Soft List Decoding of Polar Codes

Luping Xiang, Yusha Liu, Zeynep B. Kaykac Egilmez, Robert G. Maunder, *Senior Member, IEEE*, Lie-Liang Yang, *Fellow, IEEE* and Lajos Hanzo, *Fellow, IEEE*

Abstract—Soft-output (SO) decoding is proposed for the Logarithmic Successive Cancellation List (Log-SCL) polar decoder for the first time, by exploiting the left-to-right propagation of the Belief Propagation (BP) decoder, which opens new avenues for its employment in powerful turbo-receivers. In the case of decoding a half-rate polar code having a block length of 1024 bits, the proposed soft list polar decoder achieves a 1.5 **dB Block Error Ratio (BLER) performance gain,** 50% **latency improvement** and 26% **complexity reduction, compared to the state-of-the-art SO Soft Cancellation (SCAN) polar decoder in a polar-coded Multiple-Input Multiple-Output (MIMO) system. Furthermore, we conceive a Memory-Efficient (ME) soft list polar decoder, which requires only** 16% **of the soft list polar decoder's memory, at the cost of slightly increased latency and complexity.**

I. INTRODUCTION

Polar codes [1] have been selected for protecting the 5G New Radio (NR) control channels, as a benefit of their superior Block Error Ratio (BLER) performance at short block lengths [2]. Therefore, both Hard-Output (HO) and Soft-Output (SO) polar decoders have been investigated in the literature [1, 3–8]. The soft-input HO (SIHO) polar decoders typically rely on the original Successive Cancellation (SC) decoder proposed in [1] or sphere decoder of [9], whereas SISO may be produced by the Belief Propagation (BP) algorithm [3, 10].

The SC decoder is capable of approaching the Binary-Input Discrete Memoryless Channel (B-DMC) capacity for infinite block lengths [1] where the channel is sufficiently polarized, but exhibits poor BLER performance for realistic finite block lengths due to the associated error propagation. Considering a list of candidate bit sequences and operating on the basis of Logarithmic Likelihood Ratios (LLRs), the Logarithmic Successive Cancellation List (Log-SCL) polar decoder, developed from the SCL decoder of [11], improves the error-correction performance of the SC decoder in the case of practical block lengths [4]. The software and hardware implementations of the Log-SCL decoder have been investigated in [12–15], which have a low memory requirement and a low decoding latency. However, the Log-SCL decoder is not a SISO scheme, and hence fails to exploit the iterative LLR updates gleaned from the detector of a turbo-receiver [5, 16].

L. Xiang, Y. Liu, Z. B. Kaykac, R. G. Maunder, L.-L. Yang and L. Hanzo are with School of Electronics and Computer Science, University of Southampton, SO17 1BJ, UK. (E-mail: lx1g15, yl6g15, zbk1y15, rm, lly, lh@ecs.soton.ac.uk).

Furthermore, the BP and SCAN decoders of [3, 6, 7] suffer from performance loss compared to the Log-SCL decoder for transmission over Additive White Gaussian Noise (AWGN) channels [8].

With the objective of amalgamating the benefits of both the Log-SCL and BP decoders, we propose a pair of soft polar decoders for generating soft outputs for the Log-SCL decoder for the first time. The novel contributions of this paper are summarised as follows.

- We conceive a soft list polar decoder, which is achieved by intrinsically amalgamating the Log-SCL and left-to-right BP decoding, without requiring inner iterations. This approach exploits the excellent error-correction capability of the Log-SCL decoder as well as the soft-output of BP decoder. Additionally, an LLR-flip technique is applied in the proposed soft list polar decoder for matching the BLER performance of the Log-SCL decoder. The soft list decoder achieves a 50% lower decoding latency and 26% complexity reduction, when decoding a half-rate polar code having a block length of 1024, compared to the state-of-the-art SISO SCAN polar decoder [6].
- We demonstrate that the soft list polar decoder allows iterative extrinsic information exchange within a turbo receiver, which has a better BLER in a Multiple-Input Multiple-Output (MIMO) system communicating over fading channels, than the SCAN decoder of [6].
- Finally, for mitigating the memory requirement of the proposed soft list polar decoder, we propose a Memory-Efficient (ME) soft list polar decoder, reducing the memory requirement to as low as 16% of that of the soft list polar decoder, at the cost of slightly increased decoding latency.

The paper is structured as follows. Section II details the polar-coded MIMO system model, while a pair of novel polar decoders are proposed in Section III. Then their BLER, latency, complexity and memory requirement are characterised in Section IV. Finally, our conclusions and future research ideas are summarised in Section V.

II. POLAR-CODED MIMO SYSTEM MODEL

The transmitter and receiver schematic of a polar-coded MIMO system is shown in Fig. 1, where K information bits are encoded and then, using Quadrature Amplitude Modulation (QAM), they are mapped to E symbols to be transmitted by N_t transmit antennas. To be more specific, the channel polarisation weights for the $N = 2^m$ $(N > K)$ bit channels are first determined by the 3rd Generation Partnership Project (3GPP) standard of the 5G NR polar codes [2]. The resultant K most polarised channels are employed for transmitting the

information bits, whereas the remaining $(N - K)$ bit channels transmit 0-valued frozen bits.
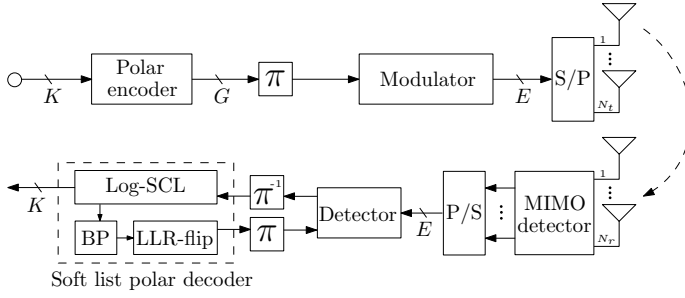

Fig. 1. Transceiver of the polar-coded MIMO system.

Then the resultant vector $\mathbf{u}_0$ comprising N bits may be polar encoded into a vector $\mathbf{u}_m$ of N encoded bits, according to the modulo-2 matrix multiplication $\mathbf{u}_m = \mathbf{u}_0 \mathbf{F}_2^{\otimes m}$, where $\mathbf{F}_2^{\otimes m}$ is the generator matrix, and the superscript $\otimes m$ indicates the mth Kronecker power of the matrix $\mathbf{F}_2$, which is expressed as $\mathbf{F}_2 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$.

Fig. 2 exemplifies this encoding operation, where the core information block is $\mathbf{u}_0 = [00000100]$, with $u_{3,0}$, $u_{5,0}$, $u_{6,0}$, and $u_{7,0}$ being the information bits, which is input at the left-hand edge of the graph and the $m = 3$ successive layers of XOR operations produce the core encoded block $\mathbf{u}_3 = [11001100]$ on its right-hand edge.

The encoded bits $\mathbf{u}_m$ are output from the encoder for subsequent operations, which typically includes rate matching, as in the 3GPP NR uplink polar code [17]. After rate matching, a polar-coded block comprising G bits is generated, giving a code rate of $R = K/G$.
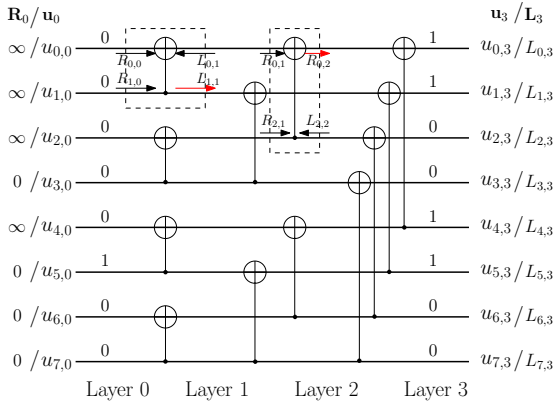

Fig. 2. The code graph of polar codes for the example of $N = 8$.

The polar encoded bits are interleaved and then mapped to $E = G/\log_2 M$ M-ary QAM symbols. After S/P conversion and further transmitter processing, these E QAM symbols are transmitted by N_t transmit antennas. The transmit signal can be represented by a $N_t \times (E/N_t)$ matrix $\mathbf{X}$.

After transmission of $\mathbf{X}$ over flat fading channels having the channel impulse response (CIR) between tth transmit antenna and the rth receive antenna at a certain time slot assumed as $h_{r,t}$, the observations at the receiver can be expressed as

$$\mathbf{Y} = \mathbf{HX} + \mathbf{N}, \qquad (1)$$

where $\mathbf{H} = \begin{bmatrix} h_{1,1} & \cdots & h_{1,N_t} \\ \vdots & \ddots & \vdots \\ h_{N_r,1} & \cdots & h_{N_r,N_t} \end{bmatrix}$ and $\mathbf{N}$ is the the additive white Gaussian noise (AWGN), which has a zero mean and a variance of $\sigma^2 = 1/(2\gamma)$, where γ denotes the signal-to-noise ratio (SNR). We assume that the CIR remains constant within a time slot.

The received signal $\mathbf{Y}$ is first input to the MIMO detector. Then P/S conversion, MQAM detection, de-interleaving, the inverse of rate matching, as well as polar decoding are carried out at the receiver. Here, the turbo-principle can be applied for extrinsic information exchange between the MIMO detector, demodulator and decoder, for further enhancing the information reliability and the resultant BLER performance [5, 18]. However, the conventional Log-SCL decoder fails to produce soft-output for exploitation by the MIMO detector. Therefore, in the following section, we propose a pair of novel polar decoders, which provide soft-output feedback for the detector.

III. POLAR DECODING ALGORITHMS

In Section III-A, we first propose a novel soft list polar decoder, which jointly performs Log-SCL and left-to-right BP decoding, followed by a ME design in Section III-B.

A. Soft list polar decoder

Algorithm 1 Soft list polar decoding algorithm

Input:
  The detected LLRs $\mathbf{L}_m$;

Output:
  The coded *a posteriori* LLRs $\bar{\mathbf{R}}_m$ and the decoded bits $\hat{\mathbf{u}}_0$;

1: Perform Log-SCL decoding according to [4], and store $\mathbf{L}_j^{(s)}$ in each layer $j \in [0, m]$;
2: Re-encode the most promising candidate sequence $\hat{\mathbf{u}}_0$;
3: Perform backwards BP decoding by employing (5a) and (5b), with the left LLRs $\mathbf{L}_j$ being those of the best candidate sequence in the Log-SCL decoding;
4: **for** $i = 0, 1..., N - 1$ **do**
5:    **if** $\text{sign}(R_{i,m}) == 2\hat{u}_{i,m} - 1$ **then**
6:       $\bar{R}_{i,m} = -R_{i,m}$;
7:    **else**
8:       $\bar{R}_{i,m} = R_{i,m}$;
9:    **end if**
10: **end for**

The operations of the proposed soft list polar decoder are summarised in Algorithm 1. Our soft list polar decoder starts by Log-SCL decoding having a list size of S performed on the LLRs $\mathbf{L}_m$ detected, where $L_{i,m} = \ln \frac{\Pr(\hat{u}_{i,m}=0)}{\Pr(\hat{u}_{i,m}=1)}$, giving S candidate decoding sequences, each comprising N decoded bits, together with a path metric [4]. Additionally, the left LLRs $\mathbf{L}_j^{(s)}, s \in [1, S]$, which are the output of the f and g functions at j-th layer, will be employed for computing the path metric during each layer of the code graph and have to be stored for subsequent calculations. More specifically, the Log-SCL decoder carries out the inverse operations of the encoding

process exemplified in Fig. 2, from right to left. For each XOR, two types of computations associated with LLRs will be performed, referred to as f and g functions, respectively [8]. The f function is performed when a pair of LLRs x and y are input to the right-hand side of a XOR, which can be expressed as

$$f(x, y) \approx \text{sign}(x)\text{sign}(y) \min\left(|x|, |y|\right). \qquad (2)$$

By contrast, the g function generates an LLR to be forwarded left, given the right input of LLRs, x and y, and the left input of a hard bit decision u, which can be expressed as

$$g(x, y, u) = (-1)^u x + y. \qquad (3)$$

After we obtained the S bit sequences decoded by the Log-SCL decoder, we re-encode the bit sequence $\hat{\mathbf{u}}_0$ having the most promising path metric to get hard decisions $\hat{\mathbf{u}}_m$ at the right-hand side of the polar code graph. The other $(S-1)$ candidate bit sequences are discarded. At the same time, BP decoding is performed from left to right (backwards) based on the LLRs of the best candidate sequence $\mathbf{L}_j, j \in [0, m]$, which have been calculated and stored during the previous Log-SCL decoding. The computations of the LLRs during BP decoding are exemplified in Fig. 3.

As shown in Fig. 3, since the BP is performed only backwards from left to right, we refer to the output LLRs of the BP decoding at the j-th layer as the right LLRs $\mathbf{R}_j$. The initialization of the left-most right LLR of the ith bit, $R_{i,0}$, is expressed as

$$R_{i,0} = \begin{cases} 0, & \text{if } \hat{u}_{i,0} \text{ is the information bit;} \\ \infty, & \text{otherwise.} \end{cases} \qquad (4)$$

Following the initialisation, for all nodes at the jth layer of the code graph, starting from $j = 0$, the right LLR of the ith and $(i+2^j)$th bit in the $(j+1)$st layer, $R_{i,j+1}$ and $R_{i+2^j,j+1}$, are calculated as

$$R_{i,j+1} = f(R_{i,j}, R_{i+2^j,j} + L_{i+2^j,j+1}), \qquad (5a)$$
$$R_{i+2^j,j+1} = f(R_{i,j}, L_{i,j+1}) + R_{i+2^j,j}. \qquad (5b)$$

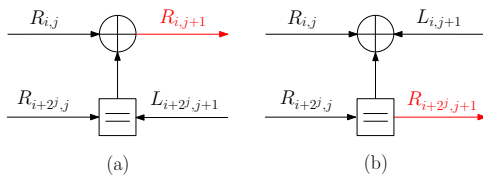Note that the left LLRs $L_{i,j+1}$ and $L_{i+2^j,j+1}$ have been recorded during the previous Log-SCL decoding.

Fig. 3. LLR propagation of the backwards BP decoding.

After the backwards BP decoding, the signs of newly generated right LLRs $\mathbf{R}_m$ are compared to the re-encoded bits $\hat{\mathbf{u}}_m$. For those LLRs having signs that are consistent with the re-encoded bit values, they are output to the channel detectors for following the iterative operations. By contrast, when the signs of the LLRs are contrary to the polar-encoded bits, we appropriately adapted the idea of bit-flipping techniques, which have been widely used in polar decoding algorithms [19, 20]. More specifically, since the signs of the LLRs are not

---

Algorithm 2 ME soft list polar decoding algorithm

Input:
The detected LLRs $\mathbf{L}_m$;

Output:
The coded *a posteriori* LLRs $\bar{\mathbf{R}}_m$ and the decoded bits $\hat{\mathbf{u}}_0$;

1: Perform Log-SCL decoding according to [4];
2: Re-encode the most promising candidate sequence $\hat{\mathbf{u}}_0$;
3: Perform the right-to-left detected LLRs $\mathbf{L}_m$ propagation alongside the code graph using BP schedule, employing f and g functions of (2) and (3), respectively;
4: Perform backwards BP decoding by employing (5a) and (5b), with the left LLRs $\mathbf{L}_j$ being those of the best candidate sequence in the Log-SCL decoding;
5: **for** $i = 0, 1..., N-1$ **do**
6:     **if** $\text{sign}(R_{i,m}) == 2\hat{u}_{i,m} - 1$ **then**
7:         $\bar{R}_{i,m} = -R_{i,m}$;
8:     **else**
9:         $\bar{R}_{i,m} = R_{i,m}$;
10:     **end if**
11: **end for**

---

consistent with the expected polar-encoded bit values, they are more likely to be wrongly decoded. Therefore, these LLRs are flipped, but the absolute values remain unchanged. This is in contrast to the bit-flipping technique, which only flips the bit values during the decoding process. In this way, the flipped LLR enhances the likelihood of the corresponding flipped bit value. After the LLR flip, the resultant *a posteriori* LLRs $\bar{\mathbf{R}}_m$ are input to the detector for iterative detection and decoding.

B. Memory-efficient (ME) soft list polar decoder

As discussed in Section III-A, each LLR $L_{i,j}^{(s)}$ generated during the first Log-SCL decoding has to be stored for the following BP decoding, which requires substantial amount of memory. In order to reduce the memory requirement, we now propose a ME soft list polar decoder, as summarised in Algorithm 2, which reduces requirement of memory to store LLRs at the cost of extra decoding latency, as will be discussed in Section IV-B.

The ME soft list polar decoder also starts with Log-SCL decoding. However, instead of storing all S candidate bit sequences and generated LLRs, the decoder only stores the S candidate bit sequences and the LLRs required to perform the Log-SCL decoding. More specifically, as shown in Fig. 2, when decoding the right-most layer, only $N/2$ LLRs output by the f function are stored so that they can be later input into the g function. Due to the serial nature of the Log-SCL decoder, the same memory can be reused for the calculation of the g function at this layer later for decoding the subsequent bits. Similarly, the decoding in the j-th layer only has to store $N/2^{(m-j)}$ LLRs to perform the f or g function, as in the Log-SCL decoder [4].

Secondly, the most promising candidate bit sequence is re-encoded backwards, as seen in the soft list polar decoder, and the intermediate bit sequence $\hat{\mathbf{u}}_j$ in each layer j is stored. In

the meantime, the input LLRs $\mathbf{L}_m$ are propagated from right to left alongside the code graph for a second time, following the BP schedule [3, 10], which propagates the N LLRs in parallel.

Now that we have the intermediate bits $\hat{\mathbf{u}}_j$ in jth layer, each successive layer of left LLRs can be calculated according to the f and g functions of (2) and (3), respectively, using the results of $\hat{\mathbf{u}}_j$ for the g functions. Only these LLRs are stored in the decoder for the subsequent backwards BP decoding employing (5a) and (5b). This is in contrast to the soft list decoder, which stores all of the N intermediate LLRs generated at each layer. Finally, when the right-most LLRs are obtained, after the backwards BP decoding, an LLR-flip operation may be performed, depending on its consistence with the legitimate encoded bit values, as in the soft list polar decoder.

IV. PERFORMANCE OF SOFT LIST POLAR DECODING

In this section, we quantify the performance of both soft list polar decoders in terms of their BLER, latency, complexity and memory requirement in a (2×2) MIMO system communicating over flat uncorrelated Rayleigh fading channels.

A. BLER performance

We first demonstrate the BLER performance of the proposed soft list polar decoder for a (2×2) MIMO V-BLAST system communicating over fast uncorrelated Rayleigh fading channels, where $I = 8$ outer iterations and Gray-coded QPSK are employed. Note that since the BP, SCAN and the proposed soft list polar decoders rely on the SISO property, the right-hand output after the backward LLR propagation is formed of extrinsic LLRs pertaining to the encoded bits, which can be interleaved and directly input to the MIMO detector for use in the next iteration. However, the conventional Log-SCL polar decoder does not have soft output capability. Hence, only a single outer iteration is employed for the polar coded MIMO system employing a Log-SCL polar decoder.

First, the BLER performance of the soft list polar decoder, BP, SCAN and Log-SCL polar decoders having coding rates of $R = 1/2$ or $1/4$ is compared in Fig. 4. Since the reduction of the memory in the ME soft list polar decoder does not
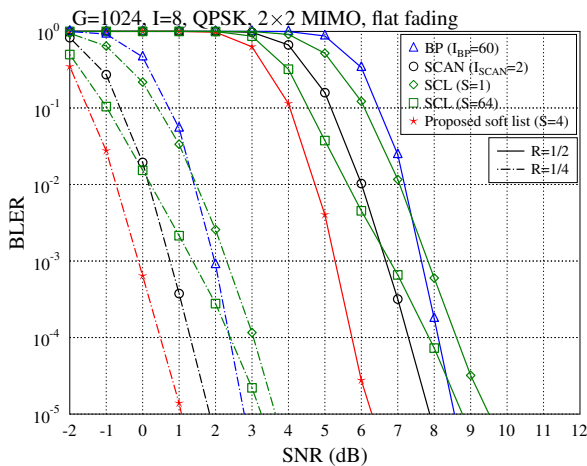

Fig. 4. BLER performance of the BP, SCAN, Log-SCL and soft list polar decoders when decoding the $G = 1024$ polar code with different coding rates using $I = 8$ iterations in a QPSK-modulated (2×2) MIMO system communicating over flat uncorrelated Rayleigh fading channels.
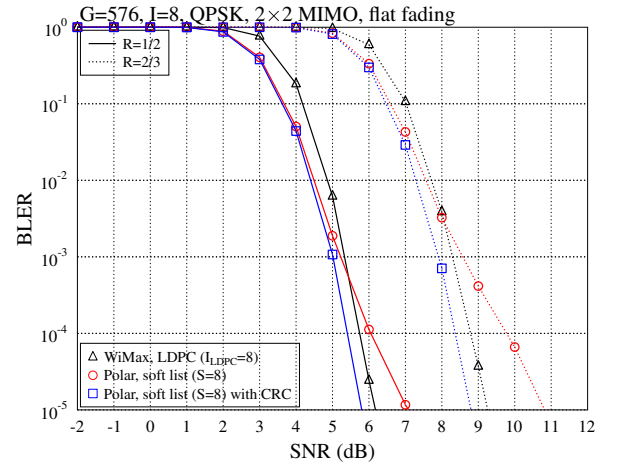

Fig. 5. BLER performance of the $G = 576$-bit polar code employing the proposed soft list decoder and the $G = 576$-bit LDPC code for Mobile WiMAX IEEE 802.16e system [17] with different coding rates using $I = 8$ iterations in a QPSK-modulated (2×2) MIMO system communicating over flat uncorrelated Rayleigh fading channels.

influence the BLER performance, we only consider the BLER performance of the soft list polar decoder in this section. We can see from Fig. 4 that both the SCAN and the proposed soft list polar decoder achieve better BLER performance than the $S = 64$ Log-SCL decoder at both coding rates, since these SISO decoders exploit the iterative soft extrinsic information exchange between the channel decoder, the MIMO detector and the demodulator. Additionally, since the proposed soft list decoder is also capable of extrinsic information exchange between the decoder and the detector, while also harnessing Log-SCL decoding, it achieves the best BLER performance among the four schemes.

Fig. 5 compares the BLER performance of polar codes employing the proposed soft list polar decoder to that of the corresponding LDPC code constructed under the Mobile WiMAX IEEE 802.16e system [21], where the same information block length of $G = 576$ is employed for both the LDPC and polar codes at a code rate of $R = 1/2$ or $2/3$. The 11 CRC bits are appended to the information block, which provide an error detection capability by performing CRC checks to determine whether the S decoded information block candidates are free of error and enhances the error correction capability by rejecting all candidates failing the CRC checks and selecting the remaining candidate having the strongest path metric. Hence, $\log_2 S$ of the 11 CRC bits are used for enhancing the error correction capability, while the remaining $(11 - \log_2 S)$ CRC bits are used for error detection. It may be observed in Fig. 5 that when a short 11-bit CRC is appended to the 565-bit information bits, which gives the same block length of $G = 11 + 565 = 576$, the polar code employing the proposed soft list decoder achieves superior BLER performance compared to the WiMAX LDPC codes. This demonstrates the advantages of polar codes over LDPC codes in the case of short block lengths.

Fig. 6 further investigates the influence of rate matching on the proposed soft list polar decoder, using different rate-matching methods standardized by 3GPP New Radio [2]. To be more specific, Fig. 6 considers an information block
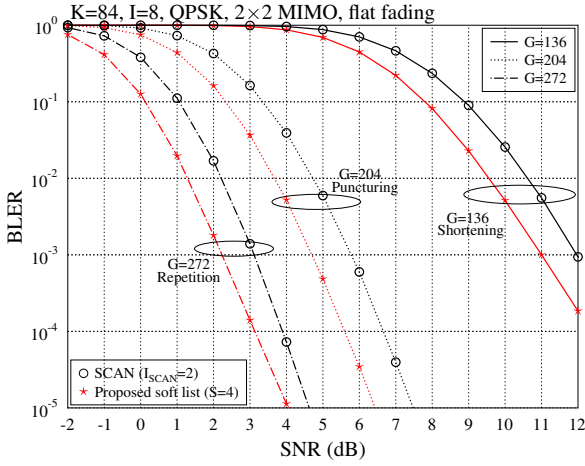
Fig. 6. BLER performance of the SCAN and the proposed soft list polar decoders, when decoding information blocks comprising $K = 84$ bits with different 5G NR rate-matching approaches using $I = 8$ iterations with a QPSK-modulated (2×2) MIMO detector and when communication over flat uncorrelated Rayleigh fading channels.

length of $K = 84$ bits with encoded block lengths of $G = 136, G = 204$ and $G = 272$ bits, which are achieved using shortening, puncturing and repetition, respectively. In each case, $I = 8$ outer iterations are employed between the polar decoder and the QPSK-modulated (2×2) MIMO detector, for communication over a flat uncorrelated Rayleigh fading channel. As expected, a lower coding rate yields superior BLER performance, but in each case, the proposed soft list decoder achieves a gain of about 1 dB at a BLER of 10^{-3} in all three scenarios, when compared to the SCAN decoder.

B. Latency, complexity and memory requirement

Table I compares the latency, computational complexity and memory of the BP, SCAN, soft list and ME soft list polar decoders[1]. to complete one iteration between the decoder and detector.

In Table I, the latency is quantified in terms of the number of Clock Cycles (CCs) required for completing the decoding operations. More specifically, when decoding an N-bit polar code, the BP decoder is required to perform right-to-left as well as left-to-right calculations across the $\log_2 N$ layers of the code graph during each iteration, as shown in Fig. 2. Hence, a total of $2I_{\text{BP}} \log_2 N$ CCs are required for the BP decoder to complete I_{BP} inner iterations. By contrast, the SCAN decoder requires $3N/2^j$ CCs for calculating the output of the j-th $(j > 1)$ layer, of which $N/2^j$ CCs are employed to calculate the backwards LLR propagation shown in Fig. 3, while the remaining $2N/2^j$ CCs are employed to calculate the reverse forward LLR propagation [6]. However, since each LLR $R_{i,0}$ $(i \in [0, N-1])$ that is input to the 1-st layer is initialized as either 0 or ∞, only $N/2$ CCs are required to calculate the backwards LLR propagation. Hence, a total number of $I_{\text{SCAN}}(N/2 + 3\sum_{j=2}^{\log_2 N} 3N/2^j)$ CCs are required to perform I_{SCAN} inner iterations of SCAN decoding. Since $\sum_{j=2}^{\log_2 N} 3N/2^j = N/2 - 1$, the latency requirement

[1]Note that there are impressive research contributions on SCL decoders which reduce the decoding latency, complexity and memory requirement [12–15]

for the SCAN decoder in terms of CCs can be expressed as $I_{\text{SCAN}}(2N - 3)$.

By contrast, both the soft list and the ME soft list polar decoders start with Log-SCL decoding, which requires $2N/2^j$ CCs, when decoding the j-th layer. Note that we assume the calculations of S candidate LLRs can run in parallel and can be completed within the same CC. Following the Log-SCL scheduling, the soft list polar decoder performs only left-to-right BP decoding, using $\log_2 N$ CCs. By contrast, the ME soft list polar decoder performs right-to-left LLR propagation and left-to-right BP decoding, resulting in a further latency requirement of $\log_2 N$ CCs. In total, the soft list and ME soft list polar decoders require $\sum_{j=1}^{\log_2 N} 2N/2^j + \log_2 N$ and $\sum_{j=1}^{\log_2 N} 2N/2^j + 2\log_2 N$ CCs, respectively. Since we have $\sum_{j=1}^{\log_2 N} 2N/2^j = N - 1$, the latency requirement for the soft list and ME soft list polar decoders in terms of CCs can be expressed as $2N - 2 + \log_2 N$ and $2N - 2 + 2\log_2 N$, respectively.

We can see from both the $N = 256$ and $N = 1024$ examples in Table I that our soft list polar decoder imposes only about half of the state-of-the-art SCAN decoder's latency, yet it significantly improves the BLER. However, the low latency of the BP decoder is achieved at the cost of much higher complexity compared to our soft list polar decoders. Note that since the proposed soft list decoder starts with the right-to-left Log-SCL operation, the fast techniques such as Rate-0, Rate-1 and Single-Parity-Check (SPC) nodes that have been proposed in [12–15] can be directly applied to the proposed soft list decoder as well as to the ME soft list decoder.

Additionally, we define the computational complexity as the number of Add, Compare and Select (ACS) arithmetic operations performed by the various decoders. Table I presents the number of operations that were counted during simulations of the various decoding processes. We can see that the original BP decoder suffers from high complexity due to the higher number of iterations required for achieving the BLER of the SCAN and of the soft list polar decoders. Furthermore, our proposed soft list polar decoder imposes the lowest complexity, which is 74% of that of the SCAN decoder. By contrast, the ME soft list polar decoder imposes slightly higher decoding complexity.

Finally, the memory requirement [4] of the different polar decoders is compared in Table I as well, where $b_{\text{LLR}} = 8$ bits are assumed for the fixed-point representation and storage of each LLR. More specifically, the memory required for the BP and SCAN polar decoder to store the LLRs can be expressed as $b_{\text{LLR}}N(\log_2 N + 1)$ and $b_{\text{LLR}}(5N - 2 + \frac{N\log_2 N}{2})$, respectively [6, 10]. In addition to this LLR storage, both the BP and SCAN decoders require an N-bit memory for storing the decoded bits $\hat{\mathbf{u}}_0$.

By contrast, the right-to-left Log-SCL operations of the soft list polar decoder having a list size of S require storage for N LLRs at the input of the decoder and for SN LLRs at the j-th $(j \in [0, \log_2 N - 1])$ layer. Hence, $[(\log_2 N - 1)SN + N]b_{\text{LLR}}$ bit storage is required to store the LLRs which are input to the decoder and generated in all $(\log_2 N - 1)$ layers [17]. Meanwhile, an extra $2SN$

TABLE I
THE LATENCY, COMPUTATIONAL COMPLEXITY AND MEMORY OF THE BP, SCAN, SOFT LIST AND ME SOFT LIST POLAR DECODERS PER ITERATION WITH THE MIMO DETECTOR.

		BP ($I_{\mathrm{BP}}=60$)	SCAN ($I_{\mathrm{SCAN}}=2$)	Soft list ($S=4$)	ME soft list ($S=4$)
Latency (CCs)	General	$2I_{\mathrm{BP}}\log_2 N$	$I_{\mathrm{SCAN}}(2N-3)$	$2N-2+\log_2 N$	$2N-2+2\log_2 N$
	$N=256$	960	1018	518	526
	$N=1024$	1200	4090	2056	2066
Complexity ($\times 10^4$)	$G=256$	49.5	1.6	1.2	1.4
	$G=1024$	245.8	8.2	6.1	7.1
Memory ($b_{\mathrm{LLR}}=8$)	General (bits)	$b_{\mathrm{LLR}}N(\log_2 N+1)+N$ [10]	$b_{\mathrm{LLR}}(5N-2+\frac{N\log_2 N}{2})+N$ [6]	$\max(M_1,M_2)$	$\max(M_2,M_3)$
	$N=256$ (KB)	2.28	2.28	7.5	1.5
	$N=1024$ (KB)	11.13	10.12	38.0	6.0

number of bits are required to store the candidate bits, giving a total of $M_1 = [(\log_2 N - 1)SN + N]b_{\mathrm{LLR}} + 2SN$ bits required for performing the right-to-left Log-SCL. Following this, the right-to-left LLR propagation requires half of the LLR memory of the BP decoder, namely $1/2 b_{\mathrm{LLR}} N(\log_2 N + 1)$ bits, since it employs the pipeline BP scheduling. In addition to this, $2N$-bits of memory is required for storing $\hat{\mathbf{u}}_0$ and $\hat{\mathbf{u}}_m$, giving a total of $M_2 = 1/2 b_{\mathrm{LLR}} N(\log_2 N + 1) + 2N$ bits. Note that the memory employed during right-to-left Log-SCL operations is released before beginning the left-to-right BP decoding. Hence, the memory employed for the soft list decoding can be expressed as $\max(M_1, M_2)$. In the examples shown in Table I, we have $M_1 > M_2$.

By contrast, for the ME soft list polar decoder, the memory reuse technique proposed in [11] is adopted, which subsequently regenerates the $N \log_2 N$ LLRs of the wining candidate, once it has been identified. Hence, the right-to-left Log-SCL operations in the soft list polar decoder require the same amount of memory as the Log-SCL decoder. As demonstrated in [11, 17], when N is sufficient large, the memory requirement of storing LLRs in a Log-SCL decoder having a list size of S can be expressed as $b_{\mathrm{LLR}}(SN + N)$. Together with the LLRs, SN decoded bit candidates and the re-encoded bits must also be stored in the decoder, which requires a further SN-bit memory. In total, $M_3 = b_{\mathrm{LLR}}(SN + N) + 2SN$ bits of memory storage are required for the Log-SCL operations. This memory can be reused for the right-to-left LLR propagation and then the backwards BP decoding. As discussed earlier, the memory required for BP decoding is $M_2 = 1/2 b_{\mathrm{LLR}} N(\log_2 N + 1) + 2N$. Hence, the memory requirement of the ME soft list decoder can be expressed as $\max(M_2, M_3)$. In the examples of Table I, we have $M_3 > M_2$.

From Table I, we can see that the proposed ME soft list polar decoder achieves a memory requirement of only 16% of that of the soft list polar decoder.

V. CONCLUSION

We proposed a pair of SISO polar decoders for employment in the turbo receivers. In the case of decoding a half-rate polar code having a block length of 1024, the proposed ME soft list polar decoder achieves a 1.5 dB BLER performance gain, 50% latency improvement and 26% complexity reduction, compared to the state-of-the-art SCAN polar decoder used in a polar-coded MIMO system. For further reducing the memory requirement, we proposed a ME soft list polar decoder, which requires only 16% of our original soft list polar decoder's memory, and yet maintains the same BLER performance.

REFERENCES

[1] E. Arikan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Transactions on Information Theory*, vol. 55, no. 7, pp. 3051–3073, 2009.

[2] 3GPP TS 38.212 V15.1.1, "NR Multiplexing and channel coding," *3rd Generation Partnership Project Std. 3GPP*, 2018.

[3] E. Arikan, "Polar codes: A pipelined implementation," in *Proc. 4th Int. Symp. on Broad. Commun. ISBC 2010*, pp. 11–14, 2010.

[4] A. Balatsoukas-Stimming, M. B. Parizi, and A. Burg, "LLR-based successive cancellation list decoding of polar codes," *IEEE Transactions on Signal Processing*, vol. 63, no. 19, pp. 5165–5179, 2015.

[5] J. Dai, K. Niu, and J. Lin, "Polar-coded MIMO systems," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 7, pp. 6170–6184, 2018.

[6] U. U. Fayyaz and J. R. Barry, "Low-complexity soft-output decoding of polar codes," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 5, pp. 958–966, 2014.

[7] B. Yuan and K. K. Parhi, "Early stopping criteria for energy-efficient low-latency belief-propagation polar code decoders," *IEEE Transactions on Signal Processing*, vol. 62, no. 24, pp. 6496–6506, 2014.

[8] Z. Babar, Z. B. K. Egilmez, L. Xiang, D. Chandra, R. G. Maunder, S. X. Ng, and L. Hanzo, "Polar codes and their quantum-domain counterparts," *IEEE Communications Surveys & Tutorials*, 2019.

[9] H. Zhou, X. Tan, W. Gross, Z. Zhang, X. You, and C. Zhang, "An improved software list sphere polar decoder with synchronous determination," *IEEE Transactions on Vehicular Technology*, 2019.

[10] J. Sha, X. Liu, Z. Wang, and X. Zeng, "A memory efficient belief propagation decoder for polar codes," *China Communications*, vol. 12, no. 5, pp. 34–41, 2015.

[11] I. Tal and A. Vardy, "List decoding of polar codes," *IEEE Transactions on Information Theory*, vol. 61, no. 5, pp. 2213–2226, 2015.

[12] G. Sarkis, P. Giard, A. Vardy, C. Thibeault, and W. J. Gross, "Fast polar decoders: Algorithm and implementation," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 5, pp. 946–957, 2014.

[13] G. Sarkis, P. Giard, A. Vardy, C. Thibeault, and W. J. Gross, "Fast list decoders for polar codes," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 2, pp. 318–328, 2015.

[14] S. A. Hashemi, C. Condo, F. Ercan, and W. J. Gross, "Memory-efficient polar decoders," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 7, no. 4, pp. 604–615, 2017.

[15] S. A. Hashemi, C. Condo, and W. J. Gross, "Fast and flexible successive-cancellation list decoders for polar codes," *IEEE Transactions on Signal Processing*, vol. 65, no. 21, pp. 5756–5769, 2017.

[16] H. Mahdavifar, M. El-Khamy, J. Lee, and I. Kang, "Polar coding for bit-interleaved coded modulation," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 5, pp. 3115–3127, 2015.

[17] L. Xiang, Z. B. K. Egilmez, R. G. Maunder, and L. Hanzo, "CRC-aided logarithmic stack decoding of polar codes for ultra reliable low latency communication in 3GPP new radio," *IEEE Access*, vol. 7, pp. 28559–28573, 2019.

[18] Y. Shen, J. Yang, X. Zhou, X. You, and C. Zhang, "Joint detection and decoding for polar coded mimo systems," in *GLOBECOM 2017-2017 IEEE Global Communications Conference*, pp. 1–6, IEEE, 2017.

[19] F. Ercan, C. Condo, and W. J. Gross, "Improved bit-flipping algorithm for successive cancellation decoding of polar codes," *IEEE Transactions on Communications*, vol. 67, no. 1, pp. 61–72, 2018.

[20] Y. Yu, Z. Pan, N. Liu, and X. You, "Belief propagation bit-flip decoder for polar codes," *IEEE Access*, vol. 7, pp. 10937–10946, 2019.

[21] G. Masera, F. Quaglio, and F. Vacca, "Implementation of a flexible LDPC decoder," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 54, no. 6, pp. 542–546, 2007.