the tables.

TABLE II: QC-OSMLD codes based on STS $v = 12t+7$

n	k	d	Rate	Base blocks
76	57	4	0.75	1 4 16
186	155	4	0.833	1 19 20 9 16 29
344	301	4	0.87	1 4 41 9 12 25 10 36 38
804	737	4	0.92	1 19 26 4 9 52 16 36 37 7 10 64 14 40 55
4564	4401	4	0.96	-

TABLE III: QC-OSMLD codes based on STS $v = 6t+3$

n	k	d	Rate	Base blocks
50	35	4	0.7	1 3 4 2 6 8
91	70	4	0.77	1 4 6 2 5 8 3 8 11
144	117	4	0.81	1 5 6 2 7 10 3 8 12 4 11 13
206	176	4	0.85	1 6 10 2 7 12 3 8 9 4 13 15 5 14 17
286	247	4	0.86	1 7 12 2 8 11 3 9 15 4 10 14 5 14 16 6 16 17
851	782	4	0.919	-
1000	925	4	0.925	-
16800	16485	4	0.981	-

TABLE IV: QC-OSMLD codes based on STS $v = 6t + 1$

n	k	d	Rate	Base blocks
39	26	4	0.667	1 3 9 2 5 6
76	57	4	0.75	1 7 11 2 3 14 4 6 9
125	100	4	0.80	1 6 11 2 12 22 4 19 24 8 13 23
186	155	4	0.833	1 5 25 3 13 15 8 9 14 11 24 27 2 10 19
344	301	4	0.875	1 6 36 3 18 22 9 11 23 26 27 33 13 35 38 19 28 39 14 30 41
1649	1552	4	0.941	-
2071	1962	4	0.947	-
64898	64021	4	0.99	-

TABLE V: QC-OSMLD codes based on STS $v = 12t + 1$

n	k	d	Rate	Base blocks
75	50	5	0.666	0 1 6 11 0 4 19 24
148	111	5	0.75	0 1 10 26 0 3 4 30 0 9 12 16
245	196	5	0.80	0 1 25 37 0 9 29 38 0 16 23 32 0 32 43 46
366	305	5	0.833	-
511	438	5	0.86	-
873	776	5	0.89	-
1331	1210	5	0.91	-
10470	10121	5	0.966	-

TABLE VI: QC-OSMLD codes based on STS $v = 20t + 1$

n	k	d	Rate	Base blocks
244	183	6	0.75	1 9 20 34 58 4 14 19 36 49 13 15 16 22 56
405	324	6	0.80	1 7 19 49 52 4 28 34 46 76 16 22 31 55 61 1 7 43 58 64
606	505	6	0.83	-
847	726	6	0.86	-
1810	1629	6	0.90	-
4215	3934	6	0.93	-
6859	6498	6	0.95	-

V. Iterative Treshold Decoding

A. Encoding

In the case of QC codes, the encoding can be achieved with simple shift registers while the complexity is linear [9][16]. Because the quasi-cyclic code is not in systematic form, an additional k-stage register is required by this encoder to store the information symbols of the next block until encoding is completed. This difficulty can be avoided by using an equivalent systematic code. In this case the codes constructed are in systematic form.

QC codes could be encoded by either generator matrix or polynomial multiplication. These codes are defined by the parity check matrix H. The generator matrix G is obtained by the following transformation :

$$H = [PI_{n-k}] \Leftrightarrow G = [I_k P^T] \tag{6}$$

The encoding algorithm consists of multiplying the message i by the generator matrix G to get the codeword v.

$$v = i * G \tag{7}$$

The Quasi-cyclic codes have a polynomial form. Consider C(n,k) a systematic quasi-cyclic code with rate $\frac{1}{2}$, and let P which defines the code be the circulant matrix. The information vector to be encoded is denoted by i, then

$$v = iG = [i \ iP] \tag{8}$$

Let i(x) and p(x) represent the information vector and the header of the circulant matrix P in polynomial form, respectively. Obviously, the remaining rows of P are:

$$(xp(x), x^2p(x), ..., x^{k-1}p(x)) \ mod \ x^k - 1 \tag{9}$$

The algebra of polynomials modulo x^m -1 is equivalent to the algebra of m x m circulant matrices; besides, the polynomial product i(x)p(x) mod x^k-1 is similar to multiplying the vector i by the circulant matrix P. Hence,

$$v(x) = [i(x), i(x)p(x)] \tag{10}$$

Example 5: Let consider the same code as in the example 2. Now, to transmit the message i=101.
i = 101 then i(x) = 1+x^2
And p = 101 then p(x) = 1+x^2
Then the codeword is:
v(x) = [i(x),i(x)*p(x)]=[1+x^2,(1+x^2)*(1+x^2)]=[1+x^2,1+x]
⇒v = 101110.
In the case of QC codes with rate of the form $\frac{n_0-1}{n_0}$, the encoding like codes with rate 1/2 can be realised by either generator matrix or polynomial multiplication [9].

These codes are defined by the parity check matrix H of the form :

$$H = [P_1 P_2 ... P_{k_0} I_{n-k}] \tag{11}$$

The generator matrix G is obtained by the following transformation :

$$H = [P_1 P_2 ... P_{k_0} I_{n-k}] \Leftrightarrow G = \begin{bmatrix} & P_1^T \\ & P_2^T \\ I_k & \vdots \\ & \vdots \\ & P_{k_0}^T \end{bmatrix}$$

In the case of codes with rate $\frac{n_0-1}{n_0}$, there are many circulant matrix (P_1 P_2 ... P_{k_0}), then for encoding the information vector 'i', it must be divided into k_0 subgroup (i_1 i_2 ... i_{k_0}) then based on the equation (10), the following equation is obtained :

$$v(x) = [i(x), \sum_{j=1}^{k_0} i_j(x)p_j(x)] \tag{12}$$

To clear this, let us consider the following example.

Example 6:

Let consider the OSMLD QC(15,10) code, with the minimum distance is d=3. The disjoint difference sets of order S=2 which define the parity matrix H of this code are D_1\{0,1\} and D_2\{0,2\}.

The parity check matrix H in systematic form is [P_1 P_2 I_5]

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

After transformation of the parity check matrix H, the generator matrix G is obtained

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \end{pmatrix}$$

Now transmitting the message i=1001001011. To know the codeword v corresponding to the message i, it's possible to use the classical method by using the generator matrix G v = i*G = 111011010110101. For using the polynomial method, the message must be divided into k_0=2 vectors
i_1=10010 ⇒ $i_1(x)=1+x^3$
i_2=01011 ⇒ $i_2(x)=x+x^3+x^4$
From the generator matrix G, headers of the two circulant matrix P_1^T and P_2^T is obtained:
c_1 = 10001 ⇒ $c_1(x) = 1+x^4$
c_2 = 10010 ⇒ $c_2(x) = 1+x^3$
Then, the equation (9) must be calculated:
$v(x)=[i(x),(i_1(x)*c_1(x))+(i_2(x)*c_2(x))]$
$=[1+x^3+x^6+x^8+x^9,((1+x^3)*(1+x^4))+((x+x^3+x^4)*(1+x^3))]$
$v(x)=[1+x^3+x^6+x^8+x^9,(1+x^2+x^3+x^4)+(x^2+x^3)]$
$=[1+x^3+x^6+x^8+x^9,1+x^4]$ Then, the codeword to transmit is :
v = 100100101110001 Which is the same as the codeword obtained by using generator matrix G.

B. ITD

Threshold decoding is simply the logical extension to soft decisions of majority decoding described above. In Massey's original work [17], he considered two different variations of the decoding algorithm. Considering here the method which uses the Bi equations that are obtained from A_i by a simple transformation [18].
Thanks to its speed and simplicity, the Majority Logic (ML) decoding of Quasi-Cyclic codes is significant. Therefore, it is worth investigating which Quasi-Cyclic codes can be decoded using ML decoder. Majority logic decoding is well described in [8,19]. It consists of cyclic shift register, XOR matrix, majority gate and XOR for correcting the codeword bit under decoding.

The ITD algorithm which is based on SISO extension of Massey threshold decoding algorithm [17] was developed mainly for decoding Parallel Concatenated Block Codes [20] and product codes constructed from OSMLD Codes [21], later it was enhanced for decoding not only Generalized Parallel Concatenated OSMLD Codes [22] but also OSMLD block codes [4]. ITD algorithm has an improvement in error correcting related to standard ML decoding.

Considering the transmission of the codeword $C(c_1,c_2,...,c_n)$ over an Additive White Gaussian Noise channel (AWGN), using BPSK modulation. The soft decision, which is the Log Likelihood Ratio (LLR), on the j^{th} bit of the received word $R(r_1,r_2,...,r_n)$ can be calculated as follows:

$$LLR_j = ln\left[\frac{p(c_j = 1/R)}{p(c_j = 0/R)}\right] \tag{13}$$

The hard decision vector corresponding to the received vector R is denoted by $H(h_1,h_2,...,h_n)$. Where c_j is the j^{th} bit of the transmitted codeword. For a code with J orthogonal parity check equations; the equation (13) can be expressed as:

$$LLRj = ln\left[\frac{p(c_j = 1/\{B_i\})}{p(c_j = 0/B_i)}\right] \tag{14}$$

Where B_i, for i in \{1, ..., J\}, are obtained from the orthogonal parity check equations on c_j bit, as follows:

B_0 = h_j and each B_i with i in \{1,...,J\}, is calculated by eliminating the term h_j from the i^{th} orthogonal parity check equation. By applying BAYES rule, (14) becomes:

$$LLR_j = ln\left[\frac{p(\{B_i\}/c_j = 1)}{p(\{B_i\}/c_j = 0)} \times \frac{p(c_j = 1)}{p(c_j = 0)}\right] \tag{15}$$

Since the parity check equations are orthogonal on the j^{th} position, so the individual probabilities P(B_i/c_j = 1 or 0) are all independent and (15) can be written as:

$$LLR_j = \sum_{i=0}^{J} ln\left[\frac{p(\{B_i\}/c_j = 1)}{p(\{B_i\}/c_j = 0)}\right] + ln\left[\frac{p(c_j = 1)}{p(c_j = 0)}\right] \tag{16}$$

Assume that the transmitted symbols are equally likely to be +1 or -1, and thus the last term in (16) is null. As a result, the equation (16) becomes:

$$LLR_j = \sum_{i=1}^{J} ln\left[\frac{p(\{B_i\}/c_j = 1)}{p(\{B_i\}/c_j = 0)}\right] + ln\left[\frac{p(\{B_0\}/c_j = 1)}{p(\{B_0\}/c_j = 0)}\right] \tag{17}$$

According to [18], (17) can be expressed as:

$$LLR_j \simeq (1-2B_0)w_0 + \sum_{i=1}^{J}(1-2B_i)w_i \tag{18}$$

Where the value of (1-2B$_i$) is equal to +1 or -1, and w$_i$ is a weighting term proportional to the reliability of the ith parity check equation. then showing that:

$$(1-2B_0)w_0 = 4\frac{E_s}{N_0}r_j \tag{19}$$

Where E$_s$ is the energy per symbol, and N$_s$ is the noise spectral density.

$$w_i = ln\left[\frac{1+\prod_{k=1,k\neq j}^{k=n_j}\tanh(\frac{L_{ik}}{2})}{1-\prod_{k=1,k\neq j}^{k=n_j}\tanh(\frac{L_{ik}}{2})}\right] \tag{20}$$

Where n$_i$ is the total number of terms in the ith orthogonal parity check equation without c$_j$, *ik* represents the kth element of the ith parity check equation and with:

$$L_{ik} = 4\frac{E_s}{N_0} \mid r_{ik} \mid \tag{21}$$

Thus the soft output can be split into two terms, namely into a normalized version of the soft input r$_j$ and an extrinsic information L$_{E_j}$ representing the estimates made by the orthogonal bits on the current bit c$_j$. Hence, (18) becomes

$$LLR_j = 4\frac{E_s}{N_0}r_j + L_{E_j} \tag{22}$$

Using the following notation:

$$L_c = 4\frac{E_s}{N_0} \tag{23}$$

Which is called the reliability value of the channel.

The algorithmic structure of the SISO threshold decoding can be summarized as follows:

For each j = 1,...,n

- Compute the terms B$_i$ and w$_i$, i ∈ {1,..,J}
- Calculate B$_0$ and w$_0$
- Compute the extrinsic information L$_{E_j}$
- The Soft-output is obtained by:

$$\text{LLR}_j = \text{L}_c\text{r}_j + \text{L}_{E_j}$$

Iterative decoding process (see Figure 1) can be described as follows:

In the first iteration, the decoder only uses the channel output as input and generates extrinsic information for each symbol. In subsequent iterations, a combination of extrinsic information and channel output is used as input.

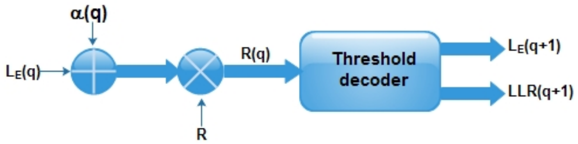

Fig. 1: Scheme of iterative threshold decoder

As shown in Figure 1, the soft input and the soft output of the qth decoder is achieved through the following equations:

$$R(q) = R + \alpha(q)L_E(q) \tag{24}$$

$$LLR(q+1) = L_cR(q) + L_E(q+1) \tag{25}$$

Where R(q) represents lines (or columns) of the received data, and L$_E$(q) is the extrinsic information computed by the previous component decoder. In the proposed procedure, a fixed value 1/J is used for the parameter α(q) and this for all iterations. The value chosen for α(q) reacts as an average of all J estimators which contribute in the computation of L$_{E_j}$.

C. Modification of Rayleigh fading channel

In the channel model, each received bit rj can be expressed as :

$$r_j = a_j\widehat{c}_j + n_j \tag{26}$$

In this representation, $\widehat{c}_j$ is a BPSK symbol associated to the transmitted bit cj, and nj is an AWGN. The Rayleigh variable aj is generated as:

$$a_j = \sqrt{x_j^2 + y_j^2} \tag{27}$$

where x$_j$ and y$_j$ are zero mean statistically independent Gaussian random variables each having a variance σ^2. Considering the power normalized to one as

$$E[a_j^2] = \sigma^2 = 1 \tag{28}$$

Which gives a variance of 0.5 for Gaussien variables.
The main matter in determining the required modification for ITD algorithm is the availability of channel side information on the Rayleigh fading channel. The threshold decoding algorithm has to be modified slightly by changing equation (23) which defines the reliability value of the channel by

$$L_c = 4\frac{E_s}{N_0}a_j \tag{29}$$

With this modification, it's possible to use the same decoder structure which was described in Figure 1.

VI. Simulation Results and analysis

This section considers simulation results and analysis for some decoding QC-OSMLD codes of rate $\frac{n_0-1}{n_0}$ and 1/2 with the Iterative Threshold Decoding algorithm. Some of our simulations are over AWGN channel , whereas others are over Rayleigh Fading channel; however, both of them are with modulation BPSK. Due to computational limitations, a minimal residual error of 200 have been used. In the simulations over Rayleigh fading channel, assuming an accurate fade estimate at the receiving and an independent Rayleigh distribution of the fades.

The performance improves with each iteration in all simulation results presented. The following results represent the performance of decoding an QC-OSMLD code with the ITD algorithm and comparison with classic Threshold decoding algorithm.

A. AWGN

The Figure 2 depicts the performance of QC-OSMLD code (366,183,15). The improvement is great for the first iterations and is negligible after the 6^{th} iteration. Figure 3 presents a comparison between three QC-OSMLD codes (366,183,15) (1106,553,25)and the(4514,2257,49). Observing that smaller codes length has best performance at low SNR, whereas at SNR>5 for the code (4514,2257,49) the performance improves quickly from 10^{-1} to 10^{-5} between SNR=5 and SNR=6. The next comparison is between the code (182,91,11) decoded with ITD 10 iterations and the code LDPC WiMax(192,96,10), which is from [23], decoded with the Belief propagation decoder. These two codes are of the same rate $\frac{1}{2}$, and they have nearly the same dimension and minimum distance.

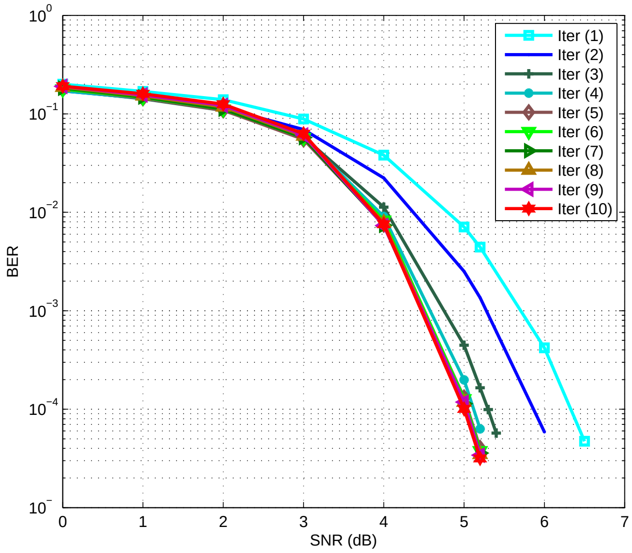

Fig. 2: The performance of the ITD algorithm with 10 iterations for decoding an QC-OSMLD code (366,183,15) over AWGN channel.

The Figure 4 shows that the LDPC WiMax code (192,96,10) decoded with BP algorithm outperforms the OSMLD-QC code (182,91,11) decoded with the ITD algorithm. However, the first decoder requires more iterations; on the other hand, the second one is less complex.

B. Rayleigh

The curves in the Figure 5 show the achieved bit error rates for the QC-OSMLD code (366,183,115) over Rayleigh fading channel. The number of iterations used is a 10, such that there is no significantly more to gain by more iterations. Observing that the performance increases with each iteration, and the improvement is negligible after the 7^{th} iteration .

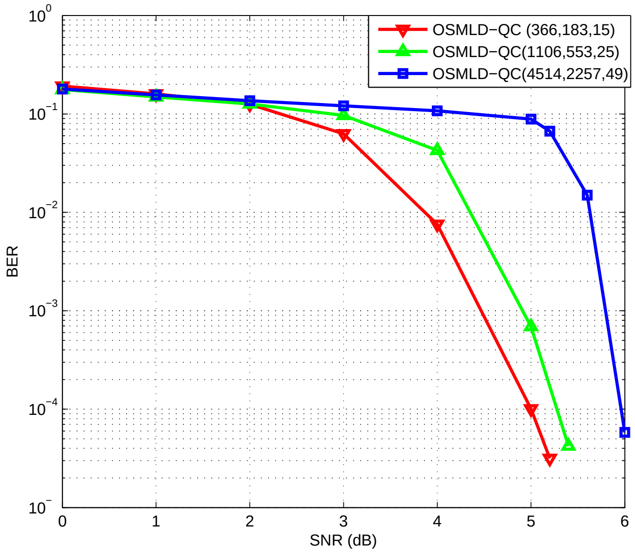

Fig. 3: Comparison between the performance of the QC-OSMLD codes (366,183,15), (1106,553,25) and (4514,2257,49) decoded with ITD 10 iteration, over AWGN channel

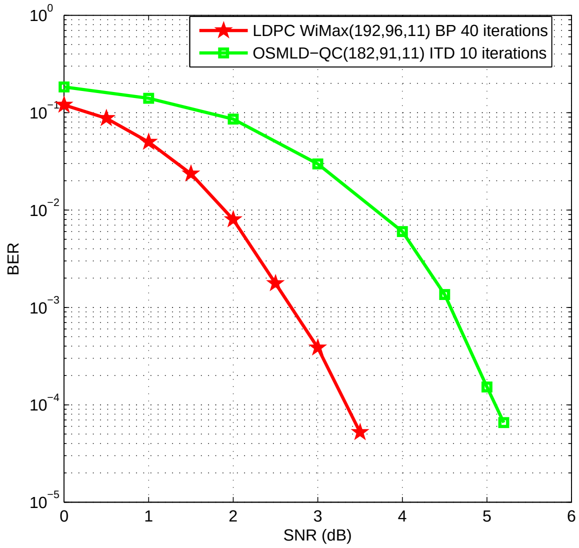

Fig. 4: Comparison between the performance of the code QC-OSMLD (182,91,11)decoded with ITD 10 iterations and and the code LDPC WiMAX(192,96,10)decoded with BP 40 iterations BP with 40 iterations

Figure 6 presents a comparison between three QC-OSMLD codes (366,183,15) (1106,553,25)and the(4514,2257,49) over Rayleigh fading channel. Observing that the same behaviour as the AWGN channel, but in high SNR.

The Figure 7 shows the performance of decoding the QC-

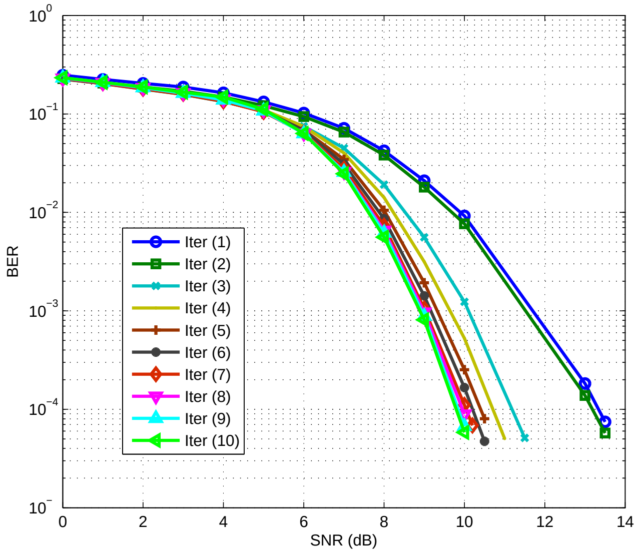

Fig. 5: The performance of the ITD algorithm with 10 iterations for decoding an QC-OSMLD code (366,183,15) over Rayleigh fading channel.

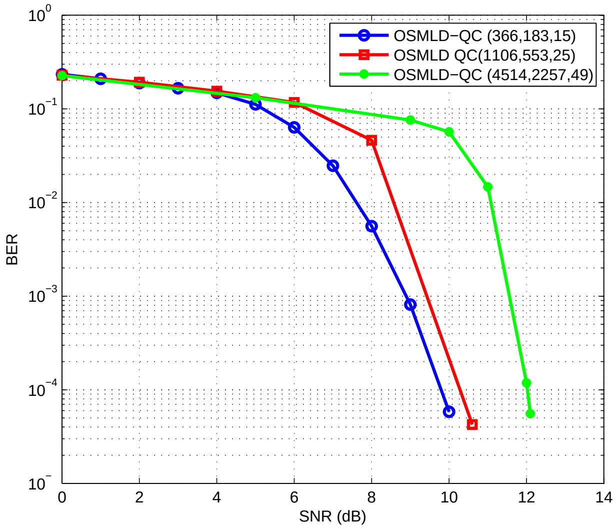

Fig. 6: Comparison between the performance of the QC-OSMLD codes (366,183,15), (1106,553,25) and (4514,2257,49) decoded with ITD 10 iterations, over Rayleigh fading channel

OSMLD code (366,183,15) on both Rayleigh and AWGN channels. As observed in the other simulations, the performance of this code in an independent Rayleigh channel is worse than that for the AWGN channel by approximately 5 dB. It is worth mentioning that the number of iterations needed is about the same for the both channels.

The figure 8 shows a comparison between the performance of decoding the QC-OSMLD codes of different rates. To simplify, two scenarios have been opted. In the first case, there are two codes (39, 26, 4) and (15, 10, 3) which have

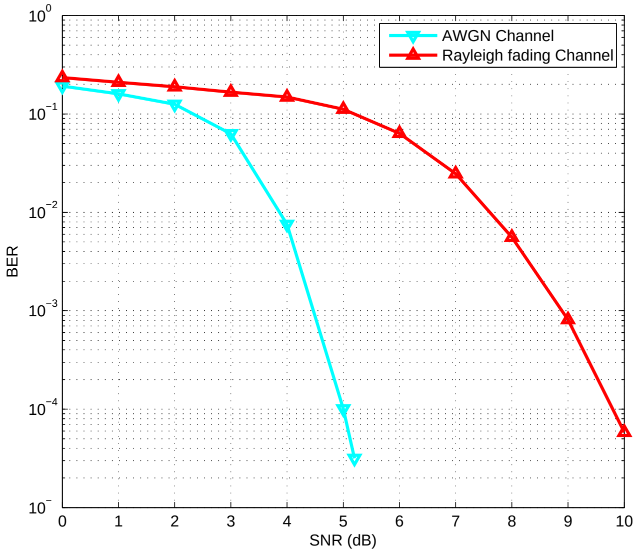

Fig. 7: Comparison between the performance of the ITD algorithm with 10 iterations for decoding an QC-OSMLD code (366,183,15) on AWGN channel and over Rayleigh fading channel

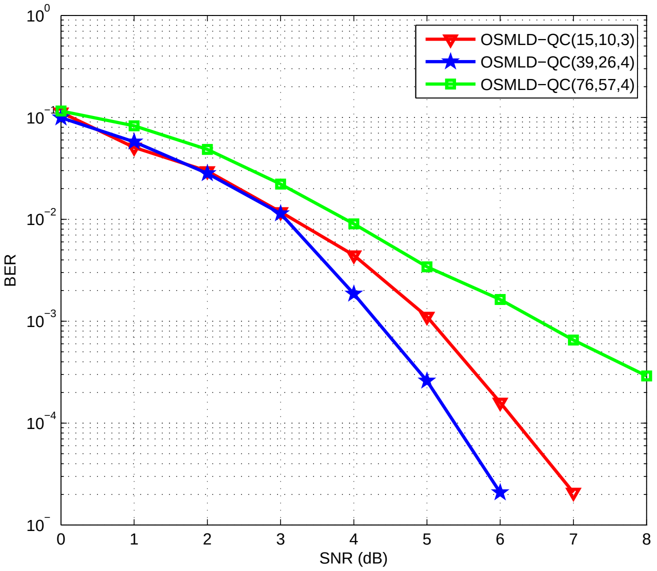

Fig. 8: Comparison between the performance of the QC-OSMLD codes (15,10,3), (39,26,4) of rate 2/3 and (76,57,4) of rate 3/4, decoded with ITD 10 iterations, over AWGN channel

the same rate 2/3. From the graph above, it's clear that as the code length increases, the performances rises, as well, which results in a gain of 1db at 10^{-5}. In the second case, two codes with different rates has been compared (76,57,4) of rate 3/4 and (39,26,4) of rate 2/3 holding the same minimum distance which is 4. As a result, even if the code length is large, the code of rate 3/4 is outperformed by the code of rate 2/3 with difference of 3db at 10^{-5}.

VII. Conclusion

In this paper, the construction of a class of QC-OSMLD codes based on Steiner triple system, and another class based on Singer difference sets has been investigated. The encoding methods has been presented for those codes. Also, the performances of decoding these codes with the ITD algorithm over AWGN channel and also over fading channel has been shown. The decoding algorithm used for AWGN channel is unchanged, and only the channel reliability factor needs to be redefined. The simulations results show that the constructed codes perform well when decoded with ITD algorithm. It is interesting to apply this iterative decoding algorithm on other channels models like Rice or Nakagami. Also as extension of this work we plan to investigate the performance of decoding rate $\frac{1}{n_0}$ QC-OSMLD codes with an adaptation of our ITD algorithm.

References

[1] R. G. Gallager,*"Low-Density Parity-Check Codes"*, Cambridge, MA: MIT Press, 1963.

[2] 802.16E-2005,802.16/COR1 IEEE Standard for Local and metropolitan area networks Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems Amendment for Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands, 2/2006.

[3] European telecommunications standards institude(etsi), *"Digital video broadcasting (dvb) second generation framing structure for broadband satellite applications"*, en 302 307 v1.1.1. URL: www.dvb.org.

[4] M. Lahmer and M. Belkasmi, *"Iterative Threshold Decoding of One Step Majority Logic Decodable block Codes"*,ISSPIT Conf, December 15-18, 2007, pp. 668 - 673 Cairo, Egypt.

[5] L. Townsend and E. Weldon, *"Self-Orthogonal Quasi-cyclic Codes"*,IEEE on Information Theory, vol. IT-13, No 2, pp. 183-195, April 1967.

[6] C. Zhi, F. Pingzhy and J. Fan, *"On Optimal Self-Orthogonal Quasi-Cyclic Codes"*, in Communications, 1990. ICC '90, Including Supercomm Technical Sessions. SUPERCOMM/ICC '90. Conference Record., IEEE International Conference on , vol., no., pp.1256-1260 vol.3, 16-19 April 1990

[7] K. Rkizat M. Lahmer and M. Belkasmi, *"Iterative Threshold Decoding of Quasi-Cyclic One Step Majority Logic Decodable Codes"*,WICT'15 Conf, December 14-16, 2015, Marrakesh, Morocco.

[8] S. Lin and D.J. Costello, *"Error Control Coding: Fundamentals and Applications"*, Prentice-Hall, Englewood Cliffs, NJ, 1983.

[9] W. W. Peterson and E. J. Weldon, *"Error-Correcting Codes"*, 2nd ed. Cambridge, MA: MIT Press, 1972.

[10] C. J. Colbourn, J. H. Dinitz, *"Handbook of Combinatorial Designs"*, Second Edition, Chapman and Hall/CRC, 2007. ISBN-13 978-1584885061.

[11] J. Singer, *"A theorem in finite projective geometry and some applications to number theory"*, AIMS Trans., vol. 43, pp. 377-385, 1938.

[12] J. Cannon and W. Bosmaresolution, *"Handbook of Magma functions"*, Version 2.10 Sydney, May 2003.

[13] K. J. C. Smith, *"An Application of Incomplete Block Designs to the Construction of Error-correcting Codes"*, University of North Carolina. Department of Statistics, 1968 - 42 pages.

[14] B. Vasic and O. Milenkovic, *"Combinatorial Constructions of Low-Density Parity-Check Codes for Iterative Decoding"*, IEEE Transactions on Information Theory, Vol 50, No 6. June 2004.

[15] A. Rosa, *"Poznámka o cyklických Steinerových systémoch trojíc"*,Mat. Fyz. Časopis 16 (1966), 285-290.

[16] C.L. Chen, W.W. Peterson E.J. Weldon Jr., *"Some results on quasi-cyclic codes"*,.

[17] J.L Massey, *"Threshold Decoding"*,Cambridge, Mass., M.I.T. Press, 1963.

[18] C. Clark and B. Cain, *"Error-Correction Coding for digital communications"*,Plenum Press, 1981.

[19] L.D. Rudolph, *"A Class of Majority Logic Decodable Codes"*, IEEE Trans. Inf. Theory, vol. IT-13, pp. 305-307, May 1967.

[20] M. Belkasmi, M. Lahmer, and M. Benchrifa, *"Iterative Threshold Decoding of Parallel Concatenated Block Codes"*,Turbo Coding 2006 Conf., 4-7 April 2006, Munich, Springer.

[21] M. Belkasmi, M. Lahmer, and F. Ayoub, *"Iterative Threshold Decoding of Product Codes Constructed from Majority Logic Decodable Codes"*, ICCTA.06 Conf., pp.2376-2381, 24-28 April 2006, Damascus Syrie.

[22] F. Ayoub, M. Belkasmi, I. Chana, *"Iterative Decoding of Generalized Parallel Concatenated OSMLD Codes"*, Applied Mathematical Sciences journal, Vol.4,no.41, pp.2021-2038, 2010.

[23] H. Michael and S. Stefan, *"Database of Channel Codes and ML Simulation Results"*,University of Kaiserslautern, 2015. URL: www.uni-kl.de/channel-codes;