A Notes on Floquet Codes

hoge

ABSTRACT: This note provides a comprehensive technical overview of Floquet quantum error-correcting codes, a novel class of dynamical quantum memories. We begin by establishing the theoretical foundations of periodically driven quantum systems via Floquet's theorem, defining the Floquet operator and the effective Hamiltonian. We then present a precise mathematical definition of Floquet codes, contrasting them with conventional static stabilizer codes. The Hastings-Haah honeycomb code [1] is analyzed in detail as a canonical example, elucidating its lattice structure, periodic measurement protocol, and the dynamics of its instantaneous stabilizer groups. A central theme is the connection between Floquet codes and non-equilibrium phases of matter, particularly their realization of Floquet Enriched Topological Order and their interpretation as discrete time crystals. We survey the broader landscape of Floquet codes, including topological, fractonic, and high-rate hyperbolic variants, contextualizing them within the unifying framework of anyon condensation. The note culminates in an analysis of their performance, hardware suitability, and current experimental status, presenting key results from numerical simulations of error thresholds. Important references are provided to guide further study.

KEYWORDS: Floquet codes, quantum error correction, topological order, time crystals, measurement-based quantum computation

Contents

1 Introduction to dynamical quantum error correction

1.1 Motivation and context

The realization of a large-scale, fault-tolerant quantum computer represents a paramount challenge in modern physics and engineering. Quantum information is notoriously fragile, susceptible to decoherence and operational errors arising from unwanted interactions with the environment. Quantum Error Correction (QEC) provides a theoretical framework to combat these effects by redundantly encoding logical information into a larger number of physical qubits. The dominant paradigm for QEC has been the use of static stabilizer codes, most notably the toric code and its planar variant, the surface code. These codes are lauded for their high error thresholds and their implementation on a two-dimensional, nearest-neighbor architecture. However, they typically require the measurement of high-weight stabilizers (e.g., four-qubit operators for the surface code), which are experimentally challenging and often necessitate complex ancilla-based circuits, contributing significantly to resource overhead. This practical difficulty has motivated the search for alternative QEC schemes.

1.2 Introducing the floquet code paradigm

Floquet codes have recently emerged as a compelling alternative that fundamentally re-imagines the process of error correction. Instead of defining a static code space protected by a fixed set of commuting stabilizers, Floquet codes utilize a periodic sequence of non-commuting, low-weight measurements to dynamically protect quantum information. This approach effectively shifts the complexity of the QEC protocol from the spatial domain (high-weight operators) to the temporal domain (a carefully orchestrated measurement sequence). Consequently, a Floquet code is not a static quantum memory but a *dynamical* one, where the logical subspace and its associated logical operators evolve periodically in time.

The fundamental innovation of Floquet codes lies in a redefinition of what constitutes a "code". Traditional stabilizer codes define a static subspace of the Hilbert space, $C = \{|\psi\rangle : S_i|\psi\rangle = |\psi\rangle, \forall i\}$, which serves as a passive safe for quantum information. Floquet codes demonstrate that logical information can be preserved without being confined to a single, fixed subspace. The protection emerges from the specific *dynamics* of the measurement sequence, which coherently shuttles the logical information through a series of distinct, instantaneous code spaces. Perhaps the most striking feature of this paradigm is that the complete set of check operators, when viewed as the generators of a static subsystem code, can define a code with *zero* logical qubits. The logical qubits are not pre-existing properties of the Hamiltonian or stabilizer set but are instead *dynamically generated* by the measurement protocol itself. The error correction protocol is thus inseparable from the definition of the code, a profound departure from the static paradigm where syndrome extraction is a procedure performed *on* a pre-defined code.

1.3 Key advantages and overview

This dynamical approach yields several practical advantages. Floquet codes can often be implemented on lattices with low qubit connectivity (e.g., trivalent lattices where each qubit connects to only three neighbors) and employ simple, two-qubit Pauli measurements. These low-weight checks are native operations in certain promising quantum computing

architectures, potentially leading to higher fidelities and reduced hardware complexity. Furthermore, numerical studies have shown that Floquet codes can achieve error thresholds competitive with, and in some contexts superior to, the surface code.

This note provides a detailed technical survey of Floquet codes. Section 2 reviews the foundational physics of periodically driven systems. Section 3 presents the precise mathematical definition of a Floquet code, using the canonical honeycomb code as a detailed example. Section 4 explores the deep connection between Floquet codes and non-equilibrium phases of matter, such as discrete time crystals. Section 5 provides a taxonomy of different Floquet code families, and Section 6 analyzes their performance and experimental status.

2 The Subsystem Code Formalism: A Foundation for Dynamical QEC

The Floquet code is built upon a more general framework known as subsystem quantum error correction [2, 3]. While standard stabilizer codes define a static code subspace as the simultaneous +1 eigenspace of a set of commuting Pauli operators, subsystem codes introduce a crucial layer of abstraction by sacrificing a portion of the information-carrying capacity to simplify the error correction procedure. This trade-off is often physically motivated: many promising stabilizer codes, such as the surface code, require the measurement of high-weight operators (e.g., four-qubit plaquette and star operators), which can be experimentally demanding and a dominant source of error [2]. Subsystem codes replaces these high-weight stabilizer measurements with a sequence of simpler, low-weight "check" measurements. This effectively shifts the complexity from the spatial domain (operator weight) to the procedural or temporal domain (measurement sequence), a conceptual step that directly paves the way for the fully dynamical nature of Floquet codes. This section provides a rigorous introduction to the subsystem formalism, establishing the algebraic structure, the physical interpretation via Hilbert space decomposition, and the key examples that motivate its application to dynamical QEC.

2.1 Generalizing Stabilizer Codes: The Gauge Group and Gauge Freedom

The algebraic structure of a subsystem code is defined by a hierarchy of groups within the n-qubit Pauli group $\mathcal{P}_n$. The foundational object is:

Definition 2.1: Gauge group and checks

The **gauge group** $\mathcal{G} \subset \mathcal{P}_n$ is a possibly non-Abelian subgroup of Pauli operators. The generators of $\mathcal{G}$ is called **checks** or **gauge generators**. The **weight** of a check $G \in \mathcal{G}$ is the number of qubits it acts on.

Unlike in the standard stabilizer formalism, these checks are not required to commute with one another.

From the gauge group, we define the **stabilizer group** $\mathcal{S}$ as its center:

$$\mathcal{S} := Z(\mathcal{G}) = \{S \in \mathcal{G} \mid [S, G] = 0, \forall G \in \mathcal{G}\}. \tag{2.1}$$

By definition, $\mathcal{S}$ is an Abelian subgroup of $\mathcal{G}$. The code space $\mathcal{C}$ is the simultaneous +1 eigenspace of all operators in $\mathcal{S}$, identical to the definition in the stabilizer formalism. If a stabilizer $S \in \mathcal{S}$ can be expressed as a product of gauge generators, $S = G_1 G_2 \cdots G_m$, its eigenvalue can be determined from the product of the individual measurement outcomes of the G_i.

Figure 1. Schematic picture of the relationship between stabilizer and subsystem code formalisms

The logical operators are those that preserve the code space $\mathcal{C}$ but are not themselves stabilizers. In the subsystem formalism, the set of logical operators is defined by the centralizer (not to be confused with the center) of the gauge group $C(\mathcal{G})$

$$\text{logical} \in C(\mathcal{G})\backslash\mathcal{S}, \qquad C(\mathcal{G}) := \{P \in \mathcal{P}_n \mid [P, G] = 0, \forall G \in \mathcal{G}\}. \tag{2.2}$$

This is a more stringent condition than in the stabilizer formalism, where logical operators are the centralizer of the stabilizer group $C(\mathcal{S})$

$$\text{logical} \in C(\mathcal{S})\backslash\mathcal{S}, \qquad C(\mathcal{S}) := \{P \in \mathcal{P}_n \mid [P, S] = 0, \forall S \in \mathcal{S}\}. \tag{2.3}$$

Since $\mathcal{S} = Z(\mathcal{G}) \subseteq \mathcal{G}$, it follows that $C(\mathcal{G}) \subseteq C(\mathcal{S})$. Put differently, the set of available logical operators $C(\mathcal{G})/\mathcal{S}$ in a subsystem code is a subset of those available $C(\mathcal{S})/\mathcal{S}$ in a stabilizer code. The schematic logic flow is depicted in Fig. 1.

The construction of logical operators in subsystem code leads to the question:

> "*Why is it advantageous to reduce the number of logical operators?*"

The answer lies in a fundamental trade-off between information capacity and operational simplicity [2, 3]. The operators in $C(\mathcal{S})\backslash C(\mathcal{G})$ are precisely those that commute with all stabilizers in $\mathcal{S} = Z(\mathcal{C})$ but anticommute with at least one gauge generator in $\mathcal{G}$. In the subsystem formalism, these operators are treated as gauge operators rather than logical operators for storing information. The degrees of freedom they act upon are sacrificed, becoming "gauge qubits" that can absorb the back-action from non-commuting check measurements without corrupting the protected logical information [4, 5]. This sacrifice is the price paid for the freedom to choose a gauge group $\mathcal{G}$ whose generators are of much lower weight and more geometrically local than the generators of the stabilizer group $\mathcal{S}$ they produce [6].

2.2 Hilbert Space Decomposition and Error Correction

The algebraic structure of a subsystem code has a direct physical interpretation in the decomposition of the underlying Hilbert space [5]. The code space $\mathcal{C}$, defined as the +1

eigenspace of the stabilizer group $\mathcal{S}$, is not an irreducible space for the logical operators. Instead, it possesses a tensor product structure that separates the information-bearing degrees of freedom from the gauge degrees of freedom [2, 6]

$$\mathcal{C} = \mathcal{H}_L \otimes \mathcal{H}_G. \tag{2.4}$$

Here, $\mathcal{H}_L$ is the 2^k-dimensional **logical subsystem**, where the k logical qubits are encoded. The bare logical operators from $C(\mathcal{G})$ act non-trivially only on this factor. $\mathcal{H}_G$ is the 2^g-dimensional **gauge subsystem**, corresponding to the g gauge qubits. The gauge operators (those in $C(\mathcal{S}) \backslash C(\mathcal{G})$) act non-trivially on this factor. The total Hilbert space of n physical qubits can thus be decomposed as $\mathcal{H}_{\text{phys}} = (\mathcal{H}_L \otimes \mathcal{H}_G) \oplus \mathcal{C}^{\perp}$, where $\mathcal{C}^{\perp}$ is the orthogonal error space.

This decomposition fundamentally alters the philosophy of error correction. For a standard stabilizer code, a recovery operation must return an errored state $E|\psi\rangle$ to the original state $|\psi\rangle \in \mathcal{C}$ (up to a stabilizer). For a subsystem code, the goal is less stringent since an error E is correctable if its effect on the logical subsystem $\mathcal{H}_L$ can be reversed, regardless of its effect on the gauge subsystem $\mathcal{H}_G$ [2]. More formally, a set of errors $\{E_a\}$ is correctable if, for any pair E_a, E_b, the operator $PE_a^{\dagger}E_b P$ (where P is the projector onto $\mathcal{C}$) acts as $I_L \otimes U_G$ for some operator $U_G : \mathcal{H}_G \to \mathcal{H}_G$. Any physical error that is equivalent to a non-trivial operation on $\mathcal{H}_G$ but the identity on $\mathcal{H}_L$ is a detectable. This provides enormous flexibility in decoding, as the recovery procedure only needs to restore the state of $\mathcal{H}_L$.

The dimensions of these spaces are related to the number of independent generators of the associated groups.

> **Proposition 2.1:**
>
> Let $s = \dim(\mathcal{S})$ be the number of independent stabilizer generators and $g = \frac{1}{2}(\dim(C(\mathcal{S})) - \dim(\mathcal{S})) - k$ be the number of gauge qubits. The number of physical qubits n is partitioned according to
>
> $$n = k + g + s. \tag{2.5}$$

This relation makes the trade-off explicit: for a fixed number of physical qubits n and stabilizers s, increasing the number of gauge qubits g (to simplify measurements) necessarily decreases the number of logical qubits k. A subsystem code is often denoted by the parameters $[[n, k, g, d]]$, where d is the code distance, defined as the minimum weight of an operator in $C(\mathcal{S}) \setminus \mathcal{G}$ [3].

2.3 Canonical Examples

We now ground this abstract formalism in three concrete examples, each chosen to illustrate a key principle and to serve as a conceptual stepping stone towards Floquet codes.

2.3.1 The Bacon-Shor Code: Locality from Gauge Freedom

The Bacon-Shor code is a canonical example of a subsystem code that demonstrates how to achieve syndrome extraction with only two-qubit, nearest-neighbor measurements on a 2D lattice, at the cost of having high-weight stabilizers [3]. Consider an $L \times L$ square lattice with a physical qubit $q_{i,j}$ at each vertex (i, j), for a total of $n = L^2$ qubits.

The gauge group $\mathcal{G}$ is generated by $2L(L-1)$ weight-2 check operators that act between adjacent qubits:

$$\text{Row checks:} \quad A_{i,j} = X_{i,j}X_{i,j+1} \quad \text{for } i \in [1, L], j \in [1, L-1], \qquad (2.6)$$

$$\text{Column checks:} \quad B_{i,j} = Z_{i,j}Z_{i+1,j} \quad \text{for } i \in [1, L-1], j \in [1, L]. \qquad (2.7)$$

These checks do not all commute. For example, $[A_{i,j}, B_{i,j+1}] \neq 0$. The stabilizer group $\mathcal{S} = Z(\mathcal{G})$ is generated by operators formed from products of these checks. One can verify that the only products of gauge generators that commute with all other gauge generators $A_{i,j}$, $B_{i,j}$ are those that span an entire row or column:

$$S_i^X = \prod_{j=1}^{L-1} A_{i,j} = X_{i,1}X_{i,2} \cdot X_{i,2}X_{i,3} \cdots = X_{i,1}X_{i,L}, \quad (\text{for } i \in [2, L]), \qquad (2.8)$$

$$S_j^Z = \prod_{i=1}^{L-1} B_{i,j} = Z_{1,j}Z_{L,j}, \quad (\text{for } j \in [2, L]). \qquad (2.9)$$

These $2(L-1)$ stabilizers are high-weight operators. However, their eigenvalues are determined by measuring the low-weight checks. For example, the eigenvalue of S_i^X is the product of the measurement outcomes of the $L-1$ checks $\{A_{i,j}\}_{j=1}^{L-1}$ in that row. This code encodes $k = 1$ logical qubit. The logical operators can be chosen as a full column of X operators and a full row of Z operators:

$$\overline{X}_L = \prod_{i=1}^{L} X_{i,1}, \qquad \overline{Z}_L = \prod_{j=1}^{L} Z_{1,j}. \qquad (2.10)$$

The code has distance $d = L$. With $n = L^2$, $k = 1$, and $s = 2L - 2$ independent stabilizers, the number of gauge qubits is $g = n - k - s = L^2 - 1 - (2L - 2) = (L-1)^2$. The code parameters are thus $[[L^2, 1, (L-1)^2, L]]$.

2.3.2 The Subsystem Toric Code: Reducing Check Weight via Qubit Overhead

While the Bacon-Shor code reduces check weight to two, it does so at the cost of having only a sub-extensive number of stabilizers, which precludes a fault-tolerant threshold in the thermodynamic limit [6]. An alternative approach, pioneered by Bravyi et al. [4], is to construct a subsystem version of the toric code that retains its topological properties but uses checks of lower weight than the standard weight-4 stabilizers (startoperators A_v and plaquette operatirs B_p). This is achieved by increasing the number of physical qubits.

The code is defined on an $L \times L$ square lattice with periodic boundary conditions. Qubits are placed on all L^2 vertices and on the center of all $2L^2$ edges, for a total of $n = 3L^2$ qubits. The gauge group $\mathcal{G}$ is generated by weight-3 Pauli operators associated with "triangles" formed by a vertex and its two adjacent edge-centers within a plaquette (See Fig. 2). For each plaquette p, there are four such triangles. The gauge generators are

$$G_X(T) = \prod_{q \in T} X_q, \qquad G_Z(T) = \prod_{q \in T} Z_q, \qquad (2.11)$$

where T denotes the set of three qubits on a triangle. The stabilizer group $\mathcal{S} = Z(\mathcal{G})$ is generated by products of these weight-3 checks. The familiar toric code stabilizers emerge as weight-6 operators:

- **Star Stabilizers:** For each vertex v, there are four incident Z-type triangles, $\{T^Z_{v,i}\}^4_{i=1}$. Their product forms the star operator

$$S^Z_v = \prod_{i=1}^{4} G_Z(T^Z_{v,i}).\qquad(2.12)$$

- **Plaquette Stabilizers:** For each plaquette p, there are two disjoint X-type triangles, $T^X_{p,1}$ and $T^X_{p,2}$. Their product forms the plaquette operator

$$S^X_p = G_X(T^X_{p,1})G_X(T^X_{p,2}).\qquad(2.13)$$

The eigenvalues of the weight-6 stabilizers are inferred from measuring the weight-3 checks. This construction exemplifies a different point in the QEC design space: compared to the standard surface code ($n \approx 2L^2$, weight-4 checks), this subsystem version uses more physical qubits ($n = 3L^2$) to achieve a reduction in check operator weight (weight 3). This trade-off can be highly advantageous in physical architectures where implementing three-body interactions is significantly more reliable than four-body interactions.
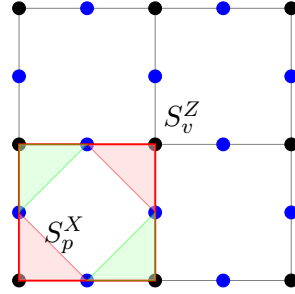
Figure 2. Lattice for the subsystem toric code. Qubits are placed on vertices (black) and edge centers (blue). The weight-6 plaquette stabilizer S^X_p (enclosed in red square) is the product of two weight-3 X-type triangle operators (red shaded). The weight-6 star stabilizer S^Z_v centered at vertex $(1.5, 1.5)$ is the product of four weight-3 Z-type triangle operators (two shown in green).

2.3.3 The subsystem (gauge) color code: A bridge to Floquet dynamics

Our final example illustrates how the properties of a complex static code can be realized through a *dynamic sequence* of simple measurements, providing a direct conceptual link to Floquet codes. The parent model is the 2D color code, a topological stabilizer code defined on a trivalent, 3-face-colorable lattice (e.g., a honeycomb lattice, see Fig. 3) [5, 6], whose Hamiltonian is given by

$$H_{CC} = -\sum_{p \in F(\Lambda)} S^p_X - \sum_{p \in F(\Lambda)} S^p_Z,\qquad(2.14)$$

where

$$S^p_X \quad := \qquad , \qquad S^p_Z \quad := \qquad \qquad (2.15)$$

A key feature of the color code is its rich $\mathbb{Z}_2 \times \mathbb{Z}_2$ topological order, but its stabilizers are high-weight operators (e.g., weight 6 on a honeycomb lattice).

Instead of being used as a static code, its error-correcting capabilities can be generated dynamically through a *periodic sequence* of weight-2 check measurements [5]. This approach is best understood through the lens of anyon condensation [2, 5]. The parent color code hosts several distinct types of anyonic excitations. A round of measurements can be designed to effectively "condense" one type of boson, which projects the system into a new effective theory. For the color code, it is possible to choose sequences of weight-2 measurements that condense different bosons in each step, causing the system to cycle through different phases that are equivalent to the standard toric code ($\mathbb{Z}_2$ topological order) [5].

A concrete protocol might involve a measurement cycle of considerable length. For instance, one could measure all $X_i X_j$ checks on edges of one color, then $Z_i Z_j$ checks on edges of a second color, and so on, cycling through different Pauli types and colors. At any given time step t, the instantaneous stabilizer group $S(t)$ is generated only by the weight-2 checks just measured. However, certain products of checks from previous rounds may commute with the current checks. These products survive as effective stabilizers, building up a "memory" of the measurement history. Over one full, long measurement cycle, the cumulative logical constraints imposed by this history of low-weight measurements become equivalent to having measured the high-weight stabilizers of the parent color code. The "fairly long measurement cycle" is the time required to cycle through a sufficient number of distinct projections (condensations) to fully specify the logical subspace of the parent theory.

This example fundamentally blurs the line between a static code and an error correction protocol. The set of all weight-2 checks measured during the cycle can be taken as the generators of a static subsystem code's gauge group. However, this static code may have zero logical qubits. The logical qubits only emerge from the specific *dynamics* of the measurement sequence. The code *is* the protocol. This is the foundational principle of Floquet codes, which we will formalize in the subsequent sections.

3 Theoretical foundations of periodically driven systems

Before we proceed to Floquet codes, let us clarify where the term "Floquet" originates in this section. It stems from the ordinary differential equation (ODE) theory describing systems governed by ODEs with time-periodic coefficients, established by a French mathematician, Gaston Floquet, in the 1880s [7]. In quantum mechanics, this applies to systems described by a time-periodic Hamiltonian, $H(t + T) = H(t)$, where T is the period of the drive.

3.1 The time-evolution operator and Floquet's theorem

The dynamics of a quantum system are governed by the time-dependent Schrödinger equation:

$$i\hbar \frac{\partial}{\partial t} |\psi(t)\rangle = H(t)|\psi(t)\rangle \tag{3.1}$$

The formal solution is given by the time-evolution operator

$$U(t_2, t_1) = \mathcal{T} \exp\left[-\frac{i}{\hbar} \int_{t_1}^{t_2} H(t')dt'\right], \tag{3.2}$$

where $\mathcal{T}$ is the time-ordering operator. This is the most general expression as it is valid even when the Hamiltonian does not commute with itself at different times $[H(t_1), H(t_2)] \neq 0$.

For a periodic Hamiltonian $H(t + T) = H(t)$, Floquet's theorem asserts the existence of a set of solutions, known as **Floquet states**, of the form

$$|\psi_\alpha(t)\rangle = e^{-i\varepsilon_\alpha t/\hbar}|\phi_\alpha(t)\rangle \tag{3.3}$$

Here, $|\phi_\alpha(t)\rangle = |\phi_\alpha(t + T)\rangle$ are the periodic **Floquet modes**, and the real-valued constants ε_α are the *quasi-energies*. This theorem implies that the wavefunction acquires a simple phase factor multiplied by a function that shares the periodicity of the drive.

3.2 The Floquet operator and effective Hamiltonian

The central object in Floquet theory is the time-evolution operator over one full period, known as the **Floquet operator** or **monodromy operator**

$$U_F \equiv U(t_0 + T, t_0)$$

The Floquet modes evaluated at time t_0 are the eigenstates of this unitary operator:

$$U_F|\phi_\alpha(t_0)\rangle = e^{-i\varepsilon_\alpha T/\hbar}|\phi_\alpha(t_0)\rangle \tag{3.4}$$

This eigenvalue equation reveals that the dynamics, when observed stroboscopically at integer multiples of the period T, are governed by a time-independent problem. This allows one to define an *effective Hamiltonian*, H_{eff}, which is a time-independent Hermitian operator that generates the same evolution over one period

$$H_{\text{eff}} := \frac{i\hbar}{T}\log(U_F) \implies U_F = e^{-iH_{\text{eff}}T/\hbar}. \tag{3.5}$$

The eigenvalues of H_{eff} are the quasi-energies ε_α. Due to the multi-valued nature of the complex logarithm, the quasi-energies are only defined up to integer multiples of $\hbar\omega$, where $\omega = 2\pi/T$. They are typically restricted to a "Floquet-Brillouin zone," $\varepsilon_\alpha \in [-\hbar\omega/2, \hbar\omega/2)$.

In the high-frequency limit where $\hbar\omega$ is much larger than other energy scales in the system, H_{eff} can be systematically approximated using the Magnus expansion:

$$H_{\text{eff}} = H^{(0)} + H^{(1)} + H^{(2)} + \cdots \tag{3.6}$$

$$H^{(0)} := \frac{1}{T}\int_0^T H(t')dt',$$

$$H^{(1)} := -\frac{1}{2i\hbar T}\int_0^T dt_1 \int_0^{t_1} dt_2 [H(t_1), H(t_2)],$$

$$\vdots$$

To zeroth order, the effective Hamiltonian is simply the time-average of the periodic Hamiltonian. Higher-order terms describe corrections arising from the non-commutativity of the Hamiltonian at different times.

4 The Mathematical and Physical Definition of a Floquet Code

While the theory of periodically driven Hamiltonians provides the conceptual underpinning, Floquet codes are typically defined not by a continuous drive $H(t)$ but by a discrete, periodic sequence of projective measurements. This can be viewed as an idealized limit of strong, periodic interactions with a measurement apparatus.

4.1 General framework: periodic measurements and ISGs

A Floquet code is specified by a quantum system, typically qubits arranged on a lattice L, and a periodic sequence of non-commuting Pauli check-operator measurements $\{C_j^{(t)}\}$. The sequence has a period of

T rounds (or time steps). The state of the system after round t is stabilized by a set of Pauli operators forming the **Instantaneous Stabilizer Group (ISG)**, denoted $\mathcal{S}(t)$. The dynamics of the code are entirely captured by the evolution of this group from one round to the next:

$$\mathcal{S}(t) \xrightarrow{\text{Measure } \{C_j\}} \mathcal{S}(t+1) \tag{4.1}$$

The logical operators at time t, denoted $\overline{L}(t)$, are Pauli operators that commute with every element of $\mathcal{S}(t)$ but are not themselves in $S(t)$, i.e.,

$$\{\overline{L}(t)\} := C(\mathcal{S}(t))\backslash\mathcal{S}(t) \tag{4.2}$$

For the code to function as a quantum memory, the measurement sequence must be designed such that the logical information is preserved through this evolution. That is, for each logical operator $\overline{L}(t)$, there must be a corresponding logical operator $\overline{L}(t+1)$ whose action on the logical subspace is predictably related to that of $\overline{L}(t)$.

4.2 Case study: the Hastings-Haah honeycomb code

The first and most studied example of a Floquet code was introduced by Hastings and Haah [1].

4.2.1 Lattice geometry and 3-coloring

The code is defined on a two-dimensional honeycomb lattice with qubits placed on the vertices. A key property of this lattice is that its faces (the hexagons or plaquettes) are 3-colorable. We can label each plaquette as Red (R), Green (G), or Blue (B) such that no two adjacent plaquettes share the same color. The edges of the lattice can then be colored according to the plaquettes they connect; for example, a "red" edge connects two red plaquettes. This structure is illustrated in Fig. 3.

4.2.2 The measurement protocol

The protocol consists of a period-3 sequence of two-qubit Pauli measurements. The initial ISG is set to $\mathcal{S}(0) = \langle I \rangle$. Each round corresponds to one of the edge colors:
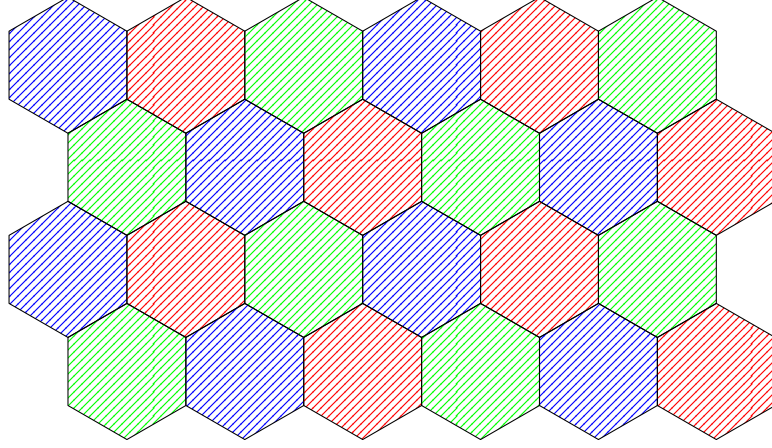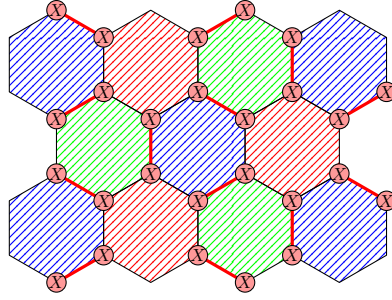
Figure 3. A portion of the honeycomb lattice with qubits (black dots) on the vertices. The hexagonal plaquettes are 3-colorable (here represented by red, green, and blue shading). The edges are implicitly colored by the type of plaquettes they connect.
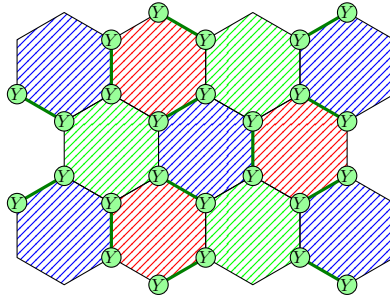
1. **Round 1 (Red):** For every red edge connecting qubits i and j, measure the Pauli operator $X_i X_j$:

$$(4.3)$$

The system is projected into a state stabilized by all the red check operators. $\mathcal{S}(1) = \langle \{X_i X_j\}_{\text{red edges}} \rangle$.

2. **Round 2 (Green):** For every green edge connecting qubits i and j, measure the Pauli operator $Y_i Y_j$:
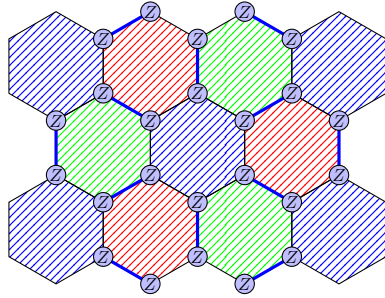

$$(4.4)$$

The new stabilizers are the green checks just measured. However, certain products of the pre-existing red and new-coming green checks commute with all green checks,

namely 6-body plaquette stabilizers around the blue plaquettes.

$$S_X^p := \qquad p \qquad (4.5)$$

Thus, $\mathcal{S}(2)$ is generated by $\langle \{Y_k Y_l\}_{\text{green edges}} \rangle$ and emergent 6-body operators for blue plaquettes. This is the crucial step where high-weight stabilizers are dynamically generated from low-weight measurements.

3. **Round 3 (Blue):** For every blue edge connecting qubits i and j, measure the Pauli operator $Z_i Z_j$:

$$ (4.6) $$

Similarly, $\mathcal{S}(3)$ is generated by the blue checks and plaquette stabilizers for both blue and red plaquettes

$$S_X^p := \qquad p \qquad (4.7)$$

This three-round sequence is then repeated indefinitely. We will clarify how the ISG changes as the rounds piles up. A schematic of one cycle is shown in Fig. 4.
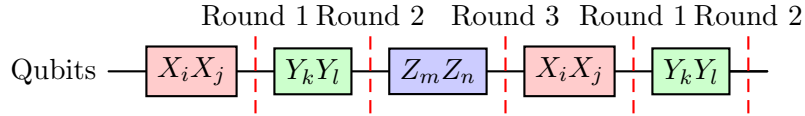

Figure 4. A schematic representation of one cycle of the honeycomb Floquet code protocol. In each round, all two-qubit checks of a given color/type are measured simultaneously.

After the first full cycle, the ISG at the end of a round of, say, red measurements, is generated by all the red check operators and the 6-body plaquette operators for all three colors. For a red plaquette p_R, the stabilizer is $B_{p_R} = \prod_{i \in p_R} X_i$, and similarly for green (Y) and blue (Z) plaquettes. These plaquette operators are precisely the stabilizers of Kitaev's toric code defined on a hexagonal superlattice.

4.2.3 Dynamically generated logical operators

On a lattice with periodic boundary conditions (a torus), the honeycomb code dynamically generates and protects two logical qubits. The logical operators correspond to string-like products of single-qubit Pauli operators that wrap around the non-trivial cycles of the torus. A critical feature is that the logical operators themselves must evolve in time.

A logical operator $\overline{L}(t)$ must commute with the current ISG $\mathcal{S}(t)$

$$\forall S^{(t)} \in \mathcal{S}(t), \; [\overline{L}(t), S^{(t)}] = 0 \tag{4.8}$$

To remain a valid logical operator after the next round of measurements, it must also commute with the checks $\{C_j^{(t+1)}\}$. Since the checks do not commute between rounds, the logical operators must be updated at each step, typically by multiplying them by some of the check operators just measured. This dynamic nature of the logical operators is intrinsically linked to the code's connection to non-equilibrium phases of matter.

5 Floquet Codes as Non-Equilibrium Phases of Matter

The periodic, non-commuting dynamics of Floquet codes give rise to phenomena that are best understood through non-equilibrium statistical mechanics, connecting QEC to concepts like time crystals and topological order.

5.1 The breaking of discrete time-translation symmetry

The measurement protocol of a Floquet code defines a discrete time-translation symmetry; the system is driven by a sequence of operations that repeats with period T. A phase of matter exhibits spontaneous symmetry breaking if its ground state (or, in a dynamical context, its steady-state) has less symmetry than the system Hamiltonian. A **_Discrete Time Crystal (DTC)_** is a non-equilibrium phase of matter that spontaneously breaks this discrete time-translation symmetry. This manifests as a robust, *subharmonic* response, meaning some local observable $\langle O(t) \rangle$ meets $\langle O(t+nT) \rangle = \langle O(t) \rangle$ $(n > 1)$ but this equality fails for any smaller positive integer $m < n$.

5.2 Floquet enriched topological order (FET)

In the honeycomb Floquet code, this time-crystalline behavior appears in the logical operators. While the measurement sequence has a period of $T = 3$ rounds, the logical operators evolve with a longer period. After one full cycle of three rounds, the logical operators corresponding to electric (e) and magnetic (m) anyon loops of the underlying toric code are exchanged:

$$\overline{L}^e(t) \xrightarrow{t \to t+T} \overline{L}^m(t+T) \quad \text{and} \tag{5.1}$$

$$\overline{L}^m(t) \xrightarrow{t \to t+T} \overline{L}^e(t+T) \tag{5.2}$$

This "*e-m* transmutation" is equivalent to applying a logical Hadamard gate to the encoded qubits. Since this transformation occurs every cycle $2T = 6$, the logical operators only return to their original form (up to a sign) after two full cycles. This robust, subharmonic evolution of non-local order parameters is the defining signature of a topological DTC.

The resulting phase is termed **Floquet Enriched Topological Order (FET)**, as the static topological order of the instantaneous code space is "enriched" by this non-trivial dynamical symmetry. The stability of the Floquet code against errors is synonymous with the stability of this non-equilibrium FET phase of matter.

This connection reveals that Floquet codes are more than just passive quantum memories. The measurement dynamics can be engineered to actively process the stored information. The e-m transmutation is, in effect, a logical Hadamard gate implemented "for free" as part of the error-correction cycle. This suggests a paradigm where complex logical operations could be compiled into specific, periodic measurement sequences. The code is not merely protecting data; it *is* the computation. This concept is generalized by Dynamic Automorphism (DA) codes, where more complex measurement sequences can implement a larger set of logical gates, potentially the entire Clifford group, purely through measurement. In this model, the distinction between error correction and computation becomes blurred, potentially leading to highly efficient fault-tolerant circuits.

5.3   Anyon automorphisms

The e-m transmutation is a specific instance of a broader concept known as an anyon automorphism, which is a permutation of the anyon labels of a topological order that preserves their fusion and braiding statistics. The periodic measurement sequence of a Floquet code can be interpreted as a physical procedure for implementing such an automorphism on the encoded topological state. This provides a powerful and general framework for designing Floquet codes that perform intrinsic logical gates, a key ingredient for fault-tolerant quantum computation.

6   A taxonomy of Floquet codes and their relation to static topologies

The principles underlying the honeycomb code have been generalized to construct a wide variety of Floquet codes with different properties, lattices, and underlying topological orders.

6.1   The principle of anyon condensation

A powerful framework for understanding and engineering Floquet codes is the theory of anyon condensation. In a static topological code, measuring a set of commuting string operators corresponding to a bosonic anyon effectively "condenses" that boson, leading to a new topological phase where that anyon becomes trivial. The non-commuting measurements in a Floquet code can be interpreted as a dynamic process that cycles the system through different condensed phases of a single, richer "parent" topological theory (see e.g. [8] for the color-code construction and [9] for CSS/CSS-like Floquet constructions). For example, the Floquet color code is understood as being derived from a parent color code theory ($\mathbb{Z}_2 \times \mathbb{Z}_2$ topological order). Its measurement sequence dynamically cycles the system through the nine distinct ways the color code can condense into a standard toric code ($\mathbb{Z}_2$ topological order).

6.2   A survey of Floquet code families

The diverse landscape of Floquet codes can be organized by their dimensionality, lattice structure, and the nature of the information they encode. Table 6.2 provides a summary of several key families.

code name	lattice	logical qubits (k)	distance (d)	parent order
2d honeycomb code [1]	honeycomb (6.6.6)	2 (on torus)	$O(L)$	$\mathbb{Z}_2$ toric code order
2d 4.8.8 Floquet TC [10]	square-octagon	2 (on torus)	$O(L)$	$\mathbb{Z}_2$ toric code order
2d Floquet color code [8, 11]	trivalent 3-colorable	2 (on torus)	$O(L)$	$\mathbb{Z}_2 \times \mathbb{Z}_2$ color code
X-cube Floquet code [12]	truncated cubic	3D	$O(L)$	switches between X-cube (fracton) and toric code layers
2d hyperbolic Floquet [13, 14]	hyperbolic tiling	$O(n)$ (finite rate)	$O(\log n)$	high encoding rate
3d Floquet TC [15]	3d generalization	$O(L)$	$O(L)$	3d toric code order

Table 1. Comparison of Major Floquet Code Families

6.2.1   2D topological Floquet codes

This class includes the honeycomb code and its variants, such as the 4.8.8 Floquet toric code defined on a square-octagon lattice. These codes are all based on 2D lattices and dynamically generate a state with $\mathbb{Z}_2$ topological order, akin to the toric code. The Floquet color code is another important 2D example, which dynamically realizes the richer $\mathbb{Z}_2 \times \mathbb{Z}_2$ topological order of the color code by cycling through its various toric code-like phases.

6.2.2   3D and fracton Floquet codes

The Floquet paradigm extends to three dimensions, leading to codes with novel properties. The **X-Cube Floquet code** is a prominent example, defined on a 3D lattice derived from coupled layers of 2D codes. Its significance lies in its ability to dynamically generate a *fracton* topological order. Fracton phases are exotic states of matter characterized by excitations (anyons) with restricted mobility, such as being confined to move along lines or planes. The X-Cube Floquet code's ISG switches between that of the X-Cube fracton model and layers of entangled 2D toric codes within a single cycle. Furthermore, it encodes a number of logical qubits that grows with the linear system size, $k \propto L$, offering a potential advantage in qubit density over 2D codes.

6.2.3   High-rate Floquet codes

A limitation of codes defined on Euclidean lattices (like square or honeycomb) is that their encoding rate k/n (logical qubits per physical qubit) vanishes in the thermodynamic limit. **Hyperbolic Floquet codes** overcome this by defining the code on a lattice embedded in a negatively curved, hyperbolic space. The geometry of hyperbolic space allows for a constant encoding rate, meaning the number of logical qubits scales linearly with the number of physical qubits, $k \propto n$. This comes at the cost of requiring non-local connections to implement on a 2D physical device, but it opens a path toward highly efficient encoding of quantum information.

7   Performance, implementation, and experimental status

7.1   Fault-tolerance and error thresholds

The ultimate utility of a QEC code depends on its performance in the presence of noise. For a Floquet code, an error (e.g., a Pauli flip on a qubit) that occurs between measurement rounds will cause subsequent check measurement outcomes to deviate from their expected

values. The error correction task is to infer the most likely error configuration from this syndrome history. This is typically modeled as a decoding problem on a three-dimensional graph, with two spatial dimensions and one time dimension, where minimum-weight perfect matching algorithms can be applied.

The **error threshold** is the critical physical error rate below which the logical error rate can be made arbitrarily small by increasing the code size. Numerical simulations are crucial for estimating these thresholds. The performance of Floquet codes is highly dependent on the underlying noise model and hardware capabilities, as summarized in Table 2.

Table 2. Performance Comparison of Honeycomb Floquet Code vs. Surface Code

Code	Noise Model	Error Threshold (p_{th})	Resource (Teraquop)
Surface Code	Circuit-level depolarizing (CNOT-based)	$\approx 0.5\%-0.7\%$	High (e.g., $>$ 2 for 10^{-12} error 0.1%)
Honeycomb Code	Circuit-level depolarizing (CNOT-based)	$\approx 0.2\%-0.3\%$	Higher than Su in this model
Surface Code	Native 2-qubit measurements (compiled)	Lower ($\approx 0.2\%$)	Very High (due tion overhead)
Honeycomb Code	Native 2-qubit measurements	$\approx 1.5\%-2.0\%$	Low (e.g., $\approx$ 90 10^{-12} error ra 0.1%)
Honeycomb Code	Photonic (loss-dominant)	6.4% (photon loss)	Significantly Surface Code

The data reveals a critical trade-off. In standard circuit models based on CNOT gates, the surface code's higher-weight stabilizers provide more robust error information, leading to a higher threshold. However, in architectures where two-qubit Pauli measurements are the native, low-error operations, the Floquet code's direct implementation gives it a significant advantage. Compiling the surface code's four-body checks into two-body operations introduces significant overhead and error channels, lowering its effective threshold in this context.

7.2 Hardware suitability and experimental status

The performance analysis highlights that Floquet codes are exceptionally well-suited for quantum computing platforms where two-qubit Pauli measurements are native operations. This includes architectures based on Majorana zero modes, where the fundamental operations are measurements of fermion parity, which map directly to Pauli products. For such platforms, Floquet codes are predicted to offer an order-of-magnitude improvement in error thresholds and substantial reductions in both space and time overheads compared to surface code implementations. Certain photonic architectures also naturally support two-qubit measurements, making Floquet codes a prime candidate for achieving fault-tolerance in these systems.

While a full, fault-tolerant logical qubit based on a Floquet code has not yet been experimentally demonstrated, significant progress has been made on key components.

- The measurement of the 6-body plaquette stabilizers for both the honeycomb code and the Floquet color code has been demonstrated on IBM's superconducting quantum processors, verifying the principle of dynamically generating high-weight checks from a sequence of two-qubit measurements.

- A small-scale Floquet-Bacon-Shor code has been realized on a 3x3 lattice of superconducting qubits.

The primary challenge for near-term devices is achieving the requisite fidelity and speed of the two-qubit measurements to surpass the performance of established static codes in dominant architectures like superconducting circuits and trapped ions.

8 Conclusion and Future Outlook

Floquet codes represent a paradigm shift in quantum error correction, moving from static data protection in a fixed subspace to dynamic preservation through a periodic sequence of non-commuting measurements. This approach has deep and fruitful connections to non-equilibrium condensed matter physics, with Floquet codes providing a tangible realization of phenomena such as discrete time crystals and Floquet enriched topological order.

The key advantage of this dynamical paradigm is the ability to construct robust QEC protocols using only simple, two-qubit Pauli measurements. This makes Floquet codes a leading candidate for fault-tolerance on hardware platforms where such operations are native, such as those based on Majorana zero modes or photonics. For these architectures, Floquet codes promise significantly higher error thresholds and lower resource overheads compared to the conventional surface code.

Despite the theoretical promise and initial experimental progress, several key challenges and research directions remain.

- **Decoding:** Developing more efficient classical decoders for the complex 3D space-time error graph is crucial for practical implementation.

- **Code Design:** There is a vast, unexplored space of possible Floquet codes. Future work will focus on systematically designing codes with tailored properties, such as robustness to specific biased noise channels or the ability to perform non-Clifford logical gates transversally.

- **Computational Power:** The interpretation of Floquet dynamics as implementing anyon automorphisms suggests that these codes can be more than just memories. Exploring the full computational power of such "dynamical automorphism codes" is an exciting frontier.

- **Experimental Realization:** Bridging the gap between theoretical proposals and experimental demonstrations of a fault-tolerant Floquet logical qubit remains the ultimate goal, requiring continued improvements in the fidelity and speed of mid-circuit measurements on a variety of quantum hardware platforms.

In conclusion, Floquet codes offer a rich and promising alternative path toward scalable, fault-tolerant quantum computation, blending the fields of quantum information, condensed matter, and non-equilibrium dynamics.

References

[1] M. B. Hastings and J. Haah, *Dynamically Generated Logical Qubits*, Quantum **5** (2021) 564, [arXiv:2107.02194].

[2] D. Poulin, *Stabilizer formalism for operator quantum error correction*, Phys. Rev. Lett. **95** (2005) 230504, [quant-ph/0508131].

[3] D. Bacon, *Operator quantum error-correcting subsystems for self-correcting quantum memories*, Phys. Rev. A **73** (2006) 012340, [quant-ph/0506023].

[4] S. Bravyi, G. Duclos-Cianci, and D. Poulin, *Subsystem surface codes with three-qubit check operators*, New J. Phys. **12** (2010) 075029, [arXiv:1002.1337].

[5] H. Bombin, *Topological subsystem codes*, Phys. Rev. A **81** (2010) 032301, [arXiv:0909.0520].

[6] M. Suchara, S. Bravyi, and B. M. Terhal, *Constructions and noise threshold of topological subsystem codes*, J. Phys. A: Math. Theor. **44** (2011) 155301, [arXiv:1012.0425].

[7] G. Floquet, *Sur les équations différentielles linéaires à coefficients périodiques*, in Annales scientifiques de l'École normale supérieure, vol. 12, pp. 47–88, 1883.

[8] M. S. Kesselring, J. C. M. de la Fuente, F. Thomsen, J. Eisert, S. D. Bartlett, and B. J. Brown, *Anyon condensation and the color code*, PRX Quantum **5** (2024) 010342, [arXiv:2212.00042].

[9] M. Davydova, N. Tantivasadakarn, and S. Balasubramanian, *Floquet codes without parent subsystem codes*, PRX Quantum **4** (2023) 020341, [arXiv:2210.02468].

[10] B. Yan, P. Chen, and S. X. Cui, *Floquet codes from coupled spin chains*, [arXiv:2410.18265].

[11] A. Townsend-Teague, J. C. Magdalena de la Fuente, and M. Kesselring, *Floquetifying the colour code*, [arXiv:2307.11136].

[12] Z. Zhang, D. Aasen, and S. Vijay, *X-cube floquet code: A dynamical quantum error correcting code with a subextensive number of logical qubits*, Phys. Rev. B **108** (Nov, 2023) 205116, [arXiv:2211.05784].

[13] O. Higgott and N. P. Breuckmann, *Constructions and performance of hyperbolic and semi-hyperbolic floquet codes*, PRX Quantum **5** (2024) 040327, [arXiv:2308.03750].

[14] A. Fahimniya, H. Dehghani, K. Bharti, S. Mathew, A. J. Kollár, A. V. Gorshkov, and M. J. Gullans, *Fault-tolerant hyperbolic floquet quantum error correcting codes*, [arXiv:2309.10033].

[15] A. Dua, N. Tantivasadakarn, J. Sullivan, and T. D. Ellison, *Engineering 3d floquet codes by rewinding*, PRX Quantum **5** (2024) 020305, [arXiv:2307.13668].