ANALYSIS OF FINITE-ALPHABET ITERATIVE DECODERS UNDER PROCESSING ERRORS

Chu-Hsiang Huang and Lara Dolecek

EE Department, University of California, Los Angeles

ABSTRACT

It is widely recognized that emerging hardware technologies will be inherently unreliable. In this paper, we study the performance of finite-alphabet iterative decoders when implemented on noisy hardware built out of unreliable components. We derive a recursive expression for the error probability in terms of both the transmission noise and processing errors. We allow different components of the decoding algorithm associated with certain computational units (i.e., bit and check nodes of varying degrees in the underlying graph) to be implemented using a collection of processors with varying levels of processing error rates. Performance analysis and optimal resource allocation of a noisy Gallager E decoder is presented as an application example of our general derivation. Simulations demonstrate that the implementation of a noisy iterative decoder according to the proposed analysis-guided optimal resource allocation outperforms implementations based on uninformed resource allocation under the common resource budget.

Index Terms— Iterative decoders; Inference on Graphs; Noisy hardware; Optimal resource assignment

1. INTRODUCTION

Characterizing reliability of systems built out of unreliable hardware components is now emerging as a central issue in implementing modern signal processing and communications algorithms [1]. The original modeling of circuits made out of faulty gates experiencing transient errors is due to von Neumann [2]. This modeling has provided a basis for fault-tolerant computing [3, 4], an active research area concerned with placing additional (redundant) processing elements to combat transient errors.

Low-density parity check (LDPC) codes and their decoding algorithms were invented by Gallager in 1960s [5]. These codes offer significant improvement in data reliability and have now found widespread use in modern data communication and storage systems. Gallager proposed certain message passing (iterative) decoding algorithms; the two algorithms with binary messages later became known as the Gallager A and Gallager B algorithms, and the algorithm with finite-alphabet messages later became known as the Gallager E algorithm [6]. Due to the compact message representation, such decoders are particularly attractive in implementing LDPC-coded systems.

In this work, we study popular finite-alphabet message passing decoders *a la* Gallager implemented on noisy hardware. Interestingly, fault tolerance of faulty memories was recently improved with a novel use of LDPC codes in [7]. The capacity and certain concentration results for a noisy LDPC message passing decoder were computed in [8]. Message repetition was explored in [9] to mitigate computational errors arising in a noisy Gallager B decoder. Our preliminary results on regular and irregular LDPC codes with binary decoders implemented on noisy hardware were reported in [10] [11].

In this paper, we generalize the error analysis presented in [10] [11] for a noisy Gallager B decoder [5], a decoder that deals with binary messages, to a noisy M-alphabet iterative decoder. As in [11], the processors with different processing error rates are assigned to different computational units (i.e., variable nodes and check nodes). Section 2 discusses recent results and how are they relate to this work. We derive an expression for the iterative error probability of a general noisy iterative decoder in Section 3. In Section 4, as an application example of the generalized error analysis, we derive the optimal assignment of processors across different components of the decoder in a noisy Gallager E decoder (with alphabet size 3). In Section 5, we present simulation results that demonstrate performance improvement achieved by the optimal resource assignment over an uninformed selection. Section 6 provides the conclusions.

2. RELATION TO PRIOR WORK

Information-theoretic results for LDPC codes implemented on noisy decoders were derived in [8]. In that work, the performance of a noisy Gallager A decoder [5] implemented using a single type of processors was analyzed. Authors in [10] [11] extended this analysis to a noisy Gallager B decoder with multiple kinds of processors. The work presented here further generalizes the analysis of noisy Gallager A and B decoders (both binary) to finite-alphabet iterative decoders. The authors in [9] considered a similar error model of variable and check nodes for a noisy Gallager B decoder, but focused on a related problem of error recovery via message repetition.

3. ERROR ANALYSIS OF A NOISY FINITE ALPHABET ITERATIVE DECODER

We consider a low-density parity check (LDPC) code $\mathcal{C}$ described a bipartite graph $G = G(V, F, E)$. Here V denotes the set of variable nodes, F denotes the set of check nodes, and E is the set of edges connecting variable nodes and check node. Let $m = |F|$ and $n = |V|$. We also let $\mathcal{N}_v$ ($\mathcal{N}_c$) denote the set of checks (variables) incident to variable node v (check node c). Suppose d_v is the largest variable node degree, and suppose d_c is the largest check node degree. For $1 \leq i \leq d_v$, following popular notation [6], we let λ_i denote the fraction of edges in G that are connected to the variable nodes of degree i. Also, for $1 \leq j \leq d_c$, we denote by ρ_j the fraction of edges in G that are connected to the check nodes of degree j. It is useful to define $\lambda(t) := \sum_{i=1}^{d_v} \lambda_i t^{i-1}$ and $\rho(t) := \sum_{j=1}^{d_c} \rho_i t^{j-1}$, respectively, as the variable and check degree polynomials. Finally, the collection of graph codes whose bipartite graphs follow $\lambda(t), \rho(t)$ distributions is referred to as the (λ, ρ) ensemble.

We use von Neumann model [2], commonly adopted for the study of transient errors in hardware [12] and interpret it as follows. We assume that iterative messages exchanged between variable and check nodes are subject to transient errors, and that these errors are independent across different computational units and across different iterations of the decoder. Following the set-up presented in [11], we consider a general framework that allows processors of different error probabilities at different check nodes and variable nodes. We remark that in practice, these error probabilities depend on the implementation choice.

Let us assume that we have L types of processors available for the implementation of variable nodes and check nodes, and that that these processors are characterized by distinct error probabilities q_i, $1 \leq i \leq L$, ordered in the ascending order from best to worst. We collectively refer to the ascending ordering of q_i's as Q. We assume (as in stochastic computing [13]) that each bit/check node acts as a probabilistic channel wherein the input and the output are different with probability of the processing error of the associated processor.

For the noisy decoder, the message exchange is iteratively performed as follows: the message from variable node v to check node c at iteration i is denoted $m_{v \to c}^{(i)}$ and the message from check node c' to variable node v' at iteration i is denoted $m_{c' \to v'}^{(i)}$. It is useful to specify two auxiliary messages, $\hat{m}_{v \to c}^{(i)}$ and $\hat{m}_{c' \to v'}^{(i)}$, which respectively represent the outgoing messages of noiseless processors. The decoder operations are shown in Fig. 1. Functions Ψ_v and Ψ_c are local functions at variable and check nodes, respectively, and they depend on the implementation choice of the decoder under study. In the Gallager B decoder, for example, the mapping Ψ_v from $m_{c' \to v}^{(i)}$, $c' \in \mathcal{N}_v \setminus c$ to $\hat{m}_{v \to c}^{(i)}$ is the majority rule, while the mapping Ψ_c from $m_{v' \to c}^{(i)}$, $c' \in \mathcal{N}_c \setminus v$ to $\hat{m}_{c \to v}^{(i)}$ is the XOR operation. Let $\tau_{j,l}^{v}$ represent the fraction of edges that are connected to the variable nodes of degree j, $1 \leq j \leq d_v$, and error q_l. Likewise, we let $\tau_{k,r}^{c}$ represent the fraction of edges that are connected to the check nodes of degree k, $1 \leq k \leq d_c$, and error q_r. In this set-up, we are interested in estimating the performance of the algorithm,

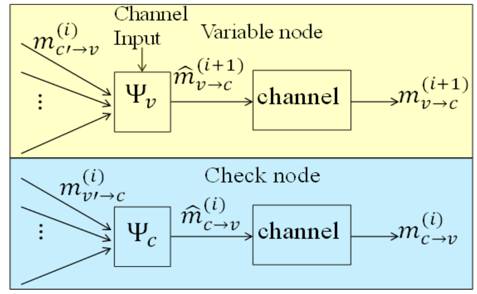

Fig. 1. General iterative decoder operations.

which we seek to express in terms of the density evolution of the propagated messages.

We remark that, as proved in [8], the density evolution under transient errors is independent of the transmitted codeword. We thus follow [8] and assume the transmission of the all-zero codeword in our analysis. As in the conventional (error-free) density evolution [6] and as in [8], and to make the analysis tractable, we moreover assume that the bipartite graph is sufficiently cycle-free.

It was shown in [8] that the well-known result on the concentration of message propagation [6] for LDPC decoding algorithms still holds true even in the presence of processing noise for finite alphabet iterative decoders. We thus focus on the average performance of our noisy decoder. We denote by p_i the average error in the messages from variable to check nodes in iteration i; i.e., $p_i = \mathbb{E}\left[\Pr\{m_{v \to c}^{(i)} \neq 0\}\right]$, where the average is taken over the (λ, ρ) code ensemble with degree and processor error rate distribution specified by $\tau_{j,l}^{v}$'s and $\tau_{k,r}^{c}$'s. In the first iteration, this error is simply the parameter p_0 of the transmission channel.

We now derive a recursive expression for the error rate p_{i+1} for a general finite-alphabet noisy decoder. The derivations generalize the previous result in [11] obtained for a noisy Gallager B decoder (under a binary message alphabet).

Denote the message alphabet by Φ with $|\Phi| = M$. (If $\Phi = \{0, 1\}$ the alphabet is binary.) The error rate p_{i+1} of a general finite-alphabet noisy iterative decoder is derived from the probability mass function (PMF) of noiseless messages, $\Pr\{\hat{m}_{v \to c}^{(i)}\}$ and $\Pr\{\hat{m}_{c \to v}^{(i)}\}$. We assume that when a processor outputs an erroneous value, it does so equiprobably over all choices,

$$\Pr\{m_{x \to y}^{(i)} = \alpha | \text{processor made an error}, \hat{m}_{x \to y}^{(i)} = \gamma\}$$
$$= \Pr\{m_{x \to y}^{(i)} = \beta | \text{processor made an error}, \hat{m}_{x \to y}^{(i)} = \gamma\},$$

for all $\alpha, \beta \neq \gamma$, $\alpha, \beta, \gamma \in \Phi$ for $x \to y$ being either $c \to v$ or $v \to c$.

Then, the PMF of variable-to-check message $m_{v \to c}^{(i+1)}$ in the noisy decoder can be expressed as a function of variable-to-check message $\hat{m}_{v \to c}^{(i+1)}$ of the noise-free decoder,

$$\begin{aligned} &\Pr\{m_{v \to c}^{(i+1)} = \alpha\} \\ &= (1 - q_l) \Pr\{\hat{m}_{v \to c}^{(i+1)} = \alpha\} + \sum_{\beta \in \Phi \setminus \alpha} \frac{q_l}{M-1} \Pr\{\hat{m}_{v \to c}^{(i+1)} = \beta\} \\ &= \left(1 - \frac{M q_l}{M-1}\right) \Pr\{\hat{m}_{v \to c}^{(i+1)} = \alpha\} + \frac{q_l}{M-1}. \end{aligned} \quad (1)$$

Similarly, after some algebra, we have

$$\Pr\{m^{(i)}_{c\to v}=\alpha\}=\left(1-\frac{Mq_l}{M-1}\right)\Pr\{\hat{m}^{(i)}_{c\to v}=\alpha\}+\frac{q_r}{M-1}. \tag{2}$$

With (1) and (2), one can then recursively compute the PMF of the messages in the noisy decoder as we now show. Let $deg(\cdot)$ denote the degree and $\varepsilon(\cdot)$ denote the error rate assigned to a particular node. The average PMF of the check-to-variable messages at iteration i can be derived by taking the average of (2) w.r.t. processor error rate distribution τ^c_{k,q_r},

$$\begin{aligned}&\mathbb{E}[\Pr\{m^{(i)}_{c'\to v}=\alpha\}] \qquad (3)\\ &=\sum_{k=1}^{d_c}\sum_{r=1}^{L}\tau^c_{k,q_r}\mathbb{E}_{deg(c')=k,\varepsilon(c')=q_r}[\Pr\{m^{(i)}_{c'\to v}=\alpha\}]\\ &=\sum_{k=1}^{d_c}\sum_{r=1}^{L}\tau^c_{k,q_r}((1-\frac{Mq_l}{M-1})\Pr\{\hat{m}^{(i)}_{c'\to v}=\alpha\}+\frac{q_r}{M-1}).\end{aligned}$$

We note that $\Pr\{\hat{m}^{(i)}_{c'\to v}=\alpha\}$ is computed as in the noiseless case, by applying the mapping Ψ_c to the inputs $\Pr\{m^{(i)}_{v'\to c'}=\beta\},\beta\in\Phi$, $v'\in\mathcal{N}_c\setminus v$ (see also Fig. 1).

Next, we derive the average PMF of the messages from the variable nodes to the check nodes,

$$\begin{aligned}&\mathbb{E}[\Pr\{m^{(i+1)}_{v\to c}=\alpha\}] \qquad (4)\\ &=\sum_{j=1}^{d_v}\sum_{l=1}^{L}\tau^v_{j,q_l}\mathbb{E}_{deg(v)=k,\varepsilon(v)=q_l}[\Pr\{m^{(i+1)}_{v\to c}=\alpha\}]\\ &=\sum_{j=1}^{d_v}\sum_{l=1}^{L}\tau^v_{j,q_l}((1-\frac{Mq_l}{M-1})\Pr\{\hat{m}^{(i+1)}_{v\to c}=\alpha\}+\frac{q_l}{M-1}).\end{aligned}$$

Similarly, $\Pr\{\hat{m}^{(i+1)}_{v\to c}=\alpha\}$ is computed as the output of Ψ_v with inputs $\Pr\{m^{(i)}_{c'\to v}=\beta\},\beta\in\Phi$ and $c'\in\mathcal{N}_v\setminus c$, analogously to the noiseless case (see also Fig. 1). Therefore, by computing $\Pr\{m^{(i)}_{v'\to c'}=\beta\}$ from $\Pr\{\hat{m}^{(i)}_{v'\to c'}=\beta\}$, then $\Pr\{\hat{m}^{(i)}_{c'\to v}=\delta\}$ from $\Pr\{m^{(i)}_{v'\to c'}=\gamma\}$, then $\Pr\{m^{(i)}_{c'\to v}=\delta\}$ from $\Pr\{\hat{m}^{(i)}_{c'\to v}=\delta\}$, and then $\Pr\{\hat{m}^{(i+1)}_{v\to c}=\alpha\}$ from $\Pr\{m^{(i)}_{c'\to v}=\varepsilon\}$, and then $\Pr\{m^{(i+1)}_{v\to c}=\alpha\}$ from $\Pr\{\hat{m}^{(i+1)}_{v\to c}=\alpha\}$ we arrive at the recursive expression relating $\Pr\{m^{(i+1)}_{v\to c}=\alpha\}$ to $\Pr\{m^{(i)}_{v'\to c'}=\beta\}$, where $\alpha,\beta,\gamma,\delta,\varepsilon\in\Phi$.

For an error-free iterative decoder, the overall error at iteration i, $\sum_{\alpha\in\Phi\setminus 0}\Pr\{\hat{m}^{(i)}_{v\to c}=\alpha\}$ converges to zero for small enough p_0. In contrast, for a noisy decoder this overall error converges to some strictly positive quantity which we call the *residual error* p. (or final BER). We note that p is at least $\sum_{j=1}^{d_v}\sum_{l=1}^{L}\tau^v_{j,l}\frac{q_l}{M-1}$.

It can be shown that p improves with higher variable node degree (see also [10]).

3.1. Noisy Gallager E Decoder

As an illustrative example, we consider a noisy Gallager E decoder [6]. The noisy Gallager E decoder has $\Phi=\{-1,0,+1\}$ and $M=3$. Denote the probabilities $\Pr\{\hat{m}^{(i)}_{c\to v}=\beta\}$ and $\Pr\{\hat{m}^{(i)}_{v\to c}=\beta\}$ of the messages of the noiseless decoder by $\hat{p}^{(i)}_{c,\beta}$ and $\hat{p}^{(i)}_{v,\beta}$, respectively, for $\beta\in\Phi$. Also denote the noisy decoder message PMFs by $p^{(i)}_{c,\beta}$ and $p^{(i)}_{v,\beta}$.

We wish to derive a recursive expression for $p^{(i+1)}_{v,\beta}$ in terms of $p^{(i)}_{v,\beta}$. From the analysis of the nominal error-free decoder [6], the noiseless check messages are

$$\begin{aligned}\hat{p}^{(i)}_{c,1}&=\frac{1}{2}[(p^{(i)}_{v,1}+p^{(i)}_{v,-1})^{d_c-1}+(p^{(i)}_{v,1}-p^{(i)}_{v,-1})^{d_c-1}],\\ \hat{p}^{(i)}_{c,-1}&=\frac{1}{2}[(p^{(i)}_{v,1}+p^{(i)}_{v,-1})^{d_c-1}-(p^{(i)}_{v,1}-p^{(i)}_{v,-1})^{d_c-1}],\\ \hat{p}^{(i)}_{c,0}&=1-(1-p^{(i)}_{v,0})^{d_c-1}.\end{aligned} \tag{5}$$

Thus, $p^{(i)}_{c,\alpha}$ is

$$p^{(i)}_{c,\alpha}=\sum_{k=1}^{d_c}\sum_{r=1}^{L}\tau^c_{k,q_r}((1-\frac{3q_l}{2})\hat{p}^{(i)}_{c,\alpha}+\frac{q_r}{2}), \tag{6}$$

where $\hat{p}^{(i)}_{c,\alpha}$ is derived from (5) and uses $M=3$.

From the standard error-free decoder analysis, we can derive the noiseless variable message, $\hat{p}^{(i+1)}_{v,\alpha}$, from $p^{(i)}_{c,\beta}$, $\alpha,\beta\in\Phi$ as shown on the top of the next page. Then, $p^{(i+1)}_{v,\alpha}$ is given by

$$p^{(i+1)}_{v,\alpha}=\sum_{j=1}^{d_v}\sum_{l=1}^{L}\tau^v_{j,q_l}((1-\frac{3q_l}{2})\hat{p}^{(i+1)}_{v,\alpha}+\frac{q_l}{2}), \tag{10}$$

where $\hat{p}^{(i)}_{v,\alpha}$ is derived by (9). Note that we put $M=3$ in (4) to derive (10). Then we have $p^{(i+1)}_{v,\alpha}$ as functions of $p^{(i)}_{v,\beta}$ from (5) to (10).

4. OPTIMAL ASSIGNMENT OF PROCESSORS

The optimal assignment of processors to check and variable nodes with different degrees can be derived by minimizing the residual error with respect to τ^v_{j,q_l} and τ^c_{k,q_r}

This minimization problem may in general be difficult to solve. Fortunately, when the channel error rate is small and the error rates of constituent processors are also sufficiently small, one can show that for a code that has all variable nodes of degree at least 3, by ignoring the second order terms involving $\hat{p}^{(0)}_{v,0}$, $\hat{p}^{(0)}_{v,-1}$, $\hat{p}^{(i)}_{c,0}$, and $\hat{p}^{(i)}_{c,-1}$, the minimization problem reduces to minimizing $\sum_{j=1}^{d_v}\sum_{l=1}^{L}\tau^v_{j,q_l}\cdot q_l$, which then simply becomes a linear programming problem.

We now study the optimal assignment of processors that offers different reliabilities of different processing nodes of the decoder. An optimal assignment is the one that minimizes the residual error p. Suppose that for every $1\le l\le L$, the cost of implementing a variable node processor of degree j for $3\le j\le d_v$ (we assume that there is no variable node of degree $1,2$) and error q_l for $1\le l\le L$ is $w^v_{j,l}$, and for every check node of degree k and error q_r for $1\le k\le d_c$ and $1\le r\le L$, the cost is $w^c_{k,r}$.

Suppose we fix the maximum allowable cost W. The total number of variable nodes of degree j and processing error q_l is $Z\tau^v_{j,l}/j$ and the total number of check nodes of degree k

$$\hat{p}_{v,0}^{(i+1)} = p_{v,0}^{(0)} \sum_{(\alpha,\beta):\alpha-\beta=0} \binom{d_v-1}{\alpha,\alpha,d_v-1-2\alpha} \cdot (p_{c,1}^{(i)})^{\alpha}(p_{c,-1}^{(i)})^{\alpha}(p_{c,0}^{(i)})^{d_v-1-2\alpha} + p_{v,1}^{(0)} \sum_{(\alpha,\beta):\alpha-\beta=-w^{(i)}} \binom{d_v-1}{\alpha,\beta,d_v-1-\alpha-\beta}$$

$$\cdot (p_{c,1}^{(i)})^{\alpha}(p_{c,-1}^{(i)})^{\beta}(p_{c,0}^{(i)})^{d_v-1-\alpha-\beta} + p_{v,-1}^{(0)} \sum_{(\alpha,\beta):\alpha-\beta=w^{(i)}} \binom{d_v-1}{\alpha,\alpha,d_v-1-\alpha-\beta} \cdot (p_{c,1}^{(i)})^{\alpha}(p_{c,-1}^{(i)})^{\beta}(p_{c,0}^{(i)})^{d_v-1-\alpha-\beta}$$

$$\hat{p}_{v,-1}^{(i+1)} = p_{v,0}^{(0)} \sum_{(\alpha,\beta):\alpha-\beta>0} \binom{d_v-1}{\alpha,\beta,d_v-1-\alpha-\beta} \cdot (p_{c,1}^{(i)})^{\alpha}(p_{c,-1}^{(i)})^{\beta}(p_{c,0}^{(i)})^{d_v-1-\alpha-\beta} + p_{v,1}^{(0)} \sum_{(\alpha,\beta):\alpha-\beta<-w^{(i)}} \binom{d_v-1}{\alpha,\beta,d_v-1-\alpha-\beta}$$

$$\cdot (p_{c,1}^{(i)})^{\alpha}(p_{c,-1}^{(i)})^{\beta}(p_{c,0}^{(i)})^{d_v-1-\alpha-\beta} + p_{v,-1}^{(0)} \sum_{(\alpha,\beta):\alpha-\beta<w^{(i)}} \binom{d_v-1}{\alpha,\beta,d_v-1-\alpha-\beta} \cdot (p_{c,1}^{(i)})^{\alpha}(p_{c,-1}^{(i)})^{\beta}(p_{c,0}^{(i)})^{d_v-1-\alpha-\beta},$$

$$\hat{p}_{v,1}^{(i+1)} = 1 - \hat{p}_{v,1}^{(i+1)} - \hat{p}_{v,0}^{(i+1)}. \tag{9}$$

and processing error q_r is $Z\tau_{k,r}^c/k$, where Z denotes the total number of edges in the LDPC graph.

By a previous discussion, our aim is then to solve the following optimization problem:

$$\text{Minimize: } \sum_{j=3}^{d_v}\sum_{l=1}^{L} \tau_{j,q_l}^v \cdot q_l$$

$$\text{Subject to: } Z\sum_{j=3}^{d_v}\sum_{l=1}^{L} \frac{\tau_{j,l}^v w_{j,l}^v}{j} + Z\sum_{k=1}^{d_c}\sum_{r=1}^{L} \frac{\tau_{k,r}^c w_{k,r}^c}{k} \le W,$$

$$\sum_{l=1}^{L} \tau_{j,l}^v = \lambda_j, \sum_{r=1}^{L} \tau_{k,r}^c = \rho_k. \tag{11}$$

We observe that the objective function and all constraints in the preceding optimization problem are linear in terms of the variables $\tau_{j,l}^v$'s and $\tau_{k,r}^c$'s so that efficient algorithms can be used to solve this linear programming problem.

It is interesting to note that the objective function in (11) does not depend on $\tau_{k,r}^c$'s. As a result, for the codes without variable node degree less than 3, all the check nodes admit the least expensive processors (of error parameter q_L) in the optimal solution.

5. SIMULATION RESULTS

In this section we report on experimental results. We tested the performance of two irregular codes proposed by MacKay (codes are available at [14]). Code 1 has 9972 variable nodes of which 9141 nodes have degree $j_1 = 3$ and 831 nodes have degree $j_2 = 9$. The code has $m = 4986$ check nodes all with degree 7. Code 2 has 1920 variable nodes of which 640 nodes have degree 14 and 1280 nodes have degree 18. This code has 5760 check nodes in total, 1280 nodes with degree 4 and 4480 nodes with degree 6.

In our MATLAB simulations, we considered the case with two kinds of available processors with error rates $q_1 = 10^{-4}$ and $q_2 = 10^{-3}$. Channel error was 2×10^{-3}. We assigned the cost of 10 (resp. cost of 1) to the variable nodes of degree j_1 and error q_1 (resp. error q_2), and we assigned the cost of 100 (resp. cost of 10) to the variable nodes of degree j_2 and error q_1 (resp. error q_2). For code 1, we assigned cost of 10 (resp. cost of 1) to the check node with error q_1(resp. error q_2). For code 2, the cost assignment is the same over all nodes.

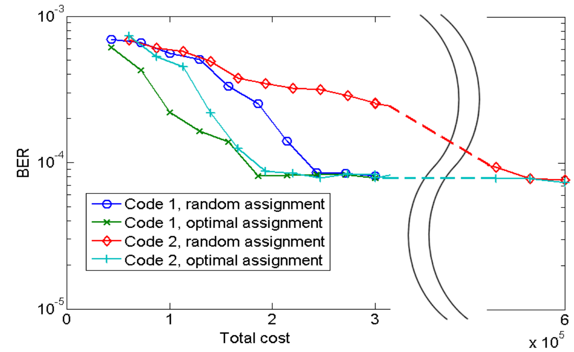

Fig. 2. The performance comparison of optimal assignment and random assignment for the noisy Gallager E decoder.

Two kinds of noisy Gallager E decoders are simulated: one based on the analysis-guided processor assignment and another one based on uninformed (random) assignment of faulty processors. We plotted the resulting BERs for the two codes in Fig. 2 for a range of total costs (a part of the plot is suppressed to highlight the difference between the allocation choices). The simulation results are presented for the finite-length case but nonetheless corroborate the analysis (valid for the infinite-length case) and demonstrate the improvement in the performance of the decoder when processors are assigned based on the solution of (11). The improvement is more pronounced for code 2, which has a higher fraction of check nodes. The results show that the same BER (at about 2×10^{-4}) can be obtained by optimal assignment at about $\frac{1}{2}$ of the cost of the random assignment.

6. CONCLUSION

In this paper, we studied a noisy finite-alphabet iterative decoder implemented on hardware built out of processors with different error rates. We derived an iterative expression for the error rates as a function of both transmission noise and processing noise. As an example, we formulated the optimal processor assignment for a noisy Gallager E decoder. Illustrative examples showed improvement in the final BER when processors were optimally assigned. Our results may serve as a basis of a future study of code design and decoding algorithms implemented on noisy hardware.

7. REFERENCES

[1] B. Shim *et al.*, "Reliable low-power digital signal processing via reduced precision redundancy," *IEEE Transactions on VLSI Systems*, vol. 12, no. 5, pp. 497–510, May 2004.

[2] J. von Neumann, "Probabilistic logic and the synthesis of reliable organisms from unreliable components," *Automata Studies*, pp. 43–98, 1956.

[3] S. Sundaram and C. Hadjicostis, "Fault-tolerant convolution via chinese remainder codes constructed from non-coprime moduli," *IEEE Transactions on Signal Processing*, vol. 56, no. 9, pp. 4244 –4254, Sep. 2008.

[4] J.-Y. Jou and J. Abraham, "Fault-tolerant matrix arithmetic and signal processing on highly concurrent computing structures," *Proceedings of the IEEE*, vol. 74, no. 5, pp. 732 – 741, May 1986.

[5] R. G. Gallager, "Low-Density Parity-Check Codes," *Cambridge, MA: MIT Press*, 1963.

[6] T. J. Richardson and R. L. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 599 –618, Feb. 2001.

[7] B. Vasic and S. Chilappagari, "An information theoretical framework for analysis and design of nanoscale fault-tolerant memories based on low-density parity-check codes," *IEEE Transactions on Circuits and Systems I*, vol. 54, no. 11, pp. 2438 –2446, Nov. 2007.

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

[9] F. Leduc-Primeau and W. J. Gross, "Faulty gallager-b decoding with optimal message repetition," in *Allerton Conference on Communication, Control, and Computing*, 2012.

[10] S. M. S. Tabatabaei *et al.*, "Probabilistic analysis of Gallager B faulty decoder," in *IEEE International Conference on Communication*, 2012.

[11] ——, "Optimal design of a Gallager B noisy decoder for irregular LDPC codes," *IEEE Communication Letters*, to be published.

[12] P. Beckmann and B. Musicus, "Fast fault-tolerant digital convolution using a polynomial residue number system," *IEEE Transactions on Signal Processing*, vol. 41, no. 7, Jul. 1993.

[13] N. R. Shanbhag *et al.*, "Stochastic computation," in *ACM/IEEE Design Automation Conference*, 2010, pp. 859–864.

[14] D. MacKay, "Encyclopedia of sparse graph codes," *Available online: http://www.inference.phy.cam.ac.uk/mackay/codes/data.html*.