# Energy Optimization of Faulty Quantized Min-Sum LDPC Decoders

Jérémy Nadal†, Mohamed Yaoumi†, Elsa Dupraz†, Frederic Guilloud†, and François Leduc-Primeau*
†IMT Atlantique, Lab-STICC, UMR CNRS 6285, F-29238 Brest, France
*Department of Electrical Engineering, École Polytechnique de Montréal, Montreal, Canada

***Abstract*—The objective of this paper is to minimize the energy consumption of a quantized Min-Sum LDPC decoder, by considering aggressive voltage downscaling of the decoder circuit. Since low power supply may introduce faults in the memories used by the decoder architecture, this paper proposes to optimize the energy consumption of the faulty Min-Sum decoder while satisfying a given performance criterion. The proposed optimization method relies on a coordinate-descent algorithm that optimizes code and decoder parameters that have a strong influence on the decoder energy consumption: codeword length, number of quantization bits, and supply voltage. Optimal parameter values are provided for several codes defined by their protographs, and significant energy gains are observed compared to non-optimized setups. Finally, further gains are obtained when the supply voltage is optimized per decoding iteration.**

## I. Introduction

Energy consumption is an important criterion in the design of electronic circuits, and can be greatly reduced by aggressive voltage scaling of the circuit. Low power supply may however introduce faults in the computation operations and memories of the circuit [1]. In this paper, we address this issue for low-density parity-check (LDPC) decoding circuits.

Two energy consumption models are provided in [2] for non-faulty LDPC decoders: the first model estimates the decoding complexity, while the second evaluates the wire length in the circuit. Then, [3] introduces a method to minimize the alphabet size of quantized messages exchanged in the decoder, aiming to lower the memories energy consumption. Finally, [4], proposes to optimize the code rate and irregular code degree distribution in order to minimize the decoder complexity and therefore its energy consumption.

In addition, the performance of LDPC decoders implemented on faulty hardware was widely studied in the literature. In [5] the authors assume that the LDPC decoder is subject to both transient and permanent errors. Transient errors make faulty gates or memory units provide an erroneous output from time to time with a non-zero probability. Permanent errors make a fraction of the gates and memories stuck at the same output. When dealing with energy consumption issues, we consider process diversity strategies, where the permanent errors turn into transient error [6]. The authors in [5]–[9] theoretically investigate the effect of transient errors on various LDPC decoders, such as Gallager A and B or quantized Min-Sum. However, none of these works relate the amount of faults introduced in the decoder to its energy consumption.

In this work, our objective is to optimize key decoding parameters, such as the code length, the quantization level and the supply voltage, so as to minimize the energy consumption of a faulty LDPC decoder while satisfying a given performance error rate criterion. For this, we consider protograph-based LDPC codes and quantized Min-Sum decoders for their easy hardware implementation [10]. These protograhs can be designed, for instance, based on one of the methods presented in [11], [12]. In addition, in [13], it is shown that memories represent around half of energy consumption of the decoder. This motivate us to consider that circuit faults are only introduced by the memory units, while processing elements are assigned to the nominal (reliable) power domain.

To estimate the LDPC decoder energy consumption, we update the non-faulty memory energy model of [14], for faulty decoders. This energy model depends on several code and decoder parameters, such as the protograph, the noise level, the number of quantization bits for the messages, the codeword length, and the number of iterations performed by the decoder. In order to properly evaluate the energy model, we consider the method of [14] which relies on Density Evolution (DE) in order to evaluate the distribution of the number of decoder iterations required for a given codeword length. Then, for a fixed protograph, we propose a method to jointly optimize the codeword length, the number of quantization bits and the noise level in order to minimize the decoder energy consumption. This method is based on a coordinate-descent algorithm that successively optimizes each parameter, assuming that the other ones are fixed, and repeats the process over several iterations. In addition, we show how to obtain further energy gain by optimizing the supply voltage per decoding iteration using the *DE-Gear-Shift* algorithm [6]. Simulation results provide the values of optimized parameters for several protographs, and show the energy gains compared to non-optimized decoders.

This paper is organized as follows. Section II reviews LDPC codes and decoders. Section III presents the method we consider to evaluate the finite-length performance of LDPC decoders. Section IV, introduces the optimization method. Section V shows simulation results.

## II. LDPC codes and decoders

We consider a codeword $\mathbf{x}$ of length $N$ to be transmitted over an additive white Gaussian noise (AWGN) channel of variance $\sigma^2$, with binary phase-shift keying (BPSK) modulation. We use $y_i$ to denote the $i$-th channel output, and $x_i \in \{-1, 1\}$ to denote the $i-$th modulated coded bit. The channel Signal-To-Noise Ratio (SNR) is defined as $\xi = 1/\sigma^2$.

In this section, we introduce protograph-based LDPC codes, and the considered faulty quantized Min-Sum decoder.

### A. Protograph-based LDPC codes

LDPC codes are represented by a sparse $M \times N$ parity-check matrix or equivalently by a Tanner graph with $N$ variable nodes and $M$ check nodes. The code rate is $R = K/N$, where $K$ is the information length. We use $\mathcal{N}_{c_j}$ to denote the set of all variable nodes (VNs) connected to check node (CN) $c_j$ in the Tanner graph , and $\mathcal{N}_{v_i}$ to denote the set of all $d_{v_i} = |\mathcal{N}_{v_i}|$ CNs connected to VN $v_i$. We consider LDPC codes constructed from protographs [15]. A protograph $S$ is an $m \times n$ matrix that gives the number of connections between each VN and CN in the reduced Tanner graph representing the protograph. We can construct an LDPC code of length $N$ by first copying the protograph $N/n$ times, and then by interleaving the edges to get the parity-check matrix.

### B. Faulty quantized Min-Sum decoder

In this paper, we consider a quantized offset Min-Sum decoder [16]. The simulated decoder is based on a bit-true version of the decoder architecture proposed in [17]. For simplicity, no pipeline stages are considered, which corresponds to a row-layered scheduling. This enables to use fewer decoding iterations and reduces the size of the circuit. The decoder messages are quantized on $q$ bits and between values $-Q$ and $Q$, where $Q = 2^{q-1} - 1$. We consider the following quantization function:

$$\Delta(x) = \text{sgn}(x) \min\left(Q, \left\lfloor |x| + \frac{1}{2} \right\rfloor\right), \tag{1}$$

where $\text{sgn}(x) = 1$ if $x \geq 0$, and $\text{sgn}(x) = -1$ if $x < 0$.

Since memory units are responsible for a large part of the decoder energy consumption [13], we assume that faults are introduced during memory read operations in the decoder. The error model corresponds to an XOR operation $\oplus$ between the memory read port output and a noise term represented by independent and identically distributed (i.i.d.) random variables $\mathcal{B}$. These random variables can equivalently be represented on $q$ bits as $(b_1, \cdots, b_q)$. We assume that the $b_k$'s are i.i.d. Bernoulli random variables with parameter $\epsilon$. Note that statistical dependencies among the bits could be considered easily by using a more general fault model.

In order to initialize the decoder, we compute log-likelihood ratios (LLR) $r_i$ for each received value $y_i$ as $r_i = 2\alpha y_i/\sigma^2$, where $\alpha$ is a scaling parameter. In the architecture proposed in [17], the a-posteriori-LLRs (AP-LLRs) $\Lambda_i^{(\ell)}$ at iteration $\ell \in [\![1, L]\!]$ are updated as

$$\Lambda_i^{(\ell)} \leftarrow \Lambda_i^{(\ell)} + \sum_{j \in \mathcal{N}_{v_i}} \left(\gamma_{j\to i}^{(\ell)} - \left(\gamma_{j\to i}^{(\ell-1)} \oplus \mathcal{B}_j^{(\ell-1)}\right)\right), \tag{2}$$

with $\Lambda_i^{(0)} = \Delta(r_i)$, $\gamma_{j\to i}^{(\ell)}$ represents the message sent from the CN $c_j$ to the VN $v_i$ at iteration $\ell$, and $\mathcal{B}_j^{(\ell-1)}$ is the noise introduced when the messages $\gamma_{j\to i}^{(\ell-1)}$ are read from their dedicated memory. The AP-LLRs $\Lambda_i^{(\ell)}$ are quantized on $q+q_s$ bits, with $q_s = \lceil \log_2(\max d_{v_i} + 1) \rceil$, in order to avoid any saturation issue when writing $\Lambda_i^{(\ell)}$ into the memory. In the considered architecture, the message $\lambda_{i\to j}^{(\ell)}$ sent from the VN $i$ to the CN $j$, at iteration $\ell$, is calculated during the CN update, which is as follows:

$$\lambda_{i\to j}^{(\ell)} = \left(\Lambda_i^{(\ell)} \oplus \mathcal{B}_i^{(\ell)}\right) - \left(\gamma_{j\to i}^{(\ell-1)} \oplus \mathcal{B}_j^{(\ell-1)}\right) \tag{3}$$

$$\begin{aligned}\gamma_{j\to i}^{(\ell)} &= \left(\prod_{i' \in \mathcal{N}_{c_j}\setminus\{i\}} \text{sgn}\left(\lambda_{i'\to j}^{(\ell)}\right)\right) \\ &\times \max\left[\min_{j' \in \mathcal{N}_{c_j}\setminus\{i\}} \left|\lambda_{i'\to j}^{(\ell)}\right| - \beta, 0\right],\end{aligned} \tag{4}$$

where $\beta$ is an offset parameter, and where $\mathcal{B}_i^{(\ell)}$ represents the noise introduced when reading the memory where the variable-node messages are stored. The decoder stops when a stopping criterion is satisfied, or when the maximum number of iterations $L$ is reached.

## III. Finite-length performance evaluation

DE allows to estimate the error probability $p_{e_\infty}^{(\ell)}(\xi)$ of an LDPC decoder, for a given protograph $S$ and at given SNR $\xi$ and iteration number $\ell$ [11]. However, DE calculates $p_{e_\infty}^{(\ell)}(\xi)$ under the assumption that the codeword length tends to infinity. As an alternative, [18] provides a method to estimate the error probability $p_{e_N}^{(\ell)}(\xi)$ of an LDPC decoder at finite length $N$. This method estimates the Bit Error Rate (BER) $p_{e_N}^{(\ell)}(\xi)$ as

$$p_{e_N}^{(\ell)}(\xi) = \int_0^{\frac{1}{2}} p_{e_\infty}^{(\ell)}(x)\, \mathcal{G}\left(x;\, p_0, \frac{p_0(1-p_0)}{N}\right) dx. \tag{5}$$

In this expression $p_0 = \frac{1}{2} - \frac{1}{2}\text{erf}\left(\sqrt{\xi/2}\right)$, and $p_{e_\infty}^{(\ell)}(x)$ is the error probability evaluated with standard DE at SNR value $2(\text{erf}^{-1}(1-2x))^2$. The function $\mathcal{G}(x;\mu,\nu^2)$ is the probability density function of a Gaussian random variable with mean $\mu$ and variance $\nu^2$.

For simplicity, we implemented the DE equations by considering a flooding scheduling. Following [19], we empirically obtained the same error probabilities as a row-layered scheduling, given that the number of iterations is doubled. In addition, the DE equations were derived by considering that the memory faults are introduced after computation of the check-to-variable messages $\lambda_{i\to j}^{(\ell)}$, as in [16]. This slightly differs from the hardware decoder of [17] described in Section II, where the faults are introduced when the AP-LLRs $\Lambda_i^{(\ell)}$ are read. Despite this difference, our bit-true simulations confirm that the DE accurately predicts the BER.

To evaluate the decoder energy consumption, the distribution of the number of decoding iterations is required. The complementary cumulative distribution function (CCDF) $\bar{\phi}_N(\ell)$ of the number of iterations at codeword length $N$ is evaluated as

$$\bar{\phi}_N(\ell) = \int_0^{\frac{1}{2}} R_\infty^{(\ell-1)}(x)\mathcal{G}\left(x; p_0, \frac{p_0(1-p_0)}{N}\right) dx. \tag{6}$$

where $R_\infty^{(\ell)}(x) = 1 - (1 - p_{e_\infty}^{(\ell)}(x))^N$. As for the error probability $p_{e_N}^{(\ell)}(\xi)$ defined in (5), the expression of $\bar{\phi}_N(\ell)$

takes into account the channel variability, but does not evaluate the effect of cycles on the decoder performance. However, as shown in [14], [18], these two formula accurately predict the finite-length waterfall performance for long codewords.

## IV. Energy Optimization

We now propose an optimization method to minimize the decoder energy consumption while satisfying a certain performance criterion.

### A. Faults and energy models

Experimental measurements show that the memory bit fault rate $\epsilon$ decreases exponentially in $V$ [20] or $V^2$ [21], depending on the technology, where $V$ is the supply voltage. We thus infer the following general model:

$$\epsilon = \min\Big(\exp\left(a + bV + cV^2\right), \frac{1}{2}\Big), \tag{7}$$

where $(a, b, c)$ are positive coefficients that depend on the circuit technology and $\min(\cdot, \frac{1}{2})$ ensures that the bit flip probability is no larger than $\frac{1}{2}$. These coefficients can be obtained through polynomial regression on measured fault rates. In this paper, we base our model on the fault rates measured in [20, Fig.11] for a recent 22nm technology. We then have $(a, b, c) = (22.12, -68.14, 0)$. The energy for reading one bit in memory is proportional to $V^2$. Thus, we define the normalized energy per memory bit as $V^2/V_{\text{nom}}^2$, where $V_{\text{nom}}$ is the nominal supply voltage. Note that the knowledge of the exact energy value is not required to minimize the energy.

The energy model proposed in [14] estimates the overall memory energy consumption of a non-faulty quantized Min-Sum decoder by counting the total number of bits written into memory during the decoding process. It is evaluated from the facts that: (i) at a VN, the AP-LLR $\Lambda_i$ is stored on $q + q_s$ bits, (ii) since we are using a row-layered scheduling, a VN updates its messages every time one of its neighboring check nodes is updated, (iii) at a CN, 1 bit is stored for the sign of the output message, and two minimum absolute values of $q-1$ bits each are stored. Hence, according to [14], the number of memory accesses $N_{\text{MA}}$ per information bits and per iteration can be expressed as

$$N_{\text{MA}}(q) = \frac{q + q_s}{Rn} \sum_{i=1}^{n} d_{v_i} + \frac{1 - R}{Rn} \sum_{j=1}^{m} \left(2q + d_{c_j} - 2\right).$$

It is worth mentioning that, for a given $q$, $N_{\text{MA}}$ only depends on the code and the decoding algorithm. Therefore, this model is valid for any row-layered hardware architectures. In order to properly capture the effect of codeword length $N$, we consider the memory energy consumption per information bits. Thus, the following energy model will be considered in the optimization:

$$\mathcal{E}(q, \boldsymbol{V_L}, N) = \frac{N_{\text{MA}}(q)}{V_{\text{nom}}^2} \sum_{\ell=1}^{L} \bar{\phi}_N(\ell) V^2(\ell), \tag{8}$$

where, for the sake of generality, $V(\ell)$ is the supply voltage at decoding iteration $\ell$, and $\boldsymbol{V_L} = [V(1), ..., V(L)]$.

### B. Optimization problem

As a performance criterion for the optimization, we fix a target error probability $p_e^\star$ to be reached at a target SNR value $\xi^\star$. For simplicity, we assume that the protograph $S$ is fixed. We propose to minimize the energy consumption $\mathcal{E}$ with respect to the quantization level, the supply voltage, and the codeword length, while satisfying the performance criterion. By considering at first that the supply voltage $V$ is fixed for all iterations ($\boldsymbol{V_L} = [V, ..., V]$)), the optimization problem can be formulated as

$$\min_{V,q,N} \mathcal{E}(q, [V, ..., V], N) \quad \text{s.t.} \quad p_{e,\text{opt}}(q, V, N) < p_e^\star \tag{9}$$

where $p_{e,\text{opt}}(q, V, N) = \min_{\alpha,\beta} p_{e_N}^{(L)}(q, V, N)$ gives the minimum BER that can be reached by optimizing the scaling parameter $\alpha$ and the offset parameter $\beta$. Note that $p_{e_N}^{(L)}(q, N)$ is calculated from (5) at SNR $\xi^\star$.

### C. Optimization method

The optimization problem (9) is difficult to solve because it involves discrete parameters $q$ and $N$. In addition, it is computationally expensive to evaluate $p_{e,N}^{(\ell)}$ for given parameters $(q, V, N)$, because this requires to numerically evaluate integrals. Therefore, we want to lower the number of evaluations of these terms. We first define search intervals for the parameters $(q, V, N)$ involved in the optimization. According to Section IV-A, the continuous parameter $V$ lies in the interval $[0, V_{\text{nom}}]$, and we assume that discrete parameters $q$ and $N$ take values in the sets $[\![q_{\min}, q_{\max}]\!]$ and $[\![N_{\min}, N_{\max}]\!]$, respectively.

We then perform a coordinate-descent (CD) optimization, which consists of optimizing alternatively each of the three parameters $V$, $q$, and $N$, over a number $I$ of iterations. Since we consider a constrained optimization problem, we verify at each CD iteration that the selected parameters meet the performance criterion of the optimization problem. For this reason, we first initialize our algorithm with the three parameters $V^{(0)} = V_{\text{nom}}$, $q^{(0)} = q_{\max}$, and $N^{(0)} = N_{\max}$. Then, at iteration $i \in [\![1, I]\!]$, we successively solve the following three optimization problems:

1) Given $V^{(i-1)}$ and $N^{(i-1)}$, solve

$$q^{(i)} = \arg\min_{q} \mathcal{E}(q) \quad \text{s.t.} \quad p_{e,\text{opt}}(q) < p_e^\star \tag{10}$$

2) Given $V^{(i-1)}$ and $q^{(i)}$, solve

$$N^{(i)} = \arg\min_{N} \mathcal{E}(N) \quad \text{s.t.} \quad p_{e,\text{opt}}(N) < p_e^\star \tag{11}$$

3) Given $q^{(i)}$ and $N^{(i)}$, solve

$$V^{(i)} = \arg\min_{V} \mathcal{E}(V) \text{ s.t. } p_{e,\text{opt}}(V) < p_e^\star \tag{12}$$

In (10), the parameter $q$ is optimized by exhaustive search since the search interval is small. Then, for the optimization of $N$ and $V$ in (11) and (12), we retain the parameters that satisfy the performance criterion $p_{e,\text{opt}}$ and minimize the energy $\mathcal{E}$ among a certain number of values between $N_{\min}$ and $N_{\max}$ and between 0 and $V_{\text{nom}}$, respectively. To further reduce the computation time, we first evaluate the performance criterion $p_{e,\text{opt}}$, and then evaluate the corresponding energy $\mathcal{E}$ only if

the performance criterion is satisfied. Therefore, the proposed CD method requires to evaluate at most $I(\Delta_N + \Delta_q + \Delta_V)$ times the term $p_{e,\text{opt}}$, where $\Delta N$, $\Delta_q$, and $\Delta_V$ are the number of times the term $p_{e,\text{opt}}$ is evaluated to respectively solve (10), (11), and (12). By comparison, an exhaustive search method requires to evaluate this term $\Delta_N \Delta_q \Delta_V$ times. Since $\Delta_N$ and $\Delta_V$ will typically be large, it is clear that the CD method is much less complex than exhaustive search. Finally, the CD approach guarantees that the energy criterion is reduced at each iteration. It also ensures that the final solution satisfies the performance criterion. However, there is a risk that the algorithm falls into a local minimum, but as discussed in Section V, we did not observe this issue in our experiments.

### D. Per-iteration voltage optimisation

Although it is generally simpler to supply the whole decoder with a single voltage level, energy can be further reduced if the supply voltage $V(\ell)$ is optimized for each decoding iteration $\ell$. Such scheme can be implemented, for instance, in unrolled decoder architectures [22]. The major interest of this scheme is that it adds the necessary degrees of freedom to adapt the memory reliability to the decoding performance at each iteration, without much penalty on the final BER. As shown in Section V, this enables large additional energy gains, especially for low BER targets.

In this case, solving problem (9) becomes computationally heavy, since $L+4$ variables must be optimized instead of 5. Instead, we propose to first solve (9) with a fixed voltage supply $V$ to find the optimal decoder parameters $V_{\text{opt}}, q_{\text{opt}}, N_{\text{opt}}, \alpha_{opt}$ and $\beta_{opt}$. Then, the optimal supply voltage vector is obtained by solving

$$\min_{\boldsymbol{V_L}} \mathcal{E}(q_{\text{opt}}, \boldsymbol{V_L}, N_{\text{opt}}) \quad \text{s.t.} \quad p_{e,\text{opt}}(q_{\text{opt}}, \boldsymbol{V_L}, N_{\text{opt}}) < p_e^\star. \quad (13)$$

However, the problem is not convex and exhaustive search is not practical. To solve (13), we propose to adapt the *DE-Gear-Shift* dynamic programming method presented in [6]. This method consists in restricting $V(\ell)$ to a finite set and finding the path with the lowest cost in a trellis graph that satisfies a constraint. In our case, each path is associated with a vector $\boldsymbol{V_L}$ and an energy cost, and the constraint is the BER $p_e^\star$. To reduce complexity, paths are removed based on rules presented in [6].

## V. Numerical results

In this section, we consider four different protographs given in Table I, all with parameters $m = 2$, $n = 4$, and code rate $R = 0.5$. The protographs $S_{17}$ and $S_{36}$ were constructed using a genetic algorithm called Differential Evolution [11] that optimizes protographs for performance only. When using this method, the protographs were optimized by considering a large quantization level $q = 8$ in order to get a performance very close to the non-quantized decoder. We also consider the protographs $S_m$ and $S_c$ that were obtained in [14] by optimizing the decoder energy consumption.

For the four protographs, we set two BER targets of $p_e^\star = 10^{-3}$ (Case 1) and $p_e^\star = 10^{-6}$ (Case 2) at respectively $\xi^\star = 1.45$ dB and $\xi^\star = 1.7$ dB SNR, with $L = 25$ layered

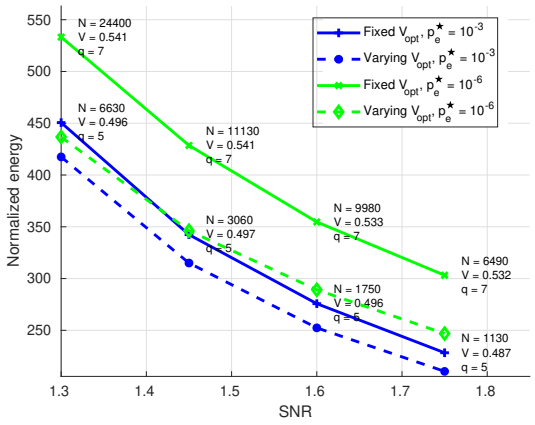


Fig. 1. Normalized energy values and optimal parameters obtained with the proposed optimisation methods, for different SNR and BER targets and when considering the protograph $S_{17}$.

decoding iterations. First, the minimum normalized energy $\mathcal{E}_{\text{fix}}$ is obtained by finding the optimum parameters $q_{\text{opt}}$, $V_{\text{opt}}$ and $N_{\text{opt}}$ using the method proposed in Section IV-C, configured to $I = 3$ CD iterations. Then, the voltage supply is optimized for each decoding iteration according to the method presented in Section IV-D, providing $\mathcal{E}_{\text{var}}$. Results are reported in Table I. For the sake of comparison, the nominal energy $\mathcal{E}_{\text{nom}}$ is provided where the supply voltage is fixed to its nominal value, *i.e.*, $V_{\text{nom}} = 0.8$V. To verify the accuracy of the proposed CD algorithm, we solved by exhaustive search the optimization problem (9) for Case 1 and protograph $S_{17}$, and found the same optimal values given in Table I.

Compared to the nominal energy, a reduction of $53\%$ to $60\%$ is obtained for all protographs when $V$ is fixed for all iterations. As expected, when $V$ is optimized at each iteration, the energy can be further reduced. For instance, depending on the protograph, an additional $20\%$ to $30\%$ energy reduction is obtained for Case 2, where $p_e^\star$ is lower. We further observe that, for all cases, the optimal code length $N_{\text{opt}}$ strongly depends on the considered protograph whereas the optimal supply voltage does not change much.

We now focus on the protograph $S_{17}$ to study the effect of the SNR on the optimization performance. Figure 1 shows the obtained normalized energy when the optimization is performed on different SNR values, for $p_e^\star = 10^{-3}$ and $p_e^\star = 10^{-6}$ BER targets. Both optimization results where $V$ is either fixed or varying at each iteration are shown. Two main observations can be made. First, only code length $N$ is significantly reduced when the SNR increase, while the other parameters $q$ and $V$ remain constant. Second, the energy is always reduced when changing the supply voltage per iteration. This reduction is also more important when the target BER is lower.

Finally, Figure 2 shows the BER with respect to SNR for protograph $S_{17}$, evaluated both from the finite-length method of Section III and from Monte-Carlo simulations. Based on the optimal parameters reported in Table I – Case 1, three cases are considered, corresponding to the following decod-

TABLE I
NORMALIZED ENERGY VALUES AND OPTIMAL PARAMETERS FOR EACH PROTOGRAPH WHEN CONSIDERING DIFFERENT BER AND SNR TARGETS.

| Protograph | Case 1: $p_e^\star = 10^{-3}$, $\xi^* = 1.45$ dB | | | | | | Case 2: $p_e^\star = 10^{-6}$, $\xi^* = 1.7$ dB | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $V_{\text{opt}}$ | $q_{\text{opt}}$ | $N_{\text{opt}}$ | $\mathcal{E}_{\text{fix}}$ | $\mathcal{E}_{\text{var}}$ | $\mathcal{E}_{\text{nom}}$ | $V_{\text{opt}}$ | $q_{\text{opt}}$ | $N_{\text{opt}}$ | $\mathcal{E}_{\text{fix}}$ | $\mathcal{E}_{\text{var}}$ | $\mathcal{E}_{\text{nom}}$ |
| $S_{17} = [2\ 3\ 1\ 1\ ;\ 0\ 1\ 4\ 1]$ | 0.497 | 5 | 3060 | 342.4 | 315 | 845.2 | 0.532 | 7 | 6490 | 303.2 | 210.4 | 710.8 |
| $S_{36} = [2\ 1\ 2\ 3\ ;\ 1\ 4\ 0\ 1]$ | 0.493 | 5 | 6170 | 377.13 | 347.1 | 931.1 | 0.539 | 6 | 7220 | 326.6 | 263.2 | 714.7 |
| $S_m = [3\ 2\ 1\ 2\ ;\ 0\ 1\ 1\ 4]$ | 0.491 | 5 | 7700 | 347.73 | 318.77 | 869.9 | 0.533 | 5 | 7410 | 270.48 | 214.7 | 606.8 |
| $S_c = [3\ 2\ 1\ 2\ ;\ 0\ 1\ 1\ 4]$ | 0.497 | 5 | 4070 | 365.93 | 332.74 | 890.2 | 0.533 | 6 | 5560 | 333.08 | 236.35 | 710.8 |

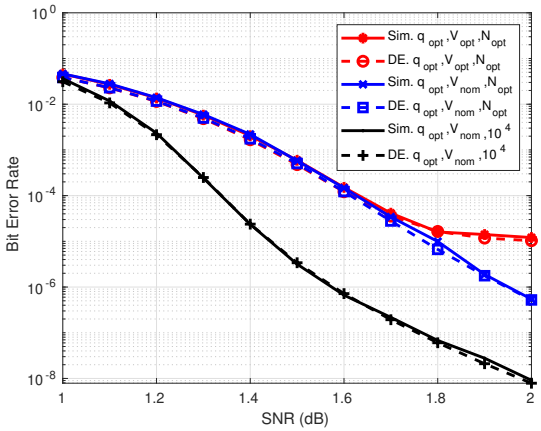


Fig. 2. BER with respect to SNR of the protograph $S_{17}$, evaluated from the finite-length DE method and from Monte Carlo simulations.

ing parameters: i) $(q_{\text{opt}}, V_{\text{opt}}, N_{\text{opt}})$, ii) $(q_{\text{opt}}, V_{\text{nom}}, N_{\text{opt}})$ iii) $(q_{\text{opt}}, V_{\text{nom}}, N = 10^4)$. Both $\alpha$ and $\beta$ are optimized using DE for each SNR value. The finite-length method of Section III accurately predicts the decoder BER in all cases.

## VI. CONCLUSION

In this paper, we introduced an energy model for faulty quantized Min-Sum decoders. We then proposed a method to optimize the number of quantization bits, the code length, and the memory fault probability, in order to minimize the energy consumption while satisfying a given decoding performance criterion. Simulation results show that using the optimal parameters greatly reduces the energy consumption while satisfying the performance criterion.

## REFERENCES


[1] P. Gupta, Y. Agarwal, L. Dolecek, N. Dutt, R. K. Gupta, R. Kumar, S. Mitra, A. Nicolau, T. S. Rosing, M. B. Srivastava *et al.*, "Underdesigned and opportunistic computing in presence of hardware variability," *IEEE Transactions on Computer-Aided Design of integrated circuits and systems*, vol. 32, no. 1, pp. 8–23, 2012.

[2] K. Ganesan, P. Grover, J. Rabaey, and A. Goldsmith, "On the total power capacity of regular-LDPC codes with iterative message-passing decoders," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 2, pp. 375–396, Feb 2016.

[3] T. T. Nguyen-Ly, K. Le, V. Savin, D. Declercq, F. Ghaffari, and O. Boncalo, "Non-surjective finite alphabet iterative decoders," in *IEEE International Conference on Communications*. IEEE, 2016, pp. 1–6.

[4] B. Smith, M. Ardakani, W. Yu, and F. R. Kschischang, "Design of irregular LDPC codes with optimized performance-complexity tradeoff," *IEEE Trans. on Communications*, vol. 58, no. 2, 2010.

[5] C.-H. Huang, Y. Li, and L. Dolecek, "Gallager B LDPC decoder with transient and permanent errors," in *IEEE Int. Symp. on Information Theory*, 2013, pp. 3010–3014.

[6] F. Leduc-Primeau, F. R. Kschischang, and W. J. Gross, "Modeling and energy optimization of LDPC decoder circuits with timing violations," *IEEE Trans. on Commun.*, vol. 66, no. 3, pp. 932–946, 2018.

[7] L. R. Varshney, "Performance of LDPC codes under faulty iterative decoding," *IEEE Transactions on Information Theory*, vol. 57, no. 7, pp. 4427–4444, 2011.

[8] A. Balatsoukas-Stimming and A. Burg, "Density evolution for min-sum decoding of LDPC codes under unreliable message storage," *IEEE Communications Letters*, vol. 18, no. 5, pp. 849–852, 2014.

[9] C. K. Ngassa, V. Savin, E. Dupraz, and D. Declercq, "Density evolution and functional threshold for the noisy min-sum decoder," *IEEE Trans. on Commun.*, vol. 63, no. 5, pp. 1497–1509, 2015.

[10] M. Karkooti and J. R. Cavallaro, "Semi-parallel reconfigurable architectures for real-time LDPC decoding," in *Int. Conf. on Information Technology: Coding and Computing*, 2004, pp. 579–585.

[11] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. on Information Theory*, vol. 47, no. 2, pp. 619–637, 2001.

[12] G. Liva and M. Chiani, "Protograph ldpc codes design based on exit analysis," in *IEEE GLOBECOM 2007-IEEE Global Telecommunications Conference*. IEEE, 2007, pp. 3250–3254.

[13] C. Roth, P. Meinerzhagen, C. Studer, and A. Burg, "A 15.8 pj/bit/iter quasi-cyclic ldpc decoder for ieee 802.11n in 90 nm CMOS," in *2010 IEEE Asian Solid-State Circuits Conference*, 2010, pp. 1–4.

[14] M. Yaoumi, E. Dupraz, F. Leduc-Primeau, and F. Guilloud, "Energy optimization of quantized min-sum decoders for protograph-based LDPC codes," *Annals of Telecommunications*, pp. 1–7, 2020.

[15] Y. Fang, G. Bi, Y. L. Guan, and F. C. Lau, "A survey on protograph LDPC codes and their applications," *IEEE Commun. Surveys & Tutorials*, vol. 17, no. 4, pp. 1989–2016, 2015.

[16] A. Balatsoukas-Stimming and A. Burg, "Density evolution for min-sum decoding of LDPC codes under unreliable message storage," *IEEE Communications Letters*, vol. 18, no. 5, pp. 849–852, 2014.

[17] E. Dupraz, F. Leduc-Primeau, and F. Gagnon, "Low-latency LDPC decoding achieved by code and architecture co-design," in *IEEE Int. Symp. on Turbo Codes & Iterative Information Processing*, 2018.

[18] F. Leduc-Primeau and W. J. Gross, "Finite-length quasi-synchronous LDPC decoders," in *9th Int. Symp. on Turbo Codes and Iterative Information Processing (ISTC)*, 2016, pp. 325–329.

[19] E. Sharon, S. Litsyn, and J. Goldberger, "Efficient serial message-passing schedules for LDPC decoding," *IEEE Transactions on Information Theory*, vol. 53, no. 11, pp. 4076–4091, 2007.

[20] A. D. Mauro, F. Conti, P. D. Schiavone, D. Rossi, and L. Benini, "Always-On 674$\mu$W@4GOP/s Error Resilient Binary Neural Networks With Aggressive SRAM Voltage Scaling on a 22-nm IoT End-Node," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 67, no. 11, pp. 3905–3918, 2020.

[21] R. G. Dreslinski, M. Wieckowski, D. Blaauw, D. Sylvester, and T. Mudge, "Near-threshold computing: Reclaiming Moore's law through energy efficient integrated circuits," *Proceedings of the IEEE*, vol. 98, no. 2, pp. 253–266, 2010.

[22] M. Milicevic and P. G. Gulak, "A multi-gb/s frame-interleaved LDPC decoder with path-unrolled message passing in 28-nm CMOS," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 26, no. 10, pp. 1908–1921, 2018.