STUDY OF DIFFERENT CODING METHODS OF POLAR CODE IN 5G COMMUNICATION SYSTEM

Atish A. Peshattiwar
Reaserch Scholar Department of Electronics Engineering, Yeshwantrao Chavan College of Engineering, Nagpur, Maharashtra, (India).
E-mail: atishp32@gmail.com

Atish S. Khobragade
Professor, Department of Electronics Engineering, Yeshwantrao Chavan College of Engineering, Nagpur, Maharashtra, (India).
E-mail: atish_khobragade@rediffmail.com

Reception: 01/12/2022 **Acceptance**: 16/12/2022 **Publication**: 29/12/2022

Suggested citation:

Peshattiwar, A. A., y Khobragade, A. S. (2022). Study of different coding methods of polar code in 5G communication system. *3C Tecnología. Glosas de innovación aplicadas a la pyme, 11*(2), 90-99. https://doi.org/10.17993/3ctecno.2022.v11n2e42.90-99

https://doi.org/10.17993/3ctecno.2022.v11n2e42.90-99

ABSTRACT

Sure, most of you are aware that right now everyone is thinking or at least the industry and researchers are thinking about the next generation, the first generation of what will 5G be, and that a significant part of the 5G telecommunication standard has been finalised, in particular the error control codes that will be used in 5G telecommunication. One of the codes that will be used is called a Low-Density Parity Check code (or LDPC code for short), and the other is called Polar Code. These two famous and celebrated codes have distinct and fascinating histories. Both technologies are now capable of providing near-capacity results, making them formidable competitors in the race to become the 5G communication system's ultimate provider. In this paper, we zeroed in on the 5G Polar codes and their encoding methods.

KEYWORDS

5G Communication System, Channel Coding, LDPC Codes, Polar Codes, Coding Methods.

https://doi.org/10.17993/3ctecno.2022.v11n2e42.90-99

1. INTRODUCTION

The POLAR codes are a new type of capacity-achieving codes developed by Arkan [1]. Since 2008, polar codes have been the subject of increasing study and interest in both the academic and professional worlds. As part of the ongoing standardisation process for 5th generation wireless systems, the 3rd Generation Partnership Project (3GPP) has certified polar codes as the channel coding for uplink and downlink control information for the enhanced mobile broadband (eMBB) communication service (5G). Polar coding is one of the proposed encoding methods for the two new frameworks that 5G expects, namely extremely reliable low-latency communications (URLLC) and massive machine-type communications (mMTC).

Creating a polar code involves determining the values of channel dependability associated with each bit of information to be encoded. This id can be accomplished with a specific signal-to-noise ratio and code length. Due to the expected wide range of code lengths, transmission rates, and channel conditions in the 5G architecture, it is impractical to calculate separate reliability vectors for each possible parameter combination. A lot of work has gone into creating polar codes that are straightforward to implement, require little in the way of descriptive complexity, and provide adequate error-correction across a wide range of code and channel parameters.

In light of their impending widespread deployment, researchers would do well to take into account the one-of-a-kind codes created for 5G and their encoding procedure when evaluating error-correction performance and constructing encoders and decoders. Almost all works of contemporary literature fall short of this goal. Polar code properties have an immediate effect on decoder performance, and there may be substantial time and effort costs associated with encoding and decoding. Publications that focus on hardware and software implementations of the 5G standard may be able to increase their readership if they emphasise compliance with the standard.

An "industry standard" is a set of guidelines for providing a service that has been accepted by a number of different companies. In most cases, an agreement between multiple manufacturers to create products that are compatible with one another leads to a standardisation of details, which is the result of a commercial trade-off. A standard is a compromise between competing goals that ultimately results in a hodgepodge of techniques that, when combined, achieve satisfactory performance.

Here, we zeroed in on one particular 5G coding method—Polar Codes—to see what all the fuss is about. 2014 saw the introduction of new encoding schemes for the polar code [2] developed by kia niu.

The remainder of the paper is organised as follows. Polar code: the basics Section II focuses on encoding, while Section III discusses the more advanced design elements and ideas used in 5G decoding, such as SC, SCF, and SCL. In Section IV, we compare the results of our Latency and Throughput measurements, and in Section V, we draw some conclusions based on our findings.

2. 5G POLAR CODE ENCODING

The method employed by the 5G standard to encode polar codes is discussed at length here. In what follows, I'll be making use of the notation defined by the 3GPP technical specification [2]. Polar codes are used to transmit uplink control data over the physical uplink shared channel and uplink control channel. Both the payload on the physical broadcast channel and the downlink control information (DCI) on the physical down-link control channel (PDCCH) are encrypted using polar codes during the downlink transmission (PBCH). For the upper communication layers in 5G applications, the required rate $R = A/E$ is specified, where A is the amount of information bits.

A mother polar code of length N = 2n is required for this purpose. When the code is too long (or too short) for the specified code length E, it is punctured, shortened, or repeated until it is. Depending on the channel, the minimum and maximum code lengths N for the uplink and downlink are 32 and 512, 1024, respectively. An additional cap is imposed by the minimum allowable coding rate of 1/8. Figure 6 depicts the various encoding methods planned for use in the 5G polar codes design. The bits of information contained in vector a, denoted by the G code, will be conveyed using the payload of G code bits. Depending on the parameters of the encoding scheme, the message could be split into two parts, each of which would be encoded separately before being sent. For every AJ-bit segmented vector, a polar code-word of length E will be generated. There is an associated L-bit CRC for each AJ-bit vector. The resulting vector c is fed into an interleave, which requires K = AJ + L bits. To generate a mother polar code of length N, we need to know the expected coding rate R, the expected codeword length E, the relative bit channel reliability sequence, and the frozen set. While the remaining bits of the N-bit u vector are held steady, the interleaved vector cJ and any parity-check bits are added to the information set.

Using the generator matrix of the selected mother code, GN = G2n, we encode the vector u as d = u GN. Sub-block interleaving is then used to divide the encoded d into 32 blocks of the same length. Then, the circular buffer receives these blocks after they have been scrambled to produce y. For rate matching, the N-bit vector y is modified in some way (puncturing, shortening, or repeating) to yield the E-bit vector e. In the event that concatenation is necessary, the computed vector f is then ready to be modulated and transmitted as g.

Having the parameters A and E in play makes it clear that there is a cap on the effectiveness of the channel being used. While A 11 uses many different block codes, the uplink uses 12 A 1706. The agreed upon codeword length range is 18 E 8192, even though G 16384 may cause the payload length G to be longer. Segmentation could be used to do this, dividing the data bits into two polar codewords. . For PDCCH in the downlink, the maximum value of A is 140, but in this case, if A is 11, the message will be zero-padded until A = 12. Although E 8192 is employed for uplink, the presence of the CRC lower limits E to 25. There is only one valid PBCH passcode that utilises the combination of A = 32 and E = 864. These flags are the Input Bits Interleaver Activation (IBIL) signal and the Channel Interleaver Activation (CIA) (IIL). There are two kinds of PC helper bits, and NPC and nwm both provide the total number of them.

Fig. 1. Yellow, red, and orange blocks are used in downlink, uplink, and both, respectively, in the 5G polar codes encoding chain.

3. DECODING CONSIDERATION

Even though the 3GPP does not provide any decoding instructions, the final code structure provides some pointers on how to decode polar codes, which are commonly used in 5G. Figure 1 shows an encoding sequence; flipping the sequence facilitates decoding. The received encoded symbols are then padded and deinterleaved to the length of the mother polar code following a second block segmentation and deinterleaving operation for the uplink. A rate-matching algorithm will be used to determine the type of padding to be used, with punctuation requiring the addition of zeros, shortening requiring the use of saturated symbols, and repetition calling for the merging of symbols. The importance of properly handling the code's helper bits at this stage will be emphasised. After padding,

https://doi.org/10.17993/3ctecno.2022.v11n2e42.90-99

the number of encoded symbols is a power of two, which is readable by common polar code decoders [1]. Next, we consider SCL in light of these repercussions and current needs.

When the check for all active pathways fails, the list decoder triggers its early decoding termination feature; decoding continues when at least one active pathway succeeds. The decoder's BLER performance may, however, depend on how it deals with routes that have become unavailable. Keeping the unsuccessful paths in the list is the simple solution to keeping the number of active paths constant and simplifying the decoder design. After each bit estimate, one strategy is to immediately turn off failed paths. As a result, the BLER performs better, and the computational complexity and energy consumption of the decoder are reduced. However, this causes a variation in the total number of possible courses of action. Last but not least, distributed assistance bits could be thought of as dynamically frozen bits that supply the bit's value for the check. The BLER's efficiency is the same using this method as it was using the ineffective path deactivation. Since all surviving paths are guaranteed to get the work done, assistance bits do not have the same impact on the decoder's computational complexity or power consumption as dynamically frozen bits do. As an added bonus, they do not lead to dismissal in the midst of the work week.

However, the number of viable paths varies based on the computational complexity and power requirements of the decoder. Finally, distributed assistance bits can be viewed as dynamically frozen bits that supply the bit with the value the check requires. The BLER's efficiency is the same using this method as it was using the ineffective path deactivation. However, unlike dynamically frozen bits, which reduce computational complexity and power consumption of the decoder and lead to early termination, assistance bits do not guarantee that all remaining pathways will pass the check.

Native successive cancellation (SC) decoding of polar codes is also recommended in [1]. Like a left-biased depth-first binary tree search, its decoding time is linear in N log N. N bits are approximable at the leaf nodes, while the root node has soft information on the received code bits. Figure 4 depicts the decoding tree for a (8, 4) polar code, where black leaf nodes represent the information bits and white nodes represent the frozen bits. Figure 3 shows how the message flow can be defined recursively with a node at the tth stage as the starting point. The node receives 2t soft inputs from its parent node and uses them to generate 2t1 soft outputs _, which are then sent to the node's left child via the formula I = f I i+2t1); later, the node combines the 2t1 hard decisions it receives from the node's left child with to generate 2t1 soft outputs r for the Finally, it combines the 2t1 hard decisions r it received from its right child with to determine the 2t hard decisions it will send to its parent node, with I = I ri if I 2t1 and I = r i2t1 otherwise. When a leaf node is accessed, the soft information is used to make a hard decision regarding the value of the information bits; frozen bits are always decoded as zeros.

In particular, the channel model affects update rules f and g for left and right child nodes. In BEC, hard decisions can take on any value from 0 to 1, while soft values are limited to 0 and 1.

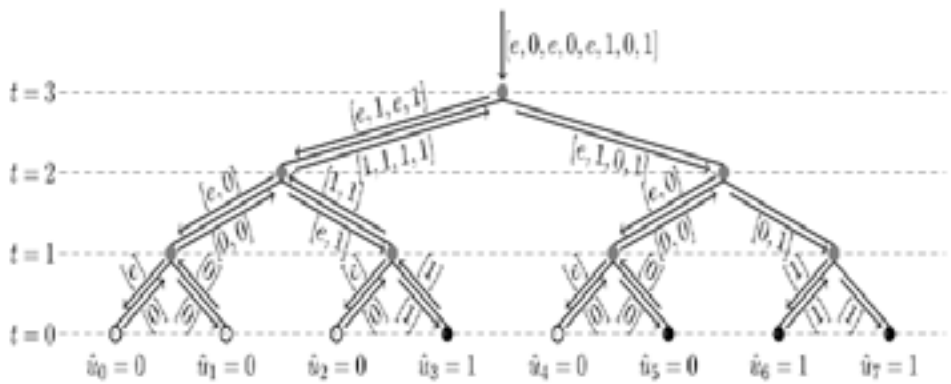

Fig. 2. White and black dots in the SC decoding of a (8, 4) polar code over a BEC stand for frozen and information bits.

https://doi.org/10.17993/3ctecno.2022.v11n2e42.90-99

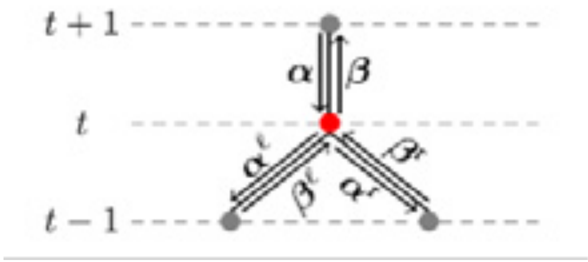

Fig. 3. SC decoding node.

3.1. DECODING FOR SUCCESSIVE-CANCELLATION (SC)

The SC decoding method, as first described [1], operates by sequentially traversing the graph representation of Fig. 2 while iteratively computing u from the noisy channel data. Do this from top to bottom and from right to left. It was first advised to calculate two bits at once in order to decrease time and boost throughput [3]. By utilising specialised, quicker decoding algorithms on selected network nodes [6] or even pruning the graph using a priori knowledge of the locations of the frozen bits [5], the SC technique was further improved. One candidate codeword is always considered, regardless of the SC algorithm version that is being used.

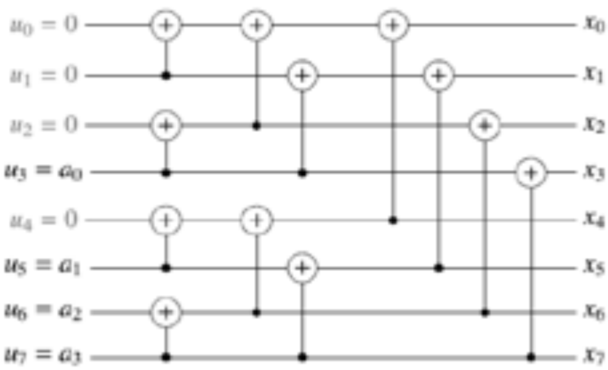

Fig. 4. Graph representation of a (8, 4) polar code.

3.2. PERFORMANCE OF SC DECODING

Fig. 5 compares the (2048, 1723) 10GBASE-T LDPC code's error-correction performance to polar codes running at the same rate. These results were obtained by modulating the binary-input additive white Gaussian noise (AWGN) channel with random codewords and binary phase-shift keying (BPSK). First, it must be stated that the (2048, 1723) polar code performs substantially poorer than the LDPC code. Up until Eb/N0 = 4.25 dB, the LDPC code works better than the PC(32768, 27568) polar code, which was intended to perform best for Eb/N0 = 4.5 dB. There is a growing performance difference over the LDPC code after that. The final polar error-rate curve, abbreviated PC*, is produced by combining the output of two polar codes (32768, 27568). (32768, 27568). The first, which has been in use up until this point, was constructed for 4.25 dB, whereas the second was designed for 4.5 dB. Polar codes can be easily decoded because of their predictable structure, which makes it easy to create a decoder that can decipher any polar code of a specific length. Therefore, polar code changes inside a system are easier than LDPC code modifications.

https://doi.org/10.17993/3ctecno.2022.v11n2e42.90-99

These findings demonstrate that a (32768, 27568) polar code built with an Eb/N0 of 4.5 dB or higher is required to surpass the (2048, 1723) LDPC one in the low error-rate zone and that this may be done even in high error-rate regions by combining numerous polar codes. Even though the polar code is longer than the LDPC code, its decoder is nevertheless built in a simpler manner. The performance difference between the (2048, 1723) code and the LDPC code with a list size of 32 and a 32-bit CRC is reduced by using the list-CRC technique [4], as illustrated in Fig. 2.

The use of some of these strategies to list decoding will need more study. Due to its serial nature, SC decoding has a limited throughput. ASIC decoders for (1024, 512) polar codes currently offer the fastest implementation, processing data at a rate of 48.75 Mbps while operating at 150 MHz [11]. The fastest decoder, on the other hand, can handle data at a rate of 26 Mbps for the (32768, 27568) code and is FPGA-based. Although it may be significantly increased by employing the SSC or ML-SSC decoding algorithms, this poor throughput renders SC decoders unusable for the bulk of systems.

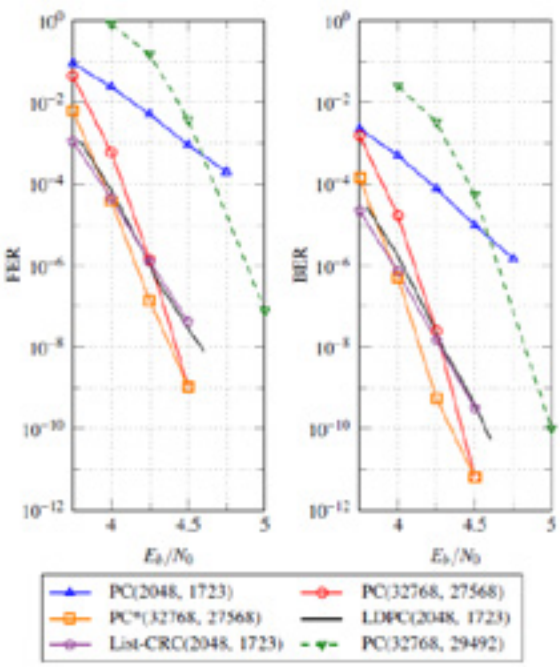

Fig. 5. Polar codes' error-correction performance is contrasted with an LDPC code of the same rate. In addition to a rate 0.9 polar code's functionality.

3.3. DECODING USING SUCCESSIVE-CANCELLATION FLIP (SCF)

Similarities exist between the SC algorithm and the SCF decoding technique [8]. At first, its operation is similar to that of SC decoding; however, in addition to decoding, it also keeps track of the bits that are the least reliable. A cyclic redundancy check must also be used to string together the polar code (CRC). The SCF decoder generates a full codeword candidate and checks if the computed CRC is the same as the expected one. When the CRC verification fails, the SC decoding process continues until the least reliable bit-decision is found. The SCF makes a contrasting conclusion and continues decoding SC. After two tries, if the CRC does not match the expected CRC, the bit-decision with the second-lowest confidence is swapped and the process is repeated. Until the CRC comparison is conclusive or until the maximum number of trials have been conducted, this technique is repeated.

3.4. DECODING OF THE SUCCESSIVE-CANCELLATION LIST (SCL)

As their names imply, the SCL algorithm [7] and the SC algorithm have some similarities. In contrast to SC decoding, the output of the SCL decoding method is a smaller set of up to L possible codewords. It considers both ui options for the locations I where data bits can be found. A path dependability metric computed during the method is used to filter out all but the L best pathways from the final list of survivors. At the end of the decoding process, the estimated codeword is selected from among the L candidates based on which one has the highest route dependability score.

If a polar code and a CRC are paired, the projected CRC and computed CRC for each of the L candidates are compared. Out of all the candidates who pass the CRC, the decoded codeword is selected as the most trustworthy candidate. If all candidates fail the CRC, the technique simply selects the candidate with the highest route dependability.

3.4. LATENCY AND THROUGHPUT

To evaluate the latency boost from the novel algorithm and implementation, we compare two unrolled decoders with an LLR-based SC-list decoder developed in accordance with the technique described in [6] in Table I. The first unrolled decoder, also called the "unrolled SC-list," does not rely on any specific set of constituent decoders. The second is called "unrolled Dec-SPC-4," and it incorporates all of the component decoders we have covered so far while limiting the number of SPC decoders to four. When the SC-list decoder is unrolled, we find that decoding time is cut by more than half. Latency is reduced to between 63% (L = 2) and 18.9% (L = 32) of the unrolled SC-list decoder when the rate-0, rate-1, repetition, and SPC-4 constituent decoders are used. In comparison to SC-list decoding, the suggested decoding strategy and implementation yield a performance boost of between 18.4 and 11.9% for list sizes of 2 and 32, respectively.

The impact of the unrolled decoder is more noticeable for shorter lists, while the impact of the new constituent decoders is greater for longer lists.

When there is no limitation on the length of the individual SPC decoders, as in the case of "Unrolled Dec-SPC-4+" in Table I, the latency for the recommended decoder is also shown. Turning on these longer constituent decoders decreases latency by 14% and 18% for L = 2 and L = 8, respectively. We do not advise using SPC-8+ component decoders because of the drastic decrease in error-correction performance for L >= 8. Consequently, we do not guarantee the lag associated with such a decoder setup. The proposed decoder has a roughly linear decrease in throughput in function of L. At L = 32 and 433 s of latency, the data rate is 4 Mbps. Performance can be improved with adaptive decoding by using a Fast-SSC decoder prior to the list decode. The results of this method's throughput are shown for L = 8 and L = 32 in Table II. According to (17), when the throughput for L = 8 and 32 is equal, the effect of the list decoder on throughput diminishes, and the Fast-SSC performs better at 4.5 dB.

4. COMPARATIVE ANALYSIS

Table 1. Latency (in µsec) for various decoding methods of polar code.

Decoder	L		
	2	8	32
SC-List	558	1450	5145
Unrolled SC-list	193 (2.9×)	564(2.6×)	2294(2.2×)
Unrolled Dec SPC-4	30.4(18.4×)	97.5(14.9×)	433(11.9×)
Unrolled Dec SPC-4+	26.3(21.2×)	80.2(18.1×)	N/A

https://doi.org/10.17993/3ctecno.2022.v11n2e42.90-99

Table 2. When calculating 524,280 information bits, the suggested list decoder's information throughput and latency were compared to those of the ldpc decoders of [32].

Decoder	N	# of N-bit frames	Rate	Latency (ms)		Info. T/P (Mbps)
				Total	Per frame	
[32]	1944	540	1/2	17.4	N/A	30.1
proposed	1024	1024	1/2	13.8	0.014	38.0
[32]	1944	405	2/3	12.7	N/A	41.0
proposed	1024	768	2/3	10.0	0.013	52.4
[32]	1944	360	3/4	11.2	N/A	46.6
proposed	1024	683	3/4	8.78	0.013	59.6
[32]	1944	324	5/6	9.3	N/A	59.4
proposed	1024	615	5/6	6.2	0.010	84.5

5. CONCLUSION

This article will help the reader understand and implement the polar code encoding process in accordance with the 5G wireless systems standard, as well as simulate it for practise. This encoding chain exemplifies the 3GPP standards body's efforts to support a variety of code lengths and speeds while satisfying the various code criteria for the eMBB control channel, such as low description complexity and low encoding complexity. Insights into the consideration given to the receiver side during the standardisation process are provided. So long as cutting-edge hardware and decoders are used, the encoder was designed to allow for the decoder to be generated with a tolerable level of complexity and perform at the required latency. But now it is possible to optimise decoding complexity or improve error-rate performance by developing new decoding structures or principles.

We also investigated potential delays introduced by various decoding strategies, including successive cancellation (SC) decoding, successive cancellation flip (SCF) decoding, and successive cancellation list (SCL) decoding. Additionally, we conducted latency analyses to determine the relative merits of each decoding technique. Decoding algorithms can be tweaked to improve delay and throughput.

REFERENCES

[1] E. Arıkan, “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 “Multiplexing and channel coding, v.15.3.0,” 3GPP, Sophia Antipolis, France, Rep. 38.212, 2018.

[2] K. Niu, K. Chen, J. Lin, and Q. T. Zhang, “Polar codes: Primary concepts and practical decoding algorithms,” IEEE Commun. Mag., vol. 52, no. 7, pp. 192–203, Jul. 2014.

[3] H. Vangala, E. Viterbo, and Y. Hong, “A comparative study of polar code constructions for the AWGN channel,” Jan. 2015. [Online]. Available: arXiv:1501.02473

[4] R. Mori and T. Tanaka, "Performance of polar codes with the con- struction using density evolution," IEEE Commun. Lett., vol. 13, no. 7, pp. 519–521, Jul. 2009.

[5] I. Tal and A. Vardy, "How to construct polar codes," IEEE Trans. Inf. Theory, vol. 59, no. 10, pp. 6562–6582, Oct. 2013.

[6] P. Trifonov, "Efficient design and decoding of polar codes," IEEE Trans. Commun., vol. 60, no. 11, pp. 3221–3227, Nov. 2012.

[7] M. Mondelli, S. H. Hassani, and R. Urbanke, "Construction of polar codes with sublinear complexity," in Proc. IEEE Int. Symp. Inf. Theory (ISIT), Aachen, Germany, Jun. 2017, pp. 1853–1857.

[8] G. He et al., "Beta-expansion: A theoretical framework for fast and recursive construction of polar codes," in Proc. IEEE Glob. Commun. Conf. (GLOBECOM), Singapore, Dec. 2017, pp. 1–6.

[9] C. Condo, S. A. Hashemi, and W. J. Gross, "Efficient bit-channel relia- bility computation for multi-mode polar code encoders and decoders," in Proc. IEEE Int. Workshop Signal Process. Syst. (SiPS), Lorient, France, Oct. 2017, pp. 1–6.

[10] M. Mondelli, S. H. Hassani, and R. L. Urbanke, "From polar to Reed–Muller codes: A technique to improve the finite-length performance," IEEE Trans. Commun., vol. 62, no. 9, pp. 3084–3091, Sep. 2014.

[11] V. Bioglio, F. Gabry, I. Land, and J.-C. Belfiore, "Minimum-distance based construction of multi-kernel polar codes," in Proc. IEEE Glob. Commun. Conf. (GLOBECOM), Singapore, Dec. 2017, pp. 1–6.

[12] M. Mondelli, S. H. Hassani, and R. L. Urbanke, "Scaling exponent of list decoders with applications to polar codes," IEEE Trans. Inf. Theory, vol. 61, no. 9, pp. 4838–4851, Sept. 2015.

[13] J. Guo, M. Qin, A. G. I. Fábregas, and P. H. Siegel, "Enhanced belief propagation decoding of polar codes through concatenation," in Proc. IEEE Int. Symp. Inf. Theory (ISIT), Honolulu, HI, USA, Jun. 2014, pp. 2987–2991.

[14] U. U. Fayyaz and J. R. Barry, "Low-complexity soft-output decoding of polar codes," IEEE J. Sel. Areas Commun., vol. 32, no. 5, pp. 958–966, May 2014.

[15] A. Balatsoukas-Stimming, A. J. Raymond, W. J. Gross, and A. Burg, "Hardware architecture for list successive cancellation decoding of polar codes," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 61, no. 8, pp. 609–613, Aug. 2014.

[16] A. Balatsoukas-Stimming, M. B. Parizi, and A. Burg, "LLR-based suc- cessive cancellation list decoding of polar codes," IEEE Trans. Signal Process., vol. 63, no. 19, pp. 5165–5179, Oct. 2015.

[17] C. Leroux, A. J. Raymond, G. Sarkis, and W. J. Gross, "A semi-parallel successive-cancellation decoder for polar codes," IEEE Trans. Signal Process., vol. 61, no. 2, pp. 289–299, Jan. 2013.

[18] B. L. Gal, C. Leroux, and C. Jego, "Software polar decoder on an embed- ded processor," in Proc. IEEE Workshop Signal Process. Syst. (SiPS), Belfast, U.K., Oct. 2014, pp. 180–185.

[19] O. Afisiadis, A. Balatsoukas-Stimming, and A. Burg, "A low-complexity improved successive cancellation decoder for polar codes," in Proc. IEEE Asilomar Conf. Signals Syst. Comput., Pacific Grove, CA, USA, Nov. 2014, pp. 2116–2120.

[20] C. Condo, F. Ercan, and W. J. Gross, "Improved successive cancellation flip decoding of polar codes based on error distribution," in Proc. IEEE Wireless Commun. Netw. Conf. (WCNC), Barcelona, Spain, Apr. 2018, pp. 19–24.

[21] I. Tal and A. Vardy, "List decoding of polar codes," IEEE Trans. Inf. Theory, vol. 61, no. 5, pp. 2213–2226, May 2015.

[22] S. A. Hashemi, A. Balatsoukas-Stimming, P. Giard, C. Thibeault, and W. J. Gross, "Partitioned successive-cancellation list decoding of polar codes," in Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP), Shanghai, China, Mar. 2016, pp. 957–960.

[23] S. A. Hashemi, M. Mondelli, S. H. Hassani, R. L. Urbanke, and W. J. Gross, "Partitioned list decoding of polar codes: Analysis and improvement of finite length performance," in Proc. IEEE Glob. Commun. Conf. (GLOBECOM), Singapore, Dec. 2017, pp. 1–7.

https://doi.org/10.17993/3ctecno.2022.v11n2e42.90-99