

Floquetifying Quantum Error Correction Code

Yusheng Zhao and Clemens Schumann

Abstract — The study of automated steps in Floquetifying Quantum Error Correction Codes.

Yusheng Zhao
HKUST(GZ), 1 Duxue Rd, Nanshan District, Guangzhou, Guangdong, China, email:
yushengzhao2020@outlook.com

Clemens Schumann
, , email:

1 Vision

“Quantum error-correcting codes are a key ingredient for fault-tolerant quantum computation.” [1]

Design of quantum error correction codes concerns the improvement of “code distance”, ease of implementing “logical gates and tradeoffs between the number of logical qubits and distance” [2].

“Recent breakthroughs in the development of quantum LDPC codes have led to significant improvements in the asymptotic parameters of quantum codes [1, 2, 3, 4, 5, 6]. However, demonstrating a significant advantage over the surface code for realistic noise models and reasonable system sizes has proven challenging. For example, Ref. [7] demonstrated a significant $14\times$ saving in resources relative to the surface code using hypergraph product codes for circuit-level noise, however this improvement was shown for very low physical error rates of around 0.01% and for very large system sizes (around 13 million qubits). This is in large part due to the high-weight stabiliser generators common in many families of quantum LDPC codes, resulting in deep and complex syndrome extraction circuits, which can introduce hook errors and lower their noise thresholds” [3]. qldpc code will have better scaling but the scaling is due to high weight measurement, in reality, this measurement translated to requiring many high fidelity measurementnets?

Surface code is good because it does not require high connectivity of qubits. “The surface code is a popular candidate because it presents an excellent compromise of good logical performance and achievable demands on hardware” [4] “Previous attempts to construct circuits with more relaxed hardware requirements have faced significant challenges. Lower connectivity either demands unreasonable overhead in the cycle depth or the use of alternative codes sacrificing logical performance [Bac05; Cha+20; Sun+22].”

Floquetifying code is a software way to “improve” performance of hardware.

What is the benefit of “high-weight stabilizer measurement”? [5] One such novel class of code is the Floquet code. A Floquet code is defined as a code “a code defined on a measurement sequence, that consists of rounds of measurements, with each round given by a set of commuting pauli measurements. After the i th round of measurements, the code is defined by the stabilizer group that has been updated by the measurements according to the stabilizer update rules in Lemma 3 defined above.” [6] (Definition of Quantum Error Correction Code should be given by Code space, measurement etc)

Origin of Floquet Code: “his model supports 6-body operators that both commute with, and can be built from, the non-commuting 2-local terms of its Hamiltonian. These operators are primed to serve as low-weight parity constraints for a quantum code requiring only 2-body measurements.”

How 2-local measurement helps “while two-local sparsely-connected codes can be desirable in hardware (e.g. to reduce crosstalk [8]), two-locality often greatly diminishes the quality of error-correction” [7] “this level of sparsity and locality has led to codes with steep losses in performance [2, 15, 16]” [8]

2 News

How are these improved in Floquet code? Are they directly related to measurement weight? Indeed, threshold was improved [7]. Another threshold numerical simulation: “Benchmarking the Planar Honeycomb Code” is for honeycomb code on planar lattice which is more physical. Record the numbers. How come it is not as robust “While it is not as robust as the surface code in the more standard SD6 and SI1000 error models, it boasts a teraquop footprint within an order of magnitude of the surface code at physically plausible error rates.” [8] “Enhanced Fault-tolerance in Photonic Quantum Computing: Floquet Code Outperforms Surface Code in Tailored Architecture” beating surface code in actual platform. Photonic system with high photon loss threshold, “we observe an improved photon loss threshold of 6.4%, compared to the original 2.8% reported in” . [5]

Stabilizer codes require large weight measurements which extra care in fault-tolerant protocols [9].

In such a code, measurement weights are lower to 2 and are native to hardware like Majorana devices [9]. Furthermore, the measurement cycle are also shorter [9]. “The sparse connectivity and two-local measurements are desirable features for certain hardware, while the weight-six parity checks enable robust performance in the circuit model.” [7].

Furthermore, “As a dynamical code consists of several rounds of ISG, it can potentially circumvent the no-go theorem by Eastin and Knill which shows that transversal operators on any non-trivial quantum code belong to a finite group, and thus cannot be universal.” [9].

“other methods that allow one to overcome the Eastin-Knill no-go theorem [9,10], such as lattice surgery and code deformation [11,12], can be unified into the framework of gauge fixing” [10]

3 Steps

4 Conclusion

5 Questions

How did “Floquet Codes without Parent Subsystem Codes” create Floquet codes?

What is the difference between “Relaxing Hardware Requirements for Surface Code Circuits using Time-dynamics” and Floquet code? What is the automated tools

they have used? They were analyzing the broken down syndrome measurement circuit like a Floquet code.

Numerical simulation of performance and explain how measurement on MZM is carried out “Performance of Planar Floquet Codes with Majorana-Based Qubits” xxx

How did people use ZX Calculus to analyze Floquet codes? “Unifying flavors of fault tolerance with the ZX calculus” xxx

Generalization of Floquet code “Quantum computation from dynamic automorphism codes” introduces a way to do universal quantum computation without code switching. “Several viewpoints have been proposed, including paths of Hamiltonians [5], other constructions from measurement-based and fusion-based quantum computation [20, 28], and a tensor-network path integral approach [20, 23]. Partial results towards a classification of Floquet codes based on unitary loops [16] and measurement quantum cellular automata [17] have also been developed.”

How did they generate Floquet code from color code “Constructions and performance of hyperbolic and semi-hyperbolic Floquet codes”? “Although these (semi-)hyperbolic Floquet codes cannot be implemented using geometrically local connections in a planar Euclidean architecture, we show how they can instead be implemented using a bilayer or modular architecture. A bilayer architecture uses two layers of qubits, where connections within each layer do not cross, but may be long-range. A modular architecture consists of many small modules, where each module only requires local 2D planar Euclidean connectivity and connections between modules may be long-range. The long-range connections between modules could be mediated via photonic links in a trapped-ion architecture [31], for example.”

What is rewinding and how did it help in generating new Floquet codes? “Engineering 3D Floquet Codes by Rewinding” xxx

Is this equivalent to compiling Floquet code from QLDPC code? “Spacetime codes of Clifford circuits” No...

Why do you need to specifically design gate for Floquet code? “Cross-Cap Defects and Fault-Tolerant Logical Gates in the Surface Code and the Honeycomb Floquet Code” How does this guide us on compiling Floquet code? “to realize the topologically ordered state by a periodic sequence of noncommuting two-qubit Pauli measurements, dubbed a honeycomb Floquet code [5].” [11]

Is the flow mentioned extensible to ZX Calculus? Can we use it as a guide for generating Floquet code in ZX Calculus? “Measurement Quantum Cellular Automata and Anomalies in Floquet Codes”

What is the significance of boundary here? “Planar Floquet Codes”

How did cultivation paper cite Floquet code? “Magic state cultivation: growing T states as cheap as CNOT gates “

How did they do construction of Floquet code from qudits? “Simple Construction of Qudit Floquet Codes on a Family of Lattices “

What is the path integral way of generating Floquet code? “The x+y Floquet code: A simple example for topological quantum computation in the path integral approach”

6 Side comments

Distance for Floquet code is not well defined since each measurement cycle, the stabilizer group changes. Consider using “mask” defined by Gottesman [6]

6.1 High level summary

“This added locality can simplify quantum error-correction circuits by either sparsifying the physical layout or compactifying the syndrome circuit.” [7]

“While both of these qualities may be desirable at the hardware-level [79], the locality often comes at significant cost to the quality of error-correction. Intuitively, by releasing degrees of freedom to increase locality, we also collect less information about the errors occurring. This manifests as high-weight parity constraints that can require a large assembly of local measurements to determine. In the most extreme case, this can eliminate a threshold altogether” [7]

Embedding was possible for open boundary condition which is more physical “embedding onto a planar lattice. We ended [9] noting that this was an important open problem, because embedding on a plane (instead of a torus) is a necessary ingredient for using the honeycomb code on two-dimensional architectures. Fortunately, this problem was quickly solved by Haah and Hastings” [8]

“First, choose a quantum error correction code to encode each qubit. Second, design a syndrome extraction circuit for this code. This quantum circuit is run regularly and returns a bit string, the so-called syndrome, which is then used to identify errors.” [12]

“Previous attempts to construct circuits with more relaxed hardware requirements have faced significant challenges. Lower connectivity either demands unreasonable overhead in the cycle depth or the use of alternative codes sacrificing logical performance [Bac05; Cha+20; Sun+22]. Further challenges are presented by various non-ideal realities of hardware, such as the presence of leakage states. Modifications to improve the code’s resilience to such effects also typically introduce additional overhead and harm logical performance [Fow13a]” [4] this is why you NEED Floquet code to implement QEC on a hardware with lower connectivity. Supposedly, it is easier to build a hardware with lower connectivity.

7 Hello

Statement: like proof
use slides to explain

Bibliography

1. Dua, A., Tantivasadakarn, N., Sullivan, J., Ellison, T. D.: Engineering 3D Floquet Codes by Rewinding. *PRX Quantum*. 5, 20305–20306 (2024). <https://doi.org/10.1103/PRXQuantum.5.020305>
2. Baspin, N., Williamson, D.: Wire Codes, <http://arxiv.org/abs/2410.10194>
3. Higgott, O., Breuckmann, N. P.: Constructions and Performance of Hyperbolic and Semi-Hyperbolic Floquet Codes, <http://arxiv.org/abs/2308.03750>
4. McEwen, M., Bacon, D., Gidney, C.: Relaxing Hardware Requirements for Surface Code Circuits Using Time-dynamics. *Quantum*. 7, 1172–1173 (2023). <https://doi.org/10.22331/q-2023-11-07-1172>
5. Hilaire, P., Dessertaine, T., Bourdoncle, B., Denys, A., Gliniasty, G. de, Valentí-Rojas, G., Mansfield, S.: Enhanced Fault-tolerance in Photonic Quantum Computing: Floquet Code Outperforms Surface Code in Tailored Architecture, <http://arxiv.org/abs/2410.07065>
6. Fu, X., Gottesman, D.: Error Correction in Dynamical Codes, <http://arxiv.org/abs/2403.04163>
7. Gidney, C., Newman, M., Fowler, A., Broughton, M.: A Fault-Tolerant Honeycomb Memory. *Quantum*. 5, 605–606 (2021). <https://doi.org/10.22331/q-2021-12-20-605>
8. Gidney, C., Newman, M., McEwen, M.: Benchmarking the Planar Honeycomb Code. *Quantum*. 6, 813–814 (2022). <https://doi.org/10.22331/q-2022-09-21-813>
9. Hastings, M. B., Haah, J.: Dynamically Generated Logical Qubits. *Quantum*. 5, 564–565 (2021). <https://doi.org/10.22331/q-2021-10-19-564>
10. Davydova, M.: Floquet Codes without Parent Subsystem Codes. *PRX Quantum*. 4, (2023). <https://doi.org/10.1103/PRXQuantum.4.020341>
11. Kobayashi, R., Zhu, G.: Cross-Cap Defects and Fault-Tolerant Logical Gates in the Surface Code and the Honeycomb Floquet Code. *PRX Quantum*. 5, 20360–20361 (2024). <https://doi.org/10.1103/PRXQuantum.5.020360>
12. Delfosse, N., Paetznick, A.: Spacetime Codes of Clifford Circuits, <http://arxiv.org/abs/2304.05943>