Stochastic Resonance Decoding for Quantum LDPC Codes

Nithin Raveendran*, Priya J. Nadkarni†, Shayan Srinivasa Garani†, and Bane Vasić*
*Department of Electrical and Computer Engineering, University of Arizona, Tucson, AZ 85721, USA.
{nithin, vasic}@ece.arizona.edu
†Department of Electronic Systems Engineering, Indian Institute of Science, Bengaluru, 560012, India.
{priya, shayan.gs}@dese.iisc.ernet.in

***Abstract*—We introduce a stochastic resonance based decoding paradigm for quantum codes using an error correction circuit made of a combination of noisy and noiseless logic gates. The quantum error correction circuit is based on iterative syndrome decoding of quantum low-density parity check codes, and uses the positive effect of errors in gates to correct errors due to decoherence. We analyze how the proposed stochastic algorithm can escape from short cycle trapping sets present in the dual containing Calderbank, Shor and Steane (CSS) codes. Simulation results show improved performance of the stochastic algorithm over the deterministic decoder.**

***Index Terms*—Faulty hardware, Gallager-B decoding, quantum LDPC codes, unreliable gates.**

I. Introduction

Quantum error correction (QEC) codes [1] are vital for protecting quantum information bits (qubits) from quantum errors for applications in quantum storage and transmission channels. Unlike classical bits, quantum states are intrinsically fragile. Interaction of the environment often results in undesirable unitary evolution of quantum states or unwanted measurements. This quantum noise, conventionally termed as decoherence, poses significant challenges for practical implementation of quantum-based computation and communication systems.

A significant milestone in the design and implementation of QEC codes is the quantum LDPC code introduced by Mackay *et al.* [2]. The sparseness property of the quantum LDPC code requires that only a small number of interactions per qubit are needed during the decoding process, thereby, facilitating efficient fault tolerant decoding. Research results starting from Mackay's work resulted in a myriad of quantum LDPC codes that were designed and analyzed by relying on classical LDPC codes [3]. Quantum LDPC codes requiring either small fraction of entangled qubits [4] or a fraction of more reliable qubits [5] have been proposed as entanglement assisted (EA) quantum codes. EA quantum LDPC codes require a single pre-shared entangled qubit (e-bit). Maintaining a noiseless entangled state is not a trivial task, making the e-bit a valuable resource. Instead, quantum LDPC codes without entanglement can be constructed based on the stabilizer formalism [6], [7] using the Calderbank, Shor and Steane (CSS) code construction. The stabilizers must satisfy the commutativity constraint which in turn is equivalent to the symplectic inner product constraint on the constituent classical LDPC codes. Bicycle codes proposed by Mackay [2] gives relatively simple, yet powerful dual containing CSS codes. These codes have shown excellent performance in comparison to other variants and constructions of CSS codes [8]. However, the symplectic criterion introduces a large number of cycles of length 4 (short cycles) in their Tanner graphs, thus making the use of iterative decoding algorithms highly suboptimal. The approach prevalent in quantum LDPC coding literature is to ignore the presence of short cycles and use the classical belief propagation (BP) algorithm. An excellent survey by Babar *et al.* [8] provides numerous examples of failures of such naive decoding approaches. Unfortunately, dealing with short cycles using the standard BP algorithm leads to very shallow error floors.

A natural way of dealing with loops in the graphical model of a code is to rely on loopy inference algorithms. There have been prior approaches for dealing with short cycles. For example, the cluster variation method by Kikuchi and its extensions such as *region graph method* proposed by Yedidia *et al.* [9] have large complexity and are suitable only for short codes. A modified belief propagation algorithm for classical LDPC decoders [10], [11] over graphs with isolated short cycles was recently proposed. However, modifications of this algorithm for nested cycles would be needed to handle generalized dual containing CSS like codes and its variants. These short cycles form harmful trapping sets for iterative decoding algorithm. The use of stochastic resonance enables the decoder to escape from trapping sets and converge to a codeword. This idea introduced in the context of classical codes [12] can be naturally extended to the quantum decoding paradigm. Similar to noisy classical gates resulting from technology shrinkage and low supply voltages within the mesoscopic physical regime, the state-of-the-art quantum logic gates using photonics have fidelities that are approximately 0.91 [13]. This inherent noisy nature of classical/quantum gates can help the classical/quantum decoder to work better at certain operating conditions where noise can benefit the decoding process. Note that the decoding algorithm proposed in this paper is entirely classical, making use of noisy classical gates for improved decoding performance in a stochastic decoding paradigm.

The paper is organized as follows. In Section II, we introduce quantum LDPC codes and focus on the properties of dual containing CSS codes and their classical analogy. We also introduce notations for describing the iterative decoding algorithm. We describe the decoding algorithm with an analysis in Section III. This is followed by simulation results for comparing the decoding performance in Section IV. Concluding remarks and future research directions are given in Section V. For completeness, the definitions and examples related to quantum codes are relegated to the Appendix.

II. Preliminaries

In this section, we introduce quantum LDPC codes based on the stabilizer formalism and dual containing CSS codes. We then introduce general notations for a syndrome based iterative decoder.

A. Stabilizer Formalism and Quantum LDPC Codes

A $[[n, k]]$ stabilizer code [6] is a 2^k-dimensional subspace $\mathcal{C}$ of a 2^n-dimensional space stabilized by a commutative group S of stabilizers of order $n-k$. By mapping each element (Pauli operators I, X, Z or Y) of generators of S to a binary tuple as follows: $\mathrm{I} \to (0,0)$, $\mathrm{X} \to (1,0)$, $\mathrm{Z} \to (0,1)$, $\mathrm{Y} \to (1,1)$, we obtain the rows of the $(n-k) \times 2n$ check matrix H_c given by

$$H_c = \left[H_{\mathrm{X}} \ H_{\mathrm{Z}}\right], \tag{1}$$

where H_{X} and H_{Z} represent binary matrices for bit flip and phase flip operators respectively.

In this paper, we consider an important class of stabilizer code namely the CSS codes [1], constructed from two classical codes $\mathcal{C}_1$ and $\mathcal{C}_2$, where $\mathcal{C}_2^\perp \subseteq \mathcal{C}_1$. Let the corresponding parity check matrices be H_1 and H_2, then the check matrix of CSS code has the form

$$H_c = \begin{bmatrix} H_1 & 0 \\ 0 & H_2 \end{bmatrix}.$$

The commutativity criterion on the stabilizers is satisfied only when $H_1 H_2^{\mathrm{T}} = 0$.
Restricting $\mathcal{C}_2 = \mathcal{C}_1$ gives us a $[[n, 2k-n]]$ dual containing CSS code with $H_1 = H_2 = H$, $HH^{\mathrm{T}} = 0$ and $\dim(\mathcal{C}_1) = k$, resulting in a simple form as follows:

$$H_{\mathcal{C}} = \begin{bmatrix} H & 0 \\ 0 & H \end{bmatrix}.$$

In a dual containing CSS code $\mathcal{C}$, each codeword is a superposition of all the states in one of the cosets of $\mathcal{C}_1^\perp$ in $\mathcal{C}_1$. Let w_i denote the coset leader of the coset i as shown in Fig. 1. The code is given by

$$\mathcal{C} = \left\{ |c_{w_i}\rangle \mid i \in [2^{2k-n}] \right\},$$

where $|c_{w_i}\rangle = 2^{-\frac{1}{2}\dim(\mathcal{C}_1{}^\perp)} \sum_{z \in \mathcal{C}_1{}^\perp} |w_i + z\rangle$, and $[a]$, $a > 0$, denotes the set of natural numbers not larger than a.

B. Channel Model

We consider a quantum depolarizing channel characterized by the depolarizing probability p. For binary decoding, depolarizing channel is isomorphic to two independent binary symmetric channels (BSCs), a simplified model ignoring the correlation between bit flip error X and phase flip error Z. The BSCs for X and Z error have a cross-over probability of $2p/3$ [2]. Hence, an error on the n qubits from the depolarizing channel can be expressed as a binary error vector of length $2n$ in the form $[\mathbf{e}_{\mathrm{Z}} \ \mathbf{e}_{\mathrm{X}}]$, where $\mathbf{e}_{\mathrm{Z}}$ and $\mathbf{e}_{\mathrm{X}}$ represent Z and X errors and are vectors of length n. This gives the syndrome measurement as $H_c[\mathbf{e}_{\mathrm{Z}} \ \ \mathbf{e}_{\mathrm{X}}]^{\mathrm{T}}$. For simulations using dual containing CSS codes, we may obtain syndrome measurements as $H\mathbf{e}_{\mathrm{X}}^{\mathrm{T}}$ and $H\mathbf{e}_{\mathrm{Z}}^{\mathrm{T}}$. Hence, X and Z errors can be decoded independently by iterative algorithms over a Tanner graph corresponding to H.

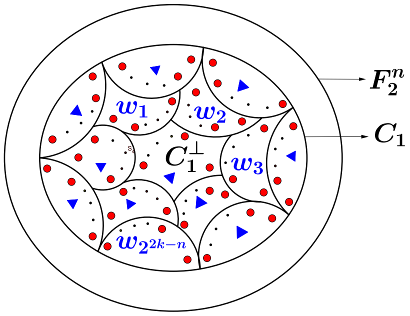

Fig. 1. Coset structure of a dual containing CSS code: The dual code $C_1^\perp$ is a subspace of code C_1 due to the dual containing property. $\mathcal{C}_1$ is partitioned into 2^{2k-n} cosets of $\mathcal{C}_1^\perp$. Every element of the coset i, where $i \in [2^{2k-n}]$ can be expressed as a sum of the coset leader w_i and an element in $\mathcal{C}_1^\perp$. ▲ denotes the coset leaders w_1, w_2, …, $w_{2^{2k-n}}$, and ● denotes the other elements in the coset.

C. Tanner Graph and Message Passing Updates

Although our decoding paradigm is applicable to any class of quantum codes, in this paper we consider a (γ, ρ)-regular binary LDPC code $\mathcal{C}$ of length n and dimension k, with code rate $R = k/n \geq 1 - \gamma/\rho$ and parity check matrix H. The parity check matrix is the bi-adjacency matrix of a bipartite (Tanner) graph $G = (V \cup C, E)$, where V represents the set of n variable nodes, C is the set of $m = n\gamma/\rho$ check nodes and E is the set of $n\gamma$ edges. Each matrix element $H_{c,v} = 1$ indicates that there is an edge between nodes $c \in C$ and $v \in V$, which are referred as neighbors. Let $\mathcal{N}_v$ ($\mathcal{N}_c$) be the set of neighbors of the variable node v (check node c). Then, $|\mathcal{N}_v| = \gamma$, $\forall v \in V$ and $|\mathcal{N}_c| = \rho$, $\forall c \in C$, where $|\cdot|$ denotes cardinality. In irregular LDPC codes, nodes do not necessarily have the same number of neighbors.

Since upon measurement, the quantum state collapses to a state consistent with measurement (often referred to as 'wave function collapse'), a syndrome-based decoder is used instead of the familiar codeword based decoder. Let $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ denote the code coordinates of a n-qubit quantum LDPC code, where x_v represents the binary value associated with the variable node v. An error vector $\mathbf{e} = (e_1, e_2, \ldots, e_n)$ is superimposed to the codeword by the depolarizing channel. The observed syndrome $\mathbf{s} = (s_1, s_2, \ldots, s_m)$ is the input to the syndrome based decoder. The syndrome measurement is based only on the check matrix H and the error vector $\mathbf{e}$. The decoding aim is to find the most likely error pattern of length n given the observed syndrome.

The message passed from a check node c to a variable node v in the t^{th} iteration is denoted by $\mu_{c\to v}^{(t)}$. The message passed from a variable node v to the check node c is denoted by $\nu_{v\to c}^{(t)}$.

Let $\mathbf{m}^{(t)} = \mu_{\mathcal{N}_v \setminus c \to v}^{(t)}$ denote all incoming messages to the variable node v except a message from the check node c. Similarly, $\mathbf{n}^{(t)} = \nu_{\mathcal{N}_c \setminus v \to c}^{(t)}$ denote all incoming messages to the check node c except from the variable node v. Finally, we denote by $\mathbf{l}^{(t)} = \mu_{\mathcal{N}_v \to v}^{(t)}$, messages incoming to v, and by $\mathbf{i}^{(t)} = \nu_{\mathcal{N}_c \to c}^{(t)}$ messages incoming to c in the t^{th} iteration (the notation is adopted from [12]).

The variable node messages are initialized to the value corresponding to the zero error vector $\mathbf{0}$, and then updated according to the variable and check node rules namely Φ and Ψ in each iteration. The function $\Psi : \{\mathcal{M}\}^{\rho-1} \times \mathcal{S} \to \mathcal{M}$ is a check node update function, and $\mathcal{M}$ and $\mathcal{S}$ are, respectively, the message and check-value alphabets. A check node c with degree ρ and observed syndrome s_c, is updated as $\mu_{c\to v}^{(t)} = \Psi(\mathbf{n}^{(t-1)}, \mathbf{R}(s_c))$, where $\mathbf{R}(s_c)$ is the reliability of the syndrome measurement of the check c. The function $\Phi : \mathcal{Y} \times \{\mathcal{M}\}^{\gamma-1} \to \mathcal{M}$ is used for updating outgoing message of a variable node v with degree γ. Variable node messages from v are updated as $\nu_{v\to c}^{(t)} = \Phi(y, \mathbf{m}^{(t)})$, where y is the reliability of the variable to take value zero, and $\mathcal{Y}$ is the variable node reliability alphabet. The alphabets $\mathcal{M}$, $\mathcal{Y}$ and $\mathcal{S}$ depend on a decoder type and quantum channel model as we describe later in the text. The "strength" of the variable node v in t^{th} iteration, $\lambda_v^{(t)}$, is used to decide the value of error $\hat{e}_v^{(t)}$ acted on the v^{th} bit position. Let $\mathbf{l} \in \{\mathcal{M}\}^{\gamma}$ be an unordered γ-tuple representing all incoming messages to the variable node v from its neighbors. For the t^{th} iteration, $\lambda_v^{(t)} = \hat{\Phi}(y, \mathbf{l}^{(t)})$ and the error value is calculated based on the sign of $\lambda_v^{(t)}$ as $\hat{e}_v^{(t)} = \mathbb{1}_{\lambda_v^{(t)} < 0}$. An estimate of the syndrome at the check node c in the t^{th} iteration is $\sigma_c^{(t)} = \hat{\Psi}\left(\mathbf{i}^{(t)}\right) = \text{sgn}\left(\prod \mathbf{i}^{(t)}\right)$.

The iterative procedure is halted when syndrome at all parity checks matches with the observed syndrome or a predefined maximum number of iterations, L, is reached. The decoding is called successful if a true error pattern is found. Otherwise, the decoding is said to have failed.

III. Decoding that breaks 4-Cycle Trapping Sets

The H matrix of dual containing codes gives rise to quantum LDPC codes. For a dual containing LDPC code, the row weights ρ of the parity check matrix H are necessarily even, and every pair of rows of H must have an even overlap of 1's. This ensures that the symplectic inner product criterion is satisfied, but results in numerous *cycles of length* 4 in the Tanner graph. We can guarantee zero length-4 cycles iff the rows of H have disjoint supports. However, in such codes, each bit takes part only in one parity check equation leading to poor error correction. Thus, short cycles become an unavoidable hurdle for the decoding of dual containing quantum LDPC codes. Iterative decoding algorithms, for instance, the Gallager-B algorithm (hard decision) or the sum-product algorithm (soft decision) suffer due to presence of such cycles and show flooring effect in their decoding performance curves. We use stochastic resonance ideas for dealing with short cycles and thereby, improving the decoding performance. The algorithm description and the short cycle trapping set analysis for the syndrome based stochastic Gallager-B decoder (hard decision) are given in this section.

A. Syndrome Based Stochastic Gallager-B Decoder

The observed syndrome $\mathbf{s} = (s_1, s_2, \ldots, s_m)$ is the input to the decoder. The syndrome based Gallager-B decoder works by sending binary messages over the edges of the graph. In other words, $\mathcal{M} = \mathcal{S} = \{0, 1\}$, and $\mathcal{Y} = \mathbf{0}$. The check reliability is the syndrome itself, i.e., $\mathbf{R}(s_c) = s_c$.

The messages are calculated based on the node update functions, following the extrinsic message passing rule that a message sent over an edge is obtained based on all received messages except the one arriving over that edge as described in Section II-C. The check node update function Ψ corresponds to the (ρ)-input XOR logic gate (uses the check reliability i.e., observed syndrome value for the respective check node in addition to the extrinsic inputs). A $(\gamma - 1)$-input majority logic (MAJ) gate is used for the variable node update function implementation. The decided value of error vector $\hat{e}_v^{(t)}$ is found at each iteration t considering the MAJ of initial value 0 and all γ incoming messages.

We conclude that the decoding attempt at the end of t iteration is a success if the calculated syndrome $\hat{\mathbf{s}}^{(t)}$ is equal to the observed syndrome $\mathbf{s}$. Then, the error pattern $\hat{\mathbf{e}}^{(t)}$ is decoded as the most likely error pattern.

To explore different decoding trajectories, noise is added to the output of the update functions. This is done by XOR-ing the noiseless output with the binary error e_{MAJ} or $e_{\oplus}$.

$$\begin{aligned} \nu_{v\to c}^{(t)} &= \Phi(\mathbf{0}, \mathbf{m}^{(t)}) + e_{MAJ}^{(t)}, \\ \mu_{c\to v}^{(t)} &= \Psi(\mathbf{n}^{(t-1)}, s_c) + e_{\oplus}^{(t)}, \end{aligned} \tag{2}$$

where e_{MAJ} and $e_{\oplus}$ are independent Bernoulli random variables with parameters α_{MAJ} and $\alpha_{\oplus}$. A gate with a smaller Bernoulli parameter introduces less perturbations in the output. Note that different realizations of Bernoulli random variables $e_{MAJ}^{(t)}$ and $e_{\oplus}^{(t)}$ correspond to different edges. We omit the indices for a simplified notation.

In addition to the logic gates needed for calculation of messages, the decoder implementation also requires perfect logic gates for the final error estimations and the syndrome calculation.

To allow the decoder to benefit from errors, a large number of iterations are needed under some conditions of gate noise/error. However, too many logic gate errors can overwhelm the decoder. The stochastic decoder is also equipped with the following key feature which prevents the accumulation of errors in the messages when the number of iterations is large. If a valid error pattern is not found after L_R iterations, where $L_R \ll L$, the decoding algorithm is re-initialized with

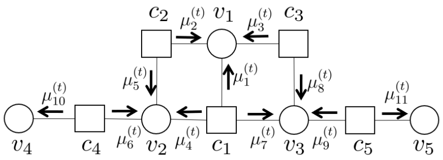

Fig. 2. The decoder state at iteration t can be expressed using $\mu^{(t)} = (\mu_1^{(t)}\mu_2^{(t)}\mu_3^{(t)}, \mu_4^{(t)} \ldots, \mu_{11}^{(t)})$.

the same observed syndrome. Instead of running the whole L iterations, the decoder runs $r = L/L_R$ very short rounds with a maximum of L_R iterations each. A decoder with such rewinding schedule is referred to as the rewind-decoder. Optimization of the parameter r is discussed in [12], and is code dependent.

Example 1 (Breaking Four-Cycle): We now illustrate an example of how the stochastic Gallager-B algorithm escapes from a 4-cycle trapping set. First, we identify a fixed set on a 4-cycle subgraph for a column-weight three quantum LDPC code in the Tanner graph representation by determining the decoder state at different iterations.

As shown in Fig. 2, the subgraph contains two 4-cycles: (v_1, c_1, v_2, c_2) and (v_1, c_1, v_3, c_3) with c_4 and c_5 as other neighbors nodes of v_2 and v_3 respectively. We use the isolation assumption [14] scenario wherein the messages generated within the trapping set do not affect the decisions in rest of the graph. This allows us to treat the trapping set in isolation to carry out its analysis.

Let $\mu^{(t)}$ denote the ordered set of all messages from the check nodes to the variable nodes in the subgraph at t^{th} iteration as shown in Fig. 2. In case of a perfect decoder, the state $\mu^{(t)}$ is completely determined by a composition of XOR and MAJ functions on $\mu^{(t-1)}$. For a perfect syndrome based Gallager-B decoder, let the observed syndrome be $\mathbf{s} = (01111)$ as shown in Fig. 3(a). ■/□ denotes that the observed syndrome value for the corresponding check node is equal to one/zero, while ●/○ denotes that the estimate of the corresponding variable node is one/zero. Initially, all the variable node messages are set to zero. Recall the decoding rule that in case of a tie while performing a majority logic operation, then the initial value of 0 is sent to the extrinsic check node. Syndrome is calculated based on the error estimates at each iteration and compared with the observed syndrome. We consider the decoding a success when the calculated syndrome is equal to the observed syndrome.

From Figs. 3(b) and 3(d), it is not difficult to find that a perfect syndrome based Gallager-B decoder fails to correct this error pattern as the decoding process within the short cycle gets trapped into the state $\mu^{(t)} = (011, 011, 011, 1, 1)$ with a syndrome estimate $\hat{\mathbf{s}} = (00000)$ and continues to remain in the same state in subsequent iterations. Here, we essentially found a trapping set for the perfect decoder.

The stochastic Gallager-B decoder can break this trapping set as shown in Figs. 3(e) and 3(f). If the XOR logic gate at check node c_1 produces erroneous values to the nodes v_2 and v_3, the decoder state changes to $\mu^{(t)} = (011, 111, 111, 1, 1)$. This new state, courtesy of the errors introduced, drives the decoder towards a different error vector with the correct syndrome estimate equal to the observed syndrome. Note that the stochastic decoder can converge towards the error vector in multiple trajectories, each successfully breaking the trapping set in different iteration steps based on the errors introduced in logic gates.

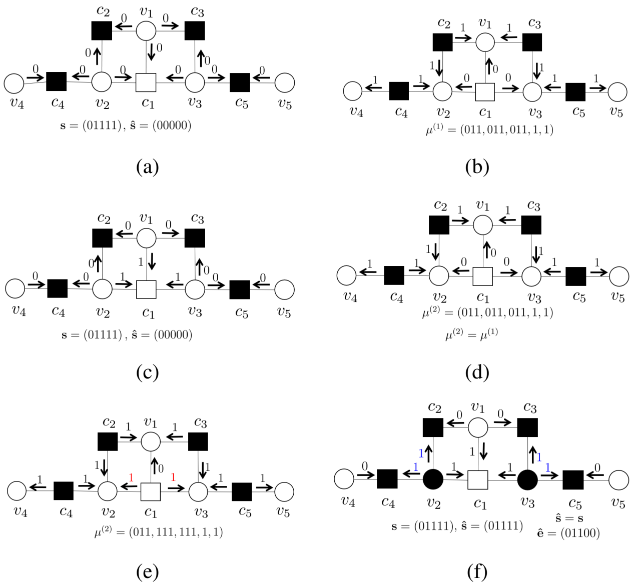
c_2
v_1
c_3
v_4
c_4
v_2
c_1
v_3
c_5
v_5
$\mathbf{s} = (01111), \hat{\mathbf{s}} = (00000)$
(a)
$\mu^{(1)} = (011, 011, 011, 1, 1)$
(b)
$\mathbf{s} = (01111), \hat{\mathbf{s}} = (00000)$
(c)
$\mu^{(2)} = (011, 011, 011, 1, 1)$
$\mu^{(2)} = \mu^{(1)}$
(d)
$\mu^{(2)} = (011, 111, 111, 1, 1)$
(e)
$\mathbf{s} = (01111), \hat{\mathbf{s}} = (01111)$
$\hat{\mathbf{s}} = \mathbf{s}$
$\hat{\mathbf{e}} = (01100)$
(f)

Fig. 3. Graphical representation of breaking of a 4-cycle trapping set by stochastic Gallager-B algorithm: (a). Initialization of variable node messages and the observed syndrome $\mathbf{s} = [01111]$, (b). The decoder state is $\mu^{(1)} = (011, 011, 011, 1, 1)$, (c). Since $\mathbf{s} \neq \hat{\mathbf{s}}$, decoding continues to iteration 2, (d). The decoder state is $\mu^{(2)} = (011, 011, 011, 1, 1)$ and the perfect decoder will be trapped in the same state for further iterations, (e). The check node update from c_1 is noisy and erroneous messages are shown in red, Observe that the state of decoder changes, (f). The errors introduced propagate to change the variable node messages (shown in blue). The syndrome estimate matches with the observed syndrome. The stochastic decoder successfully breaks the trapping set using the positive effect of noise introduced.

Stochastic decoder uses noise constructively and search for the most probable error vector possible from the observed syndrome value as a starting point. The rewinding concept helps intuitively to return back to the starting point and trace another trajectory and converge to an error vector if possible. Next, we present decoding performance curves for stochastic Gallager-B decoder.

IV. Numerical Results

In this section, we perform Monte Carlo simulation to compare the efficacy of our decoding approaches against standard decoding procedure. Dual containing quantum codes, constructed similar to Mackay's bicycle codes [2] are used for these simulations over a depolarizing channel, as described in Section II-B, for different values of crossover probabilities. The construction of an $[[n, k]]$ quantum bicycle code involves constructing a dual containing code $\mathcal{C}_1$ with parity check matrix H having row weight ρ and using the CSS construction. The parity check matrix H is obtained by first constructing a matrix $Q = [M \ M^{\mathrm{T}}]$ where M is cyclic matrix obtained from a $n/2$ binary vector of weight $\rho/2$ and then discarding $k/2$ rows.

A. *Performance of stochastic Gallager-B decoder*

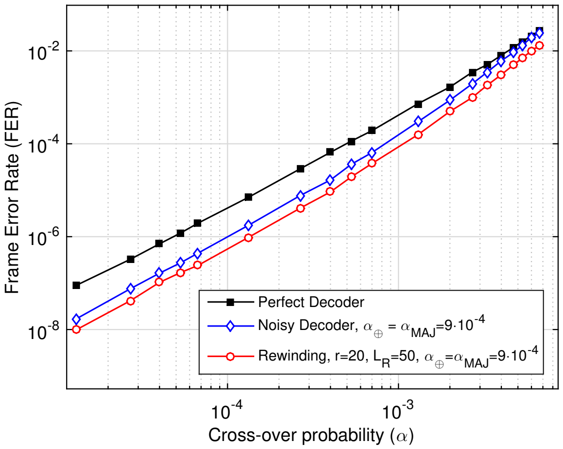

Fig. 4. FER vs. α curves comparison between Gallager-B algorithm, stochastic Gallager-B algorithm with and without rewinding for 1000 iterations on the Tanner graph of H matrix with size $(n-k) \times n$ = (100×200) and $\gamma = 3$.

In Fig. 4, we plot the decoding performance curves comparing a perfect syndrome based Gallager-B algorithm with stochastic Gallager-B algorithm with and without rewinding for 1000 iterations. The rewinding parameters chosen here are $r = 20$ and $L_R = 50$. The noise parameter used is the error introduced in the update functions, namely α_{MAJ} and $\alpha_{\oplus}$. The noise parameter chosen for curves shown in Fig. 4 is 0.0009.

There is an improvement of up to an order of magnitude (in FER value) when stochastic rewinding decoder is used in comparison to the perfect Gallager-B decoder. This shows the improved performance of the stochastic decoder and also the superior ability of the rewinding method to achieve significantly low FER values. The choice of noise parameter is code-dependent and its optimization is similar to that in [12]. Rewinding parameters were chosen in such a manner that the decoder is able to search for the most probable error vector on multiple short trajectories. The intuition behind is that hard decision decoders can get trapped into fixed points if not decoded successfully in a few iterations. Using more $r = L/L_R$ rounds give the noisy decoder alternative paths. Allowing more L_R iterations for each round helps the decoder to explore those paths and converge to an error vector.

V. Conclusions and Future Work

To summarize, we proposed a stochastic resonance based iterative decoder, built of a mixture of noisy and noiseless logic gates, which for a broad range of gate failure rates, works better than a decoder made completely of noiseless gates. A general theory validating this concept for diverse applications such as, fault tolerant quantum systems is also of interest. Many natural and artificial analog signal processing systems use noise constructively [15]. Also, this idea of using stochastic resonance based decoders to protect and correct fragile quantum states can be used towards fault tolerant circuits with significantly less circuit overhead. It is the randomness that is present in the decoding circuit gates that make these classes of decoders superior to their noiseless counterparts. A general theory for incorporating stochastic resonance within deterministic algorithms for tackling short cycles and error floors would be a natural extension of this work.

Acknowledgment

This work is supported by the IUSSTF under the Indo-US Joint R&D Network Joint Centre on Data Storage Research/16-2014 Award, and by the NSF under grants CCF-1314147 and ECCS-1500170. This work is partially supported by Shakti Foundation (SSEF) under grant G15 SSEF-154.

References

[1] A. R. Calderbank and P. W. Shor, “Good quantum error-correcting codes exist,” *Phys. Rev. A*, vol. 54, pp. 1098–1105, Aug. 1996.

[2] D. MacKay, G. Mitchison, and P. McFadden, “Sparse-graph codes for quantum error correction,” *IEEE Transactions on Information Theory*, vol. 50, no. 10, pp. 2315–2330, Oct. 2004.

[3] Z. Babar, P. Botsinis, D. Alanis, S. X. Ng, and L. Hanzo, “The road from classical to quantum codes: A hashing bound approaching design procedure,” *IEEE Access*, vol. 3, pp. 146–176, 2015.

[4] Y. Fujiwara and V. Tonchev, “A characterization of entanglement-assisted quantum low-density parity-check codes,” *IEEE Trans. on Info. Theory*, vol. 59, no. 6, pp. 3347–3353, June 2013.

[5] Y. Fujiwara, A. Gruner, and P. Vandendriessche, “High-rate quantum low-density parity-check codes assisted by reliable qubits,” *IEEE Trans. on Info. Theory*, vol. 61, no. 4, pp. 1860–1878, April 2015.

[6] D. Gottesman, “Class of quantum error-correcting codes saturating the quantum hamming bound,” *Phys. Rev. A*, vol. 54, no. 3, pp. 1862–1868, Sept. 1996.

[7] ——, “Stabilizer codes and quantum error correction,” Ph.D. dissertation, California Institute of Technology, 1997.

[8] Z. Babar, P. Botsinis, D. Alanis, S. X. Ng, and L. Hanzo, “Fifteen years of quantum LDPC coding and improved decoding strategies,” *IEEE Access*, vol. 3, pp. 2492–2519, 2015.

[9] J. S. Yedidia, W. T. Freeman, and Y. Weiss, “Constructing free energy approximations and generalized belief propagation algorithms,” *IEEE Trans. Inform. Theory*, vol. 51, pp. 2282–2312, July 2005.

[10] N. Raveendran and S. G. Srinivasa, “A modified sum-product algorithm over graphs with isolated short cycles,” *Proc. IEEE. Intl. Symp. Info. Theory*, vol. 54, pp. 2619–2623, June 2014.

[11] ——, “An analysis of the modified sum-product algorithm over graphs with isolated short cycles,” *IEEE Trans. on Info. Theory (Submitted)*, 2015.

[12] P. Ivanis and B. Vasić, “Error errore eicitur: A stochastic resonance paradigm for reliable storage of information on unreliable media,” *IEEE Trans. on Commun.*, pp. 3596–3608, Sept. 2016.

[13] A. Crespi et al., “Integrated photonic quantum gates for polarization qubits,” *Nature Comm.*, Nov. 2011.

[14] S. K. Planjery, D. Declercq, L. Danjean, and B. Vasić, “Finite alphabet iterative decoders for LDPC codes surpassing floating-point iterative decoders,” *Electron. Lett.*, vol. 47, no. 16, pp. 919–921, Aug. 2011.

[15] M. Kawaguchi, H. Mino, and D. Durand, “Stochastic resonance can enhance information transmission in neural networks,” *IEEE Trans. Biomedic. Eng.*, vol. 58, no. 7, pp. 1950–1958, July 2011.

Appendix
Quantum Codes

A qubit $|\psi\rangle$, the quantum analog of a classical bit, is a quantum system represented by a vector in a 2-dimensional complex vector space $\mathbb{C}^2$. The error operators acting on $|\psi\rangle$

are $2{\times}2$ matrices whose basis elements are from the Pauli group, namely,

$$\mathcal{P} \;=\; \{\pm \mathrm{iI}, \pm \mathrm{iX}, \pm \mathrm{iY}, \pm \mathrm{iZ}, \pm \mathrm{I}, \pm \mathrm{X}, \pm \mathrm{Y}, \pm \mathrm{Z}\}\,, \qquad (3)$$

where I denotes no error and X, Z and Y denote a bit flip, a phase flip and their combination respectively. The quantum state of n-qubits is represented by a vector in $\mathbb{C}^{2^n}$. Thus, the error operators on n qubits are $2^n{\times}2^n$ matrices with the basis obtained from the n-fold tensor product of elements of $\mathcal{P}$ represented by $\mathcal{P}^{\otimes n}$.

In quantum error correction, an $[[n,k]]$ quantum code maps 2^k messages to n qubits. The stabilizer framework [6] proposed by Gottesman gives a general method of constructing quantum codes similar to well known linear classical codes. We discuss the fundamentals of stabilizer codes in the next section.

A. Stabilizer Codes

Definition 1: An operator B is said to stabilize a quantum state $|\psi\rangle$ and B is said to be the stabilizer of $|\psi\rangle$ if

$$B\,|\psi\rangle = |\psi\rangle\,. \qquad (4)$$

Thus, an operator stabilizes $|\psi\rangle$ if the eigenvalue of the operator is 1 with $|\psi\rangle$ as the eigenstate.

Consider a group $S \subseteq \mathcal{P}^{\otimes n}$. Based on the properties of the Pauli group that each element has eigenvalues of $+1$ and -1 and any two elements commute or anti-commute, the erroneous quantum state $|\phi\rangle$ becomes an eigenstate of the elements in S with eigenvalue $+1$ or -1. The basic idea of decoding in stabilizer formalism is to correct errors based on the syndrome computed by concatenating the eigenvalues of $|\phi\rangle$ for the generators of S using the mapping $1 \to 0,\ -1 \to 1$. This gives us the definition of a codeword from a stabilizer perspective [2] as it is the simultaneous eigenstate of all generators of S with eigenvalue 1 corresponding to the all-zero syndrome. Thus, we have the following definition of a code given a stabilizer group S.

Definition 2: Given a set of stabilizers S, a codeword is defined as quantum state $|\psi\rangle$ that is a +1 eigenstate of all the stabilizers,

$$S_i\,|\psi\rangle = |\psi\rangle \quad \forall i \in [\ |S|\ ] \qquad (5)$$

The all-zero syndrome corresponds to no error as each codeword is stabilized by elements of S. The undetected errors are given by the error operators which take one codeword to another codeword. The syndrome will be all-zero as each stabilizer in S stabilizes the erroneous state $W\,|\psi\rangle$, where W is the undetectable error operator. Hence, $\forall\ S_i \in S$, $S_i W\,|\psi\rangle = W\,|\psi\rangle = W S_j\,|\psi\rangle$ for some $S_j \in S$. This means that either W commutes with each S_i or commutes with the group S as a whole. Thus, the undetectable error belongs to the centralizer or normalizer of S in $\mathcal{P}^{\otimes n}$.

Definition 3: The centralizer $C_G(S)$ of a subset S of the group G is the subset of all the elements in the group which commute with each element of S, i.e.,

$$C_G(S) \;=\; \{g{\in}G \mid gs = sg\ \forall s{\in}S\} \qquad (6)$$

Definition 4: The normalizer $N_G(S)$ of a subset S of the group G is the subset of all the elements in the group which commute with S as a whole, i.e.,

$$N_G(S) \;=\; \{g{\in}G \mid \forall s_1{\in}S\ ,\ \exists s_2{\in}S\ :\ s_1 g = g s_2\} \qquad (7)$$

As the operators of the Pauli group only commute or anti-commute, the normalizer of S is same as the centralizer of S in $\mathcal{P}^{\otimes n}$.

With these group theoretic definitions for the stabilizer code, we now discuss a code construction approach followed by error correction procedure with an example.

B. Stabilizer code construction

1) We find a commutative subgroup S of the error group $\mathcal{P}^{\otimes n}$. S forms the stabilizer group for the code. The error set $\mathcal{E}$ that this code can correct comprises of those error operators that anti-commute with at least one element of S and $\forall E_i, E_j \in \mathcal{E}, E_i^{\dagger} E_j \notin N_{\mathcal{P}^{\otimes n}}(S)$, where $N_{\mathcal{P}^{\otimes n}}(S)$ is the normalizer of S in $\mathcal{P}^{\otimes n}$.
2) Then, we find a subspace $\mathcal{C}$ of the 2^n dimensional Hilbert space whose elements are stabilized by S. $\mathcal{C}$ gives the desired stabilizer code.

C. Error correction

Due to the decoherence from the quantum channel, a quantum codeword undergoes unitary evolution to an erroneous state. Let an error operator $E_i \in \mathcal{P}^{\otimes n} \backslash N_{\mathcal{P}^{\otimes n}}(S)$ take the codeword $|\psi\rangle$ to a corrupted state $|\phi\rangle = E_i|\psi\rangle$.

1) We compute $\lambda_i \in \{+1,-1\},\ i = (1,\ldots,n-k)$, eigenvalues of $(n-k)$ stabilizer generators of the $[[n,k]]$ code, corresponding to the eigenstate $|\phi\rangle$. Based on the mapping mentioned before, we obtain the observed syndrome vector $\mathbf{s}$.
2) Based on the syndrome $\mathbf{s}$, we deduce and correct the error using syndrome based decoding techniques as demonstrated in Example 2.

Example 2 (Single bit flip correcting code):

$$\mathcal{C} \;=\; \{|000\rangle\,, |111\rangle\} \qquad (8)$$

$$S \;=\; \{\mathrm{ZZI}, \mathrm{ZIZ}, \mathrm{IZZ}, \mathrm{III}\} \qquad (9)$$

$$E \;=\; \{\mathrm{XII}, \mathrm{IXI}, \mathrm{IIX}\} \qquad (10)$$

The generators of S are $\{\mathrm{ZZI}, \mathrm{IZZ}\}$. The check matrix is

$$H \;=\; \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}. \qquad (11)$$

TABLE I
Syndrome table for a bit flip correcting code

Syndrome	Error
(0 0)	I I I
(1 0)	X I I
(1 1)	I X I
(0 1)	I I X

As an instance of error correction, suppose the error acting on the system is $E_1 \;=\; \mathrm{XII}$. We compute the syndrome to be (1 0). Based on the syndrome, from Table I, we can infer the error to be a bit flip on the first qubit. Hence, using the operator $M \;=\; \mathrm{XII}$, we can correct the error. This example shows how syndrome can be used for quantum error correction.