Real Randomized Benchmarking of the [4,2,2] Stabilizer Code on a Quantum Computer

A.J. Rasmusson
Indiana University

(Dated: April 28, 2020)

Stabilizer codes are a promising avenue for fault-tolerant quantum computing. Though today's quantum computers have tens of qubits or less, there are codes which fit on these small devices and can improve fidelities. One such code is the [4,2,2] code. I present the details for implementing the [4,2,2] code as outlined in Refs. [1, 2]. To quantify the improvement of the [4,2,2] code over not using it, I run a real randomized benchmarking test on two physical (unencoded) qubits and two logic (encoded) qubits. I find the single-qubit gate infidelity of the IBM-Burlington quantum computer for two physical qubits is 5.89%. For two [4,2,2] logic qubits the infidelity decreases to 0.36%.

INTRODUCTION

Quantum error detection and correction is critical to quantum computing especially when considering current quantum computers with high error rates and tens of qubits or less. Stabilizer codes are a promising framework to address errors. By spreading state information across many physical qubits, errors become distinguishable and correctable without loss of any information.

Though current devices are below single- and two-qubit error thresholds to truly compute fault-tolerantly, coding schemes are still useful and improve computation fidelity. Stabilizer codes map, or "encode", specific states among a group of physical qubits to one qubit state called a logic qubit state. This is useful because a group of physical qubits sharing quantum information is more robust against errors than if each physical qubit was on its on. This comes at the cost of reducing the computational space of the quantum computer as the logic qubit vector space will always be smaller than the physical qubit vector space.

Though quantum computers today only have tens of noisy qubits, stabilizer codes can offer improved computation for current algorithms and simulations. The [4,2,2] code is particularly useful because it can be used now and it has a low 2:1 physical to logical qubit ratio allowing for a vector space wherein to compute (as compared to a [5,1] code for example).

THE [4,2,2] CODE

The [4,2,2] code is a fault-tolerant code useful for error detection in small quantum computers. It can detect any single-qubit error [1, 3]. Defined in its name, the [4,2,2] takes four physical qubits to encode two logical qubits. Other codes such as the 5-qubit [4, 5], 7-qubit [6, 7], and 9-qubit [8–10] codes all require, as Gottesman points out [1], more than ten qubits to operate fault-tolerantly. Having a smaller code is a useful tool when today's quantum computers have tens of qubits or less.

The [4,2,2] code is a fault-tolerant code, but only detects errors; it does not correct them [1]. Errors are removed by post-selection: States measured outside of the code space are excluded from analysis.

The [4,2,2] code space V_s consists of logic states

$$|00\rangle_L = |0000\rangle + |1111\rangle \tag{1}$$
$$|01\rangle_L = |1100\rangle + |0011\rangle \tag{2}$$
$$|10\rangle_L = |0101\rangle + |1010\rangle \tag{3}$$
$$|11\rangle_L = |0110\rangle + |1001\rangle \tag{4}$$

with stabilizer generators

$$S = \{X \otimes X \otimes X \otimes X, Z \otimes Z \otimes Z \otimes Z\}. \tag{5}$$

S satisfies all conditions to be a stabilizer. By a quick inspection, one can see that all four logic states in V_s are invariant under both elements of S. The two generators also commute as can be shown quickly using the anti-commutation relation $\{X, Z\} = 0$. Last of all, $-\mathbb{1}$ is not a member of S.

As tabulated in Ref. [2] and shown in Table I, there are eight logic gates in the code gate set of the [4,2,2] code. The single qubit gates have the conventional definitions: $CNOT_{ij}$ has control on i and target on j; CZ_{ij} is a control-z with control on i and target on j; $R = diag(1, i)$; and $SWAP_{ij}$ swaps i with j. The first six gates are physical gates executed on the quantum computer while the last two gates can be implemented virtually by relabeling the qubits in software [2].

From the state $|00\rangle_L$, the logical gates can be used to access the other three logic states. The $|00\rangle_L$ state is prepared with the circuit shown in Fig. 1 [2].

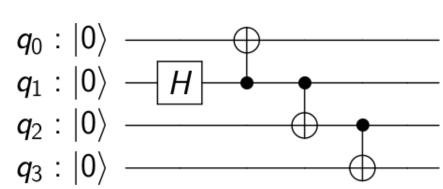

FIG. 1. Quantum circuit for preparing $|00\rangle$.

Gottesman [1] presents a fault-tolerant state preparation by use of an ancillary qubit. A parity check on qubits

Physical Gates	Logical Gates
$X \otimes X \otimes X \otimes X$	$\mathbb{1} \otimes \mathbb{1}$
$Z \otimes Z \otimes Z \otimes Z$	$\mathbb{1} \otimes \mathbb{1}$
$X \otimes \mathbb{1} \otimes X \otimes \mathbb{1}$	$X \otimes \mathbb{1}$
$X \otimes X \otimes \mathbb{1} \otimes \mathbb{1}$	$\mathbb{1} \otimes$ X
$Z \otimes Z \otimes \mathbb{1} \otimes \mathbb{1}$	$Z \otimes \mathbb{1}$
$Z \otimes \mathbb{1} \otimes Z \otimes \mathbb{1}$	$\mathbb{1} \otimes Z$
$H \otimes H \otimes H \otimes H$	$SWAP_{12}\,(H \otimes H)$
$R \otimes R \otimes R \otimes R$	$(Z \otimes Z)\,CZ_{12}$
$SWAP_{12}$	$CNOT_{12}$
$SWAP_{13}$	$CNOT_{21}$

TABLE I. Code gate set with physical implementation of logical gates. The last two gates are virtually implemented by relabeling qubits

q_0 and q_3, through the ancilla, can fault-tolerantly determine if any single-qubits errors occurred during state preparation [1]. If so, the circuit stops and retries. However, as Harper and Flammia point out in Ref. [2], randomized benchmarking is robust to spontaneously preparation and measurement (SPAM) errors, so like Harper and Flammia [2], I did not fault-tolerantly prepare the $|00\rangle_L$ state.

RANDOMIZED BENCHMARKING WITH THE REALIZABLE GROUP

Randomized benchmarking (RB) [11, 12] is becoming an industry standard for determining single- and two-qubit gate errors. For example, IBM's daily quantum computer calibrations are done using this method.

The basic idea is to apply a random set of $m+1$ gates (single or two qubit) such that the final state is identical to the initial state. In reality, the randomly chosen gates will have errors. As the errors accumulate, the probability of measuring the initial state—called the survival probability $\bar{q}$—reduces. The reduction in $\bar{q}$ is used to measure the fidelity of the gates used in the RB protocol.

Typically, the randomly selected gates are chosen from the Clifford group. However, there is no fault-tolerant phase gate for the [4,2,2] code [2]. To only use fault-tolerant gates, I (following Harper and Flammia [13]) used the Real RB [2, 13, 14] method instead. Real RB randomly draws gates from the Realizable group R(2) instead of the Clifford group. The R(2) group has 576 elements [2] which are generated from the logical gate set in Table I. For details about the R(2) group and why it is applicable see Refs [2, 13].

Over repeated E (an effect operator of a POVM) [2] measurements on ρ, the survival probability is

$$\bar{q}(m, E, \rho) = 0.25 + Bb^m + Cc^m \tag{6}$$

where, $\rho_\pm = \frac{1}{2}(\rho \pm \rho^T)$, constants B and C are defined by

$$B = \mathrm{Tr}[E\rho_+] - 0.25 \text{ and } C = \mathrm{Tr}[E\rho_-]. \tag{7}$$

For two qubits, the Real RB fidelity [2] with error $\mathcal{E}$ is

$$F(\mathcal{E}) = \frac{9b + 6c + 5}{20}. \tag{8}$$

I will use this fidelity to quantify the affect of using the [4,2,2] code vs not using the code.

IMPLEMENTATION

To do Real RB, I first computed the 576 elements of the R(2) group. To generate all 576 elements from the eight generators, I matrix multiplied every element in every possible combination. Each unique matrix was saved as an element of the group. After many iterations, all 576 elements are generated.

In order to see the advantage of two [4,2,2] encoded qubits over two unencoded qubits, I did two Real RBs. One with two physical (unencoded) qubits. The other with two logical (encoded) qubits from the [4,2,2] code.

In both cases, a single experimental run is 8192 repetitions of: initializing the state, applying $\frac{m+1}{2}$ randomly selected gates, applying the inverse of the $\frac{m+1}{2}$ gates, and then measurement. On top of the 8192 repetitions, each experimental run was executed over all four states possible states. Each state was repeated at least 2 times. To gather all the data took four days. More repetitions could be done in future for better statistics. A sample of collected data for the unencoded case can be seen in Fig. 2. A sample for the encoded case (with odd parity states removed) can be be seen in Fig. 3.

	10	00	11	01	cir_len	init_state
0	1145	4095	1500	1452	10	00
1	1059	1845	1054	4234	10	01
2	4331	1397	1099	1365	10	10
3	1430	1260	4191	1311	10	11

FIG. 2. Sample of data collected from IBM Burlington quantum computer for two unencoded qubits. Headers "10", "00", "11", "01" labels the number of times that state was measured in 8192 repetitions of the same experiment. The header "cir_len" labels the number of random gates and inverse applied. The header "init_state" labels the column of the prepared state before the random gates were applied.

For the unencoded case, the measured $\bar{q}$ is computed as the number of measured states identical to the initial state divided by the total number of executions. For example, in row 0 of Fig. 2, $\bar{q}$ = 4095 / 8192 = 0.499.

The encoded case takes advantage of the [4,2,2] code's ability to detect errors. The measured $\bar{q}$ is the number of

	0011	1001	0000	1010	0101	0110	1100	1111	cir_len	init_state
0	98	33	3821	59	44	51	294	2090	0	00
1	3078	312	282	59	126	116	1864	135	0	01
2	100	281	115	2088	3332	105	125	71	0	10
3	138	1915	150	173	201	3121	359	51	0	11

FIG. 3. Sample of data collected from IBM Burlington quantum computer for two encoded qubits. Headers with combinations of 0's and 1's label the number of times that state was measured in 8192 repetitions of the same experiment. Odd parity states have been excluded as part of the [4,2,2] code's post-selection. The header "cir_len" is the number of random gates and inverse applied. The header "init_state" is the prepared logical state before the random gates were applied.

measured states identical to the initial state divided by the total number of measured states in the code space V_s. If a state with odd parity is measured, it is removed from the statistics as part of the [4,2,2] code's post-selection error correction. For example, in row 0 of Fig. 3, $\bar{q}$ = (3821 + 2090) / 6490 = 0.911.

I then fit the measured $\bar{q}$ for the unencoded and encoded cases to Eq. (6). Extracting b and c from the fits, I computed the infidelity $1 - F(\mathcal{E})$ for each case.

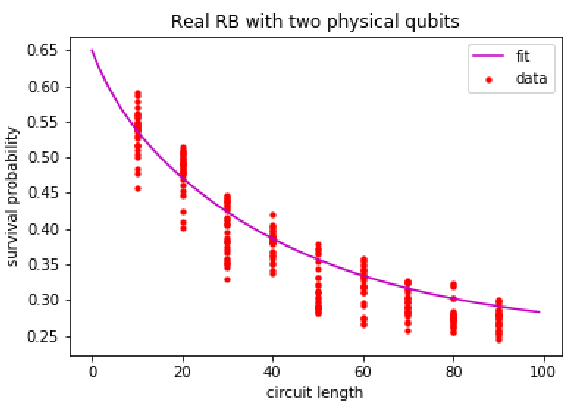

FIG. 4. Real RB with two physics qubits (i.e. unencoded). $\bar{q}$ vs circuit length

RESULTS

In Fig. 4, circuit length vs survival probability was fit for the unencdoed case. From that fit, the infidelity is measured to be 5.89%. Notice the large variance among data at each circuit length. Harper and Flammia [2] experience this as well. They employ techniques such as running RB with an adding a phase (called a Pauli twirl) which simplifies the form of $\bar{q}$ to only depend on b^m or c^m for certain states. This allows for more accurate fitting with fewer data points. I did not use these techniques as their measured infidelity 5.8(2)% [2] is close to what I measured 5.89% and to collect the additional data would have taken another 4 days.

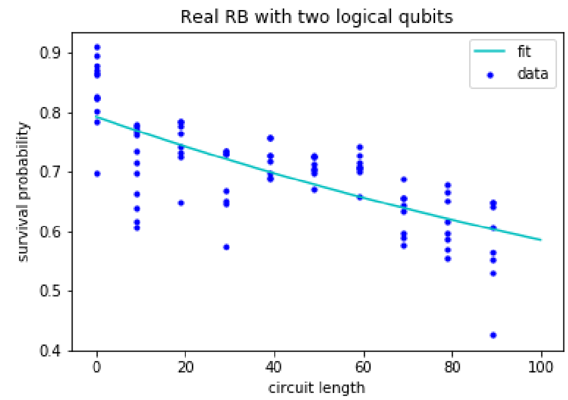

FIG. 5. Real RB with two logical qubits (i.e. encoded). $\bar{q}$ vs circuit length

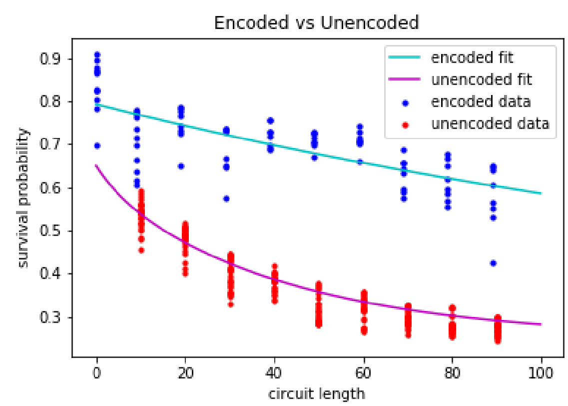

FIG. 6. Comparison between encoded and unencoded $\bar{q}$

In Fig. 5, circuit length vs survival probability was fit for the encdoed case. From that fit, the infidelity is measured to be 0.36%. Again, notice the large variance among data at each circuit length for the same reasons mentioned in the above paragraph. In comparison to Harper and Flammia [2], they measured an encoded infidelity of 0.60(3)%.

Beyond the quantitative improvement, Fig. 6 shows just how obviously different the unencoded and encoded cases are. The exponential nature of $\bar{q}$ is much less obvious in the encoded case compared to the unencoded case.

The [4,2,2] code is a useful tool for small quantum computers. It has a simple error detection scheme and an order of magnitude infidelity reduction. Though not a fully fault-tolerant code, the improved fidelities and 2:1 physical to logic qubit ratio make it an attractive option for today's noisy intermediate-scale quantum computers.

REFERENCES

[1] Daniel Gottesman, "Quantum fault tolerance in small experiments," (2016), arXiv:1610.03507 [quant-ph].

[2] Robin Harper and Steven T. Flammia, "Fault-tolerant logical gates in the ibm quantum experience," Phys. Rev. Lett. **122**, 080504 (2019).

[3] Lev Vaidman, Lior Goldenberg, and Stephen Wiesner, "Error prevention scheme with four particles," Physical Review A **54**, R1745 (1996).

[4] Raymond Laflamme, Cesar Miquel, Juan Pablo Paz, and Wojciech Hubert Zurek, "Perfect quantum error correcting code," Physical Review Letters **77**, 198 (1996).

[5] Charles H Bennett, David P DiVincenzo, John A Smolin, and William K Wootters, "Mixed-state entanglement and quantum error correction," Physical Review A **54**, 3824 (1996).

[6] A Robert Calderbank and Peter W Shor, "Good quantum error-correcting codes exist," Physical Review A **54**, 1098 (1996).

[7] Andrew Steane, "Multiple-particle interference and quantum error correction," Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences **452**, 2551–2577 (1996).

[8] Peter W Shor, "Scheme for reducing decoherence in quantum computer memory," Physical review A **52**, R2493 (1995).

[9] Dave Bacon, "Operator quantum error-correcting subsystems for self-correcting quantum memories," Physical Review A **73**, 012340 (2006).

[10] David Poulin, "Stabilizer formalism for operator quantum error correction," Physical review letters **95**, 230504 (2005).

[11] Joseph Emerson, Robert Alicki, and Karol Życzkowski, "Scalable noise estimation with random unitary operators," Journal of Optics B: Quantum and Semiclassical Optics **7**, S347 (2005).

[12] Emanuel Knill, Dietrich Leibfried, Rolf Reichle, Joe Britton, R Brad Blakestad, John D Jost, Chris Langer, Roee Ozeri, Signe Seidelin, and David J Wineland, "Randomized benchmarking of quantum gates," Physical Review A **77**, 012307 (2008).

[13] AK Hashagen, ST Flammia, D Gross, and JJ Wallman, "Real randomized benchmarking," Quantum **2**, 85 (2018).

[14] Winton G. Brown and Bryan Eastin, "Randomized benchmarking with restricted gate sets," Physical Review A **97** (2018), 10.1103/physreva.97.062323.