A Variable Node Design with Check Node Aware Quantization Leveraging 2-Bit LDPC Decoding

Philipp Mohr, Gerhard Bauch
Hamburg University of Technology
Institute of Communications
21073 Hamburg, Germany
Email: {philipp.mohr, bauch}@tuhh.de

Abstract—For improving coarsely quantized decoding of LDPC codes, we propose a check node aware design of the variable node update. In contrast to previous works, we optimize the variable node to explicitly maximize the mutual information preserved in the check-to-variable instead of the variable-to-check node messages. The extended optimization leads to a significantly different solution for the compression operation at the variable node. Simulation results for regular LDPC codes confirm that the check node aware design, especially for very coarse quantization with 2- or 3-bit messages, achieves performance gains of up to 0.2 dB - without additional hardware costs. We also show that the 2-bit message resolution enables a very efficient implementation of the check node update, which requires only 2/9 of the 3-bit check node’s transistor count and reduces the signal propagation delay by a factor of 4.

I. INTRODUCTION

Low-resolution message passing plays a key role in hardware efficient implementations of low-density parity-check (LDPC) decoders. In particular the transfer of message bits between variable and check nodes causes high energy and chip area consumption. Hence, the node updates within practical implementations include quantization operations. In recent years, finite alphabet decoders [1]–[6] that maximize the mutual information in the decoding process, have shown superior performance over conventionally designed decoders like the normalized or offset min-sum algorithms [7]. The mutual information maximizing decoders are typically optimized in an offline design phase, where discrete density evolution tracks the joint distribution of messages and relevant variables [8].

The objective of maximizing the mutual information can be solved for different local optimization levels: One option is a lookup table design that decomposes each node into concatenated two-input lookup tables [1], [2]. To avoid increasing internal resolutions that would cause non-practical table sizes, each lookup table performs a compression operation that is optimized to maximize the preserved mutual information in the output message.

Another option is a computational domain technique which improves the performance by allowing higher internal resolutions in the node update [3]. In that approach, all input messages are translated to higher resolution representation values (e.g. proportional to log-likelihood ratios) and merged with arithmetic operations. Finally, a quantization operation maximizes the mutual information in the output message.

Yet, the two aforementioned classes of discrete decoders neglect the check node behavior when optimizing the compression operation at the variable node. An important insight about the check node update is that the reliability of the output message is similar to the lowest reliability found in the extrinsic input messages. Hence, input messages that represent high reliability have a very low chance to affect the output message of the check node significantly. Ultimately, the decoder performance depends on the quality of the check node and channel messages which are processed in the variable node to compute the hard decisions. This implies potential for improvement by designing the variable node’s compression operation with the overall goal to maximize the preserved mutual information after the check node update.

In this paper, we further extend the local optimization scope to a third level where the variable and check nodes (circle and squares, respectively) are jointly considered in the optimization, as depicted in Fig. 1. We showed in [6] that the restriction to uniformly placed thresholds achieved similar performance as non-uniform quantization at significantly reduced hardware costs. In addition, the check node with the widely applied minimum approximation is an excellent performance-complexity trade-off [2], [6]. Therefore, a combination of uniform quantization in the variable node...

Fig. 1: A check node aware design of the variable node’s compression operation.
and minimum approximation in the check node is considered here as a very appealing choice for hardware implementations. Moreover, we exploit that this configuration requires a single-parameter optimization: Only the spacing between the uniform boundaries (alternatively maximum representation range) must be adjusted, which reduces the computational effort of the joint optimization of variable and check nodes. Further, we note that under symmetric 2-bit decoding, the solutions with uniform and non-uniform quantization are equivalent. The contributions of this paper can be summarized as follows:

- A check node aware design of the variable node, that explicitly maximizes the mutual information in the check node messages, is performed for the first time, to the best of our knowledge.
- We analyze the performance in terms of mutual information and error rates in detail. Especially decoding under very coarse quantization benefits from the proposed method. We reveal significant gains of up to 0.2 dB for 2-bit decoding.
- We show that a 2-bit check node update requires only 2/9 of the logic gate count and 1/4 of the propagation delay compared to a 3-bit check node update.

Next, section II presents the system structure. In particular, check and variable node operations are described in detail. Furthermore, the hardware complexity of 2-bit and 3-bit symmetric check node processing is compared. Section III explains the check node aware design of the variable node and evaluates the performance in terms of mutual information. Finally, section IV compares the error rate performance for different regular LDPC codes.

II. System Structure

We assume a binary LDPC code with parity check matrix 

$$H \in \{0,1\}^{N_c \times N}$$

which can be represented by a Tanner graph with N variable nodes (VNs) and Nc check nodes (CNs). The encoder maps the information bits 

$$u = [u_1, \ldots, u_N]$$

to code bits 

$$b = [b_1, \ldots, b_N]$$

satisfying 

$$HB^T = 0.$$ 

For the transmission, we consider binary phase-shift keying (BPSK) symbols which are disturbed by additive white Gaussian noise (AWGN) at the receiver. A mutual information maximizing symmetric channel quantizer (like in [1]) maps the received 

$$w_{ch} \in T_{w_{ch}}$$

with finite sign-magnitude alphabet 

$$T = \{-2^{l-1}, \ldots, -1, 0, 1, \ldots, 2^{l-1}\}.$$ 

In the decoder, w-bit messages are exchanged over multiple decoding iterations between variable and check nodes. We apply a flooding schedule to perform the node updates. However, the proposed techniques can also be adopted for other schedules (e.g. layered scheduling [4]). In the first decoding iteration the channel messages are directly mapped to the variable node output. If 

$$w < w_{ch}$$

an initial variable node update is designed to perform this compression mapping, which can be implemented by a small single-input lookup table.

A. Check Node Update with Minimum Approximation

A check node with degree 

$$d_c$$

exploits its underlying parity check equation 

$$b_1 \oplus b_2 \oplus \ldots \oplus b_{d_c} = 0$$

to compute extrinsic information about the participating code bits. The incoming 

$$w$$-bit variable node messages 

$$t_i^v \in T_{w_{ch}}$$

encode the probabilities 

$$p(b_i | t_i^v)$$

which can also be represented by log-probability ratios 

$$L_i^v = \log \frac{p(b_i = 0 | t_i^v)}{p(b_i = 1 | t_i^v)}.$$ 

A check node with 

$$d_c > 3$$

inputs can be decomposed into multiple degree-three check nodes [9]. Minimum delay for the full check node update is achieved with a tree structure like it is shown in Fig. 2. With proper scheduling, each factor node computes only three output messages. E.g., extrinsic probability information for the auxiliary variable 

$$x_1$$

is provided by

$$p(x_1, t_1^v, t_2^v) = \sum_{b_1, b_2} p(x_1 | b_1, b_2) p(b_1 | t_1^v) p(b_2 | t_2^v)$$

where 

$$p(x_1 | b_1, b_2)$$

is obtained from the parity check equation 

$$x_1 = b_1 \oplus b_2.$$ 

Performing the computation (1) in log-domain with 

$$L := \log \frac{p(x_1 = 0, t_1^v, t_2^v)}{p(x_1 = 1, t_1^v, t_2^v)}$$

results in the well-known box-plus operation [10]

$$L = L_1^v \oplus L_2^v = \text{sgn}(L_1^v) \text{sgn}(L_2^v) \min(|L_1^v|, |L_2^v|) + f(|L_1^v|, |L_2^v|),$$

where 

$$f(x, y) = \log(1 + e^{-|x-y|}) - \log(1 + e^{-|x+y|})$$

is a correction term. When neglecting the correction term and if the messages use a symmetric sign-magnitude format, (2) can be simplified resulting in the minimum approximation [2]:

$$t = \text{sgn}(t_1^v) \text{sgn}(t_2^v) \min(|t_1^v|, |t_2^v|).$$

![Fig. 2: Full check node update (d_c=8) in a tree structure.](image)

![Fig. 3: Circuits for a two-input check node update.](image)
In Fig. 3 we depict two optimized hardware implementations of (3) for 2- and 3-bit alphabet sizes. The circuits implement the Boolean expressions of the minimized disjunctive normal form obtained with the Karnaugh map technique. The 3-bit check node requires 9 logic gates to perform the update. The critical path consists of 4 serial logic gate operations. On the other hand, the 2-bit check node requires only 2 logic gates which can operate in parallel. The minimum delay is reduced by a factor of 4. The price to be paid is a performance loss, however, the check node aware design proposed by the paper can reduce this loss significantly.

The overall number of comparisons in a tree structure is $3(d_c - 2)$ assuming $d_c$ is a power of two. We note that the check node update could also be performed with a first and second minima search which requires only $d_c + \lfloor \log d_c \rfloor - 2$ comparisons [11]. Yet, our approach avoids usage of multiplexers and leads to less delay.

**B. Variable Node Update with Uniform Quantization**

A variable node with degree $d_v$ deals with the messages $t^v \in T_{u,v}$ and $t^c_k \in T_{u,c}, k \in \{1, \ldots, d_v\}$ from the channel and the $d_c$ connected check nodes, respectively. The messages are assumed to provide independent information about the underlying code bit $b$. For finite alphabet decoders, multiple implementation options exist for the variable node. Here, we decided for the computational domain structure [3] which potentially offers better performance than the lookup table decomposition technique [1]. For example, extrinsic information for the first connected check node in terms of an LLR yields

$$L(b | t^v, t^c_1, \ldots, t^c_{d_c}) = L(b) + L(t^v | b) + \sum_{k=2}^{d_v} L(t^c_k | b)$$

where $L(t | b) = \log \frac{p(t | b=0)}{p(t | b=1)}$. For finite alphabet decoders, the LLRs can be stored in translation tables $\phi^v(t^v) = L(t^v | b)$ and $\phi^c_k(t^c) = L(t^c_k | b)$. Finally, the resulting LLR $L(b | t^v, t^c_1, \ldots, t^c_{d_c})$ must be quantized to a $w$-bit message

$$t^c_1 = Q^v \left( \phi^v(t^v) + \sum_{k=2}^{d_v} \phi^c_k(t^c_k) \right).$$

In Fig. 4 a hardware implementation of the variable node update for all connected check nodes is depicted. We point out, that symmetric boundaries and translation tables are enforced to reduce computational and space complexity [6]. In practice, a translation table is used that outputs scaled LLRs represented by integer numbers $\phi_{\Delta}(t) = \text{sgn}(\phi(t)) \min(\lfloor \frac{1}{\Delta} \phi(t) + \frac{1}{2} \rfloor, 2^{w_c-1}-1)$ where $\phi_{\Delta}(t) \approx \frac{1}{\Delta} \phi(t)$. The step size $\Delta \in \mathbb{R}^+$ defines the internal resolution of the LLRs and $w_c$ is the translation bit width. In this paper we restrict to uniform quantization with $r \in \mathbb{N}_0$:

$$Q^v(y) = \text{sgn}(y) \min \left( \lfloor |y|/2^r \rfloor + 1, 2^{w-1} \right)$$

It was observed in [6] that uniform quantization reduces the preserved mutual information $I(B; T^v)$ only insignificantly compared to the optimal non-uniform quantization [3, 12].

**III. DESIGN OF CHECK NODE AWARE QUANTIZATION**

We consider a check node where extrinsic information about some code bit $x$ is obtained from $x = b_1 \oplus \ldots \oplus b_{d_c-1}$. Typically, the $d_c-1$ connected variable nodes are designed to preserve extrinsic information about the underlying code bit $b_i$ in the corresponding variable node message $t^v_i$. All the variable nodes within one iteration use the same design obtained for a template variable node $b$ and output message $t^v$ that is typically optimized for $\max_{Q^v} I(B; T^v)$ [1, 3]. However, as we will show, this local optimization does not achieve the maximum mutual information $I(X; T^v)$ between the code bit $x$ and the check node message $t^v$. Therefore, we extend the local optimization of the variable node by taking the check node behavior into account, aiming for $\max_{Q^v} I(X; T^v)$. Note that with restriction to uniform boundaries, $Q^v$ is defined by $\Delta$ and $r$ as depicted in Fig. 6 [6]. To simplify computations, we assume the minimum approximation (3) in the check node. Appendix VI-A describes the optimization procedure in detail.

![Fig. 4: A full variable node update [6]. The internal summation operation uses the two’s complement (2’s) format. The exchanged messages use the sign-magnitude (SM) format.](image)

![Fig. 5: Hardware schematics from [6] for performing uniform quantization with $w_y=9$ bit for $y$ and $w=4$ bit for $t$.](image)
A. Example with 2-Bit Decoding

In this example, we compare two optimization procedures. The first option (proposed) is the check node aware design, which mandates the mutual information in the messages after the check node update shall be maximized. The second option (conventional) performs the mutual information maximization in the messages before the check node update. The messages exchanged between the variable nodes \((d_v = 6)\) and check nodes \((d_c = 32)\) use 2 bits. We assume a variable node with given input joint distributions \(p(x, t^v)\) and \(p(x, t^c)\). Those distributions determine the translation tables \(\phi^c\) and \(\phi^e\) shown in the Table I.

The results from the optimization search are shown in Fig. 7a. The two vertical lines depict the optimal boundary spacing for the two optimization criteria. Clearly, the check node aware solution leads to a higher mutual information \(I(X; T^c)\) in the check-to-variable node messages, despite lower distortion magnitudes caused by clipping effects that occur for \(t^c \approx +8\) in the translation table (see Table I). However, the distortion for \(L(b|y^v) > 4.0\) is irrelevant for the performance since the threshold is placed at \(L(b|y^v) \approx 1.6\).

As described in Section I, low-reliability messages dominate high-reliability messages in the check node update. This behavior of the check node causes the check node aware design to decrease the frequency of low-reliability messages and to increase the probability for high-reliability messages, as can be observed in Fig. 7c. In this way, the probability for low-reliability check node messages \(t^c\) is reduced. On the other hand, the Kullback Leibler divergence \(D_{KL}(p(x|t^c) || p(x))\) is reduced for the high-reliability check node messages \(t^c\). The proposed method finds a trade-off, which ultimately leads to a higher preservation of mutual information \(I(X; T^c) = \sum_{t^c} p(t^c) D_{KL}(p(x|t^c) || p(x))\) in the check node messages.

The overall mutual information gain of the check node aware design is \(\Delta I(X, T^c) = 0.073 - 0.056 = 0.017\). The gains accumulate over multiple iterations as shown in Fig. 8. Remarkably, the check node aware design requires only one third of the iteration count for convergence in the considered scenario. Note, that the initial mutual information for the variable-to-check node messages is lower in the beginning. However, after a few iterations, the mutual information \(I(B, T^c)\) of the check node unaware design is surpassed.
IV. PERFORMANCE EVALUATION

To evaluate the error rate performance, we first consider a regular high-rate LDPC code [13] with $d_v=6$, $d_c=32$, $N=2048$ and rate $R=0.84$ (the parity check matrix does not have full rank). The decoder performs 10 iterations.

In Fig. 9 the bit error rates for the check node aware and unaware design are shown. In all cases the minimum approximation in the check node and computational domain with uniform quantization in the variable node are used. The design $E_b/N_0$ is optimized for best performance at bit error rates equal to $10^{-6}$. As expected from Fig. 8, the check node aware designs also show superior error rate performance over the check node unaware configurations. Especially, the 2-bit decoder with 3-bit (4-bit) channel messages achieves a gain of $0.16$ dB ($0.16$ dB). With increasing decoder bit width $w$ the gains reduce to $0.05$ dB for 3-bit decoding and to $0.02$ dB for 4-bit decoding.

Next, we take a look at the frame error rate performance in Fig. 10. The design $E_b/N_0$ is optimized for best performance at frame error rates equal to $10^{-7}$. All quantized decoders use 4 bits for the channel message. Remarkably, even the 2-bit check-node-aware decoder outperforms the 32-bit belief propagation (BP) decoder. Further, the 4-bit check-node-aware decoder achieves similar performance as the non-uniform computational domain design developed in [3], [14]. Yet, the complexity of the proposed decoder is significantly smaller by using the minimum approximation in the check node and uniform quantization in the variable node [6]. The 2-bit decoder exhibits a slightly higher error floor than the decoders which use more bits for the exchanged messages.

The second example applies a medium-rate regular LDPC code from [4] with $d_v=4$, $d_c=8$, $N=7648$, $R=0.5$ and 20 decoding iterations. Again significant gains of up to $0.2$ dB can be observed in Fig. 11, when comparing the check node aware and unaware decoders at same decoding bit width. On
the other hand, our results suggest that the performance loss between 2- and 3-bit decoding is larger for medium-rate than for high-rate codes.

V. CONCLUSIONS

In this paper, we proposed a new class of mutual information maximizing decoders, where the variable node’s quantizer design maximizes the mutual information in the check node output messages. The optimization complexity was significantly reduced by restriction to the low-complexity uniform quantization approach in the variable node [6] and the minimum approximation used in the check node [2].

Simulation results revealed significant decoding performance gains ranging from 0.02 to 0.2 dB compared to the conventional optimization, where the variable node’s quantizer design maximizes the mutual information in the check node input messages [6]. The highest gains were achieved for very coarse quantization with 2 bits. We also showed that the 2-bit check node operation offered major complexity reduction potential over the 3-bit check node. Therefore, we believe that 2-bit decoding of LDPC codes is a promising candidate for applications with low-cost, low-energy and/or high-throughput requirements.

VI. APPENDIX

A. Optimization Procedure of the Check Node Aware Design

A grid based search for \( \max_{\Delta} \ I(X; T') \) can be implemented in parallel for a certain range \( \Delta \in \mathbb{R}^+ \) and a small set of right shifts \( r \in \mathbb{N}_0 \). As a starting point, we assume that the joint distribution of the channel and extrinsic check node messages w.r.t. the code bit \( b \) are given by \( p(b, t^c_b) \) and \( p(b, t^c_i), i \in \{1, \ldots, d_v-1\} \), respectively. After initializing \( p(b, s_k) = \phi_h(k) p(t^c_b | p(b)) \), we obtain the output distribution of the summation operation recursively for \( k=1, \ldots, d_v-1 \) as

\[
p(b, s_k) = \sum_{t^c_k} p(b, s_{k-1} = s_k - \phi_h(k) p(t^c_k | p(b)).
\]

(7)

The uniform quantization \( Q(y^v) \), defined in (6), performs an \( r \)-bit right shift operation and a clipping on \( y^v := s_{d_v-1} \) which leads to the output distribution

\[
p(b, t^v) = \sum_{y^v} p(t^v | y^v) p(b, y^v).
\]

(8)

A symmetry preserving structure like in [6] is assumed where

\[
p(t^v | y^v) = \begin{cases} 1/2 & t^v = \pm 1, y^v = 0 \\ \delta(t^v - Q(y^v)) & \text{otherwise} \end{cases}
\]

(9)

At the check node, in case of regular codes, the distributions for extrinsic variable node messages are given by \( p(b_i, t^v_i) = p(b, t^v_i) | i \in \{1, \ldots, d_v\} \). The joint distribution computation for the check node message \( t^v := t^c_{d_v-1} \) is initialized with \( p(x_k, t_k) = p(b_1, t^v_1) \) and performed recursively for \( k = 2, \ldots, d_v-1 \) as

\[
p(x_k, t_k) = \sum_{t_{k-1}, t^v_k} p(t_k | t_{k-1}, y_k) p(x_k, t_{k-1}, t^v_k)
\]

(10)

where

\[
p(x_k, t_{k-1}, t^v_k) = \sum_{b_k} p(X_{k-1} = x_k \oplus b_k, t^v_k) p(b_k, t^v_k),
\]

(11)

\[
p(t_k | t_{k-1}, t^v_k) = \delta(\text{sgn}(t_k) - \text{sgn}(t_{k-1}) \text{sgn}(y_k))
\]

\[
\delta(|t_k| - \min(|t_{k-1}|, |y_k|)).
\]

(12)

With \( p(x, t^v) = p(x_{d_v}, t^v_{d_v-1}) \) and the Kullback-Leibler divergence \( D_{KL}(p(x|t^v)|p(x)) = \sum_{p(x|t^v)} \log_2(p(x|t^v)/p(x)) \), the mutual information yields

\[
I(X; T^v) = \sum_{t^v} p(t^v) D_{KL}(p(x|t^v)|p(x))
\]

(13)

REFERENCES

[1] J. Lewandowsky and G. Bauch, “Trellis based node operations for LDPC decoders from the Information Bottleneck method,” in 2015 9th International Conference on Signal Processing and Communication Systems (ICSPCS). Cairns, Australia: IEEE, Dec. 2015, pp. 1–10.

[2] M. Meidlinger, A. Balatsoukas-Stimming, A. Burg, and G. Matz, “Quantized message passing for LDPC codes,” in 2015 49th Asilomar Conf. Signals, Systems and Computers, Nov. 2015, pp. 1666–1670.

[3] X. He, K. Cai, and Z. Meas, “On Mutual Information-Maximizing Quantized Belief Propagation Decoding of LDPC Codes,” in 2019 IEEE Global Communications Conference (GLOBECOM), Dec. 2019, pp. 1–6.

[4] P. Mohr, G. Bauch, F. Yu, and M. Li, “Coarsely Quantized Layered Decoding Using the Information Bottleneck Method,” in 2021 - IEEE International Conference on Communications, Jun. 2021, pp. 1–6.

[5] L. Wang, C. Terril, M. Stark, Z. Li, S. Chen, C. Hulce, C. Kuo, R. Wesel, G. Bauch, and R. Pitchumani, “Reconstruction-Computation-Quantization (RCQ): A Paradigm for Low Bit Width LDPC Decoding,” IEEE Transactions on Communications, pp. 1–1, 2022.

[6] P. Mohr and G. Bauch, “Uniform vs. Non-Uniform Coarse Quantization in Mutual Information Maximizing LDPC Decoding,” in GLOBECOM 2022 - 2022 IEEE Global Commun. Conference, Dec. 2022, pp. 1–6.

[7] J. Chen, A. Dholakia, E. Eleftheriou, M. Fossorier, and X.-Y. Hu, “Reduced-complexity decoding of LDPC codes,” IEEE Transactions on Communications, vol. 53, no. 8, pp. 1288–1299, Aug. 2005, conference Name: IEEE Transactions on Communications.

[8] B. M. Kurkoski, K. Yamaguchi, and K. Kobayashi, “Noise Thresholds for Discrete LDPC Decoding Mappings,” in IEEE GLOBECOM 2008 - 2008 IEEE Global Telecommunications Conference. New Orleans, LA, USA: IEEE, 2008, pp. 1–5.

[9] F. R. Kschischang, B. J. Frey, and H. Loeliger, “Factor graphs and the sum-product algorithm,” IEEE Transactions on Information Theory, vol. 47, no. 2, pp. 498–519, Feb. 2001.

[10] L. Wang, C. Terril, M. Stark, Z. Li, S. Chen, C. Hulce, C. Kuo, R. Wesel, G. Bauch, and R. Pitchumani, “Reconstruction-Computation-Quantization (RCQ): A Paradigm for Low Bit Width LDPC Decoding,” IEEE Transactions on Communications, pp. 1–1, 2022.

[11] J. Lee, B. Kim, J. Jung, and I.-C. Park, “Low-Complexity Tree LDPC decoders from the Information Bottleneck method,” in 2015 9th International Conference on Signal Processing and Communication Systems (ICSPCS). Cairns, Australia: IEEE, Dec. 2015, pp. 1036–1036E.

[12] B. M. Kurkoski and H. Yagi, “Quantization of Binary-Input Discrete Memoryless Channels,” in IEEE Globecom 2005 and ACM SIGCOMM 2005. New Orleans, LA: IEEE Globecom 2005. Adelaide, Australia: IEEE, 2005, pp. 459–463.