

# Single parity check node adapted to polar codes with dynamic frozen bit equivalent to binary linear block codes

Driss Khebbou<sup>1</sup>, Idriss Chana<sup>2</sup>, Hussain Ben-Azza<sup>1</sup>

<sup>1</sup>Ecole Nationale Supérieure d'Arts et Métiers, Moulay Ismail University of Meknès, Meknès, Morocco

<sup>2</sup>Ecole Supérieure de Technologie, Moulay Ismail University of Meknès, Meknès, Morocco

## Article Info

### Article history:

Received Jul 7, 2022

Revised Sep 26, 2022

Accepted Oct 17, 2022

### Keywords:

Coding theory

Dynamic frozen bit successive-cancellation

Error correcting code

Linear block code

Polar code

## ABSTRACT

In the context of decoding binary linear block codes by polar code decoding techniques, we propose in this paper a new optimization of the serial nature of decoding the polar codes equivalent to binary linear block codes. In addition to the special nodes proposed by the simplified successive-cancellation list technique, we propose a new special node allowing to estimate in parallel the bits of its sub-code. The simulation is done in an additive white gaussian noise channel (AWGN) channel for several linear block codes, namely bose–chaudhuri–hocquenghem codes (BCH) codes, quadratic-residue (QR) codes, and linear block codes recently designed in the literature. The performance of the proposed technique offers the same performance in terms of frame error rate (FER) as the ordered statistics decoding (OSD) algorithm, which achieves that of maximum likelihood decoder (MLD), but with high memory requirements and computational complexity.

*This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.*



## Corresponding Author:

Driss Khebbou

Ecole Nationale Supérieure d'Arts et Métiers, Moulay Ismail University of Meknès

ENSAM, P.O. Box 15290 Al-Mansour, Meknès, Morocco

Email: khebbou.driss@gmail.com

## 1. INTRODUCTION

Instead of increasing the emission power to remedy the problem of alternating information sent in a communication system, redundancy must be added to the information before it is transmitted. This is the principle of error-correcting codes, which are frequently applied to detect and correct data transmission errors in communication systems. It consists of encoding the information by adding redundancy by the encoder in order to obtain a codeword, and the decoder uses this redundancy to rectify the transmission errors due to the channel noise. Generally, two types of decoding algorithms exist [1]. The first is hard decision decoding, which manipulates the received codeword in binary form. The second is soft decoding, which manipulates directly the received codeword symbols, and it offers more significant performance than a hard decision decoder.

Soft decision maximum likelihood decoding (MLD) [2] is one of the decoders that offers the most significant performance. However, it is still considered an NP-Hard problem [3]. Soft decision decoding methods that are practical and computationally efficient have attracted a lot of attention [4], [5], but it's still an open problem for linear block error-correcting codes. One of the most well-known soft decision decoders is the ordered statistical decoding (OSD) introduced in [6] or its variant proposed in [7]. It is also a starting point for many other algorithms [8]–[12]. Experiments in the literature have revealed that OSD performs similarly to MLD for several binary linear block codes. This exceptional performance can only be obtained when OSD with higher orders is used, which means a lot of memory requirement and computational complexity.

In 2008, Arikan introduced a class of error-correcting codes known as polar codes [13], which can provably attain channel capacity with explicit construction and attractive complexity of encoding/decoding,

which played a pivotal role in the selection of these codes for 5G NR deployment. To take advantage of their interesting performances and low complexity soft decoding in favor of binary linear block codes, recent work [14] proposed a method for converting a binary linear block code to a polar code with dynamic frozen bits [15], [16], which can then be decoded using successive-cancellation list (SCL) methods [17]. However, the serial nature of the SCL decoder negatively impacts latency and throughput. To mitigate the impact of the aforementioned nature, [18] proposes an adaptation of the simplified successive-cancellation list (SSCL) decoder [19], which offers parallel decoding. But the proposed adaptation does not consider the case where the frozen bits are dynamics.

In this work, we suggest a new optimization of the serial nature for the decoding binary linear block code by the polar code soft decoding techniques. In addition to the special nodes proposed by the SSCL technique, we propose in this paper an extension of the single parity check (SPC) nodes by considering those with the parity check bit as dynamic frozen bit. The rest of the paper is structured out as follows: The section 2 introduces some basic concepts in binary linear block codes and polar codes. The section 3 presents the new special node that offers a parallel estimation of a set of bits. Before concluding, the section 4 illustrates the results of applying an adaptation of SSCL with the proposed SPC-node having the dynamic parity check bit to some popular binary linear block codes.

## 2. BACKGROUND

### 2.1. Binary linear block codes

A linear block code  $C$  is an error-correcting code structured as a vector subspace  $\mathbb{F}_q^k$  over a finite field  $\mathbb{F}_q^n$ , where  $q$  is prime. If  $q = 2$ , the code is a binary linear block code. It is described by three parameters,  $[n, k, d_{min}]$  where  $n$  describes the dimension of the space that contains it. It is called the length of the code  $C$ .  $k$  is the dimension of the code, corresponding to the size of the information block to be encoded and  $d_{min}$  describes the minimum distance, in the Hamming sense, between each codeword of the code. In practice, a linear code is thus represented by a matrix  $G$ , called a generator matrix, of size  $k * n$  with  $k < n$ , which takes words from  $\mathbb{F}_q^k$  and returns codewords from  $\mathbb{F}_q^n$ . The coding is done by partitioning the original message into blocks of  $k$  bits, then multiplying each block by the matrix  $G$ , resulting in codewords of  $n$  bits as (1).

$$C = \{c = x * G \mid x \in \mathbb{F}_q^k\} \tag{1}$$

### 2.2. Polar codes

A polar code,  $C_p(n, k, \mathcal{F})$ , is a class of linear block code with length  $n = 2^m$  ( $m$  being a natural number), containing  $k$  information bits. The generator matrix  $G_p$  of the code is attained by the  $m^{th}$  Kronecker power of the Arikan Kernel  $\mathcal{K} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$ , as follow:

$$G_p = \mathcal{K}^{\otimes m} = \begin{bmatrix} \mathcal{K}^{\otimes m-1} & 0 \\ \mathcal{K}^{\otimes m-1} & \mathcal{K}^{\otimes m-1} \end{bmatrix} \tag{2}$$

The coding process consists in constructing a vector  $u$  containing  $n - k$  frozen bits set to 0 and  $k$  information bits, it is constructed in such a way that the information bits are situated on the positions  $\{i \notin \mathcal{F}\}$  of the most reliable channels, and the frozen bits in the bad channels indices  $\{i \in \mathcal{F}\}$ . The corresponding codeword can then be computed similarly to (1) as  $c_p = u * G_p$ .

From the way the generator matrix of the polar code is built, it is obvious that such a code of size  $n$  is constructed in a recursive way from a code of size  $n - 1$ . This offers the possibility of representing this codes in several forms as shown in Figure 1, as a factor graph in Figure 1(a) or in the form of a tree in Figure 1(b), decomposed into depth levels  $d \in \{0, 1, \dots, \log_2(n) + 1\}$ . There are  $2^d$  nodes in each level, each node corresponds to  $n_v = 2^{m-d}$  bits.

#### 2.2.1. Polar code decoding techniques

With the successive-cancellation (SC) technique, polar codes have been shown to attain symmetric channel capacity with low-complexity decoding. The decoding technique implies traversing the tree in the precise order illustrated in Figure 1(b), starting at the root node and progressing through the binary tree to a leaf node. However, in the case of moderate or short block length, the successive-cancellation decoding technique for polar codes is less attractive compared to other codes like low-density parity-check (LDPC) codes [20]. Therefore, it is necessary to improve the performance of polar codes in both block length regimes. The SCL decoder algorithm plays a part at this point [17]. List decoding has been used to improve the error-correction performance of block codes for a long time. Instead of considering a hard decision, the SCL decoder continues along two paths when

estimating a leaf information bit. As a result, the decoder will have two alternative outcomes for each information bit, with a certain reliability metric  $PM$  for each path, up to a predefined limit list size  $L$ . When the number of paths exceeds  $L$ , the list is reduced to the  $L$  most reliable paths. When decoding is complete, the decoder output is chosen based on the estimated codeword with the highest reliability path metric ( $PM$ ).

In SC or SCL, the estimation of a bit is linked to the estimation of the bits that precede it. This results in a serial process, which affects the algorithm's latency and throughput. Many efforts to address this issue have been the topic of research. The simplified successive-cancellation (SSC) decoder described in [21], and the simplified successive-cancellation list (SSCL) described in [19], enables estimating many bits at a time at the junction of special nodes depicted in Figure 1(b), and it is no longer required to traverse through the subtree with the special node as root, up to the leaf nodes. The different special nodes are identified in the decoding tree as follows :

- Rate-0 node, denoted by the lozenge in Figure 1(b), has all of its leaf nodes frozen. i.e., the ratio of information bits to frozen bits is null.
- Rate-1 node labelled by pentagon in Figure 1(b) is the node whose whole of leaf nodes are information bits. i.e., the ratio of frozen bits to information bits is null.
- Single parity check node or SPC node labelled by triangle in Figure 1(b), is the root node of the subtree which consists of the leaf information nodes except for the first one, which is frozen.
- Rep node labelled by square in Figure 1(b), is the root node of the subtree with all of the leaf nodes are frozen bits except the last (rightmost), which is an information bit.

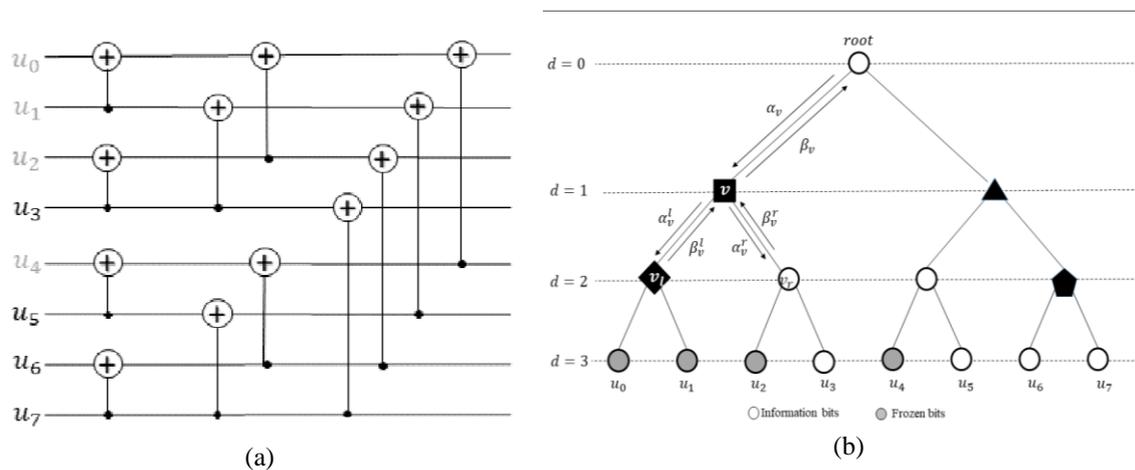


Figure 1. Representation of (8, 4) polar code (a) the factor graph representation form and (b) the tree representation form

### 3. NEW SPC-NODE WITH DYNAMIC FROZEN BIT

#### 3.1. Polar code with dynamic frozen bit

The less attractive minimum Hamming distance of polar codes is an open issue. A recent contribution [15] recommended to generate the frozen bits in a dynamic way based on the message bits, so the minimum Hamming distance can be increased. This principle suggests to inject a linear code with a significant minimum Hamming distance, having a specific upper trapezoidal generator matrix  $G_{df}$ . This matrix allows to define the linear functions defining the dynamic frozen bits according to the information bits. Thus, the polar codewords can be generated via:

$$c_p = u \cdot G_p = m_p \cdot G_{df} \cdot G_p \tag{3}$$

where  $m_p$  is the vector containing the  $k$  information bits. Trifonov and Miloslavskaya [15] provides an example of polar code with dynamic frozen bits determined using an extended bose–chaudhuri–hocquenghem (E-BCH) code, which has been proved to produce a large minimum distance.

### 3.2. New SPC-Node with dynamic frozen parity check bit

As described in [21], single parity check node or (SPC) node is the root node of the subtree which consists of the leaf information nodes except for the first one which is frozen. Since it is frozen, it must be transmitted in one of the least reliable channels. Therefore, it corresponds to the bit with the lowest absolute value  $|\alpha_{v_i}|$ .

Let's  $i_{pc}$  the position of the least reliable bit and  $n_v$  the number of bits in  $\alpha_v$ . At the encounter of a SPC-node, the estimation of  $\beta_v$  returned to the parent node is done as follows for each path:

$$\beta_v \begin{cases} h(\alpha_{v_i}) & \text{for } (i \neq i_{pc}) \\ \sum_{i=0, i \neq i_{pc}}^{n_v} \beta_{v_i} & \text{otherwise} \end{cases} \tag{4}$$

where  $h$  is the hard decision function described as (5).

$$h(x) = \begin{cases} 1, & \text{if } x < 0 \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

Note that it is important to estimate all the information bits before estimating the parity check bit. The update of the  $PMs$  for the information bits and the parity bit are respectively.

$$PM_l = PM_l + \text{sign}(\alpha_{v_i})\alpha_{v_i} - (1 - 2\beta_{v_i})\alpha_{v_i} \tag{6}$$

$$PM_l = \begin{cases} PM_l, & \text{if } \beta_{v_{i_{pc}}} = h(\alpha_{v_{i_{pc}}}) \\ PM_l + |\alpha_{v_{i_{pc}}}|, & \text{otherwise} \end{cases} \tag{7}$$

The (4), (7) are useful just in the case where the leaf frozen bit of the SPC-node is statically frozen (fixed to zero). While, for polar codes with dynamic frozen bits, it is necessary to define the estimation rules taking into account the specificity of the dynamic frozen bits. This paper aims to redefine the estimation and the  $PMs$  update in the case where the frozen bit of the SPC-node is dynamic frozen.

Let  $u_j$  be a dynamic frozen bit,  $\mathcal{F}$  the set of the frozen bits positions, and  $S_j$  a set with elements in the set of the information bits positions. By definition of a dynamic frozen bit, its value is defined according to certain information bits according to an  $S_j$ . i.e., the value of the dynamic frozen bit  $u_j$  is defined as (8).

$$\forall j \in \mathcal{F}: \exists S_j \subset \overline{\mathcal{F}} \mid u_j = \sum_{t \in S_j} u_t \tag{8}$$

If  $S_j = \emptyset$ , the bit is static frozen, i.e.  $u_j = 0$ .

The frozen position of an SPC-node is  $j = 0$ . If it is dynamic frozen, its value is no longer 0, but it is defined according to  $S_0$ . So its estimation does not depend only on the estimation of the information bits that constitute the SPC-node, but also on the function that define de dynamic frozen bit  $\sum_{t \in S_0} \beta_t$ . Thus the (4) related to the estimation of  $\beta_v$  returned to the parent node of an SPC-node, in the case where the parity bit is dynamic frozen will be as (9).

$$\beta_v \begin{cases} h(\alpha_{v_i}), & (i \neq i_{pc}) \\ \sum_{i=0, i \neq i_{pc}}^{n_v} \beta_{v_i} + \sum_{t \in S_j} \beta_t, & (i = i_{pc}) \end{cases} \tag{9}$$

And the (7) related to the  $PMs$  update must depend on the linear function that defines the dynamic frozen bit. Hence the (7) in this case can be defined as (10).

$$PM_l = \begin{cases} PM_l, & \text{if } \beta_{v_{i_{pc}}} = \sum_{t \in S_{i_{pc}}} \beta_t \\ PM_l + |\alpha_{v_{i_{pc}}}|, & \text{otherwise} \end{cases} \tag{10}$$

Figure 2 shows two types of SPC in the form of a tanner graph. For example, the estimation of generated parity bit is shown in Figure 2(a) for  $n = 4$  in the case of polar codes with static frozen bits (fixed to 0). For the case of a SPC sub-code whose parity bit is a dynamic frozen bit, which depends linearly on some information bits, the estimation of the latter can be obtained as illustrated in Figure 2(b).

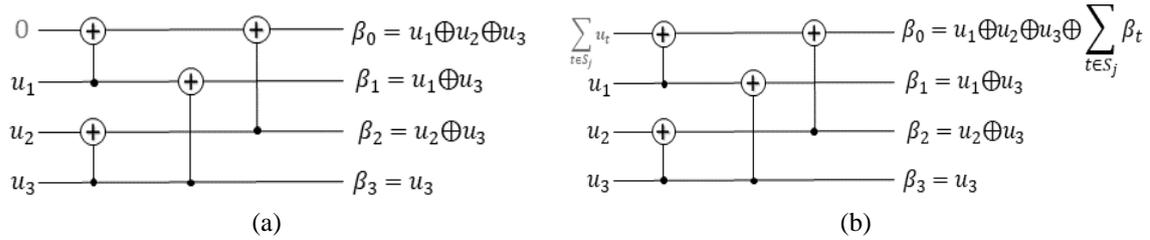


Figure 2. SPC-node adapted to polar code with dynamic frozen bits (a) SPC-node with static frozen bit and (b) SPC-node with dynamic frozen bit

**3.3. Application on polar codes equivalent to linear block codes**

Using a permutation matrix  $P$ , it was proven in [14] that the decoding space of a  $(n, k)$  binary linear block code  $C$ , defined by its generator matrix  $G$ , may be mapped into the decoding space of an appropriate polar code  $C_p$  with dynamic frozen bits defined by  $G_p$ . A codeword  $c$  of  $C$  can be stated in terms of a codeword  $c_p$  of  $C_p$ . i.e.,  $c = c_p \cdot P$ . The matrix  $G_{df}$  used in (3) and obtained by transformation of  $GP^{-1}G_p^{-1}$  into reduced row echelon form, allows to define the polar code with dynamic frozen bits equivalent to a binary linear block code  $C$ , where the index of columns with just one element not null denotes information bits, the index of columns with zero items denotes static frozen bits, and the remainder are dynamic frozen bits, obtained by linear function of some information bits.

SSCL decoding in favor of polar codes equivalents to binary linear block code has already been treated in [18]. But the authors just took into account special nodes with static frozen bits. The SPC-node proposed in this work allows to prune the decoding tree to better minimise the serial nature in the case of dynamic frozen leaf bits.

Example 1: Let's take the E-BCH code (16,7) with generator matrix  $G$  defined in [22], the generator matrix  $G_p$  of polar code of length  $n = 16$ , and the permutation matrix  $P$  defined as (11).

$$P_{i,j} = \begin{cases} 1, & (i = n - 1 - \theta^j \mid 0 \leq j \leq n - 2) \\ & \text{or } (i = j = n - 1) \\ 0, & \text{otherwise} \end{cases} \tag{11}$$

$G_{df}$ , which allows to define the polar code equivalent to the extended BCH code (16,7) defined by its generator matrix  $G$ , can be obtained by transforming  $GP^{-1}G_p^{-1}$  into reduced row echelon form.

$$G_{df} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

From  $G_{df}$ , the polar code with dynamic frozen bits equivalent to extended BCH (16,7), is defined as  $u_3, u_5, u_7, u_{11}, u_{13}, u_{14}, u_{15}$ , which are information bits,  $u_0, u_1, u_2, u_4, u_8$  which are static frozen bits and  $u_6 = u_3, u_9 = u_5, u_{10} = u_3 + u_5, u_{12} = u_{10}$  as the dynamic frozen bits. In Figure 3 which represents the SSCL decoding tree with SPC-nodes with a dynamic frozen bit for a two codes BCH and QR, Figure 3(a) represents the SSCL decoding tree of this code taking into consideration the SPC-node with dynamic frozen bits.

Example 2: Let's take  $G$  as the generator matrix of extend (32, 16) Quadratic-residue code (QR) code generated from GUAVA package [23], and same permutation matrix  $P$  defined in (11) with  $n = 32$ . By transforming  $GP^{-1}G_p^{-1}$  into a reduced row echelon form, where  $G_p$  is the 5<sup>th</sup> Kronecker power of Arikian kernel, we define the polar code equivalent to the extended QR-code as follows:  $u_0, u_1, u_2, u_4, u_8, u_{16}$  as its static frozen bits,  $u_{12}, u_{17}, u_{18}, u_{20}, u_{21}, u_{22}, u_{24}, u_{25}, u_{26}, u_{28}$  as its dynamic frozen bits and the remainder as information bits. From the representation of the equivalent polar code's tree with SPC-nodes with a dynamic frozen bit for a two codes BCH and QR in Figure 3, in additions to special nodes with static frozen bit, we notice that the fourth and the eighth node at depth 3 in Figure 3(b) are SPC-nodes with dynamic frozen bit presented in this work.

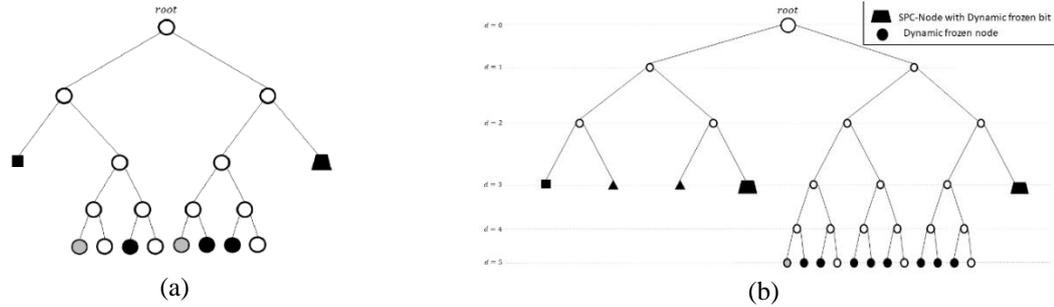


Figure 3. SSCL Decoding tree with SPC-node with dynamic frozen bit (a) (16,7) polar code equivalent to (16,7) extended BCH code and (b) (32,16) polar code equivalent to (32, 16) E-QR code

**4. RESULTS AND DISCUSSION**

In this section, we compare the frame error rate (FER) of the suggested method to the OSD algorithm for binary linear block code over additive white gaussian noise channel (AWGN) channel, in order to evaluate its performance. E-BCH codes, QR code and a binary linear block codes newly constructed in [24] are used in the simulations. Figure 4 illustrate the performances comparison between OSD and the proposed method for some popular codes. Figures 4(a) and 4(b) shows the FER of the suggested SSCL decoder including SPC-node with dynamic parity check bit, compared to the OSD decoder for two extended BCH codes of length 64 with different dimensions  $k = \{24, 51\}$ , using the same permutation matrix  $P$  described in (11), generator matrices described in [22], and the 6th Kronecker power of Arikan kernel defined in (2).

The polar code equivalent to the E-BCH code (64,24), illustrated in Figure 4(a), is defined so that the positions  $\{8,12,16,24,28,30,31,32,40,44,46,47,48,52,54,55,56,58,59,60,61,62,63,64\}$  are information bits, the positions  $\{15,20,22,23,26,27,29,36,38,42,43,45,50,51,53,57\}$  are dynamic frozen bits, and the remainder positions are static frozen bits. In addition to the optimization proposed by the SSCL technique as described in [18], the special node proposed in this work allows even more to cut the decoding tree to the met of the eighth node at depth 3 and the eighth, twelfth and fourteenth nodes at depth 4, which are SPCs with dynamic frozen bits. In terms of FER, we observe that the proposed method, with list size  $L = 16$ , achieves the performance of a second-order OSD, and it outperforms a second-order OSD decoder with list size  $L = 32$ .

For E-BCH code (64,51) illustrated in Figure 4(b), the equivalent polar code is defined in order that the positions  $\{1,2,3,5,9,17,33\}$  are static frozen bits, the positions  $\{21,34,35,37,41,49\}$  are dynamic frozen bits, and the remainder positions are information bits. The new SPC-node with dynamic frozen bit is at position 4 in depth 2, and at positions  $\{10, 11\}$  in depth 4. This allows to estimate in parallel the leaf nodes of the subtree of which SPC-node with dynamic frozen bit is root. In terms of performances, we observe that the method presented in this paper allows to reach the second-order OSD with a list size  $L = 32$ .

Figure 4(c) illustrates the performances of the polar code equivalent to the extended (32,16) QR code described in Example 2. We can notice that the performances of the proposed method join those of the OSD of order 2, that with a list size  $L = 32$ . Figure 4(d) allows to confirm that the method offers good performances, so it represents the first FER simulation of one of the codes newly built by the method proposed in [25].

For the (16, 9) code newly constructed in [25], and with a permutation matrix  $P$  defined as follows:

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

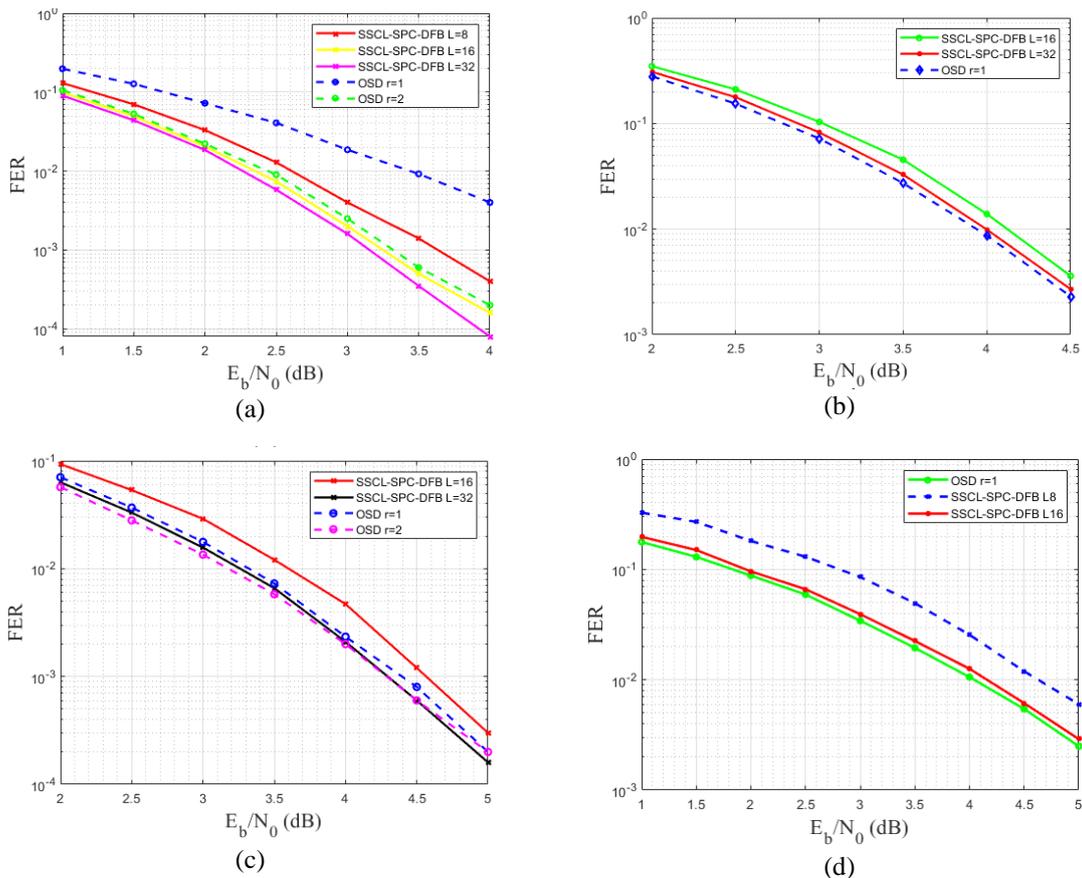
to take advantage of the performances of polar code decoders in favor of binary linear block codes, the authors in [14] proposed to decode the linear block codes with the SCL method for polar codes. In terms of performance the results obtained are good, but the serial nature of the SCL technique has an impact on latency and throughput. In

order to partially reduce the impact of the serial nature of the SCL technique in favor of linear block codes, [18] suggests to decode equivalent polar codes by the SSCL technique. The contribution allows estimating some bits related to the special nodes. But the equivalent codes to the linear block codes are polar codes with dynamic frozen bits, and the authors ignored the special nodes with dynamic frozen bits, resulting in serial estimation.

To apply parallel estimation in the case of dynamic frozen bits, we proposed in this paper the SPC node with a frozen dynamic leaf bit, as new special node offering parallel estimation instead of serial bits estimation. Since the construction of polar codes is done recursively, each non-leaf node constitutes a polar sub-code of size  $n_v$ . Therefore, for the new special node, the estimation is done in one operation instead of applying a series of operations of order  $\mathcal{O}(Ln_v \log n_v)$ . Compared to the literature [14], [18], this work offers, on the one hand, a better minimization of the serial nature and the time steps with the same performance, and less requirements in latency and computational complexity. On the other hand, it represents the first decoding simulation for newly constructed codes [25], as well as for QR codes.

In general, the OSD is the soft decoding algorithm allowing to reach the performances of an MLD for binary linear block codes, but this requires memory and exponential computing demand  $\mathcal{O}(nk^2 + 2^r nk)$ . While the technique proposed in this work is a derivative of the SC method which is characterized by an attractive computational complexity  $\mathcal{O}(n \log_2(n))$ . In addition, the principle of parallel estimation at special nodes offers a major gain in decoding latency. A table in [26], summarizes the gain in time step of the proposed method compared to the SCL technique.

Because lower complexity immediately reduces the system's cost and power consumption, coding scheme complexity is extremely important. Given that the computational complexity of the SC decoder is solely dependent on block length  $n$ , it is obvious that the complexity of the polar code with the SC decoding method is lower than that of other coding schemes. This is confirmed by [27]. The adaptation offered in this work of a derivative of the SC technique allows us to take advantage of its low complexity for the benefit of different classes of popular linear codes.



\*SSCL-SPC-DFB: Simplified Successive-Cancellation List with SPC-node having a dynamic frozen bit

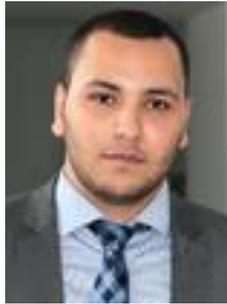
Figure 4. Performance comparison between OSD and the proposed method; (a) E-BCH (64,24), (b) E-BCH (64,51), (c) E-QR (32, 16) and (d) (16,9) linear code constructed in [24]

## 5. CONCLUSION

In this work, we propose new special node that adds to those of the SSCL technique in the case of polar codes with frozen dynamic bits. The objective is to minimize the serial nature of the basic SCL technique, offering a parallel estimation of a set of bits. It allows to decrease the decoding latency of polar codes equivalents to binary linear block codes using the SSCL algorithm. Using a set of binary linear block codes as a case study, we demonstrated that the technique allows to achieve the performances of a SCL decoder, with minimization of required time-steps. On the one hand, a significant reduction in the serial characteristic of SCL, resulting in a reduction in decoding latency. The simulation, on the other hand, demonstrates that the proposed technique can match the performances of the well-known OSD, which allows binary linear block codes to meet channel capacity while using more memory and complexity.

## REFERENCES

- [1] M. Tomlinson, C. J. Tjhai, M. A. Ambroze, M. Ahmed, and M. Jibril, "Soft and hard decision decoding performance," in *Error-Correction Coding and Decoding: Bounds, Codes, Decoders, Analysis and Applications*, Springer International Publishing, 2017, pp. 25–41, doi: 10.1007/978-3-319-51103-0\_2.
- [2] I. Sason and S. Shamai, *Performance Analysis of Linear Codes Under Maximum-likelihood Decoding: A Tutorial*. Now Publishers Inc, 2006. [Online]. Available: <https://doi.org/10.1561/0100000009>.
- [3] N. J. C. Lous, P. A. H. Bours, and H. C. A. van Tilborg, "On maximum likelihood soft-decision decoding of binary linear codes," *IEEE Trans. Inf. Theory*, vol. 39, no. 1, pp. 197–203, Jan. 1993, doi: 10.1109/18.179358.
- [4] I. Chana, H. Allouch, and M. Belkasmi, "An efficient new soft-decision decoding algorithm for binary cyclic codes," in *2011 International Conference on Multimedia Computing and Systems*, Apr. 2011, pp. 1–6, doi: 10.1109/ICMCS.2011.5945580.
- [5] H. Boualame, N. Tahiri, I. Chana, A. Azouaoui, and M. Belkasmi, "An efficient soft decision decoding algorithm using cyclic permutations and compact genetic algorithm," in *2016 International Conference on Advanced Communication Systems and Information Security (ACOSIS)*, Oct. 2016, pp. 1–6, doi: 10.1109/ACOSIS.2016.7843936.
- [6] M. P. C. Fossorier and S. Lin, "Soft-decision decoding of linear block codes based on ordered statistics," *IEEE Trans. Inf. Theory*, vol. 41, no. 5, pp. 1379–1396, Sep. 1995, doi: 10.1109/18.412683.
- [7] D. Gazelle and J. Snyders, "Reliability-based code-search algorithms for maximum-likelihood decoding of block codes," *IEEE Trans. Inf. Theory*, vol. 43, no. 1, pp. 239–249, Jan. 1997, doi: 10.1109/18.567691.
- [8] M. P. C. Fossorier, "Reliability-based soft-decision decoding with iterative information set reduction," *IEEE Trans. Inf. Theory*, vol. 48, no. 12, pp. 3101–3106, Dec. 2002, doi: 10.1109/TIT.2002.805089.
- [9] A. Valembois and M. Fossorier, "Box and match techniques applied to soft-decision decoding," *IEEE Trans. Inf. Theory*, vol. 50, no. 5, pp. 796–810, May 2004, doi: 10.1109/TIT.2004.826644.
- [10] M. Fossorier and A. Valembois, "Box and match soft decision decoding of linear block codes with iterative information set reduction," in *Proceedings of the IEEE Information Theory Workshop*, Oct. 2002, pp. 45–48, doi: 10.1109/ITW.2002.1115412.
- [11] W. Jin and M. Fossorier, "GEN03-3: enhanced box and match algorithm for reliability-based soft-decision decoding of linear block codes," in *IEEE Globecom 2006*, Nov. 2006, pp. 1–6, doi: 10.1109/GLOCOM.2006.157.
- [12] W. Jin and M. P. C. Fossorier, "Reliability-based soft-decision decoding with multiple biases," *IEEE Trans. Inf. Theory*, vol. 53, no. 1, pp. 105–120, Jan. 2007, doi: 10.1109/TIT.2006.887510.
- [13] E. Arikan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Trans. Inf. Theory*, vol. 55, no. 7, pp. 3051–3073, Jul. 2009, doi: 10.1109/TIT.2009.2021379.
- [14] C.-Y. Lin, Y.-C. Huang, S.-L. Shieh, and P.-N. Chen, "Transformation of binary linear block codes to polar codes with dynamic frozen," *IEEE Open J. Commun. Soc.*, vol. 1, pp. 333–341, 2020, doi: 10.1109/OJCOMS.2020.2979529.
- [15] P. Trifonov and V. Miloslavskaya, "Polar codes with dynamic frozen symbols and their decoding by directed search," in *2013 IEEE Information Theory Workshop (ITW)*, Sep. 2013, pp. 1–5, doi: 10.1109/ITW.2013.6691213.
- [16] P. Trifonov and V. Miloslavskaya, "Polar Subcodes," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 2, pp. 254–266, Feb. 2016, doi: 10.1109/JSAC.2015.2504269.
- [17] I. Tal and A. Vardy, "List decoding of polar codes," *ArXiv12060050 Cs Math*, May 2012, doi: 10.48550/arXiv.1206.0050.
- [18] D. Khebbou, R. Benkhouya, I. Chana, and H. Ben-azza, "Simplified successive-cancellation list polar decoding for binary linear block codes," *J. Southwest Jiaotong Univ.*, vol. 56, no. 6, Art. no. 6, 2021, doi: 10.35741/issn.0258-2724.56.6.54.
- [19] S. A. Hashemi, C. Condo, and W. J. Gross, "Simplified successive-cancellation list decoding of polar codes," in *2016 IEEE International Symposium on Information Theory (ISIT)*, Jul. 2016, pp. 815–819, doi: 10.1109/ISIT.2016.7541412.
- [20] M. Tomlinson, C. J. Tjhai, M. A. Ambroze, M. Ahmed, and M. Jibril, "LDPC Codes," in *Error-Correction Coding and Decoding: Bounds, Codes, Decoders, Analysis and Applications*, Springer International Publishing, 2017, pp. 315–354, doi: 10.1007/978-3-319-51103-0\_12.
- [21] A. Alamdar-Yazdi and F. R. Kschischang, "A simplified successive-cancellation decoder for polar codes," *IEEE Commun. Lett.*, vol. 15, no. 12, pp. 1378–1380, Dec. 2011, doi: 10.1109/LCOMM.2011.101811.111480.
- [22] R. H. Morelos-Zaragoza, "The art of error correcting coding," John Wiley & Sons, 2006, doi: 10.1002/0470847824.
- [23] D. Joyner, "GUAVA: an error-correcting codes package," *ACM SIGSAM Bull.*, vol. 39, no. 2, pp. 65–68, Jun. 2005, doi: 10.1145/1101884.1101890.
- [24] D. Khebbou, R. Benkhouya, and I. Chana, "Construction of some good binary linear codes using hadamard matrix and BCH codes," in *Proceedings of Sixth International Congress on Information and Communication Technology*, Singapore, 2022, pp. 523–532, doi: 10.1007/978-981-16-2377-6\_49.
- [25] D. Khebbou, R. Benkhouya, I. Chana, and H. Ben-azza, "Finding good binary linear block codes based on hadamard matrix and existing popular codes," *Int. J. Adv. Comput. Sci. Appl. IJACSA*, vol. 12, no. 11, Art. no. 11, Jan. 2021, doi: 10.14569/IJACSA.2021.0121150.
- [26] S. A. Hashemi, C. Condo, and W. J. Gross, "Fast and flexible successive-cancellation list decoders for polar codes," *IEEE Trans. Signal Process.*, vol. 65, no. 21, pp. 5756–5769, Nov. 2017, doi: 10.1109/TSP.2017.2740204.
- [27] S. Belhadj, A. M. Lakhdar, and R. I. Bendjillali, "Performance comparison of channel coding schemes for 5G massive machine type communications," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 22, no. 2, Art. no. 2, May 2021, doi: 10.11591/ijeecs.v22.i2.pp902-908.

**BIOGRAPHIES OF AUTHORS**

**Driss Khebbou**    he got his engineer degree in Software and Computer Systems at the Faculty of Science and Technology of Tangier, Abdelmalek Essaâdi University, Morocco in 2014. He is an engineer in information systems since 2015. Currently, he is working on his Ph.D. at the ENSAM-Ecole Nationale des Arts et Métiers, Moulay Ismail University of Meknès, Morocco, in the computer engineering department. He can be contacted at email: khebbou.driss@gmail.com.



**Prof. Idriss Chana**    received the Ph.D. degree from Mohamed V University of Rabat, Morocco in 2013. He is currently an Associate Professor at Moulay Ismail University of Meknès, Morocco. His research interests include information and communication technologies and Artificial Intelligence. A large part of his research projects is related to Error correcting codes and Turbo codes. Idriss Chana has published more than 30 papers in major journals and conferences in Information Theory and Artificial Intelligence. He can be contacted at email: idrisschana@gmail.com.



**Prof. Hussain Ben-Azza**    is a professor at ENSAM- Ecole Nationale des Arts et Métiers, Meknes, Morocco, attached to Department of Industrial and Production Engineering, Moulay Ismail University of Meknès. He obtained his Ph.D. degree in mathematics and computer science in 1995 from Claude Bernard University Lyon 1, France. His research interests include coding theory, cryptography, wireless communications, but also applications of optimization techniques to industrial engineering. He can be contacted at email hbenazza@yahoo.com.