



## High Speed Soft Decision Decoding of Linear Codes Based on Hash and Syndrome Decoding

Moulay Seddiq El Kasmi Alaoui<sup>1\*</sup>      Said Nouh<sup>1</sup>      Abdelaziz Marzak<sup>1</sup>

<sup>1</sup>*TIM Lab, Faculty of Sciences Ben M'sik, Hassan II University, Casablanca, Morocco*

\* Corresponding author's Email: [sadikkasmi@gmail.com](mailto:sadikkasmi@gmail.com)

---

**Abstract:** Digital transmission systems carry information from the source to the receiver using a physical medium such as cable, fiber optic or even propagation on a radio channel which isn't entirely reliable and causes the change of data originally emitted. Today the use of error correcting codes for protection and correction becomes an integral part in the design of communication systems and computers. In this work, we present a new interesting way to accelerate the decoding process of linear codes. The proposed method called Soft Decision Decoder by Hash Techniques (SDHT) is based on syndrome-decoding algorithm and hash techniques. The use of this latest allows reducing considerably the search time of all possible error patterns of weights less than a fixed threshold. SDHT is applicable on many linear codes and exploit the polynomial form to reduce again the run time decoding for cyclic codes. SDHT is successfully applied to decode some Bose Ray-Chaudhuri and Hocquenghem (BCH), Quadratic Residue (QR) and Extended Quadratic Residue (EQR) codes. The simulation results show that the proposed SDHT yield to good error correcting performances with reduced complexity. The comparison between SDHT and many competitors shows that it gives better performances in terms of correction rate. The experimental study of the decoding steps for the BCH(63,45,7) code shows that the time search of the most likely error pattern is reduced at about 26214153% comparing to an exhaustive search of all possible error patterns of weights less than or equal to 4. This study proves the huge success of the proposed SDHT decoder.

**Keywords:** Error correcting codes, SDHT, Hash techniques, Syndrome decoding.

---

### 1. Introduction

Protecting information on computer networks, telecommunication systems or data storage is a challenge for researchers. Generally, there are two solutions to ensure maximum reliability of the information exchanged, the first one is the increase of the emission power but this solution is very expensive. The second one is the addition of redundancy to the message to be transmitted or saved. Adding redundancy in the data to be protected is the principle of errors correcting codes.

Error correcting codes are used to ensure as maximum as possible the detection and correction of errors due to the noisy perturbation of data transmitted in communication channels or stored in digital supports. The decoding of linear codes is in general a NP-Hard problem and many decoders are

developed to detect and correct errors. The evaluation of the quality of a decoder is based on its performances in terms of Bit Error Rate and its temporal complexity.

In a communication system, before sending a message, the encoder create codewords by adding redundant information to the user vectors information and a decoder try to find the most likely transmitted message from the received sequence.

The quality of a Coding-Decoding system is measured by the bit error rate (BER) which it can be guaranteed at a given Signal-to-Noise Ratio (SNR), by its run time complexity and the hardware resources it needs.

The decoding algorithms can be separated into two categories, the first one is hard decision and the second one is soft decision. Hard decision algorithms work on the binary form of the received information and use the Hamming distance to

decode, in contrast soft decision algorithms work directly on the received symbols and generally they use the Euclidian distance as a metric to decide the most likely transmitted codeword [1].

A linear code of length  $n$  and dimension  $k$  is a linear subspace  $C$  with dimension  $k$  of the vector space  $F_q^n$  where  $F_q$  is the finite field with  $q$  elements. Generally a linear code is denoted by  $C(n, k, d)$  where  $d$  is the minimum distance between all distinct vectors of  $C$ . These vectors are called codewords, they are found using a generator matrix  $G$  or generator polynomial  $g$ . From  $d$  we can deduce the error correcting capability of a code  $C$  denoted by  $t$  and defined by  $t = \left\lfloor \frac{d-1}{2} \right\rfloor$ . We distinguish cyclic and non-cyclic codes; a cyclic code is a block code in which the cyclic permutation of any codeword gives another codeword that belongs to  $C$ .

In this paper, we present a Soft In Hard Out decoder, called Soft Decision Decoder by Hash Techniques (SDHT) which permits to rapidly correct all errors of weight less than or equal to a fixed threshold  $s$ . The proposed SDHT is based on syndrome-decoding algorithm and hash techniques. The use of this latest allows reducing considerably the search time of all possible error patterns of weights less than to a fixed threshold. SDHT is applicable on many linear codes and exploit the polynomial form to reduce again the run time decoding for cyclic codes. SDHT is successfully applied to decode some BCH, QR and EQR codes. The simulation results show that the proposed SDHT yield to good error correcting performances with reduced complexity.

The remainder of this paper is structured as follows. In section 2 we present some decoding algorithms as related works. In section 3 we present the proposed decoding algorithm. In section 4, we present the simulation results of the proposed decoder and we make a comparison with other ones. In the fifth section, we present a comparison of the complexity between SDHT and other competitors. Finally, a conclusion and possible future directions of this research are outlined in section 6.

## 2. Related works

Recently, several method deal the hard decision decoders algorithms have been published. Among the different methods developed, NESWDA (New Efficient Syndrome-Weight Decoding Algorithm) presented in [2] used to decode up to five errors for a binary systematic Quadratic Residue QR(47,24,11) code. This method is based on the weight of syndromes and proprieties of cyclic codes;

for the same code the authors of [3], have developed a Cyclic Weight (CW) algorithm for decoding. Also for the same code, in [4] the authors have presented a table lookup decoding algorithm to decode up to five errors. An approach based on the link between syndromes and correctable errors pattern is developed by using hash techniques [5]; in [6], the authors have presented an algebraic decoding algorithm to correct all patterns of four or fewer errors in the binary QR (41, 21, 9) code. In [7] the authors have proved that the extended quadratic residue binary codes are the only nontrivial extended binary cyclic codes that are invariant under the projective special linear group (PSL). In [8] a decoding of quadratic residue codes by using hashing search to determine error patterns is proposed. The authors of [9] introduced the Lagrange interpolation formula to calculate the needed primary unknown syndrome for the binary QR code and proposed hardware architecture to implement it; also, they decoded some binary QR codes by using the developed Berlekamp-Massey (BM) algorithm and Chien search.

For BCH codes we find a performances and comparative study between the BCH(15, 7, 2) and BCH(255, 231, 3) codes realized by the authors of [10]. In [11], the authors have proposed a performances study and a synthesis of some new algorithms developed in this field, for Reed Solomon (RS), BCH and Low Density Parity Check Codes(LDPC). In [12] a deep learning method to improve belief propagation algorithm was proposed, by attribution of weights to the edges of the Tanner graph the authors generalized the standard belief propagation algorithm. In [13] the authors have presented an iterative hard decision decoding algorithm for binary linear block codes over a binary symmetric channel (BSC). In [14-17] several hard decision decoders based on genetic algorithms (GA) are developed, the first one is the HDGA (Hard decision Decoder based on Genetic Algorithms) [14], the second one is the Bit Flipping decoding algorithm (BF) [15, 16] developed initially for LDPC codes and generalized after on linear block codes, its principle is the verification of many orthogonal equations. In [17], a decoder called ARDecGA (Artificial Reliabilities based Decoding algorithm by Genetic Algorithms) is presented; it uses a generalized parity check matrix to compute a vector of artificial reliabilities of the binary received sequence  $h$  and it exploits a genetic algorithm to find the maximum likelihood binary word to this vector. The algebraic hard decision decoder [18, 19] of Berlekamp-Massey is based on compute of syndromes and it has an efficient mechanism to

localize all corrigible errors, it is applicable on BCH codes. Another version of this last decoder is adapted for Quadratic residue codes [20]. In [21] we have presented two new fast and efficient hard decision decoders based on hash techniques for real time communication systems. The first one called HSDec with temporal complexity very low comparing to competitors but requires more memory space comparing to the second one whose was called HWDec, contrariwise this latest requires more run time but low memory space.

There are also several works that are interested in Soft Decision Decoders like the Soft Decoder (SD1) applied for BCH codes in Wireless Body Area Networks (WBAN) which presented in [22]. It is based on test and syndrome computing to find the error positions. Another Soft BCH decoder (SD2) was presented in [23] to improve hardware complexity and better error correcting capability. In [24], the authors have presented a Soft In-Hard Out (SIHO) version of the MacWilliams's permutation decoding algorithm (McPD) called Soft Permutation Decoding Algorithm (SPDA). In [25] authors have proposed an iterative decoding version of SPDA. In [26], Shim have proposed a forward error correction codes in communication channels to facilitate self-synchronization of digital communication systems. In [27] a compact Genetic Algorithms with larger tournament (cGA) is used to propose two dual domain Soft Decision decoders.

The Chase-2 decoding algorithm [28] is a decoder that uses a list of most likely error patterns. The main idea behind this decoder is to use a hard decision decoder (HD) in  $2^l$  times. The complexity of Chase-2 decoder is then of order  $2^l O(\text{HD})$ , the choice of HD impacts considerably the total complexity of the decoder. In [29], we have presented a soft decision decoding version the decoders presented in [21] by using the Chasing technique.

Unlike most competitors, SDHT is applicable to cyclic and non-cyclic linear codes, exploits the generator polynomial to further reduce the complexity of cyclic codes. Has an error correcting capability extended, gives better performance compared to some known decoders in the field and has a reduced time complexity.

### 3. The proposed decoding algorithm: Soft decision decoder by hash techniques (SDHT)

The syndrome decoding algorithm is used to decode linear codes; its main idea is to compute the syndrome of a binary word to determine whether an

alteration has occurred or not, and if yes in which positions. A codeword  $c(x) = c_0 + c_1x + \dots + c_{n-1}x^{n-1}$  is the product of the generator matrix G and the vector associated with the message to be transmitted  $\vec{m} = (m_0, m_1, \dots, m_{k-1})$ . When the codeword  $\vec{c}$  is sent through a noisy channel, the receiver receives a sequence  $r(x) = r_0 + r_1x + \dots + r_{n-1}x^{n-1}$  that corresponds to the sum of the  $c(x)$  and an error  $e(x) = e_0 + e_1x + \dots + e_{n-1}x^{n-1}$ .

$$\vec{r} = \vec{c} + \vec{e} \tag{1}$$

To verify if there are errors during a transmission operation or not we compute the syndrome of the received sequence as follows.

$$S(\vec{r}) = S(\vec{c}) + S(\vec{e}) = r.H^T \tag{2}$$

With  $H^T$  is the transpose of a parity check matrix of the code.

According to the value of  $S(\vec{r})$ , two cases are treated:

- ✓ If the syndrome is Null, this means that most likely, the received sequence corresponds to the codeword sent by the transmitter.
- ✓ Otherwise, the received sequence  $\vec{r}$ , does not correspond to the codeword  $\vec{c}$  sent by the transmitter; in this case, it is necessary to look for a minimal weight error pattern having syndrome equal to that of the received sequence.

The search operation of the error pattern that corresponds to the received sequence requires the prior storage of all error patterns of weight less than or equal to the threshold  $s$  ( $s \geq t$ ), in a table whose dimension grow in size relatively with  $s$ ,  $k$  and  $n$ . To alleviate the search of a codeword in the hash table (TH) we propose the use of the hash techniques by storing each error pattern in the line of number equal to the decimal value of its syndrome; and when the receiver receives a sequence it accesses directly to the error patterns corresponding to the decimal value of its syndrome. Precisely, the proposed hash function has the following algorithm.

```

Function hash (SY, POW2)
     $m \leftarrow \sum_{j=0}^{n-k-1} POW2[j].SY[n-k-1-j]$ 
    Return (m)
End Function
    
```

The proposed decoder can correct up to  $s$  errors, which mean we have to store more than one error pattern in each line of TH. The common points

between the errors pattern stored in the line number m are:

- ✓ Their error number is less than or equal to s
- ✓ They even have a syndrome of hash value equal to  $m = \text{hash}(\text{Syndrome}(e))$

Remark: In general, there are some syndromes which don't correspond to any corrigible error pattern; in these cases we propose to add the zero vector in the corresponding line in the table TH.

Example: Preparation of TH for BCH(7,4,3) code.

✓ **Case of  $s=t$**

For example  $s=t=1$ . In this case, there are  $1 + \sum_{i=1}^s C_n^i = 8$  error patterns.

Table 1 explains the process used to fill the hash table TH for BCH(7,4,3) code in the case where  $s=t=1$ . Table 2 presents its used version in the SDHT algorithm. Then the table TH is as Table 2.

Table 1. The used process to fill the hash table TH for BCH(7,4,3) code

Line number	Error pattern e	S=Syndrome(e) =e.H <sup>T</sup>	m=hash(S)
0	0000000	000	0
1	0000001	001	1
2	0000010	010	2
3	0100000	011	3
4	0000100	100	4
5	0001000	101	5
6	1000000	110	6
7	0010000	111	7

Table 2. The used form of TH in the SDHT algorithm for BCH (7, 4, 3) code, where s=1

Line number	Error pattern e
0	0000000
1	0000001
2	0000010
3	0100000
4	0000100
5	0001000
6	1000000
7	0010000

Table 3. The used form of TH in the SDHT algorithm for BCH (7, 4, 3) code, where s=2

Line number	Error pattern e			
0	0000000			
1	0010000	0000110	0100001	1001000
2	0100000	0001100	1000010	0010001
3	0000001	0110000	0001010	1000100
4	1000000	0100010	0011000	0000101
5	0001000	0100100	0000011	1010000
6	1100000	0000010	0001001	0010100
7	0000100	0010010	1000001	0101000

✓ **Case of  $s \geq t$ :**

For example  $s=t+1=2$ . In this case, there are  $1 + \sum_{i=1}^s C_n^i = 29$  error patterns. In this case Table 3 simulates the state of the TH table.

The SDHT algorithm works as follows:

- 
- 1 Inputs
    - **r**: non binary word to decode of length **n**.
    - **s**: error correction threshold.
    - The parity check matrix **H** or the generator polynomial **h(x)** of the dual code.
    - **TH**: the hash table of  $2^{n-k}$  rows.
    - **Lenghts**: a table of  $2^{n-k}$  rows and **1** column, wich contains in each position i the number of error patterns of weight less than or equal to s for whose the decimal form of the syndrome is equal to i.
    - **POW2**: the vector of **n-k** columns that contains in each position i the value of the  $2^i$
  - 2 Output: The decided codeword c.
  - 3 Begin
  - 4      $h \leftarrow$  the binary version of r
  - 5      $c \leftarrow h$
  - 6      $De \leftarrow$  infinity // a big value
  - 7     Compute the syndrome SY of h.
  - 8      $m \leftarrow \text{hash}(SY, \text{POW2})$
  - 9     For i:=1 to **Lenghts**[m] do
  - 12              $R \leftarrow h \oplus \text{TH}[m][i]$
  - 13              $d \leftarrow \text{Ed}(R, r)$
-

```

14         if d<De then
15             c←R
16             De←-d
18         End if
19     End for
20 End
    
```

With the use of the hash function, the search of the line that contains all possible error vectors of weight less than or equal to  $s$  is directly determined and the closest codeword to the received sequence can be easily found. This idea reduces considerably the search time and therefore the decoding process.

**Remarks:**

✓ The Euclidian distance is the metric noted  $Ed(h, r)$  between the hard vector  $h=\{v_1, v_2, \dots, v_n\}$ ,  $v_i \in [0,1]$  and the real vector  $r=\{r_1, r_2, \dots, r_n\}$  calculated by the following formula:  
 $Ed(h, r) = \sum_{i=1}^n (-1)^{h_i} * r_i$

✓ The codewords are modulated by a BPSK (Binary Phase Shift Keying) modulator and transmitted over an AWGN (Additive Weight Gaussian Noise) channel. When the non-binary vector  $r$  received, the demodulator prepares a hard version  $h$  for  $r$  by the follows formula.

$$h_i = \begin{cases} 1, & \text{if } r_i \geq 0 \\ 0, & \text{if } r_i < 0 \end{cases} \quad (i=1, \dots, n) \quad (3)$$

**4. Simulation results and comparison with other decoders**

In this section, we give the results of the proposed decoder SDHT for some linear codes with a comparison with other decoding algorithms over the AWGN channel. The error correcting performances will be represented in terms of Bit Error Rate (BER) in each Signal to Noise Ratio (SNR= $E_b/N_0$ ). Table 4 gives the simulations parameters.

If the data are transmitted without coding in the sending step and without correction in the receiving step over AWGN channel then the BER reaches the value  $10^{-5}$  at the SNR=9.6 dB.

Table 4. Default simulation parameters.

Simulation parameters	value
Minimum number of residual bit in errors	200
Minimum number of transmitted blocks	1000

Fig. 1 (a) and (b) represent respectively the performances of SDHT for some RQ codes of length up to 31 and for some EQR codes of length up to 48 for different thresholds. It shows that the gain of coding is about 4 dB for QR (31, 16, 7), EQR (24, 12, 8) and EQR (32, 16, 8) codes with the threshold  $s=6$ .

Fig. 2 (a) and (b) represent the performances of SDHT for some BCH codes of length up to 63 for different thresholds. It shows that the gain of coding is more than 4 dB for BCH (31, 16, 7) with  $s=7$ , about 3 dB for BCH (63, 57, 3) with  $s=4$  and about 4 dB BCH (63, 51, 5) with  $s=5$ .

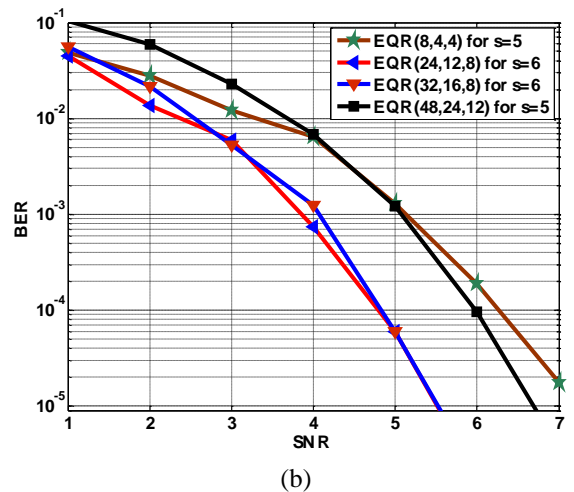
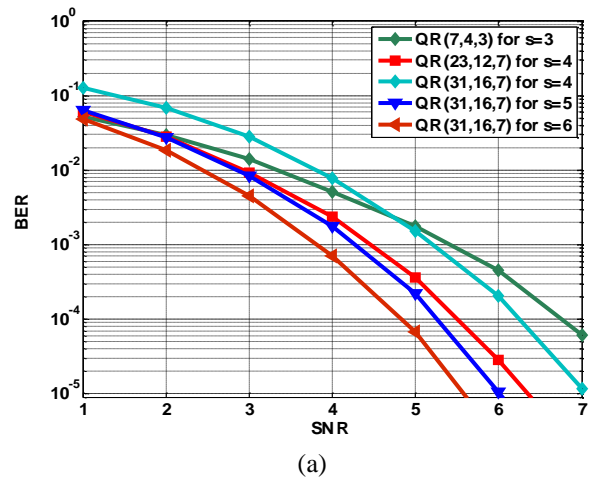


Figure.1 Performances of SDHT for: (a) QR codes of length up to 31 and (b) EQR codes of length up to 48

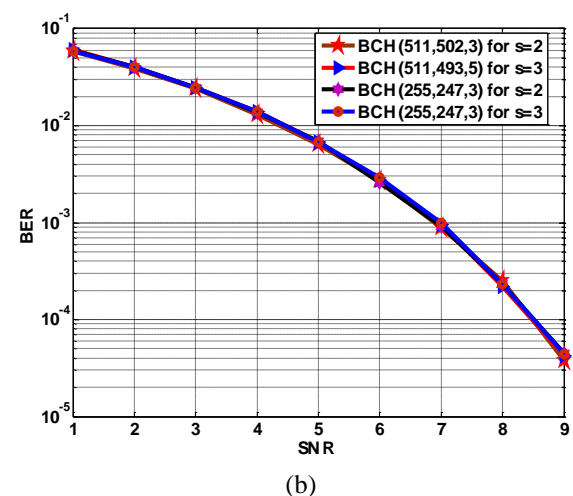
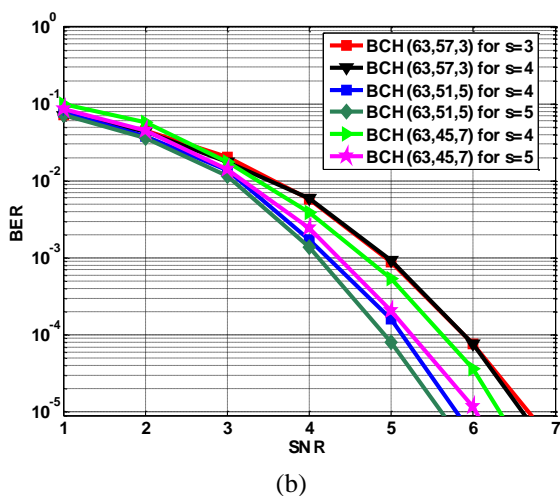
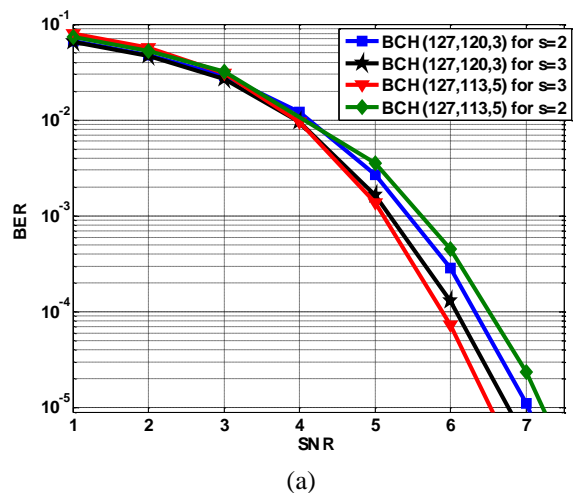
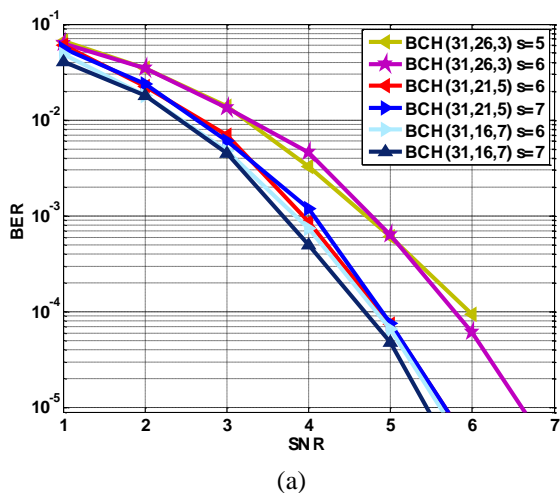


Figure.2 Performances of SDHT for some BCH codes of length: (a) 31 and (b) 63

Figure.3 Performances of SDHT: (a) for some BCH codes of length 127 and (b) for some BCH codes of length 255 and 511

Fig. 3 (a) and (b) represent the performances of SDHT for some BCH codes of length 127, 255 and 511 for different thresholds. It shows that the gain is about 3 dB for BCH (127, 113, 5) with  $s=3$ .

From these figures, we deduce that we find very good results with the increase of  $s$ . For example for the BCH(63,51,5) with  $s=5$  the gain is about 0.2 dB comparing to the same code with  $s=4$  and for the BCH(63,45,7) with  $s=4$  the gain is more than 0.3 dB comparing to the same code with  $s=4$ .

Fig. 4 (a) presents a comparison of the performances of the proposed decoder SDHT, SD1[22], OSD-1[30], Maini [32], DDGA [33] and S2W2Dec [34] decoders for BCH(63, 51, 5) code, this figure shows that our decoder passes absolutely the competitor SD1 and it has the same performances comparing to other decoders. In Fig. 4(b) we present, a comparison of the performances of the proposed decoder SDHT and a soft decoder called SPDA[24] for the EQR (24,12,8), this figure shows that our decoder passes relatively the SPDA for this code.

In Fig. 5 (a) we present a comparison of the performances of SDHT, Chase-HSDec [29], Chase-PD and SPDA for the EQR(32,16, 8) code. It shows that the proposed SDHT passes remarkably the concurrent.

Fig. 5 (b) presents a comparison of the performances of SDHT and cGAD [27] for the BCH(63, 45, 7) code. This figure shows that our decoder passes absolutely the cGAD decoder.

In Fig. 6, we present a comparison of the performances of the proposed SDHT, Chase-HSDec, Maini, OSD-1 and DDGA for BCH(31, 26, 3) and BCH(31, 21, 5) code. This figure shows that the proposed SDHT passes remarkably Chase-HSDec and it has the same performances comparing to Maini, OSD-1 and DDGA decoders.

In Fig. 7, we present a comparison of the performances of the proposed SDHT, Chase-HSDec, Maini, OSD-1 and Aut-DAG [35] for BCH(31, 16, 7). This figure shows that the proposed SDHT

passes remarkably Chase-HSDec and it has the same performances comparing to Maini, OSD-1 and Aut-DAG decoders.

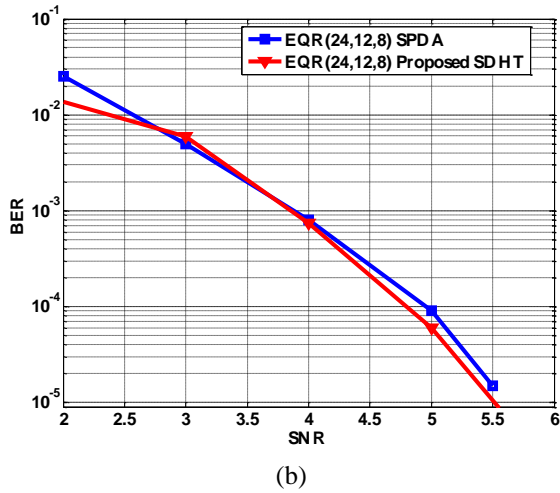
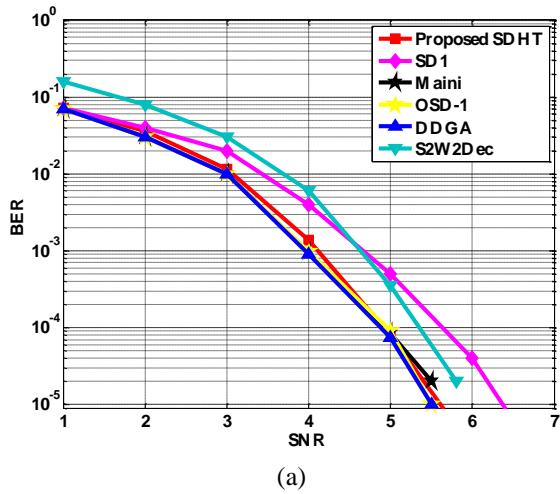


Figure.4 Comparison of the performances of the proposed SDHT with (a) SD1, Maini, OSD-1, DDGA and S2W2Dec for BCH(63,51,5) code; (b) SPDA for EQR(24,12,8) code.

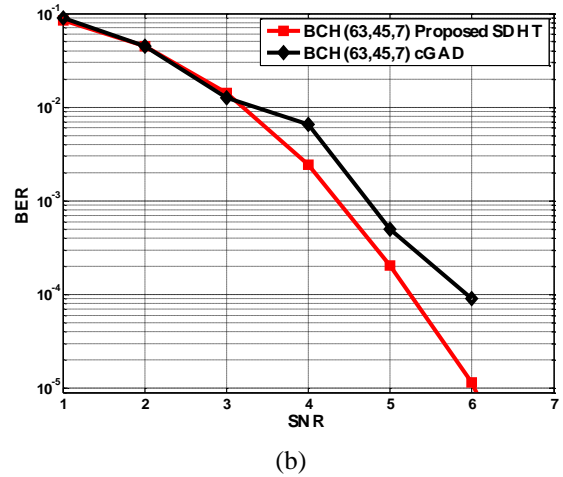
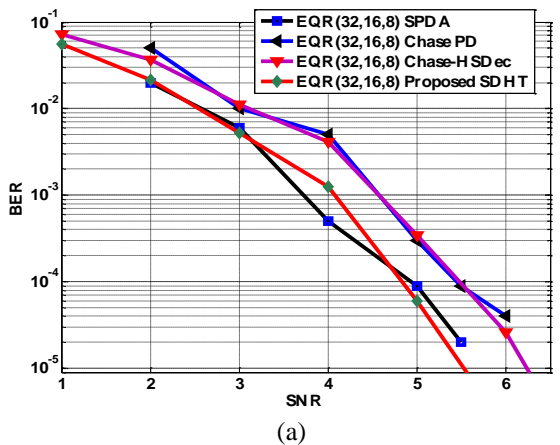


Figure.5 Comparison of the performances of the proposed SDHT with, (a) Soft-PD, Chase-PD, Chase-HSDec for EQR (32,16,8) code; (b) cGAD for BCH(63,45,7) code.

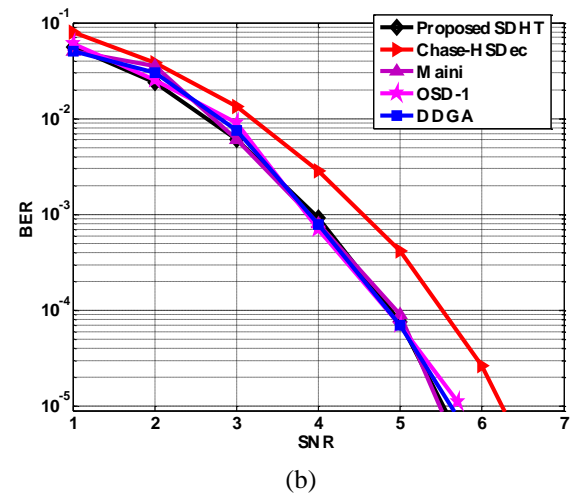
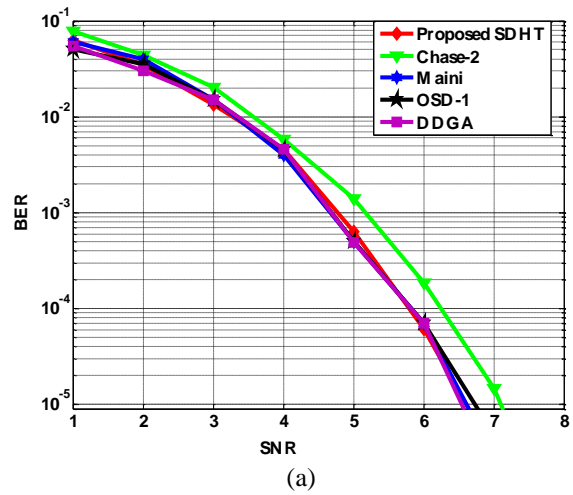


Figure.6 Comparison of the performances of the proposed SDHT, Chase-HSDec, Maini, OSD-1 and DDGA (a) for BCH(31, 26, 3); (b) for BCH(31,21,5) code

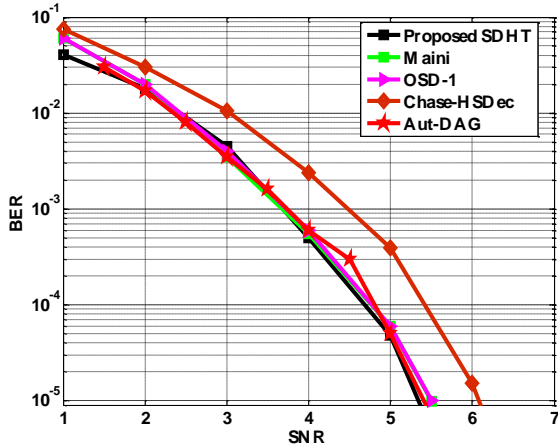


Figure.7 Comparison of the performances of the proposed SDHT, Chase-HSDec, Maini, OSD-1 and Aut-DAG for BCH(31, 16, 7) code.

### 5. Study of complexities

For linear codes, the complexity of coding is  $O(n.k)$ ; for cyclic codes, the use of the generator polynomial in encoding yield to alleviate the complexity to  $O(\log(n).log(n-k))$  [14]. We note that in this paper we use the systematic encoding for which the syndrome of a received word  $h$  is equal to the sum of its parity digits and the parity check digits recomputed by re-encoding the information part of  $h$  [36]. So, the complexity of syndrome computation is  $O(\log(n).log(n-k)+n-k)$  for cyclic codes and  $O(n.k+n-k)$  for non-cyclic codes.

The main steps of the SDHT algorithm are:

- ✓ Syndrome computation whose cost is  $O(\log(n).log(n-k) + n-k)$  for the cyclic codes and  $O(n.k + n-k)$  for non-cyclic codes this value is limited by  $O(n^2)$ .
- ✓ Hash function computation whose cost is of order  $O(n-k)$ .
- ✓ A repetitive structure "for" which iterates  $C$  ( $C$  is the maximum length of TH rows) times, each time realizes:
  - The addition of two vectors of size  $n$ ; so an operation of cost  $n$ .
  - The Euclidian distance computation between two vectors of size  $n$ ; so a cost operation  $n$  also. So the cost of the repetitive structure "for" is  $O(2nC)$

Then the complexity of SDHT for cyclic codes is  $O(\log(n).log(n-k) + 2.n-2.k+2nC)$  and  $O(n^2+2nC)$  for non-cyclic codes.

Table 5 presents the complexities of soft decoders Chase-HSDec[29], OSD-m (for an integer  $m$ ) [30], Maini[32], DDGA[33], Aut-DAG[35],

Table 5. Complexity of SDHT and other competitors.

Algorithm	Complexity
Chase-HSDec(cyclic codes)	$O(2^t(\log(n).log(n-k)+3n-2k))$
Chase-HSDec (linear not cyclic codes)	$O(2^t n^2)$
SDHT (cyclic codes)	$O(\log(n).log(n-k)+2n-2k+2nC)$
SDHT (linear not cyclic codes)	$O(n^2+2nC)$
OSD-m (cyclic or linear)	$O(n^{m+1})$
Maini (cyclic or linear)	$O(N_i.N_g(k.n + \log(N_i)))$
DDGA(cyclic or linear)	$O(N_i.N_g(k.(n - k) + \log(N_i)))$
Aut-DAG (cyclic)	$O(N_i.N_g(\log(n).log(n - k)))$
Aut-DAG (linear and not cyclic)	$O(N_i.N_g.k.n)$
S2W2Dec(PHR-SPDA)( cyclic)	$O(M.n^2 + P.log(n).log(n-k))$
S2W2Dec(PHR-SPDA)( linear and not cyclic)	$O(M.n^2 + P.n.(n-k))$

S2W2Dec[34] and SDHT. From this table, we can deduce that the SDHT has a reduced complexity comparing to their competitors and applicable on any linear block codes.  $N_i$  is the population size,  $N_g$  is the number of generations,  $m$  is the order of OSD decoder,  $t$  is the error correcting capability,  $P$  is the number of permutations used in the SPDA decoder and  $M$  is the number of dual codewords used in PHR (Partial Hartmann-Rudolf) decoder.

The comparison of complexities of SDHT and their competitors given in Table 5 proves clearly that it is a fastest decoder.

In order to show experimentally the high reduction of complexity in SDHT algorithm, we have studied it for the BCH(63,45, 7) code. For this case, the hash table TH contains  $2^{18}=262144$  lines (syndromes), the number of error patterns of weights less than or equal to 4 is 637393. The experimental study of the decoding steps of this code showed that the number of error patterns per line in TH varies between 0 and 16, which gives an average value equal to 2.4314 per line. Therefore, for decoding step instead of testing the all 637393 errors patterns, it is sufficient to use only very few number of error patterns specified by the hash function and stored in the corresponding line of TH. For this code, the time search of the most likely error pattern is reduced at about 26214153% comparing to exhaustive search off all possible errors of weights less than or equal to 4.



## 6. Conclusion and perspectives

In this article, we have presented a new efficient soft decision decoder SDHT based on the syndrome computing and exploits the hash techniques to accelerate the correction of a received data in telecommunication and storage systems. The simulation results of SDHT applied on BCH, RQ and EQR codes of different lengths show that it guarantees better performances in terms of BER in comparison with some previous known decoders; in addition, the theoretical and experimental study of complexity clearly shows its high speed compared to competitors. We also note that our decoder is applicable on any linear block code for which the error patterns can be stored in memory. The obtained results will encourage us to generalize this decoder for other families of codes like the polar codes, LDPC, ...

## References

- [1] G.C. Clark and J.B. Cain, "Error-Correction Coding for Digital Communications", first edition, Springer, New York, 1981.
- [2] Z. Yani, B. Xiaomin, Y. Zhihua, and W. Xusheng, "Decoding of the Five-Error-Correcting Binary Quadratic Residue Codes", *American Journal of Mathematical and Computer Modeling*, Vol.2, No.1, pp. 6-12, 2017.
- [3] T. Lin, H. Lee., H. Chang, and T. Truong, "A cyclic weight algorithm of decoding the (47, 24, 11) quadratic residue code", *Information Sciences*, Vol. 197, pp. 215–222, 2012.
- [4] T. Lin, H. Lee, H. Chang, S. Chu, and T. Truong, "High speed decoding of the binary (47, 24, 11) quadratic residue code", *Information Sciences*, Vol. 180, pp. 4060–4068, 2010.
- [5] C.F. Huang, W.R. Cheng, and C. Yu, "A Novel Approach to the Quadratic Residue Code", In: *Proc. of International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pp. 187-194, 2017.
- [6] T. Lin, T. Truong, H. Lee, and H. Chang, "Algebraic decoding of the (41, 21, 9) Quadratic Residue code", *Information Sciences*, Vol. 179, pp. 3451–3459, 2009.
- [7] C. Ding, H. Liu, and D.T. Tonchev, "All binary linear codes that are invariant under PSL<sub>2</sub>(n)", *IEEE Transactions on Information Theory*, Vol. 64, No. 8, pp. 5769-5775, 2017.
- [8] Y. Chen, C. Huang, and J. Chang, "Decoding of binary quadratic residue codes with hash table", *IET Communications*, Vol. 10, No. 1, pp. 122–130, 2016.
- [9] M. Jing, Y. Chang, J. Chen, Z. Chen, and J. Chang, "A new decoder for binary quadratic residue code with irreducible generator polynomial", In: *Proc. of IEEE 2008 Asia Pacific Conference on Circuits and Systems*, 2008.
- [10] M. Elghayaty, A. Hadjoudja, O. Mouhib, A. El Habti, and M. Chakir, "Performance Study of BCH Error Correcting Codes Using the Bit Error Rate Term BER", *International Journal of Engineering Research and Application*, Vol. 7, No. 2, pp.52-54, 2017.
- [11] A. El idrissi, R. El gouri, A. Lichioui, and H. Laamari, "Performance study and synthesis of new Error Correcting Codes RS, BCH and LDPC Using the Bit Error Rate (BER) and Field-Programmable Gate Array (FPGA)", *International Journal of Computer Science and Network Security*, Vol. 16, No. 5, 2016.
- [12] E. Nachmani, Y. Béery, and D. Burshtein, "Learning to Decode Linear Codes Using Deep Learning", In *Proc. of IEEE 2016 Fifty-fourth Annual Allerton Conference*, 2016.
- [13] M. Esmaeili, A. Alampour, and T. Gulliver, "Decoding Binary Linear Block Codes Using Local Search", In *Proc. of IEEE Transactions on Communications*, Vol. 61, No. 6, 2013.
- [14] A. Azouaoui, I. Chana, and M. Belkasmi, "Efficient Information Set Decoding Based on Genetic Algorithms", *International Journal of Communications Network and System Sciences*, Vol. 5, No. 7, 2012.
- [15] R. G. Gallager, "Low-Density Parity-Check Codes", *IRE Transactions on Information Theory*, Vol. 8, No. 1, pp. 21–28, 1962.
- [16] R.H. Morelos-Zaragoza, *The Art of Error Correcting Coding*, Second Edition, John Wiley and Sons, 2006.
- [17] S. Nouh, A. El khatabi, and M. Belkasmi, "Majority voting procedure allowing soft decision decoding of linear block codes on binary channels", *International Journal of Communications, Network and System Sciences*, Vol. 5, No. 9, 2012.
- [18] E. R. Berlekamp, *Algebraic Coding Theory*, rev. ed., Aegean Park Press, Laguna Hills, 1984.
- [19] J. L. Massey, "Shift-register synthesis and BCH decoding", *IEEE 1969 Transaction on Information Theory*, Vol. IT-15, No. 1, pp. 122–127, 1969.
- [20] Y.H. Chen, T.K. Truong, Y. Chang, C.D. Lee, and S.H. Chen, "Algebraic decoding of quadratic residue codes using Berlekamp-

- Massey algorithm”, *Journal of Information Science and Engineering*, Vol. 23, No. 1, pp. 127–145, 2007.
- [21] M.S. El Kasmi Alaoui, S. Nouh, and A. Marzak, “Two new fast and efficient hard decision decoders based on Hash techniques for real time communication systems”, In: *Proc. of the Second International conference on Real Time Intelligent Systems*, 2017.
- [22] B. Jung, T. Kim, and H. Lee, “Low-Complexity Non-Iterative Soft-Decision BCH Decoder Architecture for WBAN Applications”, *Journal of Semiconductor Technology and Science*, Vol. 16, No. 4, 2016.
- [23] Y. Lin, H. Chang, and C. Lee, “Improved High Code-Rate Soft BCH Decoder Architectures with One Extra Error Compensation”, *IEEE Transactions on Very Large Scale Integration Systems*, Vol. 21, No. 11, 2013.
- [24] M. Askali, S. Nouh, and M. Belkasmi, “A Soft decision version of the Permutation decoding algorithm”, In: *Proc. of NTCCCS 12 workshop*, 2012.
- [25] M. Askali, F. Ayoub, I. Chana, and M. Belkasmi, “Iterative Soft Permutation Decoding of Product Codes”, *Computer and Information Science*, Vol. 9, No. 1, 2016.
- [26] Y.G. Shim, “Forward Error Correction Codes in Communication Channels”, *International Journal of Control and Automation*, Vol. 10, No. 4, pp. 131–144, 2017.
- [27] A. Berkani, M. Azouaoui, M. Belkasmi, and B. Aylaj, “Improved Decoding of linear Block Codes using compact Genetic Algorithms with larger tournament size”, *International Journal of Computer Science Issues*, Vol. 14, No. 1, 2017.
- [28] D. Chase, “A class of algorithms for decoding block codes with channel measurement information”, *IEEE Transaction on Information Theory*, Vol. 18, pp. 170–181, 1972.
- [29] M.S. El Kasmi Alaoui, S. Nouh, and A. Marzak, “A low complexity soft decision decoder for linear block codes”, In: *Proc. of the First International Conference on Intelligent Computing in Data Sciences*, 2017.
- [30] M.P.C. Fossorier and S. lin, “Soft decision decoding of linear block codes based on ordered statistics”, *IEEE Transaction on Information Theory*, Vol. 184, pp. 1379–1396, 1995.
- [31] C. R. Hartmann and L. D. Rudolph, “An optimal symbol by symbol decoding rule for linear codes”, *IEEE Transaction on Information Theory*, Vol. 22, pp. 514–517, 1976.
- [32] H. Maini, K. Mehrotra, C. Mohan, and S. Ranka, “Soft decision decoding of linear block codes using genetic algorithms”, In: *Proc. of IEEE International Symposium on Information Theory*, 1994.
- [33] A. Azouaoui, M. Belkasmi, and A. Farchan, “Efficient Dual Domain Decoding of Linear Block Codes Using Genetic Algorithms”, *Hindawi Publishing Corporation, Journal of Electrical and Computer Engineering*, Vol. 2012, Article ID 503834, 2012.
- [34] S. Nouh and B. Aylaj, “Efficient Serial Concatenation of Symbol By Symbol and Word by Word decoders”, *International Journal of Innovative Computing, Information and Control*, Vol. 14, No. 5, 2018.
- [35] F. El Bouanani, H. Berbia1, M. Belkasmi, and H. Benazza, “Comparaison des décodeurs de Chase, l’OSD et ceux basés sur les algorithmes génétiques”, In: *Proc. of Colloque GRETSI*, 2007.
- [36] S. Lin and D.J. Costello, *Error Control Coding Fundamentals and Applications*, Prentice-Hall, Inc, 1983.