Abstract—Polar codes are linear block codes that can achieve channel capacity at infinite code length. Successive cancellation list (SCL) decoding relies on a set of parallel decoders; it yields good error-correction performance at finite code length, at the cost of increased implementation complexity and power consumption. Current efforts in literature focus on design-time decoder complexity reduction, while lacking practical run-time power reduction methods. In this work, we propose minimum-effort SCL (ME-SCL) decoding, that allows to determine the parallelism to adopt by performing simple observations on the input of the decoder. This technique guarantees fixed decoding latency and allows hardware SCL decoders to dynamically shut down part of the internal parallelism before each decoding process. It can be combined with existing complexity- and power-reduction techniques. Simulation results show that ME-SCL can reduce the run-time complexity of SCL of up to 50%.

Index Terms—Polar codes, SCL decoding, power reduction

I. INTRODUCTION

Polar codes [1] are linear block codes that are able to achieve the capacity of binary-input memoryless channels at infinite code length. They have been the subject of growing attention and research efforts in the academic and industrial research community, that have striven to improve their finite-length error correction performance, reduce their decoding latency, and design low-complexity, low-power decoder implementations. Thanks to the outstanding results obtained in the last decade, polar codes have been included in the 3GPP 5th generation wireless systems standard (5G) [2], and have made their way towards optical communications [3], [4].

Successive cancellation list (SCL) decoding has been proposed in [5] to improve on the error-correction performance of polar codes provided by their original decoding algorithm, successive cancellation (SC). It relies on a list of parallel SC decoders, each making different decoding choices; with its countless evolutions and ameliorations, it is considered the academic and industrial standard for polar code decoding. Unfortunately, the improved performance of SCL decoding comes at an increased implementation cost. Various techniques are available in literature to reduce the complexity and power consumption of SCL decoders. Design-time approaches like [6–8] modify the structure of the basic SCL decoder, reducing its implementation complexity. However, at the latest technology nodes, dynamic power dominates the total power consumption: it is thus of paramount importance to combine efficient design to run-time power-reduction techniques. The adaptive SCL decoder described in [9] can potentially reduce the average power consumption by performing sequential decoding attempts with increasing list size. However, it introduces variable decoding latency, an undesired feature in any practical decoder implementation, that needs to be timed according to the maximum latency possible.

In this work, we propose minimum-effort SCL (ME-SCL), a technique that allows to decide which list size to adopt by observing the input of the decoder. This technique guarantees fixed decoding latency and allows hardware SCL decoders to dynamically decrease the list size before each decoding by shutting down part of the internal parallelism. It relies on simple operations that have negligible implementation cost, and it can be stacked with any other complexity and power reduction technique. Simulation results over a wide set of code parameters show that ME-SCL can reduce the run-time complexity of SCL of up to 50%.

II. PRELIMINARIES

A polar code of length \( N \) relies on a transformation matrix \( T_N = T_2^{\otimes n} \), generated by the \( n \)-fold Kronecker product of a basic channel transformation kernel \( T_2 \). The resulting \( N \) bit-channels are polarized, and vary from completely noisy to completely noiseless. A polar code of length \( N \) and rate \( \hat{R} = K/N \) is constructed by creating an input vector \( u \) where the \( K \) message bits are assigned to the entries of \( u \) corresponding to the \( K \) most reliable bit-channels. The remaining entries of \( u \) are “frozen” bits, set to zero. The codeword \( x \) is then calculated as \( x = u \cdot T_N \) and transmitted.

SC decoding was introduced in [1]. It can be represented as a depth-first binary tree search with priority to the left branch. The logarithmic likelihood ratio (LLR) vector \( y \) is received from the channel and assigned to the root node: the LLRs are propagated, through node operations, downward towards the leaf nodes, each associated to an entry of the estimated input vector \( \hat{u} \). Bit values at leaf nodes are either known (in case of frozen bits) or estimated based on the sign of the incoming LLR. Bit estimations are propagated upward in the tree and combined with the descending LLRs until all leaves have been explored.

While optimal at infinite code length, SC decoding has mediocre error-correction performance at moderate code length. To improve it, a list-based decoder has been proposed in [5]. The SCL decoder relies on \( L \) parallel SC decoders, each storing a different partial decoded bits vector, called path. Every time an information leaf node is reached, each decoder splits the current decoding path, estimating the bit as 0 in one case and as 1 in the other, doubling the number of parallel decoding paths. A path metric allows to maintain only the \( L \) more likely paths, while the \( L \) less likely are discarded. The decoding continues until the last leaf node has been reached, and one as the surviving decoding paths is selected as the
TABLE I: Percentage of cases for $L_{\text{low}}$ requirement, $L_{\text{high}} = 32$.

| $L_{\text{low}}$ | $N = 128$ | $N = 256$ | $N = 512$ |
|-----------------|-----------|-----------|-----------|
|                 | $R = 1/2$ | $R = 7/8$ | $R = 1/2$ | $R = 7/8$ | $R = 1/2$ | $R = 7/8$ |
| 1               | 96.87%    | 98.60%    | 84.61%    | 94.04%    | 84.87%    | 88.78%    |
| 2               | 2.52%     | 1.10%     | 11.44%    | 3.96%     | 10.67%    | 7.89%     |
| 4               | 0.45%     | 0.23%     | 2.85%     | 1.42%     | 2.99%     | 2.28%     |
| 8               | 0.12%     | 0.05%     | 0.76%     | 0.40%     | 1.00%     | 0.65%     |
| 16              | 0.03%     | 0.02%     | 0.23%     | 0.15%     | 0.34%     | 0.28%     |
| 32              | 0.01%     | <0.01%    | 0.11%     | 0.03%     | 0.13%     | 0.12%     |

TABLE II: Average $\text{max\_err\_LLR}$ and $\text{no\_small\_LLR}$ for $N = 128$, $R = 1/2$, $C = 2$, $E_b/N_0 = 3.25$.

| $L_{\text{low}}$ | average max\_err\_LLR | average no\_small\_LLR |
|-----------------|------------------------|------------------------|
| 1               | 3.28                   | 48.40                  |
| 2               | 4.22                   | 66.86                  |
| 4               | 4.46                   | 71.74                  |
| 8               | 4.80                   | 79.54                  |
| 16              | 5.43                   | 85.43                  |
| 32              | 5.37                   | 87.50                  |

decoder output. The concatenation of polar codes with a cyclic redundancy check (CRC) of length $C$ has been proposed in [10] to help the final decoder output path selection, showing substantial error-correction performance improvement.

III. MINIMUM-EFFORT SCL DECODING

Given a polar code of code length $N$ and rate $R$, and a codeword transmitted through an additive white Gaussian noise (AWGN) channel characterized by a certain $E_b/N_0$, every received vector that can be successfully decoded through SCL with $L = L_{\text{high}}$, can be decoded by SCL with $L = L_{\text{low}} \leq L_{\text{high}}$. Table I reports the percentage of cases for which a particular $L_{\text{low}}$ value is needed to correctly decode a received vector, for various combinations of $N$, $R$ and $C$, at a block error rate (BLER) of approximately $10^{-3}$, and $L_{\text{high}} = 32$. Such a BLER is a realistic working point for many wireless applications, and for component codes in more powerful concatenated coding schemes targeting optical communications. Simulation results consider $10^6$ frames. It can be seen that the vast majority of cases does not need $L = 32$ to be correctly decoded, with the percentage rising as $N$ decreases and $R$ increases. This is also due to the fact that the required list size strongly depends on the input noise level. Given a BLER target, different combinations of code parameters will be able to achieve it at different $E_b/N_0$: in general, a higher $E_b/N_0$ will be needed for shorter, higher-rate codes.

Consequently, given the unbalanced $L_{\text{low}}$ requirements, a decoder implementation with fixed list size performs a large amount of unnecessary operations and memory accesses, resulting in power consumption that is ultimately wasted. It would be advantageous to identify the required $L_{\text{low}}$ before the start of the decoding process, so that the list size can be reduced accordingly, effectively dividing the power consumption by up to a factor $L_{\text{high}}$. To attempt to do so, it is necessary to rely on the only information available before the decoding, i.e. the channel LLRs. Through simulation, together with the $L_{\text{low}}$ requirements, we have studied the magnitudes of the channel LLRs associated to bits that have been flipped by channel-induced noise at a given $E_b/N_0$. In particular, for each received vector $y$ containing errors, we have observed:

- $\text{max\_err\_LLR}$, i.e. the largest LLR magnitude associated to an erroneous bit;
- $\text{no\_small\_LLRs}$, i.e. the number of channel LLRs with magnitude smaller or equal than $\text{max\_err\_LLR}$.

Since the channel-induced noise on the transmitted bits is modeled as i.i.d. random variables, there are great variations in both $\text{max\_err\_LLR}$ and $\text{no\_small\_LLRs}$ even when considering fixed code parameters and stable channel conditions. However, by averaging $\text{max\_err\_LLR}$ and $\text{no\_small\_LLRs}$ over a high number of instances for each $L_{\text{low}}$, it is possible to see that both quantities tend to rise as $L_{\text{low}}$ increases, as shown in Table II for the case of $N = 128$, $R = 1/2$, $C = 2$, $E_b/N_0 = 3.25$. While $L_{\text{low}}$ is shown to correlate to the magnitudes of channel LLRs, it has been instead observed that the required list size has a high degree of independence from the number of erroneous LLRs. This is due to the nature of SC decoding: even a high number of wrong bits can be corrected if the magnitude of the associated LLRs is low enough.

Based on these results, we propose two embodiments of minimum-effort SCL decoding (ME-SCL), a technique to dynamically reduce the active list size of SCL decoders before each decoding, that relies on simple observations on the channel LLRs. This technique can be combined with existing complexity- and latency reduction- methods [6], [8], [11], [12].

A. Single-layer ME-SCL

The first of the two proposed techniques is the single-layer ME-SCL, as two thresholds $\gamma$ and $\phi$ are used to identify a single $L_{\text{low}}$ as an alternative to $L_{\text{high}}$. The channel LLRs that are lower or equal than $\gamma$ are counted; if their number is lower than $\phi$, then the decoding is attempted through SCL with $L_{\text{low}}$, otherwise $L_{\text{high}}$ is used. Given a combination of code parameters and $E_b/N_0$, the single-layer ME-SCL requires the selection of $L_{\text{low}}$ and of both $\gamma$ and $\phi$, to minimize the decoding complexity in dependence of a target BLER.

An immediate way to exploit the correlation between $L_{\text{low}}$, $\text{max\_err\_LLR}$, and $\text{no\_small\_LLRs}$ would be to set $\gamma$ equal to the average $\text{max\_err\_LLR}$ for the considered $E_b/N_0$, and $\phi$ equal to the average $\text{no\_small\_LLRs}$. However, we have to take in account that the inherently random channel noise and the consequent variations in the correct and erroneous LLRs prevent from accurately identifying the exact
of active paths with respect to the total, is computed as

\[ C = \frac{\gamma}{\phi} \]

Figure 1, the following combinations of thresholds show better performance: for \( \gamma = 0.5 \) and \( \phi < 0 \), \( \gamma = 1.0 \) and \( \phi < 8 \), \( \gamma = 1.5 \) and \( \phi < 15 \). The lowest complexity among these thresholds is 88.17%, achieved with \( \gamma = 1.5 \) and \( \phi = 14 \). The difference between the BLER obtained with SCL and ME-SCL expresses the misidentification rate, i.e. the normalized number of blocks where \( L_{low} \) was selected and that resulted in a decoding error, while correct decoding could be achieved with \( L_{high} \).

The threshold selection process is repeated for all the possible \( L_{low} < L_{high} \), so that the \( L_{low} \) that minimizes the complexity is selected to be used in single-layer ME-SCL.

B. Multi-layer ME-SCL

The single-layer ME-SCL selects the best \( L_{low} \) to minimize complexity. However, by considering multiple sets of \( \gamma \) and \( \phi \), each targeting a different \( L_{low} \), it is possible to partially combine the complexity reduction effects of various single-layer ME-SCL. In this technique, that we call multi-layer ME-SCL, the decision conditions for increasing \( L_{low} \) are checked sequentially until one is met, or until \( L_{low} = L_{high} \), as shown in Algorithm 1.

As each threshold couple misidentifies a potentially different set of cases, the BLER degradation caused by each \( L_{low} \) is partially accumulated, leading to unacceptable performance. For this reason, the optimal thresholds identified for each \( L_{low} \) need to be decreased. For multi-layer ME-SCL, complexity is computed as

\[ 100 \cdot \frac{\delta}{L_{high}} \cdot \frac{L_{high} + (1 - \delta) \cdot L_{high}}{L_{high}} \]

where \( \delta \) is the fraction of times that \( L_{low} \) is chosen over \( L_{high} \). The optimal combination of \( \gamma \) and \( \phi \) is the one that minimizes the complexity given a target maximum BLER. For instance, if the maximum acceptable BLER=2 \cdot 10^{-3} in Figure 1, the following combinations of thresholds show better performance: for \( \gamma = 0.5 \), \( \phi < 3 \); for \( \gamma = 1.0 \), \( \phi < 8 \); for \( \gamma = 1.5 \), \( \phi < 15 \). The lowest complexity among these thresholds is 88.17%, achieved with \( \gamma = 1.5 \) and \( \phi = 14 \). The difference between the BLER obtained with SCL and ME-SCL expresses the misidentification rate, i.e. the normalized number of blocks where \( L_{low} \) was selected and that resulted in a decoding error, while correct decoding could be achieved with \( L_{high} \).

The threshold selection process is repeated for all the possible \( L_{low} < L_{high} \), so that the \( L_{low} \) that minimizes the complexity is selected to be used in single-layer ME-SCL.

Algorithm 1: Multi-layer ME-SCL

1: for \( i = 0 \ldots \log_2(L_{high}) - 1 \) do
2: no_small_LLRs = 0
3: for \( j = 0 \ldots N - 1 \) do
4: if \( y[j] \geq \gamma \) then
5: no_small_LLRs ++
6: end if
7: end for
8: if no_small_LLRs < \( \phi \) then
9: \( L_{low} = 2^i \)
10: return \( \hat{u} = \text{SCL decoding}(y) \)
11: end if
12: end for
13: \( L_{low} = L_{high} \)
14: return \( \hat{u} = \text{SCL decoding}(y) \)

\[ 100 \cdot \sum_{i=0}^{\log_2 L_{high}} \delta_i \cdot \frac{2^i}{L_{high}} \leq L_{high} \]

where \( \delta_i \) is the fraction of times that \( L_{low} = 2^i \leq L_{high} \) is selected.

IV. RESULTS

In this Section, simulation results for ME-SCL are reported, considering an AWGN channel and binary phase shift keying modulation. Channel LLRs have been computed as \( y = 2/\sigma^2 \cdot (1 - 2x + E) \), where \( \sigma \) is the standard deviation of the channel noise distribution, and the random variable \( E \sim \mathcal{N}(0, \sigma^2) \). Polar codes have been constructed through density evolution with Gaussian approximation with \( \sigma = 0.5 \). The presented results concern the combinations of \( N, R \) and \( C \) already shown in Table I. For each code parameter combination, Table III reports the selected \( L_{high} \) (both single- and multi-layer ME-SCL), the optimal \( L_{low} \) for single-layer ME-SCL, the \( E_b/N_0 \) at which the thresholds have been optimized (BLER≈ 10^{-3}), and the value of \( \gamma \) and \( \phi \). The target maximum BLER has been chosen as the mid-point between the BLER of SCL with \( L_{high} \) and with \( L_{low} \). The majority of results have been obtained with \( L_{high} = 32 \). However, with \( N = 128, R = 1/2 \) there is very little difference between the BLER of SCL with \( L = 16 \) and \( L = 32 \); consequently, ME-SCL uses \( L_{high} = 16 \). In the same way, for \( N = 128, R = 7/8 \) the BLER of SCL with \( L = 8, L = 16 \), and \( L = 32 \) is almost the same at the considered
Similarly, thus, $N = 128$ thresholds for the associated layer ME-SCL. Each point in the graph uses the optimal $N$ served codes at the threshold optimization decoder incurs a higher misidentification rate. Since thresholds have to be optimized for a given $\gamma$ LLRs are below the $N_{high}$ to be very restrictive, and a higher percentage of vectors is the converges to the BLER of SCL with $L$. ME-SCL thus uses $N_{high} = 8$. In the figures, ME-SCL with $L_{low} = a$ and $L_{high} = b$ is labeled as $L = a, b$.

Figure 2 and Figure 3 plot the BLER for the three considered code lengths, for $R = 1/2$ and $R = 7/8$, respectively. Both single-layer ME-SCL and standard SCL are portrayed. It can be observed that the BLER of ME-SCL follows closely that of SCL with $L_{high}$, starting to diverge when closer to the $E_b/N_0$ for which the thresholds were optimized, and converges to the BLER of SCL with $L_{low}$ at higher $E_b/N_0$. At low $E_b/N_0$, channel LLRs have smaller magnitude: $\phi$ tends to be very restrictive, and a higher percentage of vectors is decoded with $L_{high}$. In the same way, at high $E_b/N_0$ fewer LLRs are below the $\gamma$ threshold, and $L_{low}$ is prioritized; since thresholds have to be optimized for a given $E_b/N_0$, the decoder incurs a higher misidentification rate.

Figure 4 portrays the complexity percentage for the observed codes at the threshold optimization $E_b/N_0$, with single-layer ME-SCL. Each point in the graph uses the optimal thresholds for the associated $L_{low}$. As noted earlier in this section, for $N = 128$, $R = 1/2$, the selected $L_{high} = 16$; thus, $L_{low} = 16$ constitutes the 100% complexity mark. Similarly, $L_{low} = 8$ represents 100% complexity in case of $N = 128$, $R = 7/8$. It can be seen that in all observed cases, $L_{low} = L_{high}/2$ achieves the lowest decoding complexity. As the gap between $L_{high}$ and $L_{low}$ increases, the chances of misidentification rise as well, and stricter thresholds need to be used to remain within the target BLER. This leads to a

| $E_b/N_0$ | $N = 128$ | $N = 256$ | $N = 512$ |
|-----------|-----------|-----------|-----------|
| $R = 1/2$ | $C = 2$   | $R = 7/8$ C = 3 | $R = 1/2$ | $R = 7/8$ C = 8 | $R = 1/2$ | $R = 7/8$ C = 12 |
| $E_b/N_0$ | $L_{high}$ | $L_{low}$ | $\gamma$ | $\phi$ | $E_b/N_0$ | $L_{high}$ | $L_{low}$ | $\gamma$ | $\phi$ | $E_b/N_0$ | $L_{high}$ | $L_{low}$ | $\gamma$ | $\phi$ |
| 3.25 | 16 | 8 | 4.0 | 65 | 2.60 | 4.45 | 512 ME-SCL | L = 8, 16 |
| 5.30 | 8 | 4 | 4.0 | 9 | 2.60 | 4.45 | 512 ME-SCL | L = 16, 32 |
| 2.40 | 16 | 16 | 4.45 | 285 | 3.25 | 4.0 | 2.60 | 4.45 |
| 4.85 | 40 | 4.45 | 285 | 63 | 2.60 | 4.45 | 512 ME-SCL | L = 16 |
| 2.60 | 32 | 32 | 4.45 | 285 | 3.25 | 4.0 | 2.60 | 4.45 |
| 2.40 | 32 | 32 | 4.45 | 285 | 3.25 | 4.0 | 2.60 | 4.45 |
Multi-layer ME-SCL combines the complexity reduction potential of multiple single-layer thresholds. As detailed in Section III-B, the thresholds for each layer have been decreased so that the BLER of multi-layer ME-SCL matches that of single-layer ME-SCL shown in Figure 2-3. In Figure 6, the complexity of multi-layer and single-layer ME-SCL are then compared under the conditions depicted in Table III. It can be observed that multi-layer ME-SCL brings 1%-8% additional decoding complexity reduction with respect to single-layer ME-SCL.

V. CONCLUSION

In this work we have introduced minimum-effort SCL (ME-SCL) decoding of polar codes, a technique to select the list size of SCL decoders by observing the distribution of channel LLRs. It can be used as a power-reduction technique in hardware SCL decoders, allowing to by dynamically decrease the list size before each decoding, and thus deactivating part of the internal parallelism. ME-SCL is based on simple, implementation-friendly threshold comparisons, and can be combined with existing complexity- and latency-reduction techniques. Two embodiments have been proposed, and shown to reduce the decoding complexity of SCL decoding of almost 50% while meeting a given error-correction performance target.

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, July 2009.
[2] 3rd Generation Partnership Project (3GPP), “Multiplexing and channel coding,” 3GPP TR 38.212 V15.3.0, 2018.
[3] T. Mehmood, M. P. Yankov, A. Fisker, K. Gormsen, and S. Forchhammer, “Rate-adaptive concatenated polar-staircase codes for data center interconnects,” in *Optical Fiber Communication Conference (OFC) 2020*, 2020, p. Th11.6, Optical Society of America.
[4] C. Condo, V. Bioglio, H. Hafermann, and I. Land, “Practical product code construction of polar codes,” *IEEE Transactions on Signal Processing*, vol. 68, pp. 2004–2014, 2020.
[5] I. Tal and A. Vardy, “List decoding of polar codes,” *IEEE Transactions on Information Theory*, vol. 61, no. 5, pp. 2213–2226, May 2015.
[6] S. A. Hashemi, A. Balatsoukas-Stimming, P. Giard, C. Thibeault, and W. J. Gross, “Partitioned successive-cancellation list decoding of polar codes,” in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 957–960.
[7] S. A. Hashemi, M. Mondelli, S. H. Hassani, C. Condo, R. L. Urbanke, and W. J. Gross, “Decoder partitioning: Towards practical list decoding of polar codes,” *IEEE Transactions on Communications*, vol. 66, no. 9, pp. 3749–3759, 2018.
[8] W. Song, Y. Fu, Q. Chen, L. Li, and C. Zhang, “ANN based adaptive successive cancellation list decoder for polar codes,” in *2019 IEEE 13th International Conference on ASIC (ASICON)*, 2019, pp. 1–4.
[9] B. Li, H. Shen, and D. Tse, “An adaptive successive cancellation list decoder for polar codes with cyclic redundancy check,” *IEEE Communications Letters*, vol. 16, no. 12, pp. 2044–2047, 2012.
[10] K. Niu and K. Chen, “CRC-aided decoding of polar codes,” *IEEE Communications Letters*, vol. 16, no. 10, pp. 1668–1671, October 2012.
[11] C. Condo, V. Bioglio, and I. Land, “Generalized fast decoding of polar codes,” in *2018 IEEE Global Communications Conference (GLOBECOM)*, 2018, pp. 1–6.
[12] M. H. Ardakani, M. Hanif, M. Ardakani, and C. Tellambura, “Fast successive-cancellation-based decoders of polar codes,” *IEEE Transactions on Communications*, vol. 67, no. 7, pp. 4562–4574, 2019.
[13] P. Trifonov, “Efficient design and decoding of polar codes,” *IEEE Transactions on Communications*, vol. 60, no. 11, pp. 3221–3227, 2012.