Input-distribution-aware parallel decoding of block codes

Carlo Condo, Alex Nicolescu
Infinera Corporation

Abstract—Many channel decoders rely on parallel decoding attempts to achieve good performance with acceptable latency. However, most of the time fewer attempts than the foreseen maximum are sufficient for successful decoding. Input-distribution-aware (IDA) decoding allows to determine the parallelism of polar code list decoders by observing the distribution of channel information. In this work, IDA decoding is proven to be effective with different codes and decoding algorithms as well. Two techniques, M-IDA and MD-IDA, are proposed: they exploit the sampling of the input distribution inherent to particular decoding algorithms to perform low-cost IDA decoding. Simulation results on the decoding of BCH codes via the Chase and ORBGRAYD algorithms to perform low-cost IDA decoding. Simulation results on the decoding of BCH codes via the Chase and ORBGRAYD algorithms show that they perform at least as well as the original IDA decoding, allowing to reduce run-time complexity down to 17% and 67% with minimal error correction degradation.

I. INTRODUCTION

Forward error correction (FEC) is a critical component of modern communication and storage systems. Efficient implementation of FEC decoders is of paramount importance in modern applications, given the high computational cost of decoding powerful channel codes and the growing emphasis on low-power technology.

A large number of high performance FEC schemes and decoding algorithms rely on a trial and evaluation loop, in which the decoding is attempted under some conditions, and in case of failure, the decoding conditions are changed to give the decoder better chances of succeeding. For example, fountain codes can generate extra parity symbols [1], polar code flip decoders sequentially change the decoding decision on low-reliability bits [2], and low-density parity-check code decoders can inject noise and perform additional iterations to escape from decoding failure [3]. These techniques, while powerful, incur variable and possibly very long latency, something that is not always sustainable in practical systems.

To avoid such drawback, many decoding algorithms for block codes are based on parallel decoding paths, each taking different decisions or working on a modified version of the information received from the channel [4]–[7]. The higher the number of paths, the more the decoding process is able to cater to complex, less probable noise patterns. This means that most noise occurrences can be corrected with very few paths. It would be beneficial for practical decoders to be able to identify such cases, and to deactivate some of the parallel paths to reduce power consumption.

In [8], input-distribution-aware (IDA) successive cancellation list (SCL) decoding of polar codes was proposed, a technique that allows to decide which list size to adopt by observing the input of the decoder. This technique guarantees fixed, short decoding latency and allows hardware SCL decoders to dynamically decrease the list size before each decoding by shutting down part of the internal parallelism.

In this work, the generality of IDA decoding is proven by evaluating its effectiveness on different types of codes and decoding algorithms. Two low-complexity incarnations of IDA decoding are proposed. These rely on inherent characteristics of decoding algorithms to decrease the implementation complexity of IDA without impacting its performance.

II. PRELIMINARIES

A. Definitions

In this work, it is assumed that the decoder receives soft values from the channel in the form of logarithmic likelihood ratios (LLRs). Let us call the received LLR vector $y$, and $\hat{x}$ the vector of hard decisions taken on $y$:

$$\hat{x}_i = \begin{cases} 0, & \text{when } y_i \geq 0; \\ 1, & \text{otherwise.} \end{cases}$$

Moreover, let us define as $\tilde{y}$ the received vector $y$ sorted in ascending order of absolute value. Analysis and simulations all consider additive white Gaussian noise (AWGN) channel with binary phase shift keying (BPSK) modulation, for which LLRs can be computed as $y = 2/\sigma^2 \cdot (1 - 2x + E)$, where $\sigma$ is the standard deviation of the noise distribution, and the random variable $E \sim N(0, \sigma^2)$ has equiprobable sign and magnitude that increases with decreasing probability.

Finally, throughout the paper, “correct decoding” and “successful decoding” refer to the event where a decoder corrects the errors accrued during transmission, and is able to successfully retrieve the information bits that were encoded at the transmitter side.

B. Previous work

In [8], input-distribution-aware (IDA) successive cancellation list (SCL) decoding of polar codes was proposed. It relies on the observation of the link between the distribution of soft values received from the channel, usually LLRs, and the number of concurrent decoding paths employed by SCL to achieve correct decoding. Figure 1, taken from [8], shows that the average distribution of channel LLRs $y$ for a polar code of length $N = 256$ changes with the number of parallel decoding attempts $L$ necessary to correctly decode it. It was shown that the parallelism $L_{\text{high}}$ of a decoder can be reduced on-the-fly to $L_{\text{low}} < L_{\text{high}}$, with a limited cost in block error rate (BLER)
Fig. 1: LLR distribution for a polar code with $N=256$ at $E_b/N_0=2.4$ dB with $L$ necessary decoding paths.

and substantial complexity reduction. The channel LLRs that are lower or equal than a threshold $\gamma$ are counted; if their number is lower than a threshold $\phi$, then the decoding is attempted with $L_{low}$, otherwise $L_{high}$ is used. This operation can be implemented at a low cost in practical decoders. The effectiveness of IDA decoding was proven over a variety of code lengths, rates and channel conditions. The run-time complexity of such a decoder, intended as the percentage of active decoding paths with respect to the total, is computed as

$$100 \cdot \frac{\delta \cdot L_{low} + (1 - \delta) \cdot L_{high}}{L_{high}},$$

where $\delta$ is the fraction of times that $L_{low}$ is chosen over $L_{high}$.

III. IDA DECODING

IDA decoding has been applied to SCL decoding of polar codes so far; however, it is a flexible technique, and it is not exclusive to polar codes or SCL decoding. Any decoder using parallel decoding paths at the end of which a decoding candidate is chosen, can benefit from the IDA approach. As an example, let us consider Bose-Chaudhuri-Hocquenghem (BCH) codes, that are polynomial block codes widely used in communications, storage and cryptography applications. A BCH code of length $N$, with $K$ information bits, capable of correcting $t$ errors, is identified as $BCH(N,K,t)$. One of the most successful and widespread decoding algorithms for such codes is the Chase decoder [6]. Chase decoding takes the hard decision of the input LLRs, but it also identifies the $P$ bits associated to the LLRs with the smallest magnitude, i.e. $\tilde{y}_i$ with $0 \leq i < P$. It then creates the set of $2^P$ test patterns where all combinations of these bits are flipped. The resulting $2^P$ vectors are then decoded, in parallel, through bounded distance decoding. However, for a decoder that considers $P_{high}$ LLRs, the vast majority of the decoding processes can be successfully carried out with $P_{low} < P_{high}$. This can be seen in Table I that reports the percentage of cases for which a specific $P_{low}$ is sufficient for successful decoding, given $P_{high}=6$ at $E_b/N_0=6.5$ dB and a BCH$(255,239,2)$ with generator polynomial $g_C(x)=0x18DE6$ defined on GF$(2^8)$ with a field generator polynomial $g_F(x)=0x171$. Chase decoding does not change the guaranteed $t$-error correction capability of BCH codes, but can potentially correct up to $P$ additional errors, if they are associated with the least reliable LLRs. The distribution Table I is thus dependent on the number of errors in the codeword and on their average LLR. At lower $E_b/N_0$, higher $P_{low}$ will be necessary with increased probability.

Similar uneven requirement distributions can be observed with other code types and decoding algorithms; without loss of generality, the remainder of this work uses BCH codes and considers the aforementioned Chase decoder and the recently proposed ordered reliability bits guessing random additive noise decoding (ORBGRAND).

A. M-IDA and MD-IDA decoding

Observing the distribution of input LLRs is a general approach that allows to decrease the parallelism of many decoders. Some decoding algorithms, however, inherently perform a sampling of the input distribution, and perform parallel decoding on the basis of a subset of the input. This feature can be exploited to enable IDA decoding at a lower complexity cost than the observation of the full channel LLR distribution. It is the case of Chase decoding, that has been addressed earlier in this Section, and ORBGRAND [7], a universal decoder that evolves from GRAND [9]. ORBGRAND attempts to guess the noise vector that was applied during transmission, by considering nPat bit-flipping patterns and applying them to $\hat{x}$. Bit flipping patterns are determined offline based on their logistic weight $w_i$, i.e the sum of the indices of flipped bits. For example, the pattern flipping the least reliable bit has $w = 1$, the one flipping the second least reliable has weight $w = 2$, the one flipping both bits has weight $w = 1 + 2 = 3$, and so on. The nPat patterns with the smallest weight are used at decoding time, after $\tilde{y}$ is available and the actual positions of the least reliable bits in the given $\hat{x}$ is determined [7].

Both Chase and ORBGRAND decoders perform parallel decoding attempts based on a subset of LLRs with low reliability, whose cardinality is $P$ in case of Chase, and $w_nPat$ in case of ORBGRAND. This subset constitutes a sensitive sample of the total channel LLR distribution, and a clear link can be observed between the first $P$ entries of $\tilde{y}$ and the degree of parallelism required for correct decoding, much like in Fig. 1. Figure 2 plots the distribution of the average $|\tilde{y}|$ as the required $P_{low}$ for correct decoding changes in

| $P_{low}$ | Requirement % |
|-----------|---------------|
| 0         | 96.54%        |
| 1         | 2.55%         |
| 2         | 0.66%         |
| 3         | 0.18%         |
| 4         | 0.05%         |
| 5         | 0.01%         |
| 6         | < 0.01%       |
case of BCH(255,239,2). Similar observations can be made for ORBGRAND in relation to the required \( w_{\text{nPat}_{\text{low}}} \), and consequently \( \text{nPat}_{\text{low}} \). Based on this trend, it is possible to devise a simplified IDA decoding that relies on the observation of one of the entries of \( \tilde{y} \) only. To decide which one, it can be seen that \( |\tilde{y}| \) tends to decrease as \( P_{\text{low}} \) increases, until \( P_{\text{low}} = i \), whereas \( |\tilde{y}| \) starts to increase. This is due to the fact that since most correctable error events are due to errors associated with low-reliability LLRs, a decoder requiring \( P_{\text{low}} \) for successful decoding will be likely to observe a clustering of small LLRs \( |\tilde{y}| \) with \( i < P_{\text{low}} \) only. If \( |\tilde{y}| \) with \( i > P_{\text{low}} \) were small as well, it would be less likely for the errors to be clustered in the first \( P_{\text{low}} \) entries of \( |\tilde{y}| \). Consequently, to decide between \( P_{\text{low}} \) and \( P_{\text{high}} \), the most stable choice is to observe \( |\tilde{y}|_{\text{P}_{\text{high}} - 1} \). The same is true in case of \( w_{\text{nPat}} \) for ORBGRAND. This new approach to IDA decoding is called M-IDA, being based on the magnitude of a single LLR. If \( |\tilde{y}|_{\text{P}_{\text{high}} - 1} \) is higher than a threshold \( \gamma_M \), then the decoding is attempted with \( P_{\text{low}} \) (\( \text{nPat}_{\text{low}} \)), otherwise \( P_{\text{high}} \) (\( \text{nPat}_{\text{high}} \)) is used.

Following the same reasoning, an alternative metric to identify the necessary degree of parallelism is the difference between two elements of \( |\tilde{y}| \). It can give a measure of how clustered a set of LLRs are, and thus how likely it is that the decoder requires a high number of decoding attempts to succeed. Figure 3 plots the average difference between \( |\tilde{y}|_{i} \), \( 1 \leq i \leq 5 \) and \( |\tilde{y}|_{0} \), given \( P_{\text{low}} \), BCH(255,239,2), \( E_b/N_0 = 6.5 \).

As Chase decoding requires all the \( 2^P \) attempts to be completed to be able to select the best candidate, both fully parallel and partially parallel implementations can benefit from IDA decoding [10]. On the other hand, test patterns in ORBGRAND are associated to a weight known at design time. Consequently, if they are applied in ascending order of weight, the algorithm can be stopped as soon as a valid codeword is obtained. However, ORBGRAND relies on a high number of simple operations; this approach would lead to extremely long worst case latency. While currently no implementations of ORBGRAND have been published, the only hardware GRAND decoder [12] relies on a degree of parallelism of \( N \), increasing the Hamming weight of the applied test patterns over subsequent time steps. With some modification to cater to logistic weights, the same concept can be applied to ORBGRAND, and such an architecture can exploit IDA decoding to reduce its power consumption.

### IV. Results

In this Section are reported simulation results evaluating the effectiveness of the proposed techniques. BLER and complexity [2] are computed on the BCH(255,239,2) code defined in Section III under the conditions presented in Section I-A.

IDA decoding, as introduced at the beginning of Section III is applied to Chase decoding in Figure 4 for \( P_{\text{high}} = 5 \) and \( P_{\text{low}} = \{3,4\} \). Thresholds are optimized for \( E_b/N_0 = 6.5 \text{dB} \), leading to \( \gamma = \{4.5,7.5\} \) and \( \phi = \{7,12\} \). As expected, the results in [8] transfer to other types of codes and decoding algorithms. IDA-Chase can guarantee significant complexity reduction at a tunable BLER cost: at \( E_b/N_0 = 6.5 \text{dB} \), \( P_{\text{low}} = 4 \) reduces the complexity to 82.5% and \( P_{\text{low}} = 3 \) to 30.5% with negligible BLER degradation with respect to \( P = 5 \) and \( P = 4 \), respectively. Moving away from the \( E_b/N_0 \) at which \( \gamma \) and \( \phi \) have been optimized, the BLER shifts to that of \( P_{\text{high}} \) (lower \( E_b/N_0 \)) or \( P_{\text{low}} \) (higher \( E_b/N_0 \)).

Figure 5 plots the BLER and the complexity percentage for a BCH(255,239,2) code decoded with Chase, M-IDA-Chase, and MD-IDA-Chase. IDA-Chase curves have been...
than MD-IDA, going as low as $60\,\text{dB}$. At lower $E_b/N_0$, as the percentage of time where $P_{\text{high}}$ is used increases, the complexity rises and BLER drifts towards that of Chase with $P = P_{\text{high}} = 5$. An opposite effect is observed at higher $E_b/N_0$, where the BLER and complexity move toward that of Chase with $P = P_{\text{low}} = 3$. In terms of complexity M-IDA shows higher gains than MD-IDA throughout the majority of the $E_b/N_0$ range, with a minimum observed complexity of $25.6\%$ for M-IDA and $36.6\%$ for MD-IDA. Compared to IDA-Chase, both M-IDA and MD-IDA yield comparable complexity reduction, even performing better in case of M-IDA with $P_{\text{low}} = 3$.

Figure 6 portrays the BLER of the same code decoded with ORBGRAND, M-IDA-ORBGRAND, and MD-IDA-ORBGRAND. IDA-ORBGRAND curves consider $n_{\text{Pat}_{\text{high}}} = 500$ ($n_{\text{Pat}_{\text{low}}} = 252$), and have been obtained with $\gamma = 7\,\text{dB}$ for M-IDA, with $n_{\text{Pat}_{\text{low}}}=252$ ($\gamma_M = 8.2$), the BLER of IDA-ORBGRAND matches that of ORBGRAND with $n_{\text{Pat}} = 500$, with a complexity of $82.5\%$ for MD-IDA and of $89.2$ for M-IDA, while $n_{\text{Pat}_{\text{low}}}=168$ ($\gamma_M = 9.2$) allows to match the BLER of ORBGRAND with $n_{\text{Pat}} = 446$, further lowering the complexity to $75.9\%$ for MD-IDA and to $78\%$ for M-IDA. The reference curves for ORBGRAND with $n_{\text{Pat}}=252,168$ show the potential BLER degradation of IDA-ORBGRAND; however, the selection of $n_{\text{Pat}_{\text{low}}}$ is able to minimize the BLER degradation while granting substantial complexity reduction.
IDA decoding can be implemented in practical decoders at the cost of some threshold comparisons and counters, whose number depends on how many LLRs are received from the channel at each clock cycle. M-IDA and MD-IDA decoding require the identification of some of the minimum elements of $|y|$; this operation can be quite costly, depending on the number of elements in $y$ and the desired minima. However, for algorithms like Chase and ORBGRAND, this information is part of the standard decoding process, and thus comes at no extra cost. In these cases, M-IDA can be implemented at the cost of a single comparator, while MD-IDA requires a subtractor and a comparator; the latency of these computations is independent of the code length.

V. Conclusion

In this work, two low-complexity incarnations of input-distribution-aware (IDA) decoding were proposed, called M-IDA and MD-IDA. They exploit the partial sorting of channel LLRs that is inherent to some decoding algorithms, like Chase and ORBGRAND, to dynamically reduce the number of parallel decoding attempts necessary for successful decoding, leading to energy saving at a negligible implementation cost. Moreover, the original IDA decoding approach was extended to different types of codes and decoding algorithms. The proposed techniques are shown to reduce the decoding complexity of Chase and ORBGRAND down to 17% and 67%, respectively, while meeting a performance target.

REFERENCES

[1] D.J.C. MacKay, “Fountain codes,” IEE Proceedings - Communications, vol. 152, pp. 1062–1068(6), December 2005.
[2] O. Afisiadis, A. Balatsoukas-Stimming, and A. Burg, “A low-complexity improved successive cancellation decoder for polar codes,” in 2014 48th Asilomar Conference on Signals, Systems and Computers, 2014, pp. 2116–2120.
[3] F. Cochachin, D. Declercq, E. Boutillon, and L. Kessal, “Density evolution thresholds for noise-against-noise min-sum decoders,” in 2017 IEEE 29th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), 2017, pp. 1–7.
[4] I. Tal and A. Vardy, “List decoding of polar codes,” IEEE Transactions on Information Theory, vol. 61, no. 5, pp. 2213–2226, May 2015.
[5] C. Pillet, C. Condo, and V. Bioglio, “SCAN list decoding of polar codes,” in ICC 2020 - 2020 IEEE International Conference on Communications (ICC), 2020, pp. 1–6.
[6] D. Chase, “Class of algorithms for decoding block codes with channel measurement information,” IEEE Transactions on Information Theory, vol. 18, no. 1, pp. 170–182, 1972.
[7] K. R. Duffy, “Ordered reliability bits guessing random additive noise decoding,” CoRR, vol. abs/2001.00546, 2020.
[8] C. Condo, “Input-distribution-aware successive cancellation list decoding of polar codes,” IEEE Communications Letters, vol. 25, no. 5, pp. 1510–1514, 2021.
[9] K. R. Duffy, J. Li, and M. Méard, “Capacity-achieving guessing random additive noise decoding,” IEEE Transactions on Information Theory, vol. 65, no. 7, pp. 4023–4040, 2019.
[10] Y.-M. Lin, H.-C. Chang, and C.-Y. Lee, “Improved high-code-rate soft BCH decoder architectures with one extra error compensation,” IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 21, no. 11, pp. 2160–2164, 2013.
[11] Y. Wu, “Fast Chase decoding algorithms and architectures for Reed–Solomon codes,” IEEE Transactions on Information Theory, vol. 58, no. 1, pp. 109–129, 2012.
[12] S. M. Abbas, T. Tonnellier, F. Ercan, and W. J. Gross, “High-throughput VLSI architecture for GRAND,” in 2020 IEEE Workshop on Signal Processing Systems (SiPS), 2020, pp. 1–6.