

Enabling Fast Polar SC Decoding with IR-HARQ

Marwan Jalaleddine, Mohamad Ali Jarkas, Jiajie Li, Warren J. Gross

Abstract—To extend the applications of polar codes within next-generation wireless communication systems, it is essential to incorporate support for Incremental Redundancy (IR) Hybrid Automatic Repeat Request (HARQ) schemes. For very high-throughput applications, Successive Cancellation (SC) decoding is particularly appealing for polar codes owing to its high area efficiency. In this paper, we propose modifications to SC decoders that employ special nodes to accelerate decoding. Our modifications enable the use of polar IR-HARQ with SC decoding for high throughput applications. Compared to the unmodified SC IR-HARQ scheme, our proposed approach allows us to achieve a 72% reduction in node traversals with a polar code of length 2048. Simulation results confirm that the proposed special node modifications do not cause any degradation in FER performance.

Index Terms—Polar Code, Successive Cancellation Decoding, IR-HARQ, Special Nodes, 6G.

I. INTRODUCTION

Polar codes are a capacity-achieving error-correcting code [1] that has been incorporated into the 5G control channel [2]. More recently, there has been interest in incorporating polar codes in the data channel for the 6G standard [3]. This necessitates support for polar HARQ [3], which aims to enhance link reliability through re-transmissions.

HARQ schemes are typically categorized into two main categories: Chase Combining (CC) HARQ and Incremental Redundancy (IR) HARQ. In CC-HARQ, the same coded block is retransmitted across multiple transmissions, which simplifies implementation but yields minor improvements in throughput and reliability [4]. Conversely, IR-HARQ improves performance by transmitting new redundancy bits with each retransmission, thereby enhancing both coding gain and overall system reliability [4].

CC-HARQ schemes for polar codes were initially introduced in [5], where selective puncturing and re-transmissions of bits were employed based on greedy search algorithms. However, these methods require meticulous design of puncturing patterns, as such patterns critically influence the capacity of the virtual sub-channels assigned to message bits. Moreover, these approaches exhibit notable degradation in error-correction performance, primarily due to the Chase combining of log-likelihood ratios (LLRs) [4], [6].

To mitigate the degradation in decoding performance observed in CC-HARQ schemes, [7] introduced a polar IR-HARQ scheme employing an incremental freezing mechanism; however, since the two transmissions are decoded independently as short codes, this scheme fails to realize the

coding gains typically associated with longer code lengths [1]. This method was later enhanced by [6], who proposed concatenating multiple transmissions into a single, longer polar code to improve both coding and diversity gains. This enhancement was realized through the incremental extension of the polar encoding matrix, thereby exploiting the superior decoding performance characteristic of longer polar codes.

Despite the advancements made by [6], the newly introduced bit types restrict the algorithm's use of *special nodes*, which are nodes that are decoded without traversing the sub-tree. In [8], the authors described the set-based operations, used in [6] to define the bit types, in terms of binary vector-based operations which allows us to avoid the irregular memory accesses [6]. Additionally the authors devised a method capable of integrating special nodes with Successive Cancellation List (SCL) decoding through modifying the candidate generation method by representing all possible special nodes as linear combinations of the outputs of the basic special node types and the values of a newly introduced bit type. The same binary vector-based bit type generation method can be used with SC decoding; however, applying the same candidate generation method on SC decoding would introduce high latency overhead as SC decoding only chooses the most likely candidate for that node and does not need exploring all possible candidates. Hence, a completely different technique should be employed to allow for IR-HARQ enabled fast SC decoding.

This work introduces modifications to the state-of-the-art SC special node algorithms [9], [10] to enable their integration with polar IR-HARQ in SC decoding. Unlike, the method proposed for SCL decoding [8], the method introduced here is optimized for SC decoding with little complexity overhead compared to the existing SC decoding special node algorithms. The incorporation of special nodes in the IR-HARQ framework effectively reduces the number of node traversals required during SC decoding by up-to 72% with polar code of length 2048, thereby lowering the overall decoding latency. The structure of this work is as follows: first, a review of polar codes and the polar IR-HARQ scheme is provided. Next, an explanation of the necessary modifications for special nodes is detailed along with the performance of the modified nodes compared to the original nodes. Finally, the conclusions are presented.

II. PRELIMINARIES

A. Notation

Matrices are denoted by a bold upper-case letter ($\mathbf{M}$), while vectors are denoted by bold lower-case letters ($\mathbf{v}$). The i^{th} element of a vector $\mathbf{v}$ is denoted as v_i . The vector elements are numbered from left to right starting from an index of

Marwan Jalaleddine, Mohamad Ali Jarkas, Jiajie Li and Warren J. Gross are with the Department of Electrical and Computer Engineering at McGill university, Montreal, Quebec, Canada. Their emails are: marwan.jalaleddine@mail.mcgill.ca, mohamad.jarkas@mail.mcgill.ca, jiajie.li@mail.mcgill.ca, warren.gross@mcgill.ca.

0. Operation $\oplus$ represents XOR binary bit-wise operation. Furthermore, this work considers (n, k) linear block codes, where n is the code length and k is the code dimension. The Boolean complement of ζ is denoted as $\bar{\zeta}$. The sign function is defined as:

$$\text{sign}(\zeta) = \begin{cases} 0, & \zeta \geq 0, \\ 1, & \zeta < 0. \end{cases}$$

B. Polar Codes

Encoding and decoding of a polar code can be described using a binary tree representation. First, before polar encoding can happen, the k most reliable bit-channels are determined. Frozen bits are the $n - k$ least reliable bits that are set to a predefined value (typically 0) which is known to the decoder. The encoding process ascends the tree where each stage s in the tree is a polar code of length 2^s [1]. We also refer to each polar sub-code in a polar tree as a node with size N_v and to stage 0 nodes as leaf nodes.

The polar decoder receives y as its noisy channel output using which, the LLRs can be calculated. SC decoding proceeds to descend the tree to stage 0, and estimate the resulting bit values from the LLRs and the previously estimated bits at each level [1].

C. Special Nodes

Instead of traversing the entire tree down to stage 0 with SC decoding, special sub-trees/nodes of the polar decoding tree can be decoded efficiently if specific patterns of the frozen and information bits are encountered [10] [9]. The main special nodes that we will consider are the Rate 0, Rate 1, SPC and REP established in [10], and SPC-2, REP-2, PCR and RPC introduced in [9]. We provide a summary of the structure and decoding algorithms introduced in [10] and [9] below.

1) *Rate 0 Node*: A Rate 0 node is defined as a node in which all the bits at the leaf level are frozen. In this case, the decoding output is the all zero codeword.

2) *Rate 1 Node*: A Rate 1 node is defined as a node where all the bits at the leaf level are information bits. In this case, the output codeword bits for this node are found by obtaining the hard decision of the node LLRs. The hard decision of $\mathbf{l}$, is defined as $\mathbf{HD}$, given by $\text{sign}(\mathbf{l})$.

3) *REP Node*: A repetition (REP) node is defined as a node where all the leaf bits are frozen, except for the rightmost bit. After encoding, this results in a codeword where the information bit from the right-most position is repeated on every bit position of the node. In this case, decoding can proceed by summing all the LLRs of the node and by using a sign function to determine the value of the repeated bit.

4) *REP-2 Node*: A REP-2 node is defined as a node in which all leaf bits are frozen, except for the rightmost two bits. This configuration results in the formation of two repetition codes. The first repetition code corresponds to a repetition of a bit value on even-numbered indices of the node, while the second repetition code is associated with a repetition of a bit value on the odd-numbered indices of the node. Hence, two REP decoders can be used to determine the value of the repeated bits, one REP decoder used on the even indices and one REP decoder used on the odd indices.

5) *SPC Node*: A SPC (Single Parity Check) node is defined as a node where only the leftmost leaf bit is frozen, with the remaining bits being information bits. This structure enforces a parity check constraint on the resulting codeword. The leftmost bit in an SPC node at stage 0, p_0 , determines the even/odd parity, $\bigoplus_{j=0}^{N_v-1} HD_j$, of the resulting codeword:

$$p_0 = \bigoplus_{j=0}^{N_v-1} HD_j \quad (1)$$

In a regular SPC node, the frozen bit $p_0 = 0$, which enforces an even parity. To decode a regular SPC node, we first take the hard decision on the node LLRs. The parity of the bits is then calculated. If we have an even parity, the parity constraint is satisfied, and $\mathbf{HD}$ is returned as the codeword result. If we have an odd parity, the bit corresponding to the lowest LLR magnitude, $\text{argmin}(|l_j|)$, is flipped in the returned $\mathbf{HD}$ codeword result.

6) *SPC-2 Node*: A SPC-2 node is defined when the leftmost two leaf bits are frozen and all the remaining bits are information bits. This leads to the creation of two parity constraints, one on the even bits and one on the odd bits. Two SPC decoders can then be used to decode odd and even indices respectively.

7) *RPC Node*: An RPC (Repeated Parity Check) node is defined as a node where the leftmost three leaf bits are frozen, while the remaining bits are information bits. Using the notation $C^z = \bigoplus x \in X^z, z \in \{0, 1, 2, 3\}$ where set X^z contains all bit positions whose index j satisfies the following condition: $j \bmod 4 = z$, where mod is the modulo operator, the parity equations of this node are:

$$C^1 \oplus C^3 = C^2 \oplus C^3 = C^1 \oplus C^2 = C^0 \oplus C^3 = 0, \quad (2a)$$

$$C^0 = C^1 = C^2 = C^3 = 0, \quad (2b)$$

and

$$C^0 = C^1 = C^2 = C^3 = 1. \quad (2c)$$

The decoding algorithm then proceeds as described in Algorithm 1.

8) *PCR*: A Parity Checked Repetition (PCR) node is characterized as a node where the three rightmost leaf bits are information bits, while the remaining bits are all frozen bits. Similar to the RPC node, the codeword is divided into four groups X^z . The structure of this node guarantees that X^0 is the repetition of 0, X^1 is the repetition of the $N_v - 3$ codeword bit, x_{N_v-3} , X^2 is the repetition of the $N_v - 2$ codeword bit, x_{N_v-2} and X^3 is the repetition of the $N_v - 1$ codeword bit x_{N_v-1} . Decoding can proceed through the application of Algorithm 2.

D. Polar IR-HARQ by Matrix Extension

When employing IR-HARQ, the receiver requests additional redundancy from the transmitter to enhance decoding performance. In the context of polar codes, the IR-HARQ by matrix extension approach enlarges the polar encoding tree to accommodate an extended codeword length, equal to the sum of the original code length and the newly transmitted redundancy. Subsequently, the newly introduced bit-channels

Algorithm 1 Decoding a RPC Node [9]

Input: Received signal vector $l = \{l_j, j = 0, \dots, N_v - 1\}$
Output: Recovered codeword $\hat{x} = \{\hat{x}_j, j = 0, \dots, N_v - 1\}$

1. Initialize $\Delta_0, \Delta_1 = 0$
2. Initialize $\delta_z = \infty, C^z = 0, p_z = 0$ for $z = 0, \dots, 3$
3. Initialize $\hat{x}_j = HD(l_j)$ for $j = 0, \dots, N_v - 1$
4. **for** $z = 0, \dots, 3$ **do**
5. **for** $m = 0, \dots, N_v/4$ **do**
6. $j = m \times 4 + z$
7. $C^z = C^z \oplus HD(l_j)$
8. **if** $|l_j| < \delta_z$ **then**
9. $p_z = j$
10. $\delta_z = |l_j|$
11. **end if**
12. **end for**
13. **if** $C^z = 1$ **then**
14. $\Delta_0 = \Delta_0 + \delta_z$
15. **else**
16. $\Delta_1 = \Delta_1 + \delta_z$
17. **end if**
18. **end for**
19. **for** $z = 0, \dots, 3$ **do**
20. **if** $(\Delta_0 > \Delta_1 \wedge C^z = 0) \vee (\Delta_0 < \Delta_1 \wedge C^z = 1)$ **then**
21. $\hat{x}_{p_z} = \hat{x}_{p_z} \oplus 1$
22. **end if**
23. **end for**

Algorithm 2 Decoding a PCR Node [9]

Input: Received signal vector $l = \{l_j, j = 0, \dots, N_v - 1\}$
Output: Recovered codeword $\hat{x} = \{\hat{x}_j, j = 0, \dots, N_v - 1\}$

1. Initialize $\Delta = 0$ for $z = 0, \dots, 3$
2. **for** $z = 0, \dots, 3$ **do**
3. **for** $m = 0, \dots, N_v/4$ **do**
4. $j = m \times 4 + z$
5. $\Delta_z = \Delta_z + l_j$
6. **end for**
7. **end for**
8. $\{\hat{C}^0, \hat{C}^1, \hat{C}^2, \hat{C}^3\} = \text{SPC_DECODE}(\{\Delta_0, \Delta_1, \Delta_2, \Delta_3\})$
9. **for** $z = 0, \dots, 3$ **do**
10. **for** $m = 0, \dots, N_v/4$ **do**
11. $j = m \times 4 + z$
12. $\hat{x}_j = \hat{C}^z$
13. **end for**
14. **end for**

may be selected as information bits if their reliability surpasses that of previously transmitted information bits. In such cases, the original information bits are reclassified as Parity-Check frozen (PC_{Frozen}) bits, with their values mapped one-to-one to the corresponding new information bits. During encoding, each PC_{Frozen} bit shares the same value as its mapped information bit. During decoding, however, the new information bit is first estimated, after which the corresponding PC_{Frozen} bit is fixed to that decoded value. Thus, PC_{Frozen} bits constitute a special category of frozen bits that can assume binary values of either 0 or 1, with their values determined

post-decoding of their associated information bits.

III. IR-HARQ ENABLED SPECIAL NODES

IR-HARQ introduces PC_{Frozen} bits, which are frozen bits that may assume binary values of either 0 or 1. Conventional special node decoding algorithms typically assume frozen bits to be 0. Therefore, this section presents modifications to existing algorithms to extend special node decoding capabilities to support frozen bits with arbitrary binary values.

A. Rate 0 and Rate 1 nodes

The decoding of rate 0 and rate 1 nodes is trivial. For rate 0 nodes, the result of the decoding is simply the encoded PC_{Frozen} bits. This was proven in [8] which showed that the linear combination of the encoded PC_{Frozen} bit vector pc and the encoded information bits i is the resulting codeword β :

$$\beta = i \oplus pc. \quad (3)$$

For the rate 1 node, the result of the decoder is the hard decision of the received signal [11] since it does not have any PC_{Frozen} bits.

B. REP Node

The repeated bit in a repetition node can be determined through the following calculation:

$$i = \text{sign}\left(\sum_{j=0}^{N_v-1} l_j (1 - 2pc_j)\right), \quad (4)$$

where l_i denotes the LLR corresponding to the i -th bit. The corresponding codeword is then generated by XORing the encoded PC_{Frozen} vector pc with the all- i codeword, depending on the decision.

Proof. The most likely codeword corresponds to the codeword with the minimum weighted hamming distance [12] which is equivalent to the path metric (PM). For a repetition node, there are only two candidate paths—corresponding to information bit values 0 and 1—with path metrics defined as

$$PM^0 = \sum_{j=0}^{N_v-1} |l_j| ((pc_j \oplus 0) \oplus HD_i), \quad (5)$$

$$PM^1 = \sum_{j=0}^{N_v-1} |l_j| ((pc_j \oplus 1) \oplus HD_i), \quad (6)$$

where HD_j denotes the hard decision on l_j .

The ML decision corresponds to the path with the smaller path metric. Therefore, path-1 is chosen if $PM^1 \leq PM^0$, i.e.:

$$PM^1 - PM^0 = \sum_{j=0}^{N_v-1} \kappa_j < 0, \quad (7)$$

else path-0 is chosen where,

$$\kappa_j = |l_j| [(pc_j \oplus HD_j) - (pc_j \oplus HD_j)]. \quad (8)$$

We now analyze the sign of this difference term by considering both cases for pc_j .

- If $pc_j = 1$, then $p\bar{c}_j = 0$, and the expression reduces to

$$\kappa = |l_j|(HD_j - H\bar{D}_j) = -l_j. \quad (9)$$

- If $pc_j = 0$, then $p\bar{c}_j = 1$, yielding

$$\kappa = |l_j|(H\bar{D}_j - HD_j) = l_j. \quad (10)$$

Combining both cases, (7) can thus be expressed compactly as

$$PM^1 - PM^0 = \sum_{j=0}^{N_v-1} l_j (1 - 2pc_j), \quad (11)$$

If $PM^1 - PM^0 < 0$, $i = 1$ while if $PM^1 - PM^0 \geq 0$, $i = 0$. Hence determining i is equivalent to using the sign function on $PM^1 - PM^0$.

C. REP-2 Node

A REP 2 node can be considered as two REP nodes, one on the even bits which represents the repetition of the encoded i_{N_v-2} and one on the odd bits which corresponds to the repetition of the encoded i_{N_v-1} . Henceforth, it can be decoded using two repetition nodes:

$$i_{N_v-1} = \text{sign}\left(\sum_{j=0}^{\frac{N_v}{2}-1} l_{2j+1} (1 - 2pc_{2j+1})\right), \quad (12)$$

$$i_{N_v-2} = \text{sign}\left(\sum_{j=0}^{\frac{N_v}{2}-1} l_{2j} (1 - 2pc_{2j})\right). \quad (13)$$

The corresponding codeword is the repetition of i_{N_v-1} on the odd bits, the repetition of i_{N_v-2} on the even bits XORed with the PC_{Frozen} encoded vector. For $j \in [0, \frac{n}{2} - 1]$ this can be represented as:

$$\begin{aligned} \beta_{2j} &= i_{N_v-2} \oplus pc_{2j}, \\ \beta_{2j+1} &= i_{N_v-1} \oplus pc_{2j+1}. \end{aligned} \quad (14)$$

The corresponding information bits at stage 0 can be extracted by descending the graph whereby the information bit at location $N_v - 1$ at stage 0 is i_{N_v-1} , and the information bit at location $N_v - 2$ at stage 0 is $i_{N_v-1} \oplus i_{N_v-2}$.

D. SPC Node

The SPC node decoding can proceed similarly to the method described in [10]. Considering that $p_0 = pc_0$ in (1), we can use change of variable method of the original parity calculation ($\bigoplus_{j=0}^{N_v-1} HD_j$) to ($\bigoplus_{j=0}^{N_v-1} HD_j$) $\oplus pc_0$ to obtain this modified algorithm:

- 1) Find the index of the least reliable input bit:

$$z = \underset{j=0}{\operatorname{argmin}} |l_j|. \quad (15)$$

- 2) Flip bit index z if:

$$\left(\bigoplus_{j=0}^{N_v-1} HD_j\right) \oplus pc_0 \neq 0. \quad (16)$$

E. SPC-2 Node

The parity check equations for the SPC 2 node are:

$$p_0 = \bigoplus_{j=0}^{N_v-1} HD_j, \quad (17)$$

$$p_1 = \bigoplus_{j=0}^{\frac{N_v}{2}-1} HD_{2j+1} \quad (18)$$

where p_j is the j^{th} parity bits at stage 0. The encoded PC_{Frozen} bits are:

$$pc_0 = p_0 \oplus p_1 = \bigoplus_{j=0}^{\frac{N_v}{2}-1} HD_{2j}, \quad (19)$$

$$pc_1 = p_1 = \bigoplus_{j=0}^{\frac{N_v}{2}-1} HD_{2j+1}. \quad (20)$$

Hence, the same procedure taken with an SPC node can be used in this case, using two SPC modules. For the even bits, pc_0 is used as the PC_{Frozen} encoded bit while pc_1 is used for the SPC module for the odd bits.

F. RPC Node

For the RPC node, the new parity equations with the presence of PC_{Frozen} bits are:

$$\begin{aligned} p_0 &= C^0 \oplus C^1 \oplus C^2 \oplus C^3, \\ p_1 &= C^1 \oplus C^3, \\ p_2 &= C^2 \oplus C^3, \\ p_1 \oplus p_2 &= C^1 \oplus C^2, \\ p_0 \oplus p_1 \oplus p_2 &= C^0 \oplus C^3. \end{aligned} \quad (21)$$

Hence,

$$\begin{aligned} p_1 \oplus C^1 \oplus C^3 &= C^2 \oplus p_2 \oplus C^3 \\ &= C^1 \oplus p_1 \oplus C^2 \oplus p_2 \\ &= C^0 \oplus p_0 \oplus p_1 \oplus p_2 \oplus C^3 = 0. \end{aligned} \quad (22)$$

We can note that that the encoded PC_{Frozen} bits can be written as $pc_0 = p_0 \oplus p_1 \oplus p_2$, $pc_1 = p_1$ and $pc_2 = p_2$. Additionally, we can use the change of variables for the colored variables in (22) such that: $C^0 \oplus pc_0 = \tilde{C}^0$, $C^1 \oplus pc_1 = \tilde{C}^1$, and $C^2 \oplus pc_2 = \tilde{C}^2$. Hence, equation (22) can be simplified to

$$\tilde{C}^1 \oplus C^3 = \tilde{C}^2 \oplus C^3 = \tilde{C}^1 \oplus \tilde{C}^2 = \tilde{C}^0 \oplus C^3 = 0, \quad (23a)$$

$$\tilde{C}^0 = \tilde{C}^1 = \tilde{C}^2 = C^3 = 0, \quad (23b)$$

$$\text{and} \quad \tilde{C}^0 = \tilde{C}^1 = \tilde{C}^2 = C^3 = 1. \quad (23c)$$

After changing variables, (23a), (23b), and (23c) are identical to (2a), (2b) and (2c) respectively. A modification for Algorithm 1 to support decoding these new equations would be initializing C^z with $\{pc_0, pc_1, pc_2, 0\}$ on line 2. This would result in the calculation on line 7 for $\{\tilde{C}^0, \tilde{C}^1, \tilde{C}^2, C^3\}$.

TABLE I
THE NUMBER OF NODE TRAVERSALS USING DIFFERENT DECODER CONFIGURATIONS.

Configuration	IR-HARQ used	Transmitted bits	Node Designs	R0	R1	REP	REP2	SPC	SPC2	PCR	RPC	LEAF	Total
A	Yes	1024 → 1024	Proposed method	14	5	33	3	26	3	4	5	0	93
B	Yes	1024 → 1024	[10] [9]	129	79	18	1	15	0	1	2	85	330
C	No	2048	[10] [9]	14	5	33	3	26	3	4	5	0	93

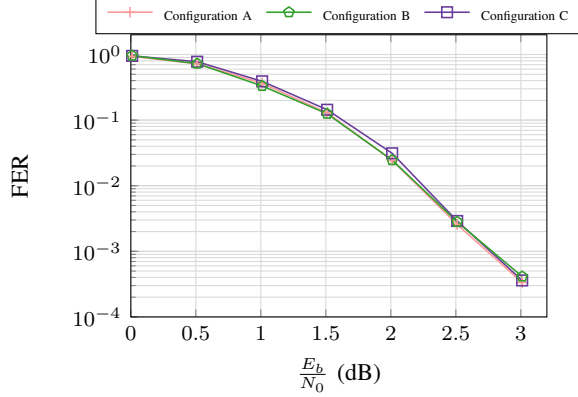


Fig. 1. Comparison of decoding performance of the three configurations discussed in Table I.

G. PCR node

We can apply the same decoding method as the modified repetition node to determine the individual bits corresponding to X^0 , X^1 , X^2 , X^3 . The necessary modifications are:

- 1) Modifying the calculation of Δ_z on line 5 of Algorithm 2 depending on the value of pc_j :

$$\Delta_z = \begin{cases} \Delta_z + l_j, & pc_j = 0, \\ \Delta_z - l_j, & pc_j = 1. \end{cases}$$

- 2) XOR-ing the result of the PCR node $\hat{x}$ with the encoded PC_{Frozen} vector pc .

IV. PERFORMANCE EVALUATION

Table I shows the three different configurations for polar SC decoders used. Configuration A and B are HARQ enabled and allow sending first a codeword of size 1024 followed by another 1024 bits according to [6], while configuration C only allows one single transmission of length 2048. In all cases a 24 bit CRC is used and the maximum total number of data bits transmitted is 2048. We can observe that our proposed method employed in configuration A provides us with the same node types as configuration C. This allows us to reduce the number of node traversals by 72% compared to configuration B which employs IR-HARQ and uses only the unmodified nodes suggested by [10] [9]. This would result in a reduction in decoding latency compared to the IR-HARQ algorithm which does not incorporate our proposed special nodes. Additionally, Fig. 1 presents the decoding performance of each of the proposed configurations in an additive white Gaussian noise channel with Quadrature Phase Shift Keying modulation. We can see that our modifications to the special nodes that enable fast IR-HARQ do not result in any performance degradation compared to the baseline schemes.

V. CONCLUSION

In this work, we present a series of modifications to the state-of-the-art special nodes to enable the IR-HARQ support. Our simulation results verify that modifications to the existing R0, R1, REP, REP 2, SPC, SPC 2, RPC and PCR nodes enable the use of special nodes in the IR-HARQ scheme without any decoding performance degradation. Additionally, our proposed modifications allow us to reduce the number of node traversals with polar code of length 2048 by 72% compared to the IR-HARQ scheme utilizing the unmodified nodes. Coupled with the hardware-friendly bit generation method suggested in [8], IR-HARQ support can be implemented in future standards for polar SC, enabling reliable communications.

REFERENCES

- [1] E. Arikan, "Channel Polarization: A Method for Constructing Capacity-Achieving Codes for Symmetric Binary-Input Memoryless Channels," *IEEE Trans. Inf. Theory*, vol. 55, pp. 3051–3073, July 2009.
- [2] 3GPP, "NR Multiplexing and channel coding," Technical Specification (TS) 3GPP.38.212, 3rd Generation Partnership Project, Nov. 2021.
- [3] Huawei and HiSilicon, "Channel coding for 6GR air interface," Discussion and Decision R1-2505185, 3GPP, 08 2025.
- [4] J.-F. Cheng, "Coding performance of hybrid ARQ schemes," *IEEE Trans. Commun.*, vol. 54, pp. 1017–1029, June 2006.
- [5] K. Chen, K. Niu, Z. He, and J. Lin, "Polar coded HARQ scheme with Chase combining," in *IEEE Wireless Commun. and Netw. Conf.*, pp. 474–479, Apr. 2014.
- [6] M.-M. Zhao, G. Zhang, C. Xu, H. Zhang, R. Li, and J. Wang, "An Adaptive IR-HARQ Scheme for Polar Codes by Polarizing Matrix Extension," *IEEE Commun. Lett.*, vol. 22, pp. 1306–1309, July 2018.
- [7] B. Li, D. Tse, K. Chen, and H. Shen, "Capacity-achieving rateless polar codes," in *IEEE International Symposium Inf. Theory*, pp. 46–50, July 2016.
- [8] M. Jalaaliddine, J. Li, and W. J. Gross, "Hardware-friendly IR-HARQ for polar SCL decoders," in *IEEE International Conf. on Commun.*, pp. 2008–2013, 2025. ISSN: 1938-1883.
- [9] J. Tong, X. Wang, Q. Zhang, H. Zhang, J. Wang, and W. Tong, "Fast polar codes for terabits-per-second throughput communications," in *IEEE International Symposium on Personal, Indoor and Mobile Radio Commun.*, pp. 1–6, 2023.
- [10] G. Sarkis, P. Giard, A. Vardy, C. Thibault, and W. J. Gross, "Fast polar decoders: Algorithm and implementation," *IEEE Journal on Selected Areas in Commun.*, vol. 32, no. 5, pp. 946–957, 2014.
- [11] 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, 2011.
- [12] F. Wang, J. Jiao, K. Zhang, S. Wu, and Q. Zhang, "Adjustable ordered statistic decoder for short block length code towards urllc," in *International Conf. on Wireless Commun. and Signal Processing*, pp. 1–5, 2021.