Quantum error correction for quantum memories

Barbara M. Terhal

JARA Institute for Quantum Information, RWTH Aachen University, 52056 Aachen, Germany

(published 7 April 2015)

Active quantum error correction using qubit stabilizer codes has emerged as a promising, but experimentally challenging, engineering program for building a universal quantum computer. In this review the formalism of qubit stabilizer and subsystem stabilizer codes and their possible use in protecting quantum information in a quantum memory are considered. The theory of fault tolerance and quantum error correction is reviewed, and examples of various codes and code constructions, the general quantum error-correction conditions, the noise threshold, the special role played by Clifford gates, and the route toward fault-tolerant universal quantum computation are discussed. The second part of the review is focused on providing an overview of quantum error correction using two-dimensional (topological) codes, in particular, the surface code architecture. The complexity of decoding and the notion of passive or self-correcting quantum memories are discussed. The review does not focus on a particular technology but discusses topics that will be relevant for various quantum technologies.

DOI: 10.1103/RevModPhys.87.307 PACS numbers: 03.67.Pp, 03.67.Lx, 42.50.−p

CONTENTS

I. INTRODUCTION

Physics in the past century has demonstrated the experimental viability of macroscopic quantum *states* such as the superconducting state or a Bose-Einstein condensate. Quantum error correction, which strives to preserve not a single macroscopic quantum state but the macroscopic states in a small subspace, can be viewed as a natural but challenging extension to this. At the same time the storage of macroscopic quantum information is a first step toward the more ambitious goal of manipulating quantum information for computational purposes.

When the idea of a quantum computer took hold in the 1990s it was immediately realized that its implementation would require some form of robustness and error correction. Kitaev proposed a scheme in which the physical representation of quantum information and realization of logical gates would be naturally robust due to the topological nature of the 2D physical system (Kitaev, 2003). Around the same time Shor formulated a first quantum error-correcting (QEC) code and proved that a quantum computer could be made fault tolerant (Shor, 1996). Several others then established the fault-tolerance threshold theorem (see Theorem 1, Sec. II.F) which shows that in principle one can realize almost noise-free quantum computation using noisy components at the cost of a moderate overhead.

The goal of this review is to discuss the basic ideas behind active quantum error correction with stabilizer codes for the purpose of making a quantum memory. In this review we also discuss how Clifford group gates (Sec. II.G) are realized on the stored quantum data. In this sense the review goes beyond a pure quantum memory perspective, but for stabilizer codes these Clifford group gates play an essential role. Clifford gates are by themselves not sufficient for realizing universal fault-tolerant quantum computation.

We distinguish schemes of active quantum error correction from forms of passive quantum error correction or self-correction. In the latter quantum information is encoded in physical degrees of freedom which are naturally protected or have little decoherence, either through topology (topological order) or physical symmetries (symmetry-protected order) at sufficiently low temperature. Even though our review focuses on active quantum error correction, we will discuss some aspects of passive protection of quantum information using quantum error-correcting codes in Sec. III.E.

In an actively corrected quantum memory, quantum information is distributed among many elementary degrees of freedom, e.g., qubits, such that the dominant noise and decoherence processes affect this information in a reversible manner. This means that there exists an error-reversal procedure that allows one to undo the decoherence. The choice of how to represent the quantum information in the state space of many elementary qubits is made through the choice of a quantum error-correcting code. In order to execute the error reversal, active quantum error correction proceeds by continuously gathering information about which errors took place (for example, by quantum measurement), classical processing of this data, and applying a corrective quantum operation on the quantum data. The active gathering of information takes place, at least for stabilizer codes, via quantum measurements which measure the parity of subsets of qubits in Pauli matrix bases. These measurements are called parity check measurements. By the active gathering of error information, entropy is effectively removed from the computation and dumped into ancilla degrees of freedom which are supplied in known states to collect the error information. This active cycling of entropy from the computation into ancillary degrees of freedom which are further processed in a classical world makes active quantum error correction very different from the notion of passively storing quantum information in a low-temperature thermal environment.

In Sec. II.A we start by discussing Shor's code as the most basic example of a quantum error-correction code. Using Shor's code we illustrate the ideas behind the general framework of stabilizer codes (Gottesman, 1997), including subsystem stabilizer codes. We then treat stabilizer and subsystem stabilizer codes on qubits more formally in Secs. II.B and II.C. In Sec. II.B.2 we also discuss various small examples of quantum error-correcting codes and the construction due to Calderbank, Steane, and Shor by which two classical codes can be used to construct one quantum code. In Sec. II.D we widen our perspective beyond stabilizer codes and discuss the general quantum error-correction conditions as well as some codes which encode qubit(s) into bosonic mode(s) (oscillators). In Sec. II.E we define D-dimensional stabilizer codes and give various examples of such codes. Roughly speaking, D-dimensional stabilizer codes are quantum error-correcting codes where the elementary qubits are laid out on a D-dimensional lattice and all the quantum operations for quantum error correction can be executed by coupling qubits only locally on this lattice.

As the procedure of detecting and correcting errors itself is subject to noise, the existence of quantum error-correcting codes by itself does not yet show that one can store or compute with quantum information for an arbitrarily long time. In Sec. II.F we review how one can, through a procedure called code concatenation, arrive at the fault-tolerance threshold theorem. In essence, the threshold theorem says that in order to combat noise and decoherence we can add redundancy, a polylogarithmic overhead in the total number of qubits and overall computation time, provided that the fundamental noise rate on the elementary qubits is below some critical value which is called the noise threshold. Topological quantum error correction discussed in Sec. III provides a different route for establishing such a threshold theorem.

In Sec. II.F we also discuss various proposals for realizing quantum error correction, including the idea of dissipative engineering. The topic of the realization of quantum error correction is again picked up in Sec. III.D, but in that section the emphasis is on D-dimensional (topological) codes. In Sec. II.G, we review constructions for obtaining a universal set of logical gates for qubits encoded with stabilizer codes and motivate our focus on 2D topological stabilizer codes for use as a quantum memory.

For stationary, nonflying qubits, an important family of codes are quantum codes in which the elementary qubits can be laid out on a two-dimensional plane such that only local interactions between small numbers of nearest-neighbor qubits in the plane are required for quantum error correction. The practical advantage of such 2D geometry over an arbitrary qubit interaction structure is that no additional noisy operations need to be performed to move qubits around. Elementary solid-state qubits require various electric or magnetic control fields per qubit, for defining the qubit subspace and/or for single- and two-qubit control and measurement. The simultaneous requirement that qubits can interact sufficiently strongly and that space is available for these control lines imposes technological design constraints; see, e.g., Levy *et al.* (2009). A two-dimensional layout can be viewed as a compromise between the constraints coming from the coding theory and those from control line and material fabrication constraints: since quantum error-correcting codes defined on one-dimensional lines have poor error-correcting properties (Sec. II.E), it is advantageous to use a two-dimensional or a more general nonlocal layout of qubits. The qubits in such a layout should be individually addressable and/or defined by local electrostatic or magnetic fields, and thus 2D structures would be favored over 3D or general nonlocal interaction structures.

These considerations are the reason that we focus in Sec. III on 2D (topological) codes, in particular, the family of 2D topological surface codes, which has many favorable properties. For the surface code we show explicitly in Sec. III.A how many noisy elementary qubits can be used to represent one "encoded" qubit that has a much lower noise rate, assuming that the noise rate of the elementary qubits is below a critical value, the noise threshold. For the surface code this threshold turns out to be very high. We review two possible ways of encoding qubits in the surface code. We also discuss how logical gates such as the controlled-NOT (CNOT) and the Hadamard gate (see Sec. II.G for definitions of these gates) can be realized in a resource-efficient way.

In Sec. III.C we review a few interesting alternatives to the surface code: the nontopological Bacon-Shor code, a surface code with harmonic oscillators, and a subsystem version of the surface code. Section III.D discusses the physical locality of the process of decoding as well as recent ideas on the realization of so-called direct parity measurements. In Sec. III.E we discuss the ideas behind passive or self-correction and its relation with topological order.

We conclude our review with a discussion of some future challenges for quantum error correction. We recommend Lidar and Brun (2013) as a broad, comprehensive, reference on quantum error correction.

A. Error mitigation

Active quantum error correction is not the only way to improve the coherence properties of elementary physical quantum systems, and various well-known methods of error mitigation exist. In a wide variety of systems there is $1/f$ noise affecting the parameters of the qubit with a noise power spectral density $S(\omega) \sim 1/\omega^{\alpha}$, $\alpha \approx 1$, favoring slow fluctuations of those parameters that lead to qubit dephasing (Weissman, 1988). Standard NMR techniques (Vandersypen and Chuang, 2005) have been adapted in such systems to average out these fluctuations using rapid pulse sequences (e.g., spin echo). More generally, dynamical decoupling is a technique by which the undesired coupling of qubits to other quantum systems can be averaged out through rapid pulse sequences (Lidar, 2014). Aside from actively canceling the effects of noise, one can also try to encode quantum information in so-called decoherence-free subspaces which are effectively decoupled from noise; a simple example is the singlet state $(1/\sqrt{2})(|\uparrow,\downarrow\rangle - |\downarrow,\uparrow\rangle)$ which is invariant under a joint (unknown) evolution $U \otimes U$. This example is not yet a code as it encodes only one state, not a qubit. One can more generally formulate decoherence-free subspaces in which the encoded qubits are protected against collectively acting noise given by a set of error operators; see Chapter 3 in Lidar and Brun (2013).

In Sec. III.E we discuss another form of error mitigation: the encoding of quantum information in a many-body quantum system with a Hamiltonian corresponding to that of a D-dimensional quantum (stabilizer) code.

B. Some experimental advances

Experimental efforts have not yet advanced into the domain of *scalable* quantum error correction. Scalable quantum error correction would mean (1) making encoded qubits with decoherence rates that are genuinely below those of the elementary qubits and (2) demonstrating how, by increasing coding overhead, one can reach even lower decoherence rates, scaling in accordance with the theory of quantum error correction.

Several experiments exist concerning the three-qubit (or five-qubit) repetition code in liquid NMR, ion-trap, optical, and superconducting qubits. Four-qubit stabilizer pumping has been realized in ion-trap qubits (Barreiro *et al.*, 2011). Some topological quantum error correction has been implemented with eight-photon cluster states by Yao *et al.* (2012), and a continuous-variable version of Shor's nine-qubit code was implemented with optical beams (Aoki *et al.*, 2009). Bell *et al.* (2014) implemented the [[4,1,2]] code in an all-optical setup using a five-qubit polarization-based optical cluster state. Nigg *et al.* (2014) used seven trapped-ion qubits to represent, using Steane's seven-qubit code (see Sec. II.B.2), one effective, encoded, qubit, and several logical gates were performed on this encoded qubit via the transversal execution of gates on the seven elementary qubits.

The book by Lidar and Brun (2013) has a chapter with an overview of experimental quantum error correction. Given the advances in coherence times and ideas of multiqubit scalable design, in particular, in ion-trap and superconducting qubits (Barends *et al.*, 2014), one can hope to see scalable error correction, fine-tuned to experimental capabilities and constraints, in the years to come.

II. CONCEPTS OF QUANTUM ERROR CORRECTION

A. Shor's code and stabilizer codes

The goal of this section is to introduce the concepts and terminology of stabilizer codes in an informal way illustrated by Shor's nine-qubit code. In Sec. II.B we discuss the formalism of stabilizer codes and give further examples.

The smallest classical code that can correct a single bit-flip error (represented by[1] Pauli X) is the three-(qu)bit repetition code where we encode $|\overline{0}\rangle = |000\rangle$ and $|\overline{1}\rangle = |111\rangle$. A single error can be corrected by taking the majority of the three bit values and flipping the bit which is different from the majority. In quantum error correction we do not want to measure the three qubits to take a majority vote, as we would immediately lose the quantum information. This quantum information is represented in the amplitude $\cos(\theta)$ and the phase $e^{i\phi}$ of an encoded qubit state $|\overline{\psi}\rangle = \cos(\theta)|\overline{0}\rangle + \sin(\theta)e^{i\phi}|\overline{1}\rangle$. If we measure the three qubits in the $\{|0\rangle, |1\rangle\}$ basis, we may get answers which depend on $\cos(\theta)$ and $e^{i\phi}$, but we also decohere the quantum state, leaving just three classical bits and losing all information about $\cos(\theta)$ and $e^{i\phi}$.

But let us imagine that we can measure the parity checks Z_1Z_2 and Z_2Z_3 without learning the state of each individual qubit, that is, without the measurement revealing any information about the eigenvalues of Z_1, Z_2, or Z_3 individually. If the parity checks Z_1Z_2 and Z_2Z_3 have eigenvalues $+1$, one concludes that there is no error as the encoded states $|000\rangle$ and $|111\rangle$ have eigenvalue $+1$ with respect to these checks.

[1] The Pauli matrices are

$$\sigma_x \equiv X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \qquad \sigma_z \equiv Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix},$$

and

$$\sigma_y \equiv Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} = iXZ.$$

An outcome of, say, $Z_1Z_2 = -1$ and $Z_2Z_3 = 1$ is consistent with the erred state $X_1|\overline{\psi}\rangle$, where $|\overline{\psi}\rangle$ is any encoded state. And $Z_1Z_2 = 1$ and $Z_2Z_3 = -1$ points to the error X_3. But how can we measure Z_1Z_2 and Z_2Z_3 without measuring the individual Z_i operators and destroying the encoded qubit? Essentially, through making sure that the "signals" from different qubits are indistinguishable and only a global property like parity is communicated to the outside world. One can realize this with a quantum circuit as follows.

One uses an extra "ancilla" qubit which will interact with the three qubits such that the value of the parity check is copied onto the ancilla qubit. A general circuit which measures a "parity check," represented by a multiqubit Pauli operator P, using an ancilla is given in Fig. 1(a). One can verify the action of the circuit by writing an arbitrary input state as a superposition of a $+1$ eigenstate ψ_{+1} and a -1 eigenstate ψ_{-1} of the Pauli operator P to be measured ($P|\psi_{\pm}\rangle = \pm|\psi_{\pm}\rangle$). A concrete example for $P = X_1X_2X_3X_4$ is given in Fig. 1(c), where we decomposed the five-qubit controlled-P gate into four two-qubit controlled-X or CNOT gates. In such a circuit, the parity information is collected in steps, via several CNOT gates, so that the state of the ancilla qubit during the execution of the gates does contain information about the individual qubits. It is thus important that this partial information on the ancilla qubit does not leak to the environment as it leads to decoherence on the encoded qubits during the parity check measurement. One can see that the parity check measurement using an ancilla qubit initially set to a fixed, known state is actively letting us remove entropy from the computation by providing us information about what errors have taken place.

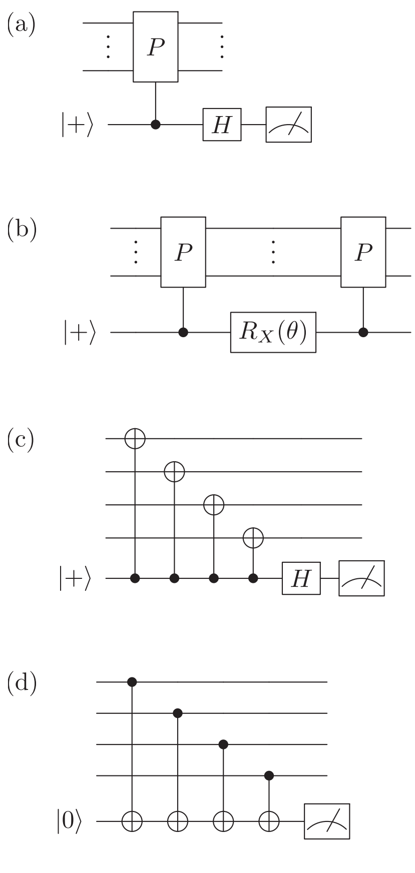

FIG. 1. Measuring parity checks in the quantum-circuit way. The meter denotes measurement in the $\{|0\rangle, |1\rangle\}$ basis or M_Z. (a) Circuit to measure the ± 1 eigenvalues of a unitary multiqubit Pauli operator P. The gate is the controlled-P gate which applies P when the control qubit is 1 and I if the control qubit is 0. (b) Realizing the evolution $\exp(-i\theta P/2)$ itself [with $R_x(\theta) = \exp(-i\theta X/2)$]. (c) Realization of circuit (a) using CNOT gates when $P = X_1X_2X_3X_4$. (d) Realization of circuit (a) using CNOT gates when $P = Z_1Z_2Z_2Z_4$.

It may be clear that the three-qubit repetition code does not protect or detect Z (dephasing) errors as these parity checks measure information only in the Z basis (M_Z). More precisely, any single-qubit Z error will harm the quantum information. We encode the qubit state $|+\rangle \equiv (1/\sqrt{2})(|0\rangle + |1\rangle)$ as $|\overline{+}\rangle = (1/\sqrt{2})(|000\rangle + |111\rangle)$ and $|-\rangle \equiv (1/\sqrt{2})(|0\rangle - |1\rangle)$ as $|\overline{-}\rangle = (1/\sqrt{2})(|000\rangle - |111\rangle)$. We can verify that any single-qubit Z error, say Z_1, maps $|\overline{+}\rangle \leftrightarrow |\overline{-}\rangle$, corrupting the quantum information.

Having seen this simple example, we informally introduce some of the notions used in describing a quantum (stabilizer) code. In general we denote logical or encoded states as $|\overline{\psi}\rangle$ and logical operators as $\overline{X}$, $\overline{Z}$, etc., where by definition $\overline{X}|\overline{0}\rangle \leftrightarrow |\overline{1}\rangle$ and $\overline{Z}|\overline{+}\rangle \leftrightarrow |\overline{-}\rangle$. The logical operators $\overline{X}$, $\overline{Z}$ can always be expressed in terms of their action as Pauli operators on the elementary qubits. For a code C encoding k qubits, one defines k pairs of logical Pauli operators $(\overline{X}_i, \overline{Z}_i)$, $i = 1, \ldots, k$, such that $\overline{X}_i\overline{Z}_i = -\overline{Z}_i\overline{X}_i$, while logical Pauli operators with labels i and i' mutually commute. The logical Pauli operators simply realize the algebra of the Pauli operators acting on k qubits. For the three-qubit code we have $\overline{X} = X_1X_2X_3$ (flipping all the bits) and $\overline{Z} = Z_1$.

The *code space* of a code C encoding k qubits is spanned by code words $|\overline{x}\rangle$, where x is a k-bit string. In general, these code words $|\overline{x}\rangle$ will be highly entangled states. All states in the code space obey the parity checks, meaning that the parity check operators have eigenvalue $+1$ for all states in the code space (we say that the parity checks act trivially on the code space). The parity checks are all represented by mutually commuting multiqubit Pauli operators. The logical operators of a quantum error-correcting code are nonunique as we can multiply them by the trivially acting parity check operators to obtain equivalent operators. For example, $\overline{Z}$ for the three-qubit code is either Z_1, or Z_2, or Z_3, or $Z_1Z_2Z_3$ as all these operators have the same action $|\overline{+}\rangle \leftrightarrow |\overline{-}\rangle$.

Shor's nine-qubit code was the first quantum error-correcting code which encodes a single qubit and corrects any single-qubit Pauli error, i.e., single-qubit bit-flip errors X, phase-flip errors Z, and bit + phase-flip errors Y. As it turns out, if one wants to correct against any single-qubit error, it is sufficient to be able to correct against any single-qubit Pauli error. We thus assume for now that the only possible errors are multiqubit or single-qubit Pauli errors and afterward we show that correcting against such Pauli errors is indeed sufficient.

Shor's code is obtained from the three-qubit repetition code by *concatenation*. Code concatenation is a procedure in which we take the elementary qubits of the code words of a code C and replace them by encoded qubits of a new code C'. In Shor's construction we choose the first code C as a "rotated" three-bit repetition code; that is, we take $|\overline{+}\rangle = |+\rangle|+\rangle|+\rangle$

and $|\overline{-}\rangle = |-\rangle|-\rangle|-\rangle$ with $|\pm\rangle = (1/\sqrt{2})(|0\rangle \pm |1\rangle)$. One can verify that the parity checks of C are X_1X_2 and X_2X_3 and the logical operators are $\overline{Z}_C = Z_1Z_2Z_3$ and $\overline{X}_C = X_1$. As the second code C' we choose the normal three-qubit repetition code, i.e., we replace $|+\rangle$ by $|\overline{+}\rangle = (1/\sqrt{2})(|000\rangle + |111\rangle)$, etc.

We get all the parity checks for the concatenated nine-qubit code by taking all the parity checks of the codes C' and the C'-encoded parity checks of C. For Shor's code this will give the Z checks Z_1Z_2, Z_2Z_3, Z_4Z_5, Z_5Z_6, Z_7Z_8, and Z_8Z_9 (from three uses of the code C') and the X checks $X_1X_2X_3X_4X_5X_6$, $X_4X_5X_6X_7X_8X_9$ (from the parity checks $\overline{X}_1\overline{X}_2$ and $\overline{X}_2\overline{X}_3$, where $\overline{X}$ is the logical operator of the code C'). The nonunique logical operators of the encoded qubit are $\overline{Z} = Z_1Z_4Z_7$ and $\overline{X} = X_1X_2X_3$.

This code can clearly correct any X error as it consists of three qubits each of which is encoded in the repetition code which can correct an X error. What happens if a single Z error occurs on any of the qubits? A single Z error will anticommute with one of the parity X checks or with both. For example, the error Z_1 anticommutes with $X_1X_2X_3X_4X_5X_6$ so that the state $Z_1|\overline{\psi}\rangle$ has eigenvalue -1 with respect to this parity check. The error Z_2 or error Z_3 would have the same syndrome: these errors have the same effect on the code space as Z_1Z_2 and Z_2Z_3 act trivially on the code space. The same holds for the three qubits in the second block and the three qubits in the third block. Thus the X parity check will tell you only whether there is a Z error in the first, the second, or the third block but this is acceptable, and any single error in the block applies the proper correction on the code space. Thus the code can correct against a single-qubit X error, or Z error and therefore also Y error.

The eigenvalues of the parity check operators are called the *error syndrome*. Aside from detecting errors (finding -1 syndrome values) the error syndrome should allow one to infer which error occurred. How do we make this inference in general? We could assign a probability to each possible error: this assignment is captured by the *error model*. Then our decoding procedure can simply choose an error, consistent with the syndrome, which has highest probability given our error model. Typically, the error model would assign a lower probability to errors that act on many qubits, and so the decoding could consist of simply picking a Pauli error, which could be responsible for the given syndrome that acts on the fewest number of qubits. This kind of decoding is called minimum-weight decoding. It is important to note that the decoding procedure does not necessarily have to point to a unique error. For example, for the nine-qubit code, the errors Z_1 and Z_2 have an equivalent effect on the code space as Z_1Z_2 is a parity check which acts trivially on the code space. The syndromes for errors that are related by parity checks are always identical: the syndrome of a Pauli error E is determined by the parity checks with which it anticommutes. Multiplying E by parity checks, which are by definition all mutually commuting operators, does not change the syndrome. This means that the classical algorithm which processes the syndrome to infer an error (this procedure is called *decoding*) does not need to choose between such equivalent errors.

But there is further ambiguity in the error syndrome. For Shor's code the errors Z_1 and Z_4Z_7 have an identical syndrome as $Z_1Z_4Z_7$ is the $\overline{Z}$ operator which commutes with all parity checks. If a single nontrivial (-1) syndrome is obtained for the parity check $X_1X_2X_3X_4X_5X_6$, we could decide that the error is Z_1 or Z_4Z_7. But if we make a mistake in this decision and correct with Z_4Z_7 while Z_1 happened, then we have effectively performed a $\overline{Z}$ without knowing it. This means that the decoding procedure should decide between errors, all of which are consistent with the error syndrome that are mutually related by logical operators. We discuss the procedure of decoding more formally in Sec. II.B. For Shor's code we can decode the syndrome by picking a single-qubit error which is consistent with the syndrome. If a two-qubit error occurred, we may thus have made a mistake. However, for Shor's code there are no two single-qubit errors E_1 and E_2 with the same syndrome whose product E_1E_2 is a logical operator as each logical operator acts on at least three qubits. This is another way of seeing that Shor's code can correct any single-qubit Pauli error. It is a $[[n,k,d]] = [[9,1,3]]$ code, encoding $k = 1$ qubit into $n = 9$ (n is called the block size of the code) and having distance $d = 3$. Having seen how this works for Shor's code, we understand the role of the distance of a code more generally as follows.

The distance d of the code is defined as the minimum weight of any logical operator [see the formal definition in Eq. (1)]. The weight of a Pauli operator is the number of qubits on which it acts nontrivially, i.e., Z_4Z_7 has a weight of 2. The definition of distance refers to a minimum weight of any logical operator as there are several logical operators, i.e., $\overline{X}$, $\overline{Z}$, etc., and we want any of them to have a high weight, and the weight of each one of them can be varied by multiplication with parity checks.

It is simple to understand why a code with distance $d = 2t + 1$ can correct t errors. Namely, errors of weight at most t have the property that their products have weight at most $2t < d$. Therefore the product of these errors can never be a logical operator as those have weight d or more. Thus if one of these errors E_1 occurs and our decoding procedure picks another error E_2 of weight at most t (both giving rise to the same syndrome) and applies E_2 to the encoded qubits, then effectively we have the state $E_2E_1|\overline{\psi}\rangle$. This state has a trivial syndrome as all parity checks commute with E_1E_2 (they either anticommute with both E_1 and E_2 or commute with both), but E_1E_2 has weight $2t < d$. Thus E_1E_2 cannot be a logical operator but has to be some product of trivially acting parity checks as E_1E_2 commutes with all parity checks.

Another direct consequence of the distance of the code is how the code can handle so-called *erasure* errors. If errors take place on only some known subset of qubits, then a code with distance d can correct (errors on) subsets of size $d - 1$ as the product of any two Pauli errors on this subset has weight at most $d - 1$. In other words, if $d - 1$ or fewer qubits of the code word are lost or their state completely erased by other means, one can still recover the entire code word from the remaining set of qubits. One could do this as follows. First one replaces the lost $d - 1$ qubits by the completely mixed state $I/2^{d-1}$.[2]

[2] The erasure of a qubit, i.e., the qubit state ρ is replaced by $I/2$, can be written as the process of applying an I, X, Y or Z error with probability $1/4$: $I/2 = (\rho + X\rho X + Z\rho Z + Y\rho Y)/4$.

Then one measures the parity checks on all qubits, which gives us a syndrome that is nontrivial only for the parity checks which act on the $d-1$ qubits that had been erased. The syndrome points to a (nonunique) Pauli operator acting on these $d-1$ qubits or fewer and applying this Pauli operator corrects the error.

1. Error modeling

Clearly, the usefulness of error correction is directly related to the error model; it hinges on the assumption that low-weight errors are more likely than high-weight errors. Error correcting a code which can perfectly correct errors with weight at most t will lead to failure with probability roughly equal to the total probability of errors of weight larger than t. This probability for failure of error correction is called the *logical error probability*. The goal of quantum error correction is to use redundancy and correction to realize logical qubits with logical error rates below the error rate of the elementary constituent qubits.

It may seem rather simplistic and limiting to use error models which assign X, Z, and Y errors probabilistically to qubits as in real quantum information, through the interaction with classical or quantum systems, the amplitude and phase of a qubit will fluctuate over time. Bare quantum information encoded in atomic, photonic, spin, or other single quantum systems is barely information as it is undergoing continuous changes. It is important to note that the ideal parity check measurement provides a discretization of the set of errors which is not naturally present in such elementary quantum systems.

Consider, for example, noise on a single qubit due the fact that its time evolution (in a rotating frame) is not completely canceled and equals $\exp(-i\delta\omega Zt/2)$ for some probability distribution over frequencies $\mathrm{Prob}(\delta\omega)$ centered around $\delta\omega = 0$. If this qubit is, say, the first qubit that is part of a multiqubit encoded state $|\overline{\psi}\rangle$, we can write $\exp(-i\delta\omega Z_1 t/2)|\overline{\psi}\rangle = [\cos(\delta\omega t)I + iZ_1 \sin(\delta\omega t)]|\overline{\psi}\rangle$, i.e., we expand the small error of strength $\delta\omega$ in a basis of Pauli errors which occur with some amplitude related to $\delta\omega$. Consider then measuring a parity X check which involves qubit 1. One obtains eigenvalue $+1$ with probability $\cos^2(\delta\omega t)$, close to 1 for small $\delta\omega t$, and we project onto the error-free state $|\overline{\psi}\rangle$. We obtain eigenvalue -1 with small probability $\sin^2(\delta\omega t)$ when we project onto the state with Pauli error $Z_1|\overline{\psi}\rangle$. Since any operator E on n qubits can be expanded in a basis of Hermitian Pauli matrices, this simple example illustrates the general principle that the correction of Pauli errors of weight less than t suffices for the correction of any error of weight less than t. This property holds in fact for arbitrary quantum codes (including nonstabilizer codes for which we may gather error information through different means than parity check measurements), as follows from the quantum error-correction conditions; see Sec. II.D.

Ideal parity measurement can induce such a discrete error model stated in terms of probabilities, but as parity measurements themselves will be inaccurate in a continuous fashion, such a fully digitized picture is an oversimplification. The theory of quantum fault tolerance (see Sec. II.F) has developed a framework that allows one to establish the results of quantum error correction and fault tolerance for very general quantum dynamics obeying physical locality assumptions [see the comprehensive results by Aliferis, Gottesman, and Preskill (2006)]. However, for numerical studies of code performance it is impossible to simulate such more general open-system dynamics, and several simple error models are used to capture the expected performance of the codes.

Two further remarks can be made with this general framework in mind. First, errors can be correlated in space and time arising from non-Markovian dynamics, but as long as (a) we use the proper estimate of the strength of the noise (which may involve using amplitudes and norms rather than probabilities) and (b) the noise is sufficiently short ranged [meaning that noisy interactions between distant uncoupled qubits are sufficiently weak (Aharonov, Kitaev, and Preskill, 2006)], fault-tolerance threshold results can be established. The second remark is that qubit coding does not directly deal with leakage errors. As many elementary qubits are realized as two-level subspaces of higher-dimensional systems to which they can leak, other protective mechanisms such as cooling (or teleporting to a fresh qubit) will need to be employed in order to convert a leakage error into a regular error which can be corrected. Aliferis and Terhal (2007) showed that one can derive general fault-tolerance threshold results for leakage errors by invoking the use of leakage reduction units (LRUs) such as quantum teleportation.

2. Shor's code as a subsystem code

We return to Shor's code and imagine that the nine qubits are laid out in a 3×3 square array as in Fig. 2. It looks relatively simple to measure the parity Z checks locally, while the weight-6 X checks would require a larger circuit using six CNOT gates between ancilla and data qubits. But why should there be such asymmetry between the X and Z checks? Imagine that instead of measuring the "double row" stabilizer operator $\mathbf{X}_{=,1} \equiv X_1X_2X_3X_4X_5X_6$, we measure (in parallel or sequentially) the eigenvalues of X_1X_4, X_2X_5, and X_3X_6 and take the product of these eigenvalues to obtain the eigenvalue of $\mathbf{X}_{=,1}$. The important property of these weight-2 operators is that they all individually commute with the logical operators $\overline{X}$ and $\overline{Z}$ of the Shor code, and hence measuring them does not change the expectation values of $\overline{X}$ and $\overline{Z}$. These weight-2 X checks do not commute with the weight-2 Z checks

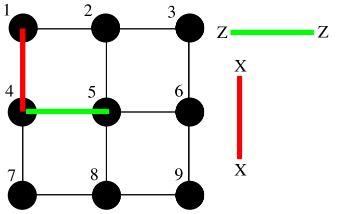

FIG. 2 (color online). The nine-qubit [[9,1,3]] Shor code with black qubits on the vertices. The stabilizer of Shor's code is generated by the weight-2 Z checks as well as two weight-6, double-row X checks $\mathbf{X}_{=,1} = X_1X_2X_3X_4X_5X_6$ and $\mathbf{X}_{=,2}$. An alternative way of measuring $\mathbf{X}_{=,1}$ and $\mathbf{X}_{=,2}$ is by measuring the weight-2 X checks. One can similarly define two weight-6, double-column Z checks $\mathbf{Z}_{\|,1}$ and $\mathbf{Z}_{\|,2}$ as products of elementary weight-2 Z checks. See also Fig. 14.

however. If we first measure all the weight-2 X checks and then measure the Z checks, then with the second step the eigenvalues of individual X checks are randomized but correlated. Namely, their product $X_1X_2X_3X_4X_5X_6$ remains fixed as $X_1X_2X_3X_4X_5X_6$ commutes with the weight-2 Z checks. By symmetry, the weight-2 X checks commute with the double-column operators $\mathbf{Z}_{\parallel,1} = Z_1Z_2Z_4Z_5Z_7Z_8$ and $\mathbf{Z}_{\parallel,2} = Z_2Z_3Z_5Z_6Z_8Z_9$. By viewing the Shor code in this way we can imagine doing error correction and decoding using the stable commuting parity checks $\mathbf{X}_{=,1}$, $\mathbf{X}_{=,2}$, $\mathbf{Z}_{\parallel,1}$, and $\mathbf{Z}_{\parallel,2}$ while we deduce their eigenvalues from measuring 12 weight-2 parity checks.

Shor's code in this form is the smallest member in the family of Bacon-Shor codes $[[n^2, 1, n]]$ (Bacon, 2006; Aliferis and Cross, 2007) whose qubits can be laid out in an $n \times n$ array as in Fig. 14; see Sec. III.C.1. The Bacon-Shor code family in which noncommuting (low-weight) parity checks are measured in order to deduce the eigenvalues of commuting parity checks is an example of a (stabilizer) subsystem code.

B. Formalism of stabilizer codes

Shor's code and many existing codes defined on qubits are examples of stabilizer codes[3] (Gottesman, 1997). Stabilizer codes are attractive as (i) they are the straightforward quantum generalization of classical binary linear codes, (ii) their logical operators and distance are easily determined, and it is relatively simple to (iii) understand how to construct universal sets of logical gates, and (iv) execute a numerical analysis of the code performance.

The main idea of stabilizer codes is to encode k logical qubits into n physical qubits using a subspace, the code space, $\mathcal{L} \subseteq (\mathbb{C}^2)^{\otimes n}$ spanned by states $|\psi\rangle$ that are invariant under the action of a stabilizer group $\mathcal{S}$,

$$\mathcal{L} = \{|\psi\rangle \in (\mathbb{C}^2)^{\otimes n}\colon\ P|\psi\rangle = |\psi\rangle \quad \forall\ P \in \mathcal{S}\}.$$

Here $\mathcal{S}$ is an Abelian subgroup of the Pauli group $\mathcal{P}_n = \langle iI, X_1, Z_1, \ldots, X_n, Z_n\rangle$ such that $-I \notin \mathcal{S}$.[4] For any stabilizer group $\mathcal{S}$ one can always choose a set of generators $S_1, \ldots, S_m$, i.e., $\mathcal{S} = \langle S_1, \ldots, S_m\rangle$, such that $S_a \in \mathcal{P}_n$ are Hermitian Pauli operators. The advantage of this stabilizer formalism is that instead of specifying the code space by a basis of 2^n-dimensional vectors, we specify the code space by the generators of the stabilizer group which fix (or stabilize) these vectors. The mutually commuting parity checks that we considered before are the generators of the stabilizer group. If there are $n - k$ linearly independent generators (parity checks) then the code space $\mathcal{L}$ is 2^k dimensional, or encodes k qubits. This description of the code space in a 2^n-dimensional vector space is thus highly efficient as it requires specifying at most n linearly independent parity checks.

The weight $|P|$ of a Pauli operator $P = P_1 \cdots P_n \in \mathcal{P}_n$ is the number of single-qubit Pauli operators P_i that are unequal to I, in other words, the number of qubits on which P acts nontrivially. If the code encodes k logical qubits, it is always possible to find k pairs of logical operators $(\overline{X}_j, \overline{Z}_j)_{j=1,\ldots,k}$. These logical operators commute with all the parity checks, i.e., they commute with all elements in $\mathcal{S}$ as they preserve the code space. However, they should not be generated by the parity checks themselves; otherwise their action on the code space is trivial. Thus these logical operators are elements of the Pauli group $\mathcal{P}_n$ which are not elements in $\mathcal{S}$ (otherwise their action is trivial), but which commute with all elements in $\mathcal{S}$. The set of operators in $\mathcal{P}_n$ which commutes with $\mathcal{S}$ is called the centralizer of $\mathcal{S}$ in $\mathcal{P}_n$, defined as $\mathcal{C}(\mathcal{S}) = \{P \in \mathcal{P}_n | \forall\ s \in \mathcal{S}, Ps = sP\}$. We thus have $\mathcal{C}(\mathcal{S}) = \langle \mathcal{S}, \overline{X}_1, \overline{Z}_1, \ldots, \overline{X}_k, \overline{Z}_k\rangle$, i.e., the logical operators of the code are elements of $\mathcal{C}(\mathcal{S})\backslash\mathcal{S}$ as they are in $\mathcal{C}(\mathcal{S})$ but not in $\mathcal{S}$.[5] The distance d of a stabilizer code can then be defined as

$$d = \min_{P \in \mathcal{C}(\mathcal{S})\backslash\mathcal{S}} |P|, \tag{1}$$

i.e., the minimum weight that any logical operator can have. As the logical operators $\overline{P} \in \mathcal{C}(\mathcal{S})\backslash\mathcal{S}$ commute with all parity check operators both the code state $|\overline{\psi}\rangle$ and $\overline{P}|\overline{\psi}\rangle$ have $+1$ eigenvalues with respect to the parity checks. Measuring the parity checks thus does not reveal whether a $\overline{P}$ has taken place or not while the quantum information is greatly changed. Clearly, these logical operators $\overline{P}$ should be prevented from happening. A good stabilizer code will have high distance d so that it is unlikely that local low-rate decoherence processes acting on a few qubits at a time will lead to a logical operator $\overline{P}$ that one cannot undo.

1. Decoding

Error correction proceeds by measuring the error syndrome $\mathbf{s}$ which is a vector of ± 1 eigenvalues of the generators of $\mathcal{S}$. As mentioned in Sec. II.A this syndrome will not point to a unique Pauli error but all $E' = EP$, where $P \in \mathcal{C}(\mathcal{S})$ give rise to the same syndrome. We now describe the formal procedure of decoding.

We define an equivalence class of errors $[E]$ consisting of errors $E' = EP$, where $P \in \mathcal{S}$; that is, elements in $[E]$ are related to E by a (trivially acting) element in the stabilizer group.[6] If error E occurs and we decide to correct this error by applying E', the sequence $EE' \in \mathcal{S}$ has been applied to the code word, leaving it unchanged. We can associate a total error probability with such a class, $\text{Prob}([E]) = \sum_{s \in S} \text{Prob}(Es)$, depending on some error model which assigns a probability $\text{Prob}(P)$ to every Pauli operator $P \in \mathcal{P}_n$. Given an error E and a syndrome, one can similarly

[3] Readers less interested in this general framework can skip the next two sections without major inconvenience.

[4] $G = \langle g_1, \ldots, g_m\rangle$ denotes a group G generated by elements $g_1, \ldots, g_m \in G$.

[5] In some quantum error-correction literature $\mathcal{C}(\mathcal{S})\backslash\mathcal{S}$ is denoted as $\mathcal{C}(\mathcal{S}) - \mathcal{S}$. Also, the centralizer $\mathcal{C}(\mathcal{S})$ of $\mathcal{S}$ in $\mathcal{P}$ is sometimes referred to as the normalizer $\mathcal{N}(\mathcal{S})$: for Pauli operators which either commute or anticommute these groups coincide.

[6] $[E]$ is a coset of the group $\mathcal{S}$ in $\mathcal{P}_n$. Note that left and right cosets are the same modulo trivial errors proportional to I.

define a discrete number of classes $[E\overline{P}]$, where the logicals $\overline{P} \in \mathcal{C}(\mathcal{S})\backslash\mathcal{S}$.

The procedure that maximizes the success probability of reversing the error while making no logical error is called *maximum-likelihood decoding*. Given a syndrome $\mathbf{s}$ and some error $E(\mathbf{s})$ which is consistent with the syndrome, a maximum-likelihood decoder compares the values of $\text{Prob}([E\overline{P}])$ for the various $\overline{P}$ and chooses the one with maximal value pointing to some $\overline{P}$. Then it applies the corrective operator $E\overline{P}$ which is by definition the most likely correction.

If $E(\mathbf{s})$ happens to be the error E that actually took place, then this decoding procedure is successful when $\text{Prob}([E]) > \text{Prob}([E\overline{P}])$ for any nontrivial $\overline{P}$.

It is important to consider how efficiently (in the number n of elementary qubits) maximum-likelihood decoding can be done since $\text{Prob}([E\overline{P}])$ is a sum over the number of elements in $\mathcal{S}$, which is exponential in n. For a simple depolarizing error model where each qubit undergoes an X, Y, or Z error with probability $p/3$ and no error with probability $1-p$, one has $\text{Prob}([E\overline{P}]) = (1-p)^n \sum_{s\in\mathcal{S}} \exp(-\beta|E\overline{P}s|)$ with inverse "temperature" $\beta = \ln[3(1-p)/p]$.

We can define a classical Hamiltonian $H_{E\overline{P}}(s) \equiv |E\overline{P}s|$ which acts on spin variables $s_i \in \{-1, 1\}$ each of which corresponds to a generator S_i of the stabilizer group $\mathcal{S}$. The Hamiltonian will be a sum of terms, each corresponding to a single qubit in the code and contributing either 0 or 1. Each term can be written as a function of the stabilizer generators $s_i = \pm 1$, E, and $\overline{P}$ which act on the particular qubit, making it trivial (weight 0) or nontrivial (weight 1). We can view $Z_{E\overline{P}} \equiv \sum_{s\in\mathcal{S}} \exp[-\beta H_{E\overline{P}}(s)]$ as a partition function of the Hamiltonian $H_{E\overline{P}}(s)$ at a temperature related to the error probability.

For small error rates $p \ll 1$ corresponding to low temperatures $\beta \to \infty$, the value of this partition function is dominated by the spin configuration s that minimizes $H_{E\overline{P}}(s) = |E\overline{P}s|$. Thus for sufficiently low error rates, instead of maximum-likelihood decoding, which compares the relative values of $\text{Prob}([E\overline{P}])$, one can also opt for minimum-weight decoding. In minimum-weight decoding one simply picks an error $E(\mathbf{s})$, consistent with the syndrome $\mathbf{s}$, which has minimum weight $|E|$. We discuss this decoding method for the surface code in Sec. III.

For topological codes, the criterion for successful maximum-likelihood decoding and the noise threshold of the code can be related to a phase transition in a classical statistical model with quenched disorder (Dennis *et al.*, 2002; Katzgraber and Andrist, 2013). This can be readily understood as follows. A probabilistic noise model such as the depolarizing noise model induces a probability distribution $\text{Prob}(E)$ over the errors E. For a given error E we can decode successfully when $Z_E > Z_{\overline{P}E}$, where Z_E is the partition function of the quenched-disorder Hamiltonian $H_E(s) = |Es|$ defined previously. We want to be able to decode successfully for typical errors E; hence we are interested in looking at averages over the disorder E. Assume that one has a family of codes for which one can define a thermodynamic limit in which the number of qubits $n \to \infty$. One can define a critical, say depolarizing, error rate p_c by the following condition:

$$\begin{aligned} p < p_c &\to \lim_{n\to\infty} \sum_E \text{Prob}(E) \log\left(\frac{Z_E}{Z_{\overline{P}E}}\right) = \infty, \\ p > p_c &\to \lim_{n\to\infty} \sum_E \text{Prob}(E) \log\left(\frac{Z_E}{Z_{\overline{P}E}}\right) = 0. \end{aligned} \tag{2}$$

One thus studies the behavior of the free energy of the statistical model with quenched disorder (which is determined by the error probability p), i.e., $\langle \log Z_E \rangle_p = \sum_E \text{Prob}(E) \log Z_E$, to determine the value of p_c. The temperature β and the quenched disorder are not independent but directly depend on the same error probability p. For this reason one identifies p_c with a phase transition of the quenched-disorder model along the so-called Nishimori line on which β is a function of the strength of the error probability which is also the disorder parameter. One of the first studies of this sort was done by Wang, Harrington, and Preskill (2003).

2. Stabilizer code examples and the Calderbank-Shor-Steane construction

We discuss a few small examples of stabilizer codes to illustrate the formalism. For classical error correction the smallest code that can detect an X error is a two-bit code and the smallest code that can correct any X error is the three-qubit code. As a quantum error-correcting code has to correct both X and Z errors, the smallest quantum error-correcting code will have more qubits.

We consider first the two-qubit code. For the two-qubit code with $|\overline{0}\rangle = (1/\sqrt{2})(|00\rangle + |11\rangle)$ and $|\overline{1}\rangle = (1/\sqrt{2})(|01\rangle + |10\rangle)$ we have $\overline{X} = X_1$ or $\overline{X} = X_2$ and $\overline{Z} = Z_1 Z_2$. The code can detect any single Z error as such an error maps the two code words onto the orthogonal states $(1/\sqrt{2})(|00\rangle - |11\rangle)$ and $(1/\sqrt{2})(|01\rangle - |10\rangle)$ (as $\overline{Z}$ is of weight 2). The code cannot detect single X errors as these are logical operators.

The smallest nontrivial quantum code is the [[4,2,2]] error-detecting code. Its linearly independent parity checks are $X_1X_2X_3X_4$ and $Z_1Z_2Z_3Z_4$: the code encodes $4-2=2$ qubits. One can verify that one can choose $\overline{X}_1 = X_1X_2$, $\overline{Z}_1 = Z_1Z_3$ and $\overline{X}_2 = X_2X_4$, $\overline{Z}_2 = Z_3Z_4$ as the logical operators which commute with the parity checks. The code distance is 2, which means that the code cannot correct a single-qubit error. The code can however still detect any single-qubit error as any single-qubit error anticommutes with at least one of the parity checks, which leads to a nontrivial -1 syndrome. Alternatively, we can view this code as a subsystem code (see Sec. II.C) which has one logical qubit, say, qubit 1, and one gauge qubit, qubit 2. In that case $\mathcal{G} = \langle X_1X_2X_3X_4, Z_1Z_2Z_3Z_4, Z_3Z_4, X_2X_4\rangle = \langle Z_1Z_2, Z_3Z_4, X_1X_3, X_2X_4\rangle$, showing that measuring weight-2 checks would suffice to detect single-qubit errors on the encoded qubit 1. The smallest stabilizer code that encodes one qubit and corrects one error is the [[5,1,3]] code; one can find its parity checks in Nielsen and Chuang (2000).

In order to make larger codes out of small codes one can use the idea of code concatenation which we first illustrate with an explicit example. We take a small stabilizer code C_6 [defined in Knill (2005)] with parity checks $X_1X_4X_5X_6$, $X_1X_2X_3X_6$,

$Z_1Z_4Z_5Z_6$, and $Z_1Z_2Z_3Z_6$ acting on six qubits. This code has four independent parity checks; hence it encodes $6-4=2$ qubits with the logical operators $\overline{X}_1 = X_2X_3$, $\overline{Z}_1 = Z_3Z_4Z_6$ and $\overline{X}_2 = X_1X_3X_4$, $\overline{Z}_2 = Z_4Z_5$. As its distance is 2, it can detect only single X or Z errors.

One can concatenate this code C_6 with the code [[4,2,2]] [called C_4 by Knill (2005)] by replacing the three pairs of qubits, i.e., the pairs (12), (34), and (56), in C_6 by three sets of C_4-encoded qubits, to obtain a new code. This code has thus $n = 12$ qubits and encodes $k = 2$ qubits. We can represent these 12 qubits as three sets of four qubits such that the X checks read

$$\mathcal{S}(X) = \begin{pmatrix} X & X & X & X & I & I & I & I & I & I & I & I \\ I & I & I & I & X & X & X & X & I & I & I & I \\ I & I & I & I & I & I & I & I & X & X & X & X \\ X & X & I & I & I & X & I & X & X & I & I & X \\ X & I & I & X & X & X & I & I & I & X & I & X \end{pmatrix}.$$

The Z checks are

$$\mathcal{S}(Z) = \begin{pmatrix} Z & Z & Z & Z & I & I & I & I & I & I & I & I \\ I & I & I & I & Z & Z & Z & Z & I & I & I & I \\ I & I & I & I & I & I & I & I & Z & Z & Z & Z \\ Z & I & Z & I & I & I & Z & Z & Z & I & I & Z \\ Z & I & I & Z & Z & I & Z & I & I & I & Z & Z \end{pmatrix},$$

and the logical operators are

$$\begin{aligned} \overline{X}_1 &= & I \quad X \quad I \quad X \quad X \quad X \quad I \quad I \quad I \quad I \quad I \quad I, \\ \overline{Z}_1 &= & I \quad I \quad I \quad I \quad Z \quad I \quad I \quad Z \quad I \quad I \quad Z \quad Z, \\ \overline{X}_2 &= & X \quad X \quad I \quad I \quad X \quad I \quad I \quad X \quad I \quad I \quad I \quad I, \\ \overline{Z}_2 &= & I \quad I \quad I \quad I \quad I \quad I \quad Z \quad Z \quad Z \quad I \quad Z \quad I. \end{aligned}$$

One can verify that the minimum weight of the logical operators of this concatenated code is 4. Thus the code is a [[12,2,4]] code, able to correct any single error and to detect any three errors.

One could repeat the concatenation step and recursively concatenate C_6 with itself (replacing a pair of qubits by three pairs of qubits, etc.) as in Knill's C_4/C_6 architecture (Knill, 2005) or, alternatively, recursively concatenate C_4 with itself as considered by Aliferis and Preskill (2009).

In general when we concatenate an $[[n_1, 1, d_1]]$ code with an $[[n_2, 1, d_2]]$ code, we obtain a code which encodes one qubit into $n = n_1 n_2$ qubits and has distance $d = d_1 d_2$. Code concatenation is a useful way to obtain a large code from smaller codes as the number of syndrome collections scales linearly with the number of concatenation steps while the number of qubits and the distance grows exponentially with the number of concatenation steps. In addition, decoding of a concatenated code is efficient in the block size n of the code, and the performance of decoding can be strongly enhanced by using message passing between concatenation layers (Poulin, 2006).

Another way of constructing quantum error-correcting codes is by using two classical binary codes in the Calderbank-Shor-Steane (CSS) construction (Nielsen and Chuang, 2000). Classical binary linear codes are fully characterized by their parity check matrix H. The parity check matrix H_1 of a code C_1 encoding k_1 bits is an $(n - k_1) \times n$ matrix with 0,1 entries where linearly independent rows represent the parity checks. The binary vectors $c \in \{0,1\}^n$ that obey the parity checks, i.e., $Hc = 0$ (where addition is modulo 2), are the code words. The distance $d = 2t + 1$ of such classical code is the minimum (Hamming) weight of any code word and the code can correct t errors.

We represent a row r of H_1 of a code C_1 by a parity check operator $s(Z)$ such that for the bit $r_i = 1$ we take $s(Z)_i = Z$ and for bit $r_i = 0$ we set $s(Z)_i = I$. These parity checks generate some stabilizer group $\mathcal{S}_1(Z)$. In order to make this into a quantum code with distance larger than 1, one needs to add X-type parity checks. These could simply be obtained from the $(n - k_2) \times n$ parity check matrix H_2 of another classical code C_2. We obtain the stabilizer parity checks $\mathcal{S}_2(X)$ by replacing the 1's in each row of this matrix by Pauli X and I otherwise. But in order for $\mathcal{S} = \langle \mathcal{S}_1(Z), \mathcal{S}_2(X) \rangle$ to be an Abelian group the checks all have to commute. This implies that every parity X check should overlap on an even number of qubits with every parity Z check. In coding words it means that the rows of H_2 have to be orthogonal to the rows of H_1. This in turn can be expressed as $C_2^{\perp} \subseteq C_1$, where $C_2^{\perp}$ is the code dual to C_2 (code words of $C_2^{\perp}$ are all the binary vectors orthogonal to all code words $c \in C_2$).

In total $\mathcal{S} = \langle \mathcal{S}_1(Z), \mathcal{S}_2(X) \rangle$ will be generated by $2n - k_1 - k_2$ independent parity checks so that the quantum code encodes $k_1 + k_2 - n$ qubits. The distance of the quantum code is the minimum of the distance $d(C_1)$ and $d(C_2)$ as one code is used to correct Z errors and the other code is used to correct X errors.

A good example of this construction is Steane's seven-qubit code [[7,1,3]] which is constructed using a classical binary code C that encodes four bits into seven bits and has a distance of 3. Its parity check matrix is

$$H = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}. \tag{3}$$

The code words c that obey $Hc = 0$ are linear combinations of the $7 - 3 = 4$ binary vectors (1,1,1,0,0,0,0), (0,0,0,1,1,1,1), (0,1,1,0,0,1,1), and (1,0,1,0,1,0,1), where the last three are the rows of the parity check matrix: these are also code words of $C^{\perp}$. Hence $C^{\perp} \subseteq C$, and we can use the CSS construction with $C_1 = C$ and $C_2 = C$ to get a quantum code. As $C^{\perp}$ (as well as C) has a distance of 3, the quantum code will have a distance of 3 and encodes one qubit. The parity checks are $Z_4Z_5Z_6Z_7$, $Z_2Z_3Z_6Z_7$, and $Z_1Z_3Z_5Z_7$ and $X_4X_5X_6X_7$, $X_2X_3X_6X_7$, and $X_1X_3X_5X_7$.

Steane's code is the smallest example in a family of two-dimensional color codes (Bombin and Martin-Delgado, 2006). Codes obtained using the CSS construction have some useful properties in terms of what logical gates can be realized easily on the encoded qubits; see Sec. II.G. Homological codes discussed in Sec. III.A.1 represent another interesting class of CSS codes.

C. Formalism of subsystem stabilizer codes

Subsystem stabilizer codes can be viewed as stabilizer codes in which some logical qubits, called gauge qubits, are not used to encode information (Poulin, 2005). The state of these extra qubits is irrelevant and is in principle left to vary. The presence of the gauge qubits sometimes lets one simplify the measurement of the stabilizer parity checks as the state of the gauge qubits is allowed to freely change under these measurements.

To define a subsystem code, one can thus takes a stabilizer code $\mathcal{S}$ and split its logical operators $(\overline{X}_i, \overline{Z}_i)$ into two groups: the gauge qubit logical operators $(\overline{X}_i, \overline{Z}_i)$ with $i = 1, \ldots, m$ and the remaining logical operators $(\overline{X}_i, \overline{Z}_i)$ with $i = m+1, \ldots, k$. Then we define a new subgroup $\mathcal{G} = \langle \mathcal{S}, \overline{X}_1, \overline{Z}_1, \ldots, \overline{X}_m, \overline{Z}_m \rangle$ which contains $\mathcal{S}$ but also the logical operators of the irrelevant gauge qubits. We note that $\mathcal{G}$ is non-Abelian as the logical $\overline{X}$ and $\overline{Z}$ operators of a gauge qubit do not mutually commute. However, all elements in the center of this group, defined as $\mathcal{G} \cap \mathcal{C}(\mathcal{G}) = \{P \in \mathcal{G} | \, \forall \, g \in \mathcal{G}, Pg = gP\} = \mathcal{S}$ (modulo trivial elements), will commute with all elements in $\mathcal{G}$.

One could do error correction by measuring the parity check operators in $\mathcal{S}$ but imagine that instead we measure the (noncommuting) generators of the group $\mathcal{G}$. As some of these operators are noncommuting, their ± 1 eigenvalues cannot simultaneously be fixed. However, by choosing the proper order to measure these noncommuting checks, we can determine the eigenvalues of the generators for $\mathcal{S}$ since $\mathcal{S} \subseteq \mathcal{G}$. For example, for a code $\mathcal{G} = \langle \mathcal{G}_1(X), \mathcal{G}_2(Z) \rangle$ where the group $\mathcal{G}_1(X)$ $[\mathcal{G}_2(Z)]$ consists only of X checks (Z checks), one can first measure all the generators of $\mathcal{G}_1(X)$ and then all the generators of $\mathcal{G}_2(Z)$. For more general gauge groups $\mathcal{G}$ which do not split up into an X and a Z part, there is a simple condition which constrains the order in which the gauge checks have to be measured [see, e.g., Suchara, Bravyi, and Terhal (2011)] in order to derive stable values for the stabilizer checks in $\mathcal{S}$. Note that the $k - m$ logical operators $(\overline{X}_i, \overline{Z}_i)$, $i = m+1, \ldots, k$, of the logical qubits in use commute with $\mathcal{G}$ and so these logical operators are unaffected by the measurement of elements in $\mathcal{G}$.

A priori there is no reason why measuring the generators of $\mathcal{G}$ would be simpler than measuring the generators of the stabilizer $\mathcal{S}$. In interesting constructions such as the Bacon-Shor code and the subsystem surface code discussed in Sec. III.C, we gain because we measure very low-weight parity checks in $\mathcal{G}$, while often we lose by allowing more qubit overhead or declining noise threshold.

The perspective of viewing a subsystem code merely as a partially used stabilizer code is useful for understanding the role of $\mathcal{G}$ vs $\mathcal{S}$. It is not in general the way one wants to construct such a code as creating $\mathcal{G}$ from an arbitrary stabilizer code $\mathcal{S}$ (generated by low-weight checks) by adding some logical operators of gauge qubits gives no guarantee that $\mathcal{G}$ is itself generated by low-weight parity checks.

When we measure the eigenvalues of the noncommuting generators of $\mathcal{G}$, the gauge check operators, we are affecting the state of the gauge qubits. Consider a subsystem code $\mathcal{G} = \langle \mathcal{G}_1(X), \mathcal{G}_2(Z) \rangle$. If we measure the gauge Z checks, we fix the state of the gauge qubits to be an eigenstate of these Z checks, and hence the gauge qubits are eigenstates of their logical $\overline{Z}$ operators. If we then measure all the gauge X checks, we project the gauge qubit states onto eigenstates of their logical $\overline{X}$ operators, thus actively changing their logical state. We can make a new stabilizer code out of a subsystem code by "fixing the gauge" as follows. In order to fix, say, an X gauge, we add all the logical $\overline{X}$ operators of the gauge qubits $\overline{X}_1, \ldots, \overline{X}_m$ to $\mathcal{S}$; we call this the stabilizer code $\mathcal{S}_{X\,\text{fix}} = \langle \mathcal{S}, \overline{X}_1, \ldots, \overline{X}_m \rangle$. For this stabilizer code, all the gauge qubits are prepared in their logical $|\overline{+}\rangle$ state. Gauge fixing is a concept that can be useful in the efficient realization of a universal set of logical gates; see Sec. II.G.

The distance of a subsystem code is not the same as that of a stabilizer code, Eq. (1), as we should consider only the minimum weight of the genuine $k - m$ logical operators. These logical operators are not unique as they can be multiplied by elements in $\mathcal{S}$ but also by the logical operators of the irrelevant gauge qubits (which change the state of the gauge qubits). This motivates the definition of the distance as $d = \min_{P \in \mathcal{C}(\mathcal{S}) \setminus \mathcal{G}} |P|$. We can further distinguish logical operators as either so-called bare or dressed logical operators. Bare logical operators do not change the state of the gauge qubits: they commute with all elements in $\mathcal{G}$; in other words, they are contained in $\mathcal{C}(\mathcal{G})$. Dressed logical operators can be obtained from bare logical operators by multiplication with elements in $\mathcal{G}$, in particular, by multiplication with the gauge qubit logical operators which are in $\mathcal{G}$ but not in $\mathcal{S}$.

It is the distance and properties of the dressed logical operators that define the qualities of the code. For example, one can easily construct a "Heisenberg" subsystem code of n qubits on a line with n odd. Let $\mathcal{G} = \langle X_1 X_2, Z_1 Z_2, \ldots, X_{n-1} X_n, Z_{n-1} Z_n \rangle$. The operators $\overline{X} = X^{\otimes n}$ and $\overline{Z}^{\otimes n}$ mutually anticommute and commute with all elements in $\mathcal{G}$ but are not elements of $\mathcal{G}$ as they are of odd weight. Hence they are bare logical operators of weight n, but multiplying these operators by elements in $\mathcal{G}$ will result in dressed logical operators which have weight 1, and hence a low-distance code.

As errors on the gauge qubits are harmless, it means that equivalent classes of errors are those related to each other by elements in $\mathcal{G}$ (instead of $\mathcal{S}$ for stabilizer codes). Given the eigenvalues of the stabilizer generators, the syndrome $\mathbf{s}$, the decoding algorithm considers equivalence classes defined as $[E] = \{E' | \exists g \in \mathcal{G}, gE' = E\}$. Maximum-likelihood decoding (or minimum-weight decoding) can proceed similarly as for stabilizer codes: one determines which class $[E\overline{P}]$ has a maximum value for $\text{Prob}([E\overline{P}]) = \sum_{g \in \mathcal{G}} \text{Prob}(E\overline{P}g)$, where $\overline{P}$ varies over the possible logical operators.

D. Quantum error-correction conditions and other small codes of physical interest

One may ask what properties a general (not necessarily stabilizer) quantum code, defined as some subspace C of a physical state space, should have in order for a certain set of errors to be correctable. These properties are expressed as the quantum error-correction (QEC) conditions which can hold exactly or only approximately.

We encode some k qubits into a code space C which is a subspace of an n-qubit space such that $|\overline{x}\rangle$ are the code words encoding the k-bit strings x. Assume there is a set of errors $\mathcal{E} = \{E_i\}_{i=1}^{I}$ that we want to correct. We can capture the action of these errors by a superoperator $\mathcal{S}(\rho) = \sum_{i=1}^{I} E_i \rho E_i^\dagger$ which is not necessarily trace preserving. We are seeking a trace-preserving reversal superoperator $\mathcal{R}$ such that $\mathcal{R} \cdot \mathcal{S}(\overline{\rho}) \propto \overline{\rho}$ for any encoded density matrix $\overline{\rho}$ (which is supported only on the code space). The quantum error-correction conditions (Bennett *et al.*, 1996; Knill and Laflamme, 1997) say that there exists such an error-correcting reversal operation $\mathcal{R}$ if and only if the following conditions are obeyed for all errors $E_i, E_j \in \mathcal{E}$:

$$\forall\ x, x', \quad \langle \overline{x} | E_i^\dagger E_j | \overline{x'} \rangle = c_{ij} \delta_{xx'}. \tag{4}$$

Here c_{ij} is a constant *independent* of the code word $|\overline{x}\rangle$ with $c_{ij} = c_{ji}^*$. The condition for $x = x'$ informally requires that the code words are not distinguished by the error observables. The condition for $x \neq x'$ indicates that the orthogonal code words need to remain orthogonal after the action of the errors (otherwise we could not undo the effect of the errors). One can understand these conditions as arriving from the requirement that, in order for a reversal operation $\mathcal{R}$ to exist, no quantum information should leak to the environment. These conditions are derived in Nielsen and Chuang (2000) directly by demanding that $\mathcal{R} \cdot \mathcal{S}(\overline{\rho}) \propto \overline{\rho}$.

If a code can correct the error set $\{E_i\}$, it can also correct an error set $\{F_j\}$, where each F_j is any linear combination of the elements E_i, as one can verify that the set $\{F_j\}$ will also obey the quantum error-correction conditions in Eq. (4). This means that if a code can correct against Pauli errors on any subset of t qubits, it can correct against any error on t qubits, as the Pauli matrices form an operator basis in which one can expand the errors. Stabilizer codes are generally designed such that the code has distance $d = 2t + 1$, which implies that it can correct any Pauli error on any subset of t qubits (and thus any other error on these subsets as well).

These QEC conditions can be generalized to the unified framework of operator quantum error correction (Kribs, Laflamme, and Poulin, 2005; Nielsen and Poulin, 2007) which covers both subsystem codes and error-avoidance techniques via the use of decoherence-free subspaces and noise-free subsystems. Another generalization of the stabilizer framework is the formalism of code-word-stabilized quantum codes (Cross *et al.*, 2009) which also includes nonstabilizer codes.

1. Physical error models

How do we determine the set of error operators $\{E_i\}$ for a given set of qubits? In principle, one could start with a Hamiltonian description of the dynamics of the qubits, the system S, coupled to a physically relevant part of the rest of the world, which we call the environment E.

One has a Hamiltonian $H(t) = H_S(t) + H_{SE}(t) + H_E(t)$, where $H_S(t)$ $[H_E(t)]$ acts on S (E) and $H_{SE}(t)$ is the coupling term. We assume that the qubits of the system and environment are initially $(t = 0)$ in some product state $\rho_S \otimes \rho_E$ and then evolve together for time τ. The dynamics due to the $U(0, \tau) = \mathcal{T} \exp[-i \int_0^\tau dt' H(t')]$ for the system alone can then be described by the superoperator $\mathcal{S}_\tau$:

$$\mathcal{S}_\tau(\rho_S) = \mathrm{Tr}_E U(0, \tau) \rho_S \otimes \rho_E U^\dagger(0, \tau) = \sum_i E_i \rho_S E_i^\dagger,$$

where $\{E_i\}$ with $\sum_i E_i^\dagger E_i = I$ are the so-called Kraus operators determining the action of the superoperator. These Kraus operators $\{E_i\}$ are thus the error operators. The Kraus operators of a given superoperator are not unique. One can define a different set of error operators $F_j = \sum_i U_{ji} E_i$ with a unitary matrix U which realizes the same superoperator [see, e.g., Nielsen and Chuang (2000)], but as noted before if the set $\{E_i\}$ is correctable then the set $\{F_j\}$ is correctable as well.

This derivation of the error operators is appropriate when the system-environment interaction is memoryless or Markovian beyond a time scale τ so that it is warranted that we start the evolution with an initial product state between system and environment. For general non-Markovian noise such a description is not directly appropriate. Instead of finding a map describing the dynamics of the system by itself, one can always consider the unitary dynamics of the system and environment together and expand this in terms of errors. One writes the joint unitary transformation between times t_1 and t_2 as $U(t_1, t_2) = U_{\mathrm{ideal}}(t_1, t_2) + E_{SE}$ with $U_{\mathrm{ideal}}(t_1, t_2)$ as the ideal faultless evolution. The operator E_{SE} can always be expanded as $E_{SE} = \sum_i E_i \otimes B_i$, where $\{E_i\}$ can be identified as a set of error operators on the system. See Lidar and Brun (2013) for a more extensive treatment of non-Markovian noise models.

Quite commonly one can describe the open-system dynamics by a Markovian master equation of Lindblad form,

$$\frac{d\rho}{dt} = -i[H(t), \rho] + \mathcal{L}(\rho) \equiv \mathcal{L}_{\mathrm{tot}}(\rho), \tag{5}$$

where $\mathcal{L}(\rho) = \sum_j L_j \rho L_j^\dagger - (1/2)\{L_j^\dagger L_j, \rho\}$ with quantum jump or Lindblad operators L_j.[7] Here $H(t)$ is the Hamiltonian of the quantum system which could include some time-dependent driving terms. For short times τ we have $\rho(\tau) = \mathcal{S}_\tau(\rho(0)) = E_0 \rho E_0^\dagger + \sum_i E_i \rho E_i^\dagger$ with $E_0 \approx I - i\tau H - (1/2)\tau \sum_i L_i^\dagger L_i = I - O(\tau)$ and $E_i \approx \sqrt{\tau} L_i$. Thus the error set is given by the quantum-jump operators L_i and the no-error operator E_0 which is nontrivial at order $O(\tau)$.

A special simple case of such a Lindblad equation leads to the Bloch equation which is used to describe qubit decoherence at a phenomenological level. We consider a qubit, described by a Hamiltonian $H = -(\omega/2)Z$, which exchanges energy with a large Markovian environment in thermal equilibrium at temperature $\beta = 1/kT$. One can model such open-system dynamics using a master equation of Lindblad form with quantum-jump operators with $L_- = \sqrt{\kappa_-}\sigma_-$ and $L_+ = \sqrt{\kappa_+}\sigma_+$ where $\sigma_- = |0\rangle\langle 1|$ and $\sigma_+ = |1\rangle\langle 0|$. Here the rates κ_+, κ_- obey a detailed balance condition $\kappa_+/\kappa_- = \exp(-\beta\omega)$. The resulting Lindblad equation has the thermal state $\rho_\beta = \exp(-\beta H)/\mathrm{Tr}[\exp(-\beta H)]$ as

[7] Using the definition of the anticommutator $\{A, B\} = AB + BA$.

its unique stationary point for which $\mathcal{L}_{\text{tot}}(\rho_\beta) = 0$. We can include additional physical sources of qubit dephasing modeled by the quantum-jump operator $L_Z = \sqrt{\gamma_Z} Z$ in the Lindblad equation; this, of course, does not alter its stationary point.

We can parametrize a qubit as $\rho = (1/2)(I + \mathbf{r} \cdot \sigma)$ with Bloch vector $\mathbf{r}$ and Pauli matrices $\sigma = (X, Y, Z)$ and reexpress this Lindblad equation as a differential equation for $\mathbf{r}$, the Bloch equation. Aside from the process of thermal equilibration and dephasing, one can add time-dependent driving fields in the Bloch equation (which are assumed not to alter the equilibration process) so that the general Hamiltonian is $H(t) = (1/2)\mathbf{M}(t) \cdot \sigma$.

The Bloch equation then reads

$$\frac{d\mathbf{r}}{dt} = \mathbf{r}(t) \times \mathbf{M}(t) + \mathbf{R}\big(\mathbf{r}(t) - \mathbf{r}_\beta\big), \tag{6}$$

where the first (second) part describes the coherent (dissipative) dynamics. Here the equilibrium Bloch vector $\mathbf{r}_\beta = \big(0, 0, \tanh(\beta\omega/2)\big)$ and the diagonal relaxation matrix $\mathbf{R} = \text{diag}(-1/T_2, -1/T_2, -1/T_1)$, where the decoherence time T_2 and relaxation time T_1 characterize the basic quality of the qubit.

We now consider two simple codes which approximately obey the conditions in Eq. (4). The first code protects against amplitude damping, which models T_1 relaxation for qubits. In the second code, the cat code, one encodes a qubit into a bosonic mode so as to be partially protected against photon loss. We continue in Sec. II.D.3 with another stabilizer code that encodes a qubit in a bosonic mode, which demonstrates that one can also apply the stabilizer formalism to phase space.

2. Amplitude-damping code

Even though the [[5,1,3]] code is the smallest code that can correct against any single-qubit error, one can use four qubits to approximately correct any *amplitude-damping* error which can model energy loss (Leung *et al.*, 1997). The noise process for amplitude damping on a single qubit is given by the superoperator $\mathcal{S}(\rho) = \sum_i A_i \rho A_i^\dagger$ with

$$A_0 = \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{1-\kappa} \end{pmatrix} \approx I - O(\kappa)$$

and $A_1 = \sqrt{\kappa}\sigma_-$. The code words for the four-qubit amplitude-damping code are $|\overline{0}\rangle = (1/\sqrt{2})(|0000\rangle + |1111\rangle)$ and $|\overline{1}\rangle = (1/\sqrt{2})(|0011\rangle + |1100\rangle)$.

We assume that each qubit in this code is subjected to amplitude-damping noise. We want to approximately correct against the error set $E_0 = A_0^{\otimes 4}$, $E_1 = A_1 \otimes A_0^{\otimes 3}$, $E_2 = A_0 \otimes A_1 \otimes A_0^{\otimes 2}$, $E_3 = A_0^{\otimes 2} \otimes A_1 \otimes A_0$, and $E_4 = A_0^{\otimes 3} \otimes A_1$, corresponding to no damping and single-qubit damping on any of the four qubits, respectively. Leung *et al.* (1997) showed that this code obeys the QEC conditions approximately, with $O(\kappa^2)$ corrections, which is a quadratic improvement over the basic error rate κ.

Clearly, when one uses an approximate error-correction code, one can only approximately undo the errors. Determining an optimal recovery (defined as optimizing a worst-case or average-case fidelity) is more involved; see the most recent results on this code and the general approach by Bény and Oreshkov (2010) and Ng and Mandayam (2010).

3. Qubit-into-oscillator codes

Another interesting example is that of a single bosonic mode (with creation and annihilation operators $a^\dagger$ and a) that is used to encode a qubit in two orthogonal states which are approximately protected against photon loss. The damping process can be modeled with the Lindblad equation, Eq. (5), with $L = \sqrt{\kappa} a$, while $H = \omega(a^\dagger a + 1/2)$ (which we can transform away by going to the rotating frame at frequency ω). One can choose two Schrödinger cat states as encoded states $|\overline{0}_+\rangle$ and $|\overline{1}_+\rangle$ with

$$\begin{aligned} |\overline{0}_\pm\rangle &= \frac{1}{\sqrt{N_\pm}}(|\alpha\rangle \pm |-\alpha\rangle), \\ |\overline{1}_\pm\rangle &= \frac{1}{\sqrt{N_\pm}}(|i\alpha\rangle \pm |-i\alpha\rangle). \end{aligned} \tag{7}$$

Here $|\alpha\rangle$ is a coherent state

$$|\alpha\rangle = \exp(-|\alpha|^2/2) \sum_{n=0}^{\infty} \frac{\alpha^n}{\sqrt{n!}} |n\rangle,$$

and $N_\pm = 2[1 \pm \exp(-2|\alpha|^2)] \approx 2$. For sufficiently large photon number $\langle n \rangle = |\alpha|^2$, the states $|\pm\alpha\rangle$ and $|\pm i\alpha\rangle$, and therefore $|\overline{0}_+\rangle$ and $|\overline{1}_+\rangle$, are approximately orthogonal [as $|\langle\alpha|\beta\rangle|^2 = \exp(-|\alpha - \beta|^2)$].

The creation and manipulation of cat states has been actively explored; see an extensive discussion on cavity-mode cat states in microwave cavities (Haroche and Raimond, 2006). The code states are chosen such that loss of a photon from the cavity maps the states onto (approximately) orthogonal states. As $a|\alpha\rangle = \alpha|\alpha\rangle$, we have

$$a|\overline{0}_+\rangle = \alpha\sqrt{N_-/N_+}|\overline{0}_-\rangle, \qquad a|\overline{1}_+\rangle = i\alpha\sqrt{N_-/N_+}|\overline{1}_-\rangle, \tag{8}$$

with $|\overline{0}_-\rangle$ and $|\overline{1}_-\rangle$ defined in Eq. (7). The preservation of orthogonality is a prerequisite for these code states to be correctable. More precisely, one can verify that in the unphysical limit $|\alpha| \to \infty$ one obeys the QEC conditions,[8] Eq. (4), for $E_0 = \sqrt{\kappa} a$ and $E_1 = I - (\kappa/2)a^\dagger a$.

The code space (spanned by $|\overline{0}_+\rangle$ and $|\overline{1}_+\rangle$) is distinguished from the orthogonal erred space (spanned by $|\overline{0}_-\rangle$ and $|\overline{1}_-\rangle$) by the photon parity operator $\exp(i\pi a^\dagger a) = \sum_n (-1)^n |n\rangle\langle n| = P_{\text{even}} - P_{\text{odd}}$. This parity operator has $+1$ eigenvalue for the even-photon-number states $|\overline{0}_+\rangle$, $|\overline{1}_+\rangle$, and -1 eigenvalue for the odd-photon-number states $|\overline{0}_-\rangle$, $|\overline{1}_-\rangle$. By continuously monitoring the value of the parity operator one could track the occurrence of errors (Haroche, Brune, and Raimond, 2007; Sun *et al.*, 2014). Even better would be the realization of a restoring operation which puts an erred state

[8] If we use two coherent states as code states, say, $|\overline{0}\rangle = |\alpha\rangle$ and $|\overline{1}\rangle = |-\alpha\rangle$, the QEC conditions would not be obeyed, as $\langle\alpha|E_1^\dagger E_0|\alpha\rangle \neq \langle -\alpha|E_1^\dagger E_0|-\alpha\rangle$ for any α.

with decayed amplitude $\alpha e^{-\kappa t/2}$ back into the code space while restoring the amplitude back to α. However, such a restorative process will always add noise to the code words as it is physically impossible to increase the distinguishability between (decayed) nonorthogonal code words. Thus, starting with cat states with finite α, after repeated cycles of errors followed by, we assume, perfect error detection and correction, the cat states will gradually lose their intensity and thus their approximate protection. Leghtas *et al.* (2013) and Mirrahimi *et al.* (2013) proposed the interaction of superconducting qubits coupled to 2D or 3D microwave cavities (circuit QED) to be used for encoding, correction, and decoding of such cat states while Sun *et al.* (2014) showed how this was done in an experiment.

One can generalize the stabilizer formalism to continuous-variable systems characterized by an infinite-dimensional Hilbert space (Braunstein, 1998; Lloyd and Slotine, 1998). Of particular interest are codes which encode a discrete amount of information, a qubit say, in a harmonic oscillator (Gottesman, Kitaev, and Preskill, 2001). Harrington (2004) constructed more general "symplectic" codes which encode multiple qubits in multiple oscillators.

Given are two conjugate (dimensionless) variables $\hat{p}$ and $\hat{q}$ that represent a generalized momentum and position, obeying $[\hat{q}, \hat{p}] = i$. The idea is to encode the information such that small shifts in position or momentum correspond to correctable errors while logical operators are represented as large shifts. For a harmonic oscillator space, the Pauli group P_n can be generalized to the Weyl-Heisenberg group generated by the unitary shift operators $\exp(it\hat{p})$ and $\exp(is\hat{q})$ for real s and t. These operators form a basis for the space of operators and thus any error E (any operator) can be written as $E = \int ds \int dt c(s,t) e^{it\hat{p}} e^{is\hat{q}}$ with complex coefficients $c(s,t)$. Small shifts in p and q may not *a priori* seem like a very natural noise model, but one can show that generic errors of low rate and low degree in $\hat{p}$ and $\hat{q}$ can be expanded into linear combinations of products of these shifts. One should compare this to the similar expansion of any low-weight error in terms of low-weight Pauli errors.

In order to define a qubit in this infinite-dimensional space we select a set of commuting check generators whose $+1$ eigenvalue space is two dimensional. We observe that the operators $\exp(it\hat{p})$ and $\exp(is\hat{q})$ commute if and only if $st = 0 \bmod 2\pi$: this follows from the fact that $e^A e^B = e^{[A,B]} e^B e^A$ when A and B are linear combinations of $\hat{q}$ and $\hat{p}$. We consider two examples.

In our first trivial example the code space is a single state. We choose $S_q = e^{2i\hat{q}}$ and $S_p = e^{-i\pi\hat{p}}$ as commuting check operators and we seek the states that have eigenvalue $+1$ with respect to the S_p check operator. When $S_p = 1$ the eigenvalues of $\hat{p}$ are even integers. We can define $\hat{n} = \hat{p}/2$ and $\hat{\phi} = 2\hat{q}$ so that for $S_p = 1$ we have $\hat{n} = 0, \pm 1, \ldots$.

The eigenvalue of the commuting operator S_q can be simultaneously fixed to be $e^{i\phi}$ so that we should identify $\phi = \phi \bmod 2\pi$. Thus we have the state space of a quantum rotor which is described by conjugate variables $\hat{n}$ taking integer values and a 2π-periodic phase $\hat{\phi}$ with $[\hat{\phi}, \hat{n}] = i$.

A physical realization of these degrees of freedom is the quantization of a superconducting circuit, where ϕ is the superconducting phase (difference of phase across a Josephson junction) and $\hat{n}$ represents the number of Cooper pairs (difference in the number of Cooper pairs across a Josephson junction). Fixing the eigenvalues for both S_p and S_q leads to a single state characterized by its superconducting phase $\phi \bmod 2\pi$. Small shifts $\exp(i\epsilon\hat{\phi})$ for small ϵ do not commute with S_p and gets one out of the "Cooper pair" code space fixed by $S_p = 1$. This in some sense represents the phase stability of the superconducting state at a purely mathematical level.

If we want to use this state space to represent a qubit, we have to use (linear combinations of) such states characterized by their phase. For example, the Hamiltonian of the multilevel transmon qubit (Koch *et al.*, 2007) is given by $H_{\text{transmon}} = 4E_C\hat{n}^2 - E_J \cos(\phi)$, where E_C (E_J) is the capacitive (inductive) energy. This Hamiltonian has been interpreted as that of a charged quantum rotor in a magnetic field by Koch *et al.* (2007). The lowest two energy levels of the system can define a qubit, the transmon qubit. If we expand $\cos(\phi) \approx 1 - \phi^2/2 + \phi^4/4!$, we obtain the Hamiltonian of an anharmonic (Duffing) oscillator with eigenstates which are superpositions of ϕ eigenstates. This type of qubit has thus no intrinsic protection against dephasing, i.e., the value of the energy-level splitting is affected by charge and flux noise (represented as linear combinations of small shifts in $\hat{p}$ and $\hat{q}$).

A different choice of S_q and S_p leads to a real code that encodes a single qubit and has built-in protection. We choose as checks the operators $S_q = e^{2i\hat{q}}$ and $S_p = e^{-2i\pi\hat{p}}$. Fixing the eigenvalues of these operators to be $+1$ leads to the discretization $\hat{p} = 0, \pm 1, \pm 2, \ldots$ and again $\hat{q}$ should have eigenvalues that are multiples of π. Now there are two operators which commute with S_q and S_p but which mutually anticommute: these are $\overline{Z} = e^{i\hat{q}}$ and $\overline{X} = e^{-i\pi\hat{p}}$.

The state $|\overline{0}\rangle$ (defined by $\overline{Z}|\overline{0}\rangle = |\overline{0}\rangle$ and $S_p|\overline{0}\rangle = |\overline{0}\rangle$) is a uniform superposition of states with $\hat{q} = 0, \pm 2\pi, \ldots$. Similarly, $|\overline{1}\rangle$ corresponds to a uniform superposition of $\hat{q} = \pm\pi, \pm 3\pi, \ldots$; see Fig. 3 with $\alpha = \pi$. Consider the effect of shifts of the form $e^{i\delta\hat{p}}$ where $|\delta| < \pi/2$, which are correctable. Such shifts map the code words outside the code space as they do not commute with the stabilizer operator S_q. Error correction thus takes place by measuring $q \bmod \pi$ and applying the smallest shift which resets $q = 0 \bmod \pi$. Similarly $|\overline{+}\rangle$ is a uniform superposition of states with $\hat{p} = 0, \pm 2, \pm 4, \ldots$, while $|\overline{-}\rangle$ is a uniform superposition of

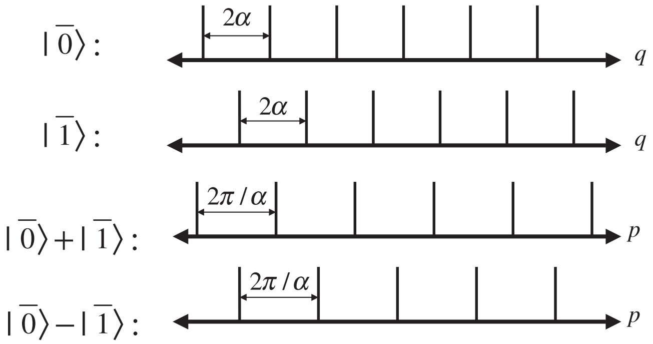

FIG. 3. Amplitude of code words for the stabilizer code with commuting checks $S_q(\alpha) = e^{2i\pi\hat{q}/\alpha}$ and $S_p(\alpha) = e^{-2i\hat{p}\alpha}$ which encodes a qubit in an oscillator. From Gottesman, Kitaev, and Preskill, 2001.

states with $\hat{p} = \pm 1, \pm 3, \ldots$; see Fig. 3. The qubit is protected against shifts $e^{i\epsilon\hat{q}}$ with $|\epsilon| < 1/2$.

This code space can be viewed as the state space of a Majorana fermion qubit (Alicea, 2012), where $\hat{p} = \hat{n}$ counts the total number of electrons while $\hat{q} = \hat{\phi}$ is the π-periodic conjugate phase variable. The $|\overline{+}\rangle$ eigenstate of $\overline{X}$ with an even number of electrons corresponds to the Majorana mode unoccupied while $|\overline{-}\rangle$ is the state with an odd number of electrons as the Majorana mode is occupied. The protection of the Majorana fermion qubit can thus also be understood from this coding perspective although the perspective sheds no light on how to physically realize this qubit nor on the effect of noise which cannot be represented by shifts.

Another representation of this code space, which does not use Majorana fermion qubits but superconducting circuits, is the 0-π qubit (Kitaev, 2006) which is designed such that the superconducting phase difference between terminals has degenerate energy minima at 0 and π corresponding to the approximate code words $|\overline{0}\rangle$ and $|\overline{1}\rangle$.

More generally, we can parametrize this code by a real number α by taking the stabilizer checks as $S_q = e^{2i\pi\hat{q}/\alpha}$ and $S_p = e^{-2i\hat{p}\alpha}$ (previously we took $\alpha = \pi$). The logical operators are $\overline{Z} = e^{\pi i\hat{q}/\alpha}$ and $\overline{X} = e^{-i\hat{p}\alpha}$ (Gottesman, Kitaev, and Preskill, 2001); see the code words in Fig. 3. The code can correct against shifts $e^{i\epsilon\hat{q}}$ with $|\epsilon| < \pi/2\alpha$ and $e^{-i\delta\hat{p}}$, where $|\delta| < \alpha/2$.

One can use this code for encoding a qubit in a bosonic mode where $\hat{q}$ and $\hat{p}$ arise as quadrature variables, i.e., $\hat{q} = (1/\sqrt{2})(a^\dagger + a)$ and $\hat{p} = (i/\sqrt{2})(a^\dagger - a)$. The free Hamiltonian $H_0 = \omega(a^\dagger a + 1/2)$ will periodically transform $\hat{q}$ into $\hat{p}$ and vice versa, so it is natural to let S_q be of the same form as S_p and choose $\alpha = \sqrt{\pi}$.

Gottesman, Kitaev, and Preskill (2001) showed explicitly how errors such as photon loss $L_- = \sqrt{\kappa_-}a$, photon gain $L_+ = \sqrt{\kappa_+}a^\dagger$, dephasing (or decay) of the oscillator $e^{i\theta a^\dagger a}$ (or $e^{-\kappa a^\dagger a}$), or a nonlinearity $e^{iK(a^\dagger a)^2}$ for sufficiently small parameters $\kappa_\pm$, θ, and K can be expanded into the small shift operators and can thus be corrected. The level of protection thus goes well beyond that of the cat state code.

However, the code words of this code in Fig. 3 are not physically achievable as it requires an infinite amount of squeezing (and thus an infinite amount of energy) to prepare (superpositions of) quadrature eigenstates such as $|q\rangle$ or $|p\rangle$. Gottesman, Kitaev, and Preskill (2001) proposed using approximate code words, e.g., the approximate code word $|\tilde{0}\rangle$ is a superposition of Gaussian peaks in q space, each one centered at integer multiples of $2\sqrt{\pi}$ with width Δ, in a total Gaussian envelope of width $1/\kappa$. Viewed as a superposition of p eigenstates, such a state is a superposition of peaks with width κ and total envelope of width Δ^{-1}. An error analysis of this approximate encoding was done by Glancy and Knill (2006), while Vasconcelos, Sanz, and Glancy (2010) considered the preparation of the encoded states using cat states as in Eq. (7), squeezing, and homodyne detection.

Menicucci (2014) showed how one can use a continuous-variable cluster state and homodyne measurements to perform quantum error correction on these approximate GKP (Gottesman, Kitaev, Preskill) code words and realize a universal set of gates assuming that the noise is due only to the finite amount of squeezing in the preparation of the GKP code words and the cluster state. For squeezing levels of 21 dB, Menicucci estimates that the (worst-case) effective gate error rate is 10^{-6}, sufficiently below the noise threshold of the surface code discussed in Sec. III.A. In Sec. III.C.2 we consider a version of the surface or toric code which encodes an oscillator in a 2D coupled array of harmonic oscillators, which can also be viewed as a way to concatenate the GKP code with the surface code.

E. *D*-dimensional (stabilizer) codes

Of particular practical interest are D-dimensional stabilizer codes. These are stabilizer code families on qubits located at vertices of some D-dimensional cubic lattice (with or without periodic boundary conditions). The parity checks involve $O(1)$ qubits which are within $O(1)$ distance of each other on this lattice, where $O(1)$ means that this quantity is a constant independent of block size n. One can easily prove that one-dimensional stabilizer codes have distance $O(1)$, independent of block size (Bravyi and Terhal, 2009), showing that without concatenation, such codes offer little fault-tolerant protection. Various two-dimensional topological stabilizer codes are discussed in Sec. III, while some 3D and 4D examples of topological codes are the Haah code (Haah, 2011), the Chamon code (Bravyi, Leemhuis, and Terhal, 2011), the 3D toric code (Castelnovo and Chamon, 2008), and the 4D toric code (Dennis *et al.*, 2002), discussed in Sec. III.A.1.

There are of course many codes which are not captured by the stabilizer formalism. Here I briefly mention the class of 2D topological qubit codes where the stabilizer checks are still commuting, but are no longer simple Pauli operators. As Hamiltonians these correspond to the so-called 2D Levin-Wen models (Levin and Wen, 2005); as codes they are called Turaev-Viro codes (Koenig, Kuperberg, and Reichardt, 2010). The advantage of these codes which generalize the 2D surface code in Sec. III is that universal quantum computation can be achieved by purely topological means. The disadvantage from the coding perspective is that (1) the stabilizer checks are more complicated as operators, e.g., for the so-called Fibonacci code on a hexagonal lattice, the stabilizer checks act on three and 12 qubits, and (2) decoding and determining a noise threshold for these codes has only recently begun (Brell *et al.*, 2014; Wootton *et al.*, 2014).

F. Error correction and fault tolerance

We understand from the previous discussions that the crucial element of quantum error correction for stabilizer codes is the realization of the (parity) check measurement as in Fig. 1. The immediate problem is that the parity check measurement suffers from the same imperfections and noise as any other gate or measurement that one may want to do.

In practice a parity check measurement may arise as a continuous weak measurement, leaving a classical stochastic data record which hovers around the value $+1$ (pointing to the state being in the code space) while occasionally *jumping* to a value centered around -1, modeled using a stochastic master equation. One can imagine that such a continuously acquired record is immediately fed back to unitarily steer the qubits to the code space (Ahn, Doherty, and Landahl, 2002). The feedback cannot just rely on the instantaneously measured noisy signal but should integrate over a longer measurement record to estimate the current conditional quantum state of the system (Wiseman and Milburn, 2010). However, tracking the entire quantum state in real time is computationally expensive and defeats the purpose of quantum computation. For the realization of quantum error correction, van Handel and Mabuchi (2005) described a filter in which one tracks only the probability that the quantum system at time t is in a state with particular error syndrome $\mathbf{s}$ given the continuous measurement record in time. Chase, Landahl, and Geremia (2008) improved on this construction by explicitly including the effect of the feedback Hamiltonian in the stochastic analysis.

Another model of feedback is one in which no weak measurements are performed and processed, but rather the whole control loop is a dissipative quantum computation. One could set up a simple local error-correction mechanism by explicitly engineering a dissipative dynamics which drives or corrects the qubits toward the code space as proposed by Barreiro *et al.* (2011) and Müller *et al.* (2011). We assume that the open-system dynamics of code qubits and environment is described by a Lindblad equation as in Eq. (5). For simplicity, we consider the case in which we want to pump or drive four qubits into a state with even parity so that the four-qubit parity Z check $Z_1Z_2Z_3Z_4$ has eigenvalue $+1$. Imagine that we can engineer the dissipation (in the interaction picture) such that there is a single quantum-jump operator $L = \sqrt{\kappa}X_1P_{\text{odd}}$ with $P_{\text{odd}} = (1/2)(I - Z_1Z_2Z_3Z_4)$, the projector onto the odd-parity space, and $H \propto Z_1Z_2Z_3Z_4$. Integration of the Lindblad equation gives rise to the time dynamics $\rho(t) = \exp(t\mathcal{L}_{\text{tot}})[\rho(t=0)]$ with stationary states ρ determined by $\mathcal{L}_{\text{tot}}(\rho) = 0$. States supported on the even-parity subspace are “dark” states with $\mathcal{L}(\rho) = 0$ and $[H, \rho] = 0$. The odd-parity subspace is not stationary as the quantum jump operator L flips the first qubit so that an odd-parity state becomes an even-parity state pumping the system toward the stationary dark subspace.

Müller *et al.* (2011) considered the following stroboscopic evolution using an ancillary dissipative qubit or mode which approximately gives rise to such Lindblad equation. The idea is to alternate (or *trotterize*) the coherent evolution with H and the dissipative evolution with $\mathcal{L}$ for short periods of time τ so that $\exp(\tau\mathcal{L}_{\text{tot}}) \approx \exp(-i\tau[H, \cdot])\exp(\tau\mathcal{L})$. The dynamics of H can be obtained by a small modification of the parity check measurement circuits in Fig. 1: for the evolution $\exp(-i\theta P)$, where P is a multiqubit Pauli operator, we can use the circuit in Fig. 1(b).

The dissipative evolution $\mathcal{L}$ could be implemented for short times $\tau \ll 1$ using a circuit consisting of a dissipative ancilla coupled to the four qubits as in Fig. 1(d). At the end of this circuit, instead of immediately measuring the ancilla qubit, we apply a CNOT operation with the ancilla qubit as control and qubit 1 as target (to change the parity of the odd states to even). This is then followed by natural dissipation of the ancilla qubit (T_1 process) so any amplitude in the $|1\rangle$ state is transferred to $|0\rangle$. This means that the ancilla qubit is effectively reset and can be used for the next round of application of $\exp(\tau\mathcal{L})$.

These ideas of stabilizer pumping were experimentally tested on two and four ion-trap qubits by Barreiro *et al.* (2011). The use of this kind of extremely local feedback is limited as the dissipative evolution applies a correction depending only on the outcome of a single parity check, whereas classical decoding in general makes decisions on the outcomes of many parity checks. We continue the discussion on local decoders for the surface code in Sec. III.D.

As any realization, closed or open loop, of quantum error correction will suffer from inaccuracies there is no guarantee that one will improve coherence times by encoding a qubit in a code as it may introduce more errors than it takes away. And if coding leads to a lower logical error rate, then how does one proceed to get an even lower logical error rate? In topological code families such as the surface code in Sec. III, the logical error rate decreases exponentially as some function of the block size n, once one is below a critical error rate. This implies that the more qubit overhead one is willing to tolerate the smaller the logical error rate will be. Another way of obtaining a decreasing logical error rate is through recursively applied code concatenation of codes of a fixed block size n. The main ideas of this mathematical theory of quantum fault-tolerant computation by means of code concatenation are the following.

For simplicity, we assume that every elementary gate, idling step, or measurement (these are called *locations* in the circuit) can fail independently with some error probability p (independent stochastic noise). In a concatenation scheme every qubit and operation in a quantum circuit is replaced by an encoded qubit and an encoded operation, respectively, and the process is recursively repeated. The encoded operation consists of an error-correction step and a fault-tolerant realization of the operation (see Fig. 4), which together constitute a rectangle.

For a code such as Steane's [[7,1,3]] code, which can correct a single error, the fault tolerance of the rectangle should be such that a single error in any of the locations of the rectangle cannot lead to two, *incorrectable* errors in one code

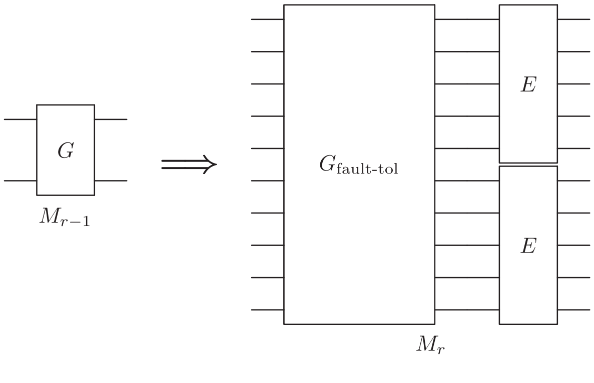

FIG. 4. Code concatenation: each qubit in the circuit on the left is replaced by an encoded block of qubits in the circuit on the right. The gate G in the circuit M_{r-1} is replaced by a rectangle consisting of the fault-tolerant encoded realization of the gate ($G_{\text{fault-tol}}$) followed by error-correcting steps (E). The process can be repeated for every elementary qubit and gate in the new circuit M_r.

block. Then if the elementary error rate scales as p, it follows that the encoded error rate scales as Cp^2 as two elementary errors are required for a logical error. Here C is a constant which roughly counts the number of pairs of locations in the rectangle where failure can lead to a logical error. If $Cp^2 < p$ the concatenation step helps and r steps of concatenation will drive down the error rate to $\sim p^{2^r}$, while the overhead in terms of qubits and gates increases only exponentially in r. The equality $Cp^2 = p$ sets the noise threshold p_c.

If we have a code with higher distance which can, say, correct t errors, then fault tolerance of a rectangle means that any error of weight $k \leq t$ in this rectangle spreads to at most k qubits in a block. This will ensure that the logical error rate is $O(p^{t+1})$.

It is not trivial to make sure that a single error in a rectangle can lead to at most one error in the block for a code such as Steane's code. Consider the parity check circuit in Fig. 1(c) which is used to measure the weight-4 X-check operators of this code, where the data qubits are some subset of the seven qubits. One needs to ensure that a single error on the ancilla qubit cannot spread to two errors in the block. However, a single X error on the ancilla between the first and the last two CNOT gates will directly spread to two X errors on the data. We can see this by commuting the Pauli X on the ancilla through the two CNOT gates (and note that the X error on the ancilla does not affect the outcome of the measurement). Thus the bare parity check measurement circuit is not fault tolerant for the Steane code and one needs to modify this. Three methods have been devised to deal with making parity check circuits fault tolerant. This first method is called Shor error correction; it replaces the ancilla qubits by a k-qubit verified cat state $(1/\sqrt{2})(|00\cdots 0\rangle + |11\cdots 1\rangle)$, where k is the weight of the check to be measured [see, e.g., Preskill (1998) for details]. The second method is Steane error correction for CSS codes. In this method the ancilla is replaced by an encoded verified ancilla $|\overline{0}\rangle$ (or $|\overline{+}\rangle$) and a logical CNOT gate is executed between the encoded data qubit and the encoded ancilla (Steane, 1997; Cross, DiVincenzo, and Terhal, 2009). A third method is Knill error correction which uses quantum teleportation into a new encoded qubit such that the logical Bell measurement outcomes reveal the error syndrome (Knill, 2005).

The idea of repeated code concatenation was used in the early days of quantum error correction to prove the threshold theorem (Aharonov and Ben-Or, 1997; Kitaev, 1997; Knill, Laflamme, and Zurek, 1998; Aliferis, Gottesman, and Preskill, 2006). This theorem says that fault-tolerant computation is possible with arbitrarily small error rate if one is willing to tolerate an overhead which scales polylogarithmically with the size N of the computation to be performed (the size of a quantum circuit is the number of locations in it).

Theorem 1: An ideal circuit of size N can be simulated with arbitrary small error δ by a noisy quantum circuit subjected to independent stochastic noise of strength less than $p < p_c$, where the noisy quantum circuit is of size $O(N(\log N)^c)$ with some constant c.

It should be noted that this theorem assumes that "fresh" ancillas can be added during the quantum computation or quantum storage for doing parity check measurements. This means that these ancillas or qubit preparations have an error rate similar to those of other elementary components in the computation. The same assumption underlies the results on the asymptotic noise threshold for topological quantum error correction. Another assumption underlying the threshold results for concatenated and topological codes is that qubits can be acted upon in parallel. In practice simultaneous readout or control of, say, multiple superconducting qubits using only a few microwave lines can be achieved by using qubits operating at sufficiently different microwave frequencies and frequency division multiplexing.

Another typical assumption is that classical processing of error information is fast and accurate, imposing no delay in the execution of the quantum computation. We return to the demands on classical processing in Secs. II.G.3 and III.D.

Practically relevant questions with respect to the threshold theorem are as follows: how high is the value of the noise threshold p_c, how large is the constant c, and what is the value of the constant in $O(\cdot)$? These numbers determine when quantum error correction will be useful and how large an overhead one should actually expect. The constant c in the theorem is roughly $c \approx \log_2 S$, where S is the number of locations in a rectangle.

The best-performing concatenated coding scheme to date is the C_4/C_6 scheme of Knill (2005). For this scheme, which assumes nonlocal interactions between qubits, Knill has numerically estimated a noise threshold as high as $p_c \approx 3\%$ albeit at the cost of huge overheads. Aliferis, Gottesman, and Preskill (2008) derived a rigorous lower bound of the noise threshold of this scheme of 0.1%.

Gottesman (2000) showed that the threshold theorem still holds if all interactions between elementary qubits are local on a one-, two-, or higher-dimensional lattice. In such a scheme nonlocal interactions between elementary qubits are assumed to be realized via chains of noisy swap gates. This result means that, even though a one-dimensional quantum error-correcting code (see Sec. II.E) has a distance $O(1)$, one can use a small 1D code and concatenate it with itself to obtain a fully fault-tolerant one-dimensional scheme. The additional noisy movement via swap gates will negatively impact the noise threshold. For example, in an entirely 2D realization of the concatenated Steane [[7,1,3]] code in which movement of data qubits via noisy swap gates is explicitly included (Svore, DiVincenzo, and Terhal, 2007), the fault-tolerant CNOT gate has $S = O(10^3)$ so that $c \approx 10$ demonstrating the potential inefficiency of code concatenation. For this scheme, the threshold was estimated as $p_c \approx 1.85 \times 10^{-5}$, while for the same nonlocal scheme the analysis resulted in a threshold of 3.61×10^{-5}. These fairly low numbers should be contrasted with the noise threshold of about 1% for the 2D surface code in Sec. III.A.

One may at first sight expect that the overhead incurred by code concatenation is worse than the overhead that is incurred with topological error correction (Sec. III). One possible reason is that in topological quantum error correction parity check measurements are simply made robust by repeating the measurement needing no additional qubits. In contrast, in code concatenation the parity check measurements are realized using more complicated ancillas as in Steane error correction. However, this picture is too simplistic: the comparative study by Suchara *et al.* (2013) showed that for a computational task such as factoring the number 1024, concatenated Bacon-Shor codes perform better than the surface code at low error rates below 1×10^{-7} while at high error rates the surface code performs better. Other studies of coding overhead for several families of codes were undertaken by Steane (2003) and Cross, DiVincenzo, and Terhal (2009). Fowler, Mariantoni *et al.* (2012) estimated that in order to factor a 2000-bit number one needs about 10^4 physical qubits per logical qubit using the double-defect encoding of the surface code described in Sec. III.B.4.

One can ask whether it is, in principle, possible to realize fault-tolerant computation with constant overhead, meaning that the number of qubits of the noisy fault-tolerant circuit scales with the number of qubits of the original circuit. This question was analyzed and answered in the affirmative by Gottesman (2014). The fault-tolerant construction by Gottesman (2014) can be based upon any family of quantum low-density parity check (LDPC) codes with constant *rate* $R = k/n \geq c$ and, loosely speaking, finite noise threshold (when the block size $n \to \infty$) even if parity check measurements are faulty.

LDPC stabilizer qubit codes are codes such that all parity checks (stabilizer generators) act on $O(1)$ qubits, independent of block size. Several codes with such properties have recently been developed (Tillich and Zémor, 2014; Freedman and Hastings, 2013; Guth and Lubotzky, 2014) which have distances $d = O(n^{\alpha})$ with $0 < \alpha \leq 0.5$. For such LDPC codes it has been shown (Kovalev and Pryadko, 2013) that having a distance scale as some function of n guarantees the existence of a finite noise threshold, assuming that we can do minimum-weight decoding. In order to be of practical interest, decoding of such LDPC codes with constant rate should be computationally efficient. However, efficient minimum-weight decoders are not known to exist for quantum LDPC codes in general. Hastings (2013) showed how one can decode a 4D hyperbolic code with an efficient local decoder running in time $O(n \log n)$ to get a logical error rate $\overline{p} \sim p^{c \log n}$ with p representing a basic error rate and c a constant, thus falling off only polynomially (instead of exponentially) with n.

It was proven by Bravyi, Poulin, and Terhal (2010) that 2D stabilizer codes (which are LDPC codes with qubits on a 2D regular lattice) obey the trade-off $kd^2 = O(n)$. This result demonstrates that 2D codes such as the surface codes discussed in Sec. III do not allow for fault-tolerant computation with constant overhead. More generally, the results by Bravyi, Poulin, and Terhal (2010) showed that any D-dimensional stabilizer code family which has distance scaling with lattice size will have a vanishing rate (when $n \to \infty$), showing that nonlocal parity checks [between $O(1)$ but distant qubits] on such lattices are necessary in order to achieve a constant overhead. We note that it is an open question whether there exist quantum LDPC stabilizer codes with constant rate and distance scaling as $n^{1/2+\beta}$ for some $\beta > 0$.

G. Universal quantum computation

In quantum error correction with stabilizer (subsystem) codes a special role is played by logical gates which are elements of the Clifford group. The Clifford group $\mathcal{C}_n$ is a finite subgroup of the group of unitary transformations $\mathcal{U}(2^n)$ on n qubits. It is defined as the normalizer of the Pauli group $\mathcal{C}_n = \{U \in \mathcal{U}(2^n) | \, \forall \, P \in \mathcal{P}_n, \exists P', UPU^{\dagger} = P'\}$, meaning that it maps Pauli operators onto Pauli operators. An overcomplete set of generators of the Clifford group are the two-qubit CNOT gate, the Hadamard H gate, the phase gate S,[9] and Pauli operators X and Z. Note that $S^2 = Z$, so that S and H and CNOT suffice to generate the whole group.

The Knill-Gottesman theorem (Gottesman, 1999b) proves that one can efficiently classically simulate any quantum circuit which employs gates only from the Clifford group. One does this by tracking the stabilizer group, or more precisely its generators, which has the input state of the quantum circuit as its unique $+1$ state. Every Clifford gate and measurement maps the stabilizer generators, which are Pauli operators, onto new stabilizer generators, providing an efficient representation of the action of the quantum circuit. Thus if a quantum circuit with Clifford gates contains additional known Pauli errors, one can easily represent these Pauli errors by additional updates of the stabilizer generators in the classical simulation.

For universal quantum computation one needs additional gates such as the T gate ($\pi/8$ rotation). Examples of universal gate sets are $\{H, T, \text{CNOT}\}$, $\{H, \text{Toffoli}\}$, and $\{H, \Lambda(S)\}$, where $\Lambda(S)$ is the two-qubit controlled-S gate.[10] Even though Clifford group gates have no quantum computational power they can be used to develop a *quantum substrate* on which to build universal computation using stabilizer codes. This comes about by combining the following sets of ideas.

First, note that stabilizer error correction by itself uses only CNOT gates, preparations of $|+\rangle$, $|-\rangle$, $|0\rangle$, $|1\rangle$, and measurements in the Z and X basis as is clear from Fig. 1. The T, $\Lambda(S)$, and Toffoli gates, each of which can be used with Clifford gates to get universality, are special unitary gates as they map Pauli errors onto elements of the Clifford group. One can define a Clifford hierarchy (Gottesman and Chuang, 1999) $\mathcal{C}(j) = \{U \in \mathcal{U}(2^n) | U\mathcal{P}_n U^{\dagger} \subseteq C(j-1)\}$ such that $\mathcal{C}(0) = \mathcal{C}(1) = \mathcal{P}_n$, $\mathcal{C}(2) = \mathcal{C}_n$. The T, $\Lambda(S)$, and Toffoli gates are thus members of $\mathcal{C}(3)$. Such gates in $\mathcal{C}(3)$ [and similarly gates in $C(j)$ for $j > 3$] can be realized with ancillas and Clifford group gates using quantum teleportation ideas (Gottesman and Chuang, 1999; Zhou, Leung, and Chuang, 2000). The idea is illustrated in Fig. 6 for the T gate.

One teleports the qubit on which the T gate has to act, prior to applying the gate, using the bottom one-bit teleportation circuit in Fig. 5. We first put a T gate at the end of that teleportation circuit so that the output is $T|\psi\rangle$. Now we modify this circuit and commute the T gate backward. In the quantum circuit we insert $I = TT^{\dagger}$ prior to the corrective Pauli X so that we can use $TXT^{\dagger} = e^{-i\pi/4}SX$.[11] Hence the correction (in case we measure $M_Z = -1$) is now the Clifford gate SX. As a last

[9]
$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}, \quad T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}.$$

[10] $\Lambda(S)|b_1, b_2\rangle = |b_1\rangle S^{b_1}|b_2\rangle$ for $b_1, b_2 = 0, 1$.

[11] Note that in the quantum circuit gates are applied from the left to the right while in equations gates are applied from the right to the left.

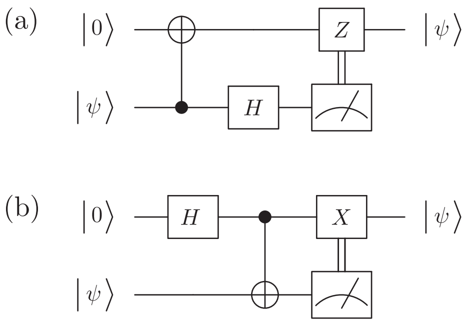

FIG. 5. The so-called one-bit teleportation circuits. The measurement denoted by the meter is a measurement in the Z basis and determines whether to do a Pauli operation on the output qubit: for outcome $M_Z = +1$ no correction is performed. From Zhou, Leung, and Chuang, 2000.

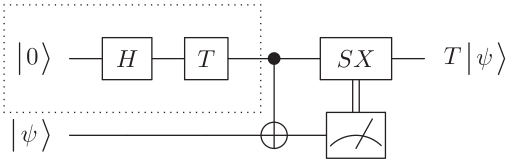

FIG. 6. Using the ancilla $T|+\rangle$ in the dashed box, one can realize the T gate by doing a corrective operation SX.

step, we note that the T gate can be commuted through the control line of the CNOT as both gates are diagonal in the Z basis on the control qubit. In this way we obtain the circuit in Fig. 6. Note that if we do not apply the correction, we obtain the state $XT^{\dagger}|\psi\rangle$.

We can use the same trick for the $S = T^2$ gate; that is, we can reduce the S gate to the preparation of a $|+i\rangle = (1/\sqrt{2})(|0\rangle + i|1\rangle)$ ancilla, a CNOT gate, and a corrective Pauli Y. We get this from starting with the bottom circuit in Fig. 5 to which we apply the S gate at the output. We insert $SS^{\dagger}$ in the quantum circuit before the corrective Pauli X and use that $SXS^{\dagger} \propto Y$. We thus need the ancilla $SH|0\rangle = (1/\sqrt{2})(|0\rangle + i|1\rangle)$.

1. Fault-tolerant logical gates

How do we realize a universal set of logical fault-tolerant gates for a code? Fault tolerance means that such logical gates do not spread errors; ideally errors of weight t remain errors of weight t. In principle, fault-tolerant gate constructions can be made for any stabilizer code (Gottesman, 1997). The question is how to do computation with minimal resource requirements and overheads, that is, as close as possible to the resources needed for a quantum memory alone. Ideally, the computation threshold, i.e., the performance of the code when used for computation, is close to the memory noise threshold, the performance of the code as a pure quantum memory.

An example of a gate which does not require additional qubits and does not spread errors is a *transversal* CNOT between two code blocks (each block encoding a single qubit into n qubits). In such transversal CNOT operations every qubit in the block is paired with a qubit in the other block in a CNOT gate such that the encoded CNOT is realized by doing n two-qubit CNOT operations in parallel. A logical CNOT gate can be performed transversally for any CSS stabilizer code with $\mathcal{S} = \langle \mathcal{S}_1(X), \mathcal{S}_2(Z)\rangle$ (Gottesman, 1997). One can understand this by observing that the product of the two stabilizer groups $\mathcal{S} \times \mathcal{S}$ of the encoded logical qubits is preserved by performing CNOT gates between all elementary qubits in the blocks. Thus the code does not change by adding these gates. Second, one can always assume that the logical $\overline{X}$ of a CSS code is only a product of Pauli X's and the logical $\overline{Z}$ is only a product of Pauli Z's. Doing CNOT gates transversally then has the same action on these logical operators as doing the CNOT on a pair of qubits.

In the CSS code construction when the classical codes $C_2 = C_1 = C$ (and thus the CSS constraint $C_2^{\perp} \subseteq C_1$ implies that $C^{\perp} \subseteq C$), the Hadamard gate H on the code block encoding a single qubit is also transversal. For this code the stabilizer has identical X and Z parts, $\mathcal{S} = \langle \mathcal{S}(X), \mathcal{S}(Z)\rangle$. The gate $H^{\otimes n}$ maps these stabilizers onto each other and similarly $H^{\otimes n}\colon \overline{X} \leftrightarrow \overline{Z}$ as these operators have the same support. An example of a code with a transversal Hadamard and CNOT gate is Steane's [[7,1,3]] code.

Eastin and Knill (2009) showed that if a quantum code can detect at least any error on a single qubit (meaning that it is a nontrivial code), then it does not have a transversally realizable universal set of gates. A somewhat weaker version of this theorem, namely, that qubit stabilizer codes do not allow for a universal set of gates to be realized via transversal unitary gates, was proved by Zeng, Cross, and Chuang (2011).

Bravyi and Koenig (2013) showed for any 2D stabilizer code that the logical gates, which can be performed by constant-depth circuits employing only local gates (between neighboring qubits), are members of the Clifford group. The reason to focus on constant-depth local circuits is that such circuits are small, naturally fault tolerant, and provide a simple extension of the idea of a transversal gate. For a constant-depth local circuit any number of errors that occurs in the circuit will affect only a patch of $O(1)$ qubits on the 2D lattice, and such $O(1)$ error patches are correctable when the code distance scales with the lattice size. Hence we expect that such constant-depth implementation of gates does not negatively impact the noise threshold or qubit overhead. The result of Bravyi and Koenig (2013) is subtle as we can realize a fault-tolerant set of universal gates for any stabilizer code, but apparently we cannot do this by composing a sequence of constant-depth encoded gates.

The results of Bravyi and Koenig (2013) also hold if we try to perform a gate by a constant-depth circuit while at the same time altering the stabilizer code to a new stabilizer code. Transforming one stabilizer code into a new one in a sequence of steps is sometimes called "code deformation." The idea is that after the entire sequence of deformations one comes back to the original code but with a logical operation applied to the encoded qubits. Code deformation is a very useful concept for topological codes. As discussed in Sec. III, one can use the code deformation technique to implement the logical H and CNOT gates in the 2D surface code. For the surface code it is not clear how one can do a logical S gate in this manner, as $S^{\otimes n}$ maps the X checks of the surface code onto Y checks. The stabilizer code with Z and Y checks is not simply related to the original stabilizer code by some code deformation, translation, or rotation. For the surface code one can do the logical S in the same fashion as the logical T; see Sec. II.G.2 [for a different logical S trick, see Aliferis (2007)]. Other stabilizer codes, 2D color codes, have been found which do allow for an efficient fault-tolerant realization of the full Clifford group (Bombin and Martin-Delgado, 2006).

2. *T* gate

Bravyi and Koenig (2013) suggest that with 2D stabilizer codes it is not possible to realize a T gate without large overhead. However, for gates such as the T gate [or Toffoli gate, also in $\mathcal{C}(3)$] the method of magic state distillation has been developed (Bravyi and Kitaev, 2005). This method shows how to realize these gates fault tolerantly by assuming that noiseless Clifford group operations and a supply of noisy unencoded $T|+\rangle$ ancillas are available. Thus, once we have built a low-noise Clifford computation substrate, universal quantum computation can be bootstrapped from it. In a nutshell, the ideas are as follows. We implement the T gate at the logical level using Fig. 6 which requires the preparation of low-noise logical ancillas $|A\rangle \equiv \overline{T}|\overline{+}\rangle$. We can obtain such an ancilla in a non-fault-tolerant noisy manner by, for example, injecting several noisy unencoded ancillas into the code (Knill, 2005). From many of these noisy encoded ancillas we distill using logical H, CNOTs, and measurements, a single low-error encoded ancilla. The strength of the distillation scheme is that the noise rate which one can tolerate on the unencoded $T|+\rangle$ ancillas for the scheme to work and produce very-low-noise encoded $|A\rangle$ ancillas is extremely high. Distillation can succeed if and only if the unencoded ancilla ρ has a fidelity $F = \langle A|\rho|A\rangle$ above approximately 0.854 (Reichardt, 2005).

The downside of this scheme is that the qubit or gate overhead per logical T gate is orders of magnitude larger than that of a "topological" CNOT [see, e.g., Fig. 11 in Raussendorf, Harrington, and Goyal (2007)]. Current work is ongoing to design alternative schemes to reach universal computation with reduced overhead [see, e.g., Jones (2013a, 2013b) and references therein].

It is worthwhile to mention a family of 3D color codes introduced by Bombin and Martin-Delgado (2007) that allow for the transversal realization of the T gate, thus requiring no additional qubit overhead. As these codes have stabilizers $\mathcal{S} = \langle \mathcal{S}_1(X), \mathcal{S}_2(Z)\rangle$ one has a transversal CNOT gate. For the Hadamard gate one can use the gate teleportation ideas above if one can prepare an ancilla in a $|\overline{+}\rangle$ state. In Sec. III.B.2 we discuss how to prepare the state $|\overline{+}\rangle$ for the surface code; such a technique also works for these color codes.

The smallest member of this class of color codes [[15,1,3]] is a quantum Reed-Muller code which has been known to have a transversal T gate due to the special symmetry which is inherent in its construction via classical Reed-Muller codes (Steane, 1999). A possibly even more attractive family of 3D codes are the gauge color codes (Bombin, 2013) for which the Hadamard gate is transversal. By fixing the logical state of the gauge qubits, one obtains a 3D color code for which the T gate is transversal. The idea of gauge fixing as a means of getting around the Eastin-Knill theorem was first explored by Paetznick and Reichardt (2013).

3. The logical Pauli frame

In this section we discuss how, during a fault-tolerant computation, one can handle the logical and elementary Pauli operators which are inferred from the syndrome measurement data. The idea is that the decoding procedure gives both a logical and a physical Pauli error that can be interpreted as a frame, the so-called Pauli frame (Knill, 2005) which we can classically track during the quantum computation.

First, we note that in principle it is never necessary to physically correct Pauli errors to map back to the $+1$ eigenspace of the stabilizer $\mathcal{S}$. This is because any syndrome eigenspace of $\mathcal{S}$ is a good code and we simply need to know which code space we are using. Second, note that if the quantum circuit consists entirely of Clifford gates, both at the logical and at the physical level, the classical information of the logical and physical Pauli frame does not necessarily need to be available during the execution of the circuit as the entire Pauli frame can be commuted through the circuit (as Clifford gates map Pauli frames onto Pauli frames). The Pauli frame simply alters the interpretation of the final measurement outcome of the computation. This is different if the circuit consists of non-Clifford gates as we will now discuss.

Imagine that syndrome data are collected and processed and every so often it is deduced that a logical (or physical) Pauli has happened on the coded data. It may also be that the quantum circuit we want to realize includes some logical Pauli operations. Consider what happens when a logical Pauli X occurs on the data prior to doing a T gate, as in Fig. 6. The X commutes through the CNOT gate and then effectively changes the way we should interpret the measurement outcome M_Z. Now, in case $M_Z = -1$ we have $T|\psi\rangle$ and we do not need to do a correction. If $M_Z = +1$ we need to correct with SX.

This means that the original Pauli X is mapped onto a logical Clifford error on the data which will subsequently need to be corrected. If we do not correct the Clifford error, it may spread and become a more complicated multiqubit error which is even harder to correct. This implies that it is best to know the logical Pauli frame of the data qubit and the ancilla qubit before we move on to the next gate after the T gate.

Once we are done determining the logical Pauli frame, we know which Clifford error took place and we should then try to correct it right away. Concerning knowing the logical Pauli frame, what is important is that one needs to know only the logical Pauli frame which influences the outcome of the measurement M_Z. For example, the results of parity check measurements after the CNOT gate on the ancilla qubit in Fig. 6 will not influence this logical Pauli frame and hence can be processed later. The outcome of these measurements may of course cause a change in the logical Pauli frame, but as SX itself (but not controlled SX) is a Clifford gate, this change can be commuted through and remain a logical Pauli frame.

If we know the logical Pauli frame on time, that is, before we move to the next gate, we can handle this logical Pauli frame in the classical control software as discussed, for example, by Fowler, Mariantoni *et al.* (2012). If we want to handle any logical Pauli in the classical control software, we just use this logical Pauli frame information to correct the interpretation of M_Z (or M_X measurement), thus changing which correction we do. In addition since we do not want to do any logical Pauli operations, we can replace the correction gate SX by the correction S (using that $SXT^{\dagger} \propto YT$ so that we have realized T modulo an additional Y).

In conclusion, one can argue that one does not need to physically implement any logical or elementary Pauli operation, but one does need to know the logical Pauli operation before one can proceed further with the computation. If determining the logical Pauli frame takes some time, as, for example, the quantum measurement is slow or the processing of the parity checks using classical computation is slow, one is thus required to wait before doing the classically controlled-SX gate. This additional delay is not problematic as was observed by DiVincenzo and Aliferis (2007) because parity checks are collected during this delay time and thus the qubits are protected. DiVincenzo and Aliferis (2007) showed that a slow measurement will not lead in general to a lower noise threshold but can be accommodated by small modifications in the fault-tolerant (concatenated code) architecture. Slow measurement here means a quantum measurement with a long latency: the rate at which parity checks are collected is not changed, but it takes a while to measure an ancilla qubit that has been coupled to the data (as in Fig. 1). It may be clear that acquiring syndrome data at a slower rate will lead to a lower noise threshold as it effectively corresponds to a higher error rate. Thus in order to keep the rate of syndrome data acquisition high if the measurement of the ancillas is slow (say 10 times slower than the gate time) we have to couple ten different ancillas to the data in sequence so that we get the measurement outcome of one of those ten ancillas at the rate of (roughly) the inverse gate time.

Given these considerations concerning the logical Pauli frame, we see an important distinction between the complexity of building a quantum memory (including only Clifford gates) versus building a quantum fault-tolerant computer using stabilizer codes. In a fault-tolerant computer, one may allow for slow measurements with some latency, but the classical processing of the syndrome data record, the decoding, should never lead to an increasing backlog of syndrome data. Let r_{proc} be the rate (in bauds) at which syndrome bits are processed and r_{gen} be the rate at which these syndrome bits are generated. We argue that if $r_{\mathrm{gen}}/r_{\mathrm{proc}} = f > 1$, a small initial backlog in processing syndrome data will lead to an exponential slowdown during the computation, for the following reasons.

Given that $f > 1$, there will be some time t_0 at which there is enough backlog in our syndrome record for us to have to delay executing the corrective gate after the T gate as we do not know whether a logical Pauli error happened, which influences what correction we should do. Let t_0^{proc} be the time up to which we have processed the syndrome data at time t_0, so $\Delta_{\mathrm{gen}} = |t_0 - t_0^{\mathrm{proc}}|$ is large enough that it is likely that a logical Pauli error has happened in the time interval Δ_{gen}. In this time interval we have generated an additional $D_1 = r_{\mathrm{gen}}\Delta_{\mathrm{gen}}$ bits. We now process this record at a rate r_{proc}, and hence this takes time $\Delta_{\mathrm{proc}} = f\Delta_{\mathrm{gen}}$. The problem is that during this delay time Δ_{proc} a new data record is generated of $D_2 = \Delta_{\mathrm{proc}} r_{\mathrm{gen}} = D_1 f > D_1$ bits. If there was a sufficient possibility for a logical Pauli error in the original data record of size D_1, then this also holds for data record D_2. Hence at some next T gate which is impacted by this Pauli frame information, we need to have at least processed the D_2 record. This implies again a delay in executing the gate during which one acquires a new data record $D_3 = fD_2$, etc. We assume that the number of T gates on a logical qubit is some polynomial in n, poly(n), e.g., for Shor's factoring algorithm $O(n^3)$ Toffoli gates are needed on n qubits. Then the backlog data record that we have acquired at the $k = \mathrm{poly}(n)$th gate is $D_k = f^k D_1$ which is exponential in n. Hence in order to execute the kth T gate, one has to wait for a time $r_{\mathrm{proc}} D_k$, an exponential amount of time in n.

The conclusion is that the syndrome data acquisition through quantum measurement and the classical processing should be fast enough to let the logical Pauli frame be "retarded" by only a constant amount of time, i.e., not increasing during the time of the computation. In order to achieve this one needs to decode using maximum classical parallelism, possibly using an on-chip decoder. Whether the backlog question is a practical problem thus depends on how fast one can decode as compared to the physical error rate of the elementary qubits; see the further discussion in Sec. III.D.

We should contrast this backlog issue with the case of a quantum memory (including Clifford gates) in which the computation never has to wait for the classical processing of the logical Pauli frame. Such a stored qubit could be measured at the end of its storage time T_{store}: in case of slow classical processing the outcome of the measurement may not be immediately available (as it depends on the syndrome record), but it would just mean that the computation, including the processing of syndrome data, is finished in time fT_{store} which is just a constant slow down.

The upshot of these considerations is that 2D and 3D stabilizer codes will be most suitable for building a quantum memory and performing Clifford group operations. The goal of universal quantum computation within the same platform can be reached using methods such as injection and distillation or using a code with a transversal T gate, but the additional overhead and complexity of distillation and demands for fast decoding are considerable.

III. 2D (TOPOLOGICAL) ERROR CORRECTION

In this section we discuss several stabilizer and subsystem codes in which the parity checks act locally on qubits laid out on a 2D lattice. Before we discuss these codes, we make a few comments on noise models and noise thresholds.

In numerical or analytical studies of code performance, one uses simple error models such the independent depolarizing noise model to assess the performance of the code. Independent depolarizing noise assumes that every qubit independently undergoes an X, Y, or Z error with equal probabilities $p/3$ and no error with probability $1-p$. Similarly, if qubits undergo single-, or two-qubit gates or measurement and preparation steps, one assumes that the procedure succeeds with probability $1-p$ while with total probability p (tensor products of) Pauli errors X, Y, and Z are applied.

A related noise model is that of independent X and Z errors in which a qubit can independently undergo an X error with probability p and a Z error with probability p in each time step. In all codes that we discuss in this section the parity checks are either X or Z like, detecting either Z or X errors. In addition, the parity Z and X checks have the same form; hence the simplest form of error correction is to correct X and Z errors in the same fashion but independently. For depolarizing noise, this means that we effectively neglect correlations between X and Z errors. It is also possible to decode the surface code taking these correlations into account; see Fowler (2013b) and references therein.

We consider codes which encode a single qubit in a block of n qubits with $n = O(L^2)$ with L the linear size of the 2D array. Several parameters can characterize the code performance. One is the so-called pseudothreshold $p_c(L)$ for which $p_c(L) = \overline{p}(p, L)$, i.e., the logical error rate equals the elementary error rate given a fixed block size. This logical error rate $\overline{p}(p, L)$ could be separately split into a logical, X, Z, or total error rate, all being functions of the block size and the elementary error rate p. In the definition of the pseudothreshold we can assume that the elementary error rate is less than 50% (otherwise the qubits would be completely randomized and no coding would help) and note that the logical error rate is also maximally equal to 50%. When the elementary error rate is less than the logical error rate, coding is not helpful. When the logical error rate is less than the elementary error rate, coding is helpful. The pseudothreshold thus captures the crossover point. This crossover point depends on L and gives more information than the typically stated asymptotic threshold $p_c = \lim_{L\to\infty} p_c(L)$. Svore *et al.* (2006) considered the behavior of pseudothresholds for concatenated code schemes.

For the Bacon-Shor code in Sec. III.C.1, the asymptotic threshold $p_c = 0$; hence it is of interest to consider what is the optimal block size for this code. Another interesting class of 2D topological stabilizer codes is the color codes (Bombin and Martin-Delgado, 2006). The color codes offer little practical advantage over the surface code if the goal is to build a quantum memory as some of the parity checks involve more than four qubits. Having higher-weight parity checks negatively impacts the noise threshold as we assume each gate in the parity check measurement circuit can fail. This is likely to be the reason that the phenomenological threshold of the color code obtained as approximately 0.082% in the detailed study by Landahl, Anderson, and Rice (2011) is lower than the surface code threshold (about 1%). The higher-dimensional color codes may be of interest in schemes for universal encoded computation; see the discussion in Sec. II.G.

A. Surface code

The surface code is a version of Kitaev's toric code (Kitaev, 2003) in which the periodic boundaries of the torus have been replaced by open boundaries (Bravyi and Kitaev, 1998; Freedman and Meyer, 2001). Many of its properties and ideas for its use as a quantum memory were first analyzed in the seminal paper by Dennis *et al.* (2002). The topological 2D realization of the CNOT gate (Sec. III.B.4) was first proposed by Raussendorf and Harrington (2007) and Bombin and Martin-Delgado (2009).

There are several different ways of encoding and representing qubits in the surface code. We start by discussing how to encode a single logical qubit in a sheet or patch and then show in Sec. III.B.3 how a CNOT gate can be performed between such qubits in patches using logical Pauli measurements. In Sec. III.B.4 we discuss encoding single qubits into the surface code using so-called smooth and rough defects, here called smooth and rough qubits. The performance of this encoding in terms of logical error rate and overhead has been studied extensively by Fowler *et al.* The review by Fowler, Mariantoni *et al.* (2012) gives an excellent overview of methods of fault-tolerant quantum computation using this encoding. Another way of encoding qubits in the surface code is by means of pairs of distant dislocations (Bombin, 2010a); the use of this scheme for fault-tolerant quantum computation was analyzed by Hastings and Geller (2014).

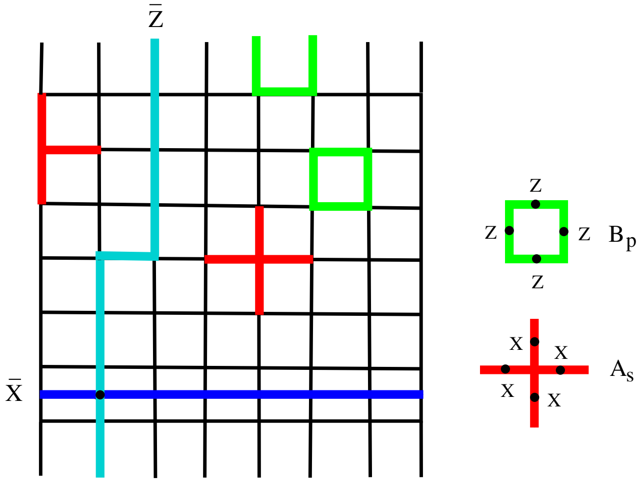

FIG. 7 (color online). Surface code on an $L \times L$ lattice. On every edge of the (black) lattice there is a qubit, in total $L^2 + (L-1)^2$ qubits (depicted is $L = 8$). Two types of local parity checks A_s and B_p each act on four qubits, except at the boundary where they act on three qubits. The subspace of states which satisfy the parity checks is two dimensional and hence represents a qubit. $\bar{Z}$ is any Z string connecting the north to the south boundary, which is referred to as "rough", while $\bar{X}$ is any X string connecting the east to west "smooth" boundary running through vertices of the dual lattice.

A simple continuous sheet, depicted in Fig. 7, can encode one logical qubit. The linearly independent parity checks are weight-4 plaquette Z checks B_p and star X checks A_s which mutually commute and are modified at the boundary to act on three qubits; see Fig. 7. Note that the star operators are just plaquette operators on the dual lattice when also interchanging $X \leftrightarrow Z$. The smallest surface code encoding one logical qubit that can correct one error is the code [[13,1,3]].[12] $\overline{Z}$ is any Z string which connects the north and south rough boundaries; we can deform this string by multiplication by the trivially acting plaquette operators. $\overline{X}$ is any X string (on the dual lattice) connecting the smooth east and west boundaries. As these strings have to connect boundaries in order to commute

[12] One can minimize the qubit overhead while keeping the distance equal to 3 by rotating the lattice and chopping off some qubits at the corners to get a total of nine qubits. This rotation + chopping, while leaving the distance unchanged, can be done for arbitrary-sized lattices (Horsman *et al.*, 2012).

with the check operators, their minimum weight is L. Thus for general L, the code parameters are $[[L^2+(L-1)^2, 1, L]]$.

Using 13 qubits to correct one error does not seem very efficient, but the strength of the surface code is not fully expressed in its distance which scales only as the square root of the number of qubits in the block.[13]

Kitaev's original toric code is defined on a 2D lattice with periodic boundary conditions (a torus). For the toric code, there is a linear dependency between all the Z checks (the product of the Z checks is I) and a similar linear dependency between all the X checks. With this linear dependency it follows that the number of logical qubits is 2. The torus has two nontrivial loops: the logical $\overline{Z}_1$ is one nontrivial loop of Z's and the logical $\overline{Z}_2$ corresponds to the other nontrivial loop of $\bar{Z}$'s. The matching logical $\overline{X}_1$ and $\overline{X}_2$ are similar loops running over the dual lattice.

For the toric code it may be clear that the logical operators are directly connected to the homology of the torus. One can deform a logical $\overline{Z}$ by multiplying it with Z checks B_p but it will remain a noncontractible loop on the torus, as products of plaquette B_p checks correspond to trivial, contractible loops. This holds analogously for the $\overline{X}$ loops and products of star A_s checks on the dual lattice.

1. Viewing the toric code as a homological quantum code

The toric code is a simple example of a homological (CSS) quantum code[14] (Freedman and Meyer, 2001; Kitaev, 2003; Guth and Lubotzky, 2014) in which the logical $\overline{Z}$ ($\overline{X}$) operators correspond to the homology (cohomology) groups of the underlying manifold. In the surface code one can view the homology as being relative to a boundary (Bravyi and Kitaev, 1998). In this section we discuss the framework of homological stabilizer codes and illustrate the concepts with the toric code in two, three, and four dimensions.

For the toric code one takes a flat two-dimensional manifold with periodic boundaries, a torus. One has to fix a triangulation of the manifold resulting in a so-called simplicial complex which consists of 0-simplices (vertices), 1-simplices (edges), and 2-simplices (faces), etc. The toric code corresponds to taking a square lattice with faces which consist of four edges.

In the general construction, with each type of object, e.g., vertices, edges, or faces, or generally i-simplices, one associates a $\mathbb{Z}_2$-vector space C_i. Elements of C_0 are thus a collection of vertices, elements of C_1 are collections of edges, etc. In a $\mathbb{Z}_2$-vector space C_i, addition is mod 2. Two binary vectors a and b are orthogonal if and only if $\sum_i a_i b_i = 0 \bmod 2$ or the number of bits i for which $a_i = b_i = 1$ is even. If one represents such binary vector a by a Pauli X operator $P_X(a) = \Pi_i X_i^{a_i}$ and b by a Pauli Z operator $P_Z(b) = \Pi_i Z_i^{b_i}$, then the inner product between a and b is 0 if and only if $P_X(a)$ and $P_Z(b)$ commute.

For the toric code we associate the qubits with the 1-simplices (the edges), but for more general homological codes in higher dimensions one can associate qubits with i-simplices. The stabilizer generators and logical operators of the CSS code are then constructed using subspaces of the vector space C_i such that the required commutation relations between these operators hold and the logical operators directly relate to topological properties of the manifold. This comes about as follows.

One starts by defining boundary operators $\partial_i\colon\ C_i \to C_{i-1}$ which act as the name suggests. The boundary operator ∂_2 maps a face onto the collection of edges which are incident to the face, and the boundary operator ∂_1 maps a collection of edges onto a collection of vertices, namely, the end points of these edges. For qubits associated with 1-simplices, the Z checks are obtained as the boundary space $B_1 = \mathrm{Im}(\partial_2)$, i.e., generating vectors in B_1 correspond to the boundary of a face. For the square lattice, these generators are thus the Z plaquettes acting on the four edges of every face of the lattice.

An important property of the boundary operator is that the boundary of an i-simplex does not have a boundary; mathematically this is expressed as $\partial_{i-1} \circ \partial_i = 0$ applied to any vector in C_i.

A 1-cycle is defined to be a collection of edges without a boundary. This means that the vector space of 1-cycles is $Z_1 = \ker(\partial_1)$. Any element of B_1 is a 1-cycle, or $B_1 \subseteq Z_1$; these are the trivial cycles that correspond to products of the Z-check operators. The first homology group $H_1(T, \mathbb{Z}_2) = Z_1/B_1$ of the torus T is generated by 1-cycles which are not the boundaries of plaquettes, i.e., the two nontrivial cycles around the torus. These cycles correspond to the logical $\overline{Z}$ operators.

In the general construction when qubits are associated with i-simplices, the Z checks correspond to the generators of $B_i = \mathrm{Im}(\partial_{i+1})$, the i-cycle space is $Z_i = \mathrm{Ker}(\partial_i)$, and the ith homology group $H_i(M, \mathbb{Z}_2) = Z_i/B_i$ captures the logical $\overline{Z}$ operators.

In order to define the X checks for the quantum code one makes the same construction on the dual lattice or, equivalently, one uses cohomology. One can define the coboundary operator $\delta_i\colon\ C_i \to C_{i+1}$ which maps an i-simplex onto the set of $(i+1)$-simplices incident to it. Thus the coboundary operator δ_1 maps an edge onto the faces which are incident to this edge, δ_0 maps a vertex onto the edges emanating from it, etc. With the coboundary operator one can define a cocycle space $Z^i = \mathrm{Ker}(\delta_i)$ and a coboundary space $B^i = \mathrm{Im}(\delta_{i-1})$. For qubits associated with i-simplices, the generators of B^i correspond to the X checks and the generators of the ith cohomology group $H^i(M, \mathbb{Z}_2) = Z^i/B^i$ are the logical $\overline{X}$ operators. For the toric code one has $i = 1$ and so the X checks correspond to the generators of B^1, which are obtained from taking the edges incident to a vertex; hence the star operators.

For the toric code it is easy to verify that the logical operators and the checks are all mutually commuting. For a general homological CSS quantum code, this essential

[13] One can prove that the distance of any 2D stabilizer code is at most $O(L)$ (Bravyi and Terhal, 2009). However, one can also show (Bravyi, Poulin, and Terhal, 2010) that any block of size $R \times R$, where R is less than some constant times the distance, is correctable, i.e., all errors in such an $R \times R$ patch can be corrected. These arguments show that there are no other 2D stabilizer codes with better distance scaling and that this scaling allows one to correct failed blocks of size beyond the distance.

[14] Readers less interested in this mathematical framework can skip this section without major inconvenience.

property comes about from the fact that $\delta_i = \partial_{i+1}^T$ (where T is the matrix transposition if we view these linear maps as matrices acting on a finite-dimensional space). Then $B^i = \text{Im}(\delta_{i-1}) = \text{Im}(\partial_i^T) = [\text{Ker}(\partial_i)]^\perp = Z_i^\perp$ [and similarly $B_i = (Z^i)^\perp$]. As $B_i \subseteq Z_i$, the spaces B^i and B_i are orthogonal, so the check operators all commute and $B^i = Z_i^\perp$ implies that the X checks commute with the logical $\overline{Z}$, etc. Thus in general the ith homology groups $H_i(M, \mathbb{Z}_2)$ and cohomology groups $H^i(M, \mathbb{Z}_2)$ and their dimensions $\dim(H_i) = \dim(H^i)$ determine the number of logical qubits and also the character of the logical operators, meaning the dimensionality of their support (one-dimensional stringlike or two-dimensional surfacelike, etc.).

Instead of using the coboundary operator, one can also consider the dual of a simplicial complex of an n-dimensional manifold. Going to the dual means that an i-simplex is mapped onto an $(n-i)$-simplex, i.e., in two dimensions a vertex becomes a face, while in three dimensions a vertex becomes a 3-simplex. For the toric code, a face (2-simplex) thus gets mapped onto a vertex (0-simplex) and vice versa, and edges (1-simplices) remain the same. In order to obtain the X-check operators and the logical $\overline{X}$, we can define boundary operators on the dual lattice. If we associate qubits with i-simplices on the primal lattice, the boundary space B_{n-i}^{dual} generates the X checks and H_{n-i}^{dual} (which is isomorphic to H^i) is generated by the logical $\overline{X}$ operators.

We illustrate the construction with the 3D and 4D toric codes defined on cubic lattices with periodic boundaries in all directions so that one has a 3-torus T^3 and a 4-torus T^4, respectively. For the 3D toric code (Castelnovo and Chamon, 2008) the 3D cubes are the 3-simplices, their faces are the 2-simplices, and we associate the qubits with the 1-simplices or edges. For the 3D toric code, $\text{Im}(\partial_2)$ is generated by the four-qubit Z-plaquette operators in the x-y, x-z, and y-z planes. The logical $\overline{Z}$ operators are elements in $H_1(T^3, \mathbb{Z}_2)$, the three noncontractible Z loops around the 3-torus.

If we take the boundary of the boundary of a 3D cube, i.e., apply the map $\partial_2 \circ \partial_3$, on the cube, we get 0. This implies that the product of Z plaquettes which make up the boundary of the cube has no support on the edges; in other words the product of these stabilizer checks is I. This is a local linear dependency or a redundancy among the stabilizer checks which ensures that for any X error, the Z checks that are nontrivial form a connected string. To see this note that if one plaquette of the cube has nontrivial eigenvalue -1, some other plaquette of this cube must also have -1 eigenvalue as the product of all plaquettes which make up the cube is always I. We can also understand this property as a Gauss's law for $\mathbb{Z}_2$ charges: $\mathbb{Z}_2$ flux lines (lines of nontrivial syndromes) form closed loops which have no sources and do not terminate. This kind of redundancy is not present for the two-dimensional toric or surface code as no set of edges is the boundary of a boundary. We discuss in Secs. III.D and III.E how this redundancy and the lack thereof plays a role in the complexity of locally decoding and the question of self-correction and finite-temperature topological order.

We consider the X checks of the 3D toric code. The X checks can be obtained from the coboundary operator δ_0 [taking $\text{Im}(\delta_0)$] which maps a vertex on the set of six edges emanating from this vertex. Hence the X check is a star operator centered on a vertex acting on six qubits. The logical $\overline{X}$ operators are elements in $H^1(T^3, \mathbb{Z}_2)$, i.e., they are xy-oriented, yz-oriented, or xz-oriented planes of X's on the dual lattice.

We observe that as the edges (1-simplices) become faces [(3–1)-simplices] on the dual lattice, one does not have a local linear dependency for the X checks, as the faces are only the boundary of some three-dimensional objects and never the boundary of a boundary. One thus needs to go to four dimensions in order for there to be a local linear dependency for both X and Z checks. In the 4D toric code we associate qubits with the faces of a four-dimensional cubic lattice. Each X check is associated with an edge such that the X check acts on the qubits on the six faces which touch the edge [elements of $\text{Im}(\delta_1)$]. Similarly, the Z checks are obtained as $\text{Im}(\partial_3)$, i.e., as the collections of faces which form the boundary of a three-dimensional cube. Hence the Z check also acts on six qubits. The logical operators are associated with (co)homology groups $H_2(T^4, \mathbb{Z}_2)$ which has rank 6, so the code encodes six logical qubits, and $H^2(T^4, \mathbb{Z}_2) \simeq H_2(T^4, \mathbb{Z}_2)$. Now both X and Z checks have a local linear dependency, as $\partial_3 \circ \partial_4 = 0$ (the boundary of a four-dimensional cube is a collection of three-dimensional cubes which has no boundary) and $\delta_1 \circ \delta_0 = 0$. Both logical operators are surfacelike (have a two-dimensional support) as they are elements in $H_2(T^4, \mathbb{Z}_2)$.

B. Quantum error correction with the surface code

We first consider how quantum error correction can take place for the surface code assuming that the parity check measurements are noise free. If a single X error occurs on an edge in the bulk of the system, then the two plaquette operators next to it will have an eigenvalue of -1. The places where these plaquette eigenvalues are -1 are sometimes called defects. A connected string of X errors will produce only two defects at its boundary. If the X error rate p per qubit is sufficiently small, one obtains a low density of close-by defects. Such errors are correctable as defects can be locally paired without much ambiguity. As we know, inferring an error E' which differs from the real error E by only stabilizer operators (plaquette operators in this case) is harmless. Here it means that we decode correctly as long as $E'E$ does not form an X string $(\overline{X})$ which goes from one smooth boundary to the other smooth boundary. For a sufficiently low rate, the operators $E'E$ will instead form small closed loops which are products of check operators. From this picture it may be intuitively clear that there should be a finite asymptotic threshold p_c for noise-free error correction.

For the bulk system the error syndrome thus gives us the location of the defects. A minimum-weight decoding algorithm then corresponds to finding a minimum-weight error string $E(X)$ which has these defects as end points. This decoding algorithm can be implemented using Edmond's minimum-weight matching (Blossom) algorithm (Edmonds, 1965). Open-source software Autotune has been developed specifically for surface code decoding (Fowler, Whiteside *et al.*, 2012) and is available on GitHub. We note that at the boundary of the lattice an error can produce a single defect: in a minimum-weight matching decoder one can match these defects with a possible ghost defect beyond the boundary.

Fowler, Whiteside, and Hollenberg (2012) demonstrated empirically that the number of steps in the minimum-weight matching algorithm scales as $O(L^2)$ per round of error correction.

Ideal decoding is not minimum-weight decoding, but maximum-likelihood decoding as described in Sec. II.B.1. As argued in Sec. II.B.1, one can estimate the maximally achievable threshold p_c with any decoder by relating the maximum-likelihood decoding problem to a phase transition of a classical Hamiltonian with quenched disorder. For the surface code this Hamiltonian is the 2D random-bond Ising model (Dennis *et al.*, 2002; Wang, Harrington, and Preskill, 2003). Assuming noise-free parity checks and independent X errors with probability p, the critical value has been numerically estimated as $p_c \approx 11\%$ (Dennis *et al.*, 2002). For a depolarizing noise model with error probability p, this threshold has been shown to increase to $p_c \approx 18.9\%$ in the numerical study by Bombin *et al.* (2012).

These thresholds for independent X errors or depolarizing noise are the best one can expect with any code (within numerical accuracy) as they saturate the so-called Hashing bound for these error channels. The Hashing bound for, say, a depolarizing channel says that the channel cannot preserve or transmit any quantum information, its channel capacity is zero, when $1 - H_{\mathrm{depol}}(p) \leq 0$, where $H_{\mathrm{depol}}(p)$ is the Shannon entropy of the depolarizing channel $H_{\mathrm{depol}}(p) = -(1-p)\log_2(1-p) - p\log_2(p/3)$, implying that $p_c \gtrapprox 18.9\%$. For independent X errors with probability p, the Hashing bound $1 - 2H_2(p) = 0$ gives $p_c \approx 11\%$ $[H_2(p) = -p\log_2 p - (1-p)\log_2(1-p)]$.

This picture becomes modified when the parity checks are inaccurate. A simple way to model noisy parity checks is to assign a probability q for the parity check outcome to be inaccurate while in between the parity checks qubits undergo X and Z errors with probability p as before. In practice, one would expect the parity check measurements to induce some correlated errors between the qubits of which we take the parity. For example, for the parity Z check one may expect additional qubit dephasing if more information than merely the parity is read out.

As the parity check measurements are no longer reliable, one needs to change their use as an error record. For example, a single isolated defect which appears for a few time steps and then disappears for a long time is likely to be caused by a faulty parity measurement outcome instead of an error on the data qubits. The strength of topological codes for sufficiently large L (as compared to using small codes and code concatenation) is that noisy parity checks can be dealt with by repeating their measurement as the additional noise which the parity checks produce on the code qubits is local and, at sufficiently low rate, correctable.

Both minimum-weight decoding and maximum-likelihood decoding can be generalized to the noisy parity check measurement setting. We extend the lattice into the third (time) dimension (Dennis *et al.*, 2002); see Fig. 8. Vertical links, corresponding to parity check measurements, fail with probability q while horizontal links fail with probability p. In minimum-weight decoding the goal is now to find a minimum-weight error E which has vertical defect links, where the parity check is -1 as its boundary; see Fig. 8. When we match the defects in 3D we obtain an inferred error E' which can have a vertical time component (corresponding to a measurement error) as well as horizontal space components (corresponding to qubit errors). We can visualize the difference between errors E which get properly corrected and errors for which decoding fails by considering $E'E$. When error correction succeeds, $E'E$ is a trivial loop in the 3D lattice, but decoding fails when $E'E$ is some nontrivial space-time loop which winds around the torus (or for the surface code which connects the proper two boundaries).

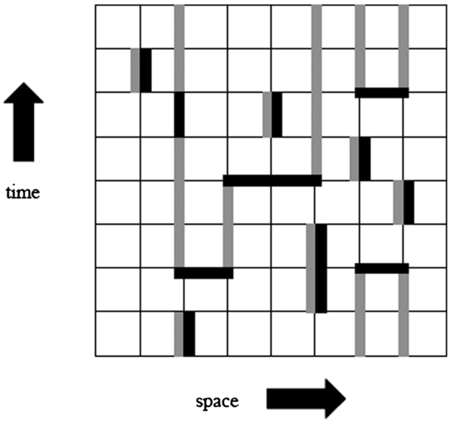

FIG. 8. 1D cross section of the lattice in space and time. Gray links correspond to nontrivial -1 syndromes. Errors that could have caused such a syndrome are represented by black links. Horizontal black links are qubit errors while vertical black links are parity check measurement errors. Note that a possible error E has the same boundary as the gray defect links: a likely error E (in the bulk) can be found by looking for a minimum-weighted matching of the end points of the gray links. From Dennis *et al.*, 2002.

If the parity check measurements are ongoing, one needs to decide how long a time record to keep in which one matches defects in the time direction; this length depends on the failure probability q. In the simple case when $q = p$ the record length is taken as L (Wang, Harrington, and Preskill, 2003).

An analytical lower bound on the noise threshold for $q < p$ is derived by Dennis *et al.* (2002) with the value $p_c \geq 1.1\%$. Numerical studies by Wang, Harrington, and Preskill (2003) (using minimum-weight decoding) showed a threshold of $p_c \approx 2.9\%$ for $p = q$.

If we assume that the parity check measurement errors are due to depolarizing noise on all elementary gates, measurement, and preparations with depolarizing probability p, Raussendorf, Harrington, and Goyal (2007) found a threshold of 0.75%. Below the noise threshold the logical error rate $\overline{p}(p, L) \sim \exp[-\kappa(p)L]$, where $\kappa(p) \approx 0.8$–0.9 at $p = p_c/3$ (Wang, Harrington, and Preskill, 2003; Raussendorf, Harrington, and Goyal, 2007). Wang *et al.* (2011) even estimated the depolarizing noise threshold to be in the range of 1.1%–1.4%. All these results have been obtained for toric codes, assuming periodic boundary conditions: one may expect results to be somewhat worse for surface codes (Fowler, 2013a).

These results indicate that the surface code, even with noisy parity check measurements, has a very high threshold as compared to other coding schemes; see, e.g., those studied by Cross, DiVincenzo, and Terhal (2009).[15] A practically relevant question is how much overhead L is needed before one is in the scaling regime where the pseudothreshold is close to the asymptotic threshold $p_c(L) \approx p_c$?

The pseudothreshold for a small code such as [[13,1,3]] is very small, certainly no higher than 0.1%. Using the results in Fowler (2013a), one can estimate that the [[25,1,4]] ($L = 4$) surface code has a pseudothreshold [defined by $\overline{p} = Ap^{L/2}$ with $A = A_X$, A_Z given in Table I in Fowler (2013a)] of approximately 0.2% and [[61,1,6]] has a pseudothreshold of approximately 0.7%. Thus with a depolarizing error rate $p = 5 \times 10^{-4}$ [[25,1,4]] gives a logical X or Z error rate $\overline{p}_X \approx \overline{p}_Z \approx Ap^2 \approx 1 \times 10^{-4}$ which is barely lower than the bare depolarizing rate. Even though small surface codes have worse performance than large codes they could still be used as test beds for individual components and error scaling behavior.

Minimum-weight decoding with Edmonds' matching algorithm is a good decoding method if our goal is to realize a quantum memory (with or without encoded Clifford group operations). As one never needs to physically do any correction (see the notion of the Pauli frame discussed in Sec. II.G), the measurement record can be stored and the data record can be processed at leisure and used to interpret a final $M_{\overline{X}}$ or $M_{\overline{Z}}$ measurement on the qubits. The realization of such a quantum memory will require that the record of parity check measurements is obtained at a sufficiently high rate compared to the error rate, since a low-rate stroboscopic picture of the defects (even if they are obtained perfectly) could potentially miss the occurrence of a logical error. Fowler, Mariantoni *et al.* (2012) proposed a 200 ns surface code cycle time (based on a 10–100 ns elementary gate time) meaning that every 200 ns both X and Z parity check measurements over the whole lattice are executed.

Researchers have developed potentially more efficient renormalization-group (RG) decoders (Bravyi and Haah, 2013; Duclos-Cianci and Poulin, 2014) which process the defects using parallel processing over the 2D or 3D lattices in time $O(\log L)$ (not taking into account a finite speed of communication). The idea of the simple decoder in Bravyi and Haah (2013) which works for any D-dimensional stabilizer code is to recursively match defects locally. For a 2D surface code with perfect parity check measurements, one starts by dividing up the defect record into local clusters of $O(1)$ size. In each cluster the algorithm tries to find a local error which removes the defects. If a cluster contains a single defect, for example, then no such local error can be found. Thus the next step is to enlarge the linear size of the cluster by a factor of 2 and reapply the same procedure on the leftover defect record. The decoder stops when no more defects are present or when one has reached a certain maximum number of iterations $r = O(\log L)$. For the toric code with perfect parity checks, Bravyi and Haah (2013) obtained a noise threshold of $p_c = 6.7\%$ using this RG decoder while the RG decoder in Duclos-Cianci and Poulin (2010) achieves 9% (minimum-weight decoding via matching gives 10.3%).

As mentioned before, there are various ways in which we can encode multiple qubits in the surface code and do a logical Hadamard or CNOT gate. The simplest method is to encode multiple qubits in multiple separate sheets (as in Fig. 7) laid out next to each other in a 2D array as in Fig. 12. Using operations on a single sheet one can do a logical Hadamard gate; see Sec. III.B.2. A CNOT gate between qubits in separate sheets can be realized using the idea of lattice surgery in which sheets are merged and split as proposed by Horsman *et al.* (2012).

The important point of doing a CNOT or Hadamard gate using these code deformation methods is that their implementation does not affect the surface noise threshold as error correction is continuously taking place during the implementation of the gates and the single qubit noise rate is not substantially changed. In addition, the realization of these gates does not require a large overhead in terms of space, meaning additional qubits, but the gates do require some overhead in time, as compared to transversal or constant-depth gates.

Another method of encoding qubits is to have one sheet for all qubits in the computation such that logical qubits are represented by holes in the lattice; see Sec. III.B.4. Given this encoding, it is possible to disconnect and then deform the encoding of a single qubit so that it becomes a single disconnected sheet [see details in Fowler, Mariantoni *et al.* (2012)] on which we can do the Hadamard gate or do a preparation or measurement step.

1. Preparation and measurement of logical qubits

How do we prepare the surface code memory in the states $|\overline{0}\rangle$, $|\overline{1}\rangle$, or $|\overline{\pm}\rangle$? And how do we read out information, that is, realize $M_{\overline{X}}$ and $M_{\overline{Z}}$? In order to prepare $|\overline{0}\rangle$, we initialize all elementary qubits in Fig. 7 to $|0\rangle$ and start measuring the parity checks. The state $|00\cdots0\rangle$ has $B_p = +1$ and $\overline{Z} = +1$ while the star operators A_s have random eigenvalues ±1 corresponding to the presence of many Z errors. Thus we choose some correction E for these Z errors (we pick a Pauli frame): the choice will not matter as E commutes with $\overline{Z}$. If the preparation of $|0\rangle$ and the parity check measurements are noisy, one needs to measure the parity checks for a while before deciding on a Pauli frame for both X and Z errors. The preparation of $|\overline{1}\rangle$ and $|\overline{\pm}\rangle$ can be performed analogously using the ability to prepare the elementary qubits in $|1\rangle$ and $|\pm\rangle$, respectively. Instead of preparing the quantum memory in one of these four fixed states, there are also methods for encoding a single unencoded qubit $|\psi\rangle$ into a coded state $|\overline{\psi}\rangle$; see Dennis *et al.* (2002) and Horsman *et al.* (2012). Of course, during this encoding procedure, the qubit to be stored is not fully protected, as the qubit starts off in a bare, unencoded state.

A projective destructive measurement in, say, the $|\overline{0}\rangle$, $|\overline{1}\rangle$ basis ($M_{\overline{Z}}$) proceeds essentially in reverse order. One measures all qubits in the Z basis. Using the past record of parity Z-check measurements and this last measurement, one infers what X errors have taken place and corrects the outcome of $\overline{Z} = \pm1$ accordingly.

[15] One has to be careful in comparing noise-threshold values across publications as slightly different methods, noise model, decoding strategy, and code usage can impact the results.

2. Hadamard gate

Consider doing a Hadamard rotation on every elementary qubit on a sheet encoding one logical qubit. The resulting state is a $+1$ eigenstate of the Hadamard-transformed parity checks $HA_sH^\dagger$ and $HB_pH^\dagger$ which are the plaquette Z check (respectively, the star X check) of the code $\mathcal{S}_{\rm dual}$ defined on the dual lattice. The dual lattice is defined by placing a vertex on each plaquette in Fig. 7 and connecting these vertices by edges on which the qubits are defined. On the dual lattice the rough and smooth boundaries are thus interchanged so that the lattice (code) is effectively rotated by 90°. The Hadamard gates map $\overline{Z}$ onto $\overline{X}_{\rm dual}$ and $\overline{X}$ onto $\overline{Z}_{\rm dual}$. We have thus performed a Hadamard transformation but we have also rotated our code. In principle one can work with this rotated code as long as we can connect the qubits of another, say, nonrotated sheet, with this rotated sheet via some long-range interactions. However, it is more practical if we rotate the code back to its initial orientation. The original procedure described by Dennis *et al.* (2002) showed how by a sequence of ancilla preparations at the boundaries and local CNOT gates one can modify the boundaries so that a rough boundary becomes smooth again and vice versa. In this procedure one removes qubits from the code at the west and south boundaries and one adds qubits at the north and east boundaries so that the lattice is effectively shifted upward. Instead of using CNOT gates one can add ancilla qubits and immediately measure the new plaquette and star operators. What is important is that this rotation over 90° can be done only gradually in $O(L)$ steps as sketched in Fig. 9 so that the distance between two rough boundaries or two smooth boundaries remains L (so as to protect the encoded qubit). The overall shift of the lattice can be repaired either by swapping qubits in the southwest direction or by keeping the shifted lattice and making sure other sheets connect to qubits on the shifted sheet. It means that some flexibility or nonlocality in the coupling structure is required at these boundaries.

It is simple to show that the Hadamard gate for the surface code requires a quantum circuit of depth scaling with L. For the Hadamard gate we have $\overline{H}\,\overline{X}\,\overline{H}=\overline{Z}$ and $\overline{H}\,\overline{Z}\,\overline{H}=\overline{X}$, and $\overline{X}$ and $\overline{Z}$ are the strings going from boundary to boundary. Imagine $\overline{H}$ implemented by a constant-depth circuit; it implies that $\overline{H}\,\overline{Z}\,\overline{H}$ is a fattened (by some constant factor) string going from the north to the south boundary in Fig. 7. The original $\overline{Z}$ was any string going from the north to the south boundary and hence it is simple to see that the operator $\overline{H}\,\overline{Z}\,\overline{H}$ will commute with the original $\overline{Z}$ and therefore $\overline{H}\,\overline{Z}\,\overline{H}$ cannot be the logical $\overline{X}$ operator. This argument fails only when the depth of the circuit is of the order of L so that the string might be completely spread over the lattice.

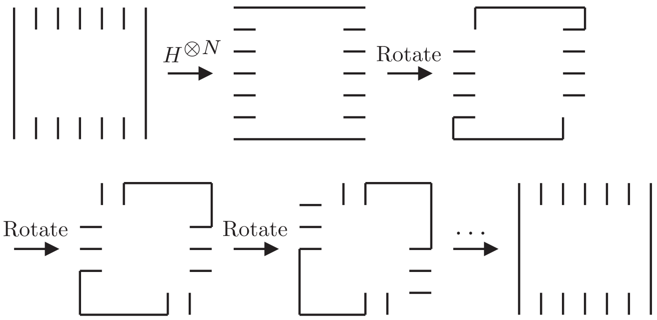

FIG. 9. The Hadamard gate on a sheet which encodes a single-logical qubit as in Fig. 7; see also Horsman *et al.* (2012). After performing a Hadamard gate on each qubit, the rotated code lattice is gradually rotated back to its original orientation by adding and taking away qubits and stabilizer checks at the boundaries.

3. CNOT gate via lattice surgery

This construction for the logical CNOT gate is based on the circuit in Fig. 10 which implements the CNOT gate through two-qubit parity measurements, originally described by Gottesman (1999a). In the dislocation encoding of Hastings and Geller (2014) this quantum circuit is similarly used to reduce a CNOT gate to the measurement of logical XX and ZZ operators. For the dislocation encoding of Hastings and Geller (2014) one also has the ability to measure the logical ZX and YZ, that is, any product of two logical Pauli operators. One can then observe that any single (logical) qubit Hadamard H gate or the S gate ($SXS^\dagger=Y$) can be absorbed into either the following logical single-qubit Pauli measurement (if the logical qubit is to be measured) or a modified two-qubit logical Pauli measurement of a CNOT gate. This implies that in such a scheme executing such gates does not cost any additional time.

To verify the CNOT circuit one can consider the evolution of the input $|c\rangle_1|0\rangle_2|t\rangle_3$ for bits $c=0$, 1 and $t=0$, 1 explicitly (here 1 denotes the top qubit in Fig. 10). For $M_{XX}=+1$, we have a bit $b_{xx}=0$ and $M_{XX}=-1$ corresponds to $b_{xx}=1$, etc. We have the overall evolution

$$|c\rangle_1|0\rangle_2|t\rangle_3\rightarrow|c\rangle_1Z_2^{b_x}|+\rangle_2Z_3^{b_{xx}}X_3^{b_{zz}}|c\oplus t\rangle_3. \tag{9}$$

We observe the logic of the CNOT gate on qubits 1 and 3 in addition to corrective Pauli's $Z_3^{b_{xx}}X_3^{b_{zz}}$ which depend on the outcomes b_{xx} and b_{zz} of the measurements M_{XX} and M_{ZZ}, respectively. The measurement M_X on the second qubit ensures that no information leaks to that qubit so that the CNOT gate properly works on any superposition of inputs.

This circuit identity implies that we can realize a logical CNOT gate if we have the capability of projectively measuring the operators $\overline{X}\otimes\overline{X}$ and $\overline{Z}\otimes\overline{Z}$ of two qubits encoded in different sheets. The capability to prepare a sheet in $|\overline{0}\rangle$ and the measurement $M_{\overline{X}}$ was discussed before. The realization of such joint measurement, say, $\overline{X}\otimes\overline{X}$, is possible by

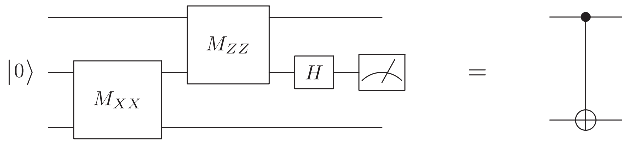

FIG. 10. A CNOT gate via two-qubit quantum measurements. Here M_{XX} measures the operator $X\otimes X$, etc. The ancilla qubit in the middle is discarded after the measurement disentangles it from the other two input qubits. Each measurement has equal probability for outcome ±1 and Pauli corrections [not shown, see Eq. (9)] depending on these measurement outcomes are done on the output target qubit.

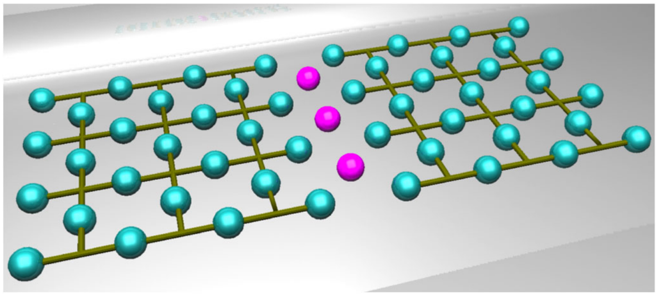

FIG. 11 (color online). Two sheets (outer) are merged at their rough boundary by placing a row of (center) ancilla qubits in the $|0\rangle$ state at their boundary and measuring the parity checks of the entire sheet. For a similar smooth merge, the ancillary qubits in between the two sheets are prepared in the $|+\rangle$ state; see the INT and C sheets in Fig. 12. From Horsman *et al.*, 2012.

temporarily merging the two sheets, realizing the measurement, and then splitting the sheets as follows. Consider two sheets laid out as in Fig. 11 where a row of ancillary qubits is prepared in $|0\rangle$ between the sheets. We realize a rough merge between the sheets by including the parity checks, and plaquette, and star operators at this boundary. If the parity check measurements are perfect, the new weight-4 plaquette Z checks have $+1$ eigenvalue as the ancilla qubits are prepared in $|0\rangle$. The four new star boundary checks have random ± 1 eigenvalues subject to the constraint that the product of these boundary checks equals the product of $\overline{X}$s of the two sheets. Hence a perfect measurement would allow us to perform a $\overline{X} \otimes \overline{X}$ measurement. As the parity check measurements are imperfect, one needs to repeat the procedure in the usual way to reliably infer the sign of $\overline{X} \otimes \overline{X}$.

We are however not yet done as we want to realize a projective $\overline{X} \otimes \overline{X}$ measurement on the qubits encoded in two separate sheets. This means that we should split the two sheets again: we can do this by reversing the merge operation and measure the ancillary qubits in the Z basis and stop measuring the four boundary X checks. Again, if the parity check measurements are perfect, the eigenvalues of the plaquette Z checks at the boundary of both sheets will take random values, but both are correlated with the outcome of the Z measurement on the ancillary qubits. Hence the individual $\overline{X}$ eigenvalues of the separate sheets may be randomized, but they are correlated so that $\overline{X} \otimes \overline{X}$ remains fixed. Similarly, a smooth merging and splitting (as between qubits C and INT in Fig. 12) with the ancillary qubits prepared and measured in the X basis accomplishes a $\overline{Z} \otimes \overline{Z}$ measurement.

The procedure for a CNOT gate in Fig. 12 then consists of first a preparation of the INT qubit in $|\overline{0}\rangle$, then a rough merge and split of qubits T and INT followed by a smooth merge and split between qubits INT and C and finally an $M_{\overline{X}}$ measurement of qubit INT.

4. Topological qubits and CNOT gate via braiding

A different way of encoding multiple qubits and realizing a CNOT gate was first proposed by Raussendorf, Harrington, and Goyal (2007) and Bombin and Martin-Delgado (2009). In this method one considers a single sheet for the whole computation in which holes are made which encode logical qubits.

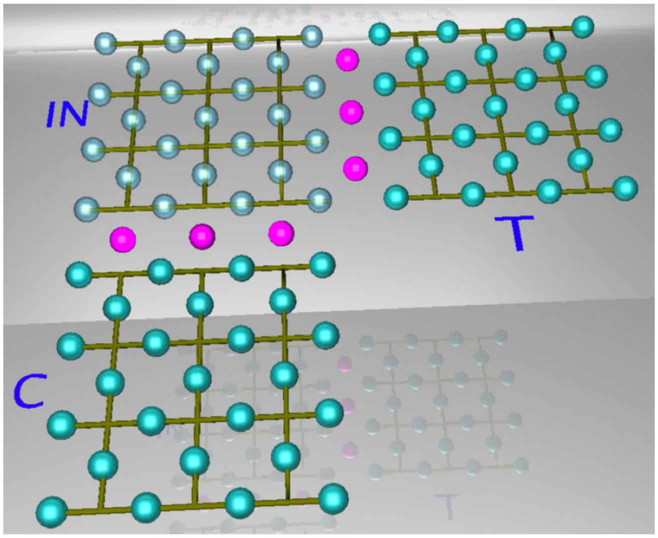

FIG. 12 (color online). Using an ancilla (INT) qubit sheet we can perform a CNOT gate between the control (C) and target (T) sheets by a sequence of mergings and splittings between the sheets. From Horsman *et al.*, 2012.

By moving holes around, or "deforming the stabilizer," one can execute a CNOT gate. This method is also the one that is analyzed by Fowler, Mariantoni *et al.* (2012) with the goal of giving a detailed overview of the procedures and practical space-time overhead. One possible disadvantage of this method is that it has an additional qubit overhead. A distance-3 smooth hole qubit (see the description below) costs many more than 13 physical qubits. A detailed comparative overhead analysis has not yet been performed between the separate sheet layout + lattice surgery scheme and this scheme.

In order to see how to encode multiple qubits, we start with a simple square sheet with all smooth boundaries which encodes no qubits; see Fig. 13(a).[16] To encode qubits one makes a hole in the lattice; that is, one removes some checks from the stabilizer $\mathcal{S}$. This is a change in topology which affects the code space dimension. In stabilizer terms: when we remove one plaquette, say, B_{p_*} for some p_* from the stabilizer $\mathcal{S}$, then B_{p_*} is no longer an element in $\mathcal{S}$ but still commutes with $\mathcal{S}$; therefore B_{p_*} is a logical operator. The matching logical operator which anticommutes with it starts at the hole and goes to the boundary. This encoded qubit has poor distance, namely, $d = 4$, as B_{p*} is of weight 4. We can modify this procedure in two ways such that logical qubits have a large distance and its logical operators do not relate to the boundary. The particular choice of logical qubits allows one to execute a CNOT gate by moving holes.

To get a logical qubit with large distance we simply make a bigger hole. We remove all, say, k^2, plaquette operators in a block [and all $(k-1)^2$ star operators acting in the interior of this block] and modify the star operators at the boundary to be of weight 3, no longer acting on the qubits in the interior; see

[16] On an $L \times L$ lattice there are $2L(L+1)$ qubits, $L^2 + (L+1)^2$ stabilizer checks, and one linear dependency between the star operators, and hence zero encoded qubits.

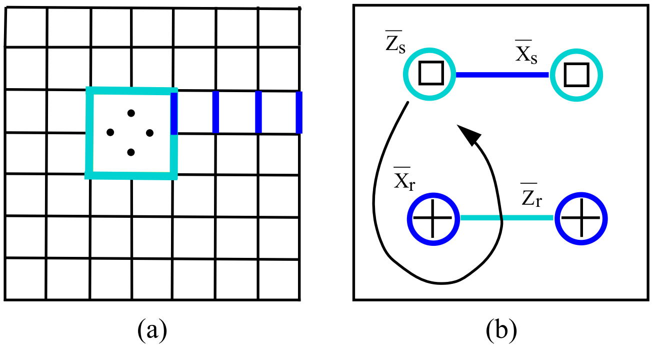

FIG. 13 (color online). (a) A smooth hole is created by removing a block of plaquette operators and the star operators acting on qubits in the interior of the block. The Z loop around the hole is $\bar{Z}$ while $\bar{X}$ is an X string to the boundary. The qubits inside the hole (four in the picture) are decoupled from the lattice. (b) Two smooth holes can make one smooth qubit and two rough holes can make one rough qubit so that moving a smooth hole around a rough hole realizes a CNOT gate.

Fig. 13(a). The qubits in the interior of the block are now decoupled from the code qubits. The procedure creates one qubit with $\overline{Z}$ equal to any Z loop around the hole. The $\overline{X}$ operator is an X string which starts at the boundary and ends at the hole. Clearly, the distance is the minimum of the perimeter of the hole and the distance to the boundary. We call this a smooth hole as the hole boundary is smooth. Of course, we could do an identical procedure on the star operators, removing a cluster of stars and a smaller subset of plaquette operators and adapting the plaquette operators at the boundary. Such a qubit will be called a rough hole and its $\overline{X}$ operator is an X string around the hole (a string on the dual lattice) and $\overline{Z}$ is a Z string to the boundary.

In order to be independent of the boundary, we use two smooth holes to define one smooth or *primal* qubit and use two rough holes to define one rough or *dual* qubit as follows. Consider two smooth holes 1,2 and define a new smooth qubit as $|\overline{0}\rangle_s = |\overline{0},\overline{0}\rangle_{1,2}$ and $|\overline{1}\rangle_s = |\overline{1},\overline{1}\rangle_{1,2}$. For this smooth qubit s we have $\overline{Z}_s = \overline{Z}_i$, $i = 1, 2$ (we can deform $\overline{Z}_1$ into $\overline{Z}_2$ by plaquette operators) and $\overline{X}_s = \overline{X}_1\overline{X}_2$ which we can deform to an X string that connects the two holes; see Fig. 13(b). The distance of this smooth qubit is the minimum of the distance between the holes and the perimeter of one of the holes (assuming the boundary is sufficiently far away). Similarly, we can create a rough qubit by taking two rough holes and defining

$$|\overline{0}\rangle_r = \frac{1}{\sqrt{2}}(|\overline{0},\overline{0}\rangle_{3,4} + |\overline{1},\overline{1}\rangle_{3,4}),$$

$$|\overline{1}\rangle_r = \frac{1}{\sqrt{2}}(|\overline{0},\overline{1}\rangle_{3,4} + |\overline{1},\overline{0}\rangle_{3,4}).$$

With this choice $\overline{X}_r$ is the loop $\overline{X}_3$ (or equivalently $\overline{X}_4$) while $\overline{Z}_r = \overline{Z}_1\overline{Z}_2$ is equivalent to the Z string connecting the holes.

Imagine moving one smooth hole around a rough hole as in Fig. 13(b). After the move, the X string connecting the smooth holes will additionally go around the rough hole enacting the transformation $\overline{X}_s \to \overline{X}_r \otimes \overline{X}_s$. This can be understood by noting that an $\overline{X}$ string with some end points a and b that loops around a rough hole is equivalent (modulo stabilizer operators) to an $\overline{X}$ loop around the rough hole disconnected from a direct $\overline{X}$ string between the end points a and b. Similarly, the Z string $\overline{Z}_r$ connecting the rough holes will, after the move, wind around the smooth hole, leading to the transformation $\overline{Z}_r \to \overline{Z}_s \otimes \overline{Z}_r$. The loops $\overline{Z}_s$ and $\overline{X}_r$ are not changed by the move. This action precisely corresponds to the action of a CNOT gate with a smooth qubit as control and a rough qubit as target.[17]

The ability to perform a CNOT gate with a smooth qubit as control and a rough qubit as target qubit seems limited as all such gates commute. However, one can use the one-bit teleportation circuits in Fig. 5 to convert a smooth qubit into a rough qubit and a rough qubit into a smooth qubit, using only CNOT gates with smooth qubits as controls. We have already shown how to realize the other components in the one-bit teleportation circuit such as $M_{\overline{X}}$ and $M_{\overline{Z}}$. Thus by composing these circuits we can execute a CNOT gate between smooth qubits alone (or rough qubits alone).

How is the braiding done using elementary gate operations? The advantage of realizing topological gates in stabilizer codes (as opposed to braiding of Majorana fermions or non-Abelian anyons in quantum Hall systems) is that braiding can be realized by changing where we measure the parity checks, or deforming the code. For example, one can enlarge the hole in Fig. 13 to include, say, two more plaquettes and three more qubits in the interior. We stop measuring those two plaquette checks and the star checks in the interior, modify the star boundary measurements, and measure the qubits in the interior in the X basis. The modified weight-3 boundary checks will have random ±1 eigenvalues as their previous eigenstates were perfectly entangled with the qubits in the interior. This corresponds to a high Z error rate around the modified boundary. By repeating the measurement to increase the confidence in their outcome one can correct these Z errors, but of course we may partially complete a $\overline{Z}$ loop this way. The protection against a full $\overline{Z}$ loop around the hole is thus provided by the part of the hole boundary which remains fixed.

This implies that the hole can safely be moved and braided in the following *caterpillar* manner. One first enlarges the hole (while keeping its "back end" fixed, providing the protection) so that it reaches its new position (the caterpillar stretches out the front part of its body to a new position). In terms of parity check measurements it means that from one time step to the next, one switches from measuring the small-hole to measuring the large-hole parity checks. Because of this extension errors will occur along the path over which the hole is moved and if error correction is noisy we should not act immediately to infer the new Pauli frame, but repeat the new check measurements to make this new frame more robust. Then, as a last step, we shrink the hole to its new position and corroborate the new measurement record by repetition

[17] The action of the CNOT gate in the Heisenberg representation is $X_c \otimes I_t \to X_c \otimes X_t$, $I_c \otimes X_t \to I_c \otimes X_t$, $Z_c \otimes I_t \to Z_c \otimes I_t$, and $I_c \otimes Z_t \to Z_c \otimes Z_t$ where X_c (X_t) stands for Pauli X on control qubit c (target qubit t).

(the caterpillar brings its rear end next to its front end again). Figures 19–23 of Fowler, Mariantoni *et al.* (2012) depict the enlargement of the hole and its subsequent shrinkage and its effect on the logical operators.

Alternatively, one can move the hole by a sequence of small translations, so that the hole never becomes large. The speed at which the hole can then be safely moved is determined by the time it takes to establish a new Pauli frame (eliminate errors) after a small move. Details of hole moving schemes are discussed in, e.g., Fowler, Stephens, and Groszkowski (2009) and Fowler, Mariantoni *et al.* (2012).

C. Different 2D code constructions

In this section we discuss a few 2D quantum codes which are variations of the surface code. These codes may have advantages over the surface code depending on physical hardware constraints. These codes are two competitive examples of 2D subsystem codes, as well as a surface code using harmonic oscillators instead of qubits.

1. Bacon-Shor code

An interesting family of subsystem codes are the Bacon-Shor codes (Bacon, 2006). For the $[[m^2,1,m]]$ Bacon-Shor code the qubits are laid out in a 2D $m \times m$ square array; see Figs. 2 and 14. The stabilizer parity checks are the double-Z-column operators $\mathbf{Z}_{\|,i}$ for columns $i=1,\ldots,m-1$ and double-X-row operators $\mathbf{X}_{=,j}$ for rows $j=1,\ldots,m-1$.

It is also possible to work with asymmetric Bacon-Shor codes with qubits in an $n \times m$ array. Asymmetric codes can have better performance when, say, Z errors are more likely than X errors (when $T_2 \ll T_1$); see Brooks and Preskill (2013). The gauge group $\mathcal{G}$ (see Sec. II.C) is generated by weight-2 vertical XX links and horizontal ZZ links and contains the parity checks. The bare logical operators (which commute with $\mathcal{G}$ but are not in $\mathcal{S}$) are the single Z column $\overline{Z}$ and a single X row $\overline{X}$.

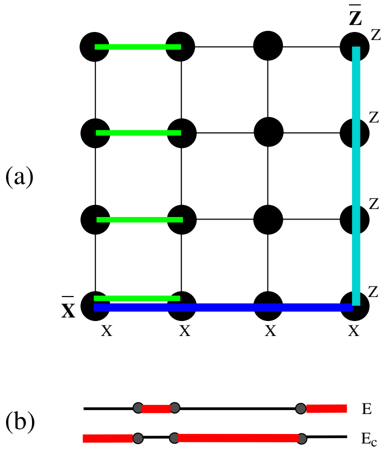

FIG. 14 (color online). (a) [[16,1,4]] Bacon-Shor code with $\bar{X}$, a row of X's, and $\bar{Z}$, a column of Z's. The stabilizer generators are double columns of Z's, $\mathbf{Z}_{\|,i}$ (one is depicted) and double rows of X's, $\mathbf{X}_{=,j}$. (b) Decoding for X errors (or Z errors in the orthogonal direction). Black dots denote the places where the double-column parity checks $\mathbf{Z}_{\|,i}$ have eigenvalue -1 (defects). The X error string E has X errors in the fattened region and no errors elsewhere and E_c is its complement. Clearly the string E has lower weight than E_c and is chosen as the likely error.

Consider the correction of X errors sprinkled on the lattice, assuming for the moment that the parity check measurement of $\mathbf{Z}_{\|,i}$ is noise free. For each column we note that an even number of X errors is a product of the vertical XX gauge operators and therefore does not affect the state of the logical qubit. This means that per column only the parity of the number of X errors is relevant. The double-column operator $\mathbf{Z}_{\|,i}$ determines whether this parity flips from column i to column $i+1$. The interpretation of the eigenvalues of $\mathbf{Z}_{\|,i}$ is then the same as for a 1D repetition code (or 1D Ising model) with parity checks $Z_i Z_{i+1}$. Double columns where $\mathbf{Z}_{\|,i} \equiv Z_i Z_{i+1} = -1$ are defects marking the end points of X strings (domain walls in the 1D Ising model). Minimum-weight decoding is very simple as it corresponds to choosing the minimum weight one between two possible X-error strings: E or the complement string E_c which both have the faulty double-column defects as end points; see Fig. 14(b). The code can thus correct all errors of weight at most $\lfloor m/2 \rfloor$ for odd m. Higher-weight errors can also be corrected as long they induce a low density of defects on the boundary. Note however that the number of syndrome bits scales as m, whereas the number of errors scales with m^2. This means that in the limit $m \to \infty$ the noise-free pseudothreshold $p_c(m) \to 0$ as the fraction of uncorrectable errors will grow with m. So, how do we choose m in order to minimize the logical error rate $\overline{p}(p,m)$? Napp and Preskill (2013) found that the optimally sized Bacon-Shor code for equal X and Z error rates p is given by $m=(\ln 2)/4p$ and for that optimal choice they can bound the logical X (or Z) error rate as $\overline{p}(p) \lesssim \exp(-0.06/p)$.

How does one acquire the nonlocal parity check values? One can either measure the XX and ZZ gauge operators and use this information to get the eigenvalues of $\mathbf{X}_{=,i}$ and $\mathbf{Z}_{\|,j}$, or one can measure the parity checks directly. The first method has the advantage of being fully local: the ancilla qubits for measuring XX and ZZ can be placed in between the data qubits; see Fig. 15(a). In the second method we prepare an

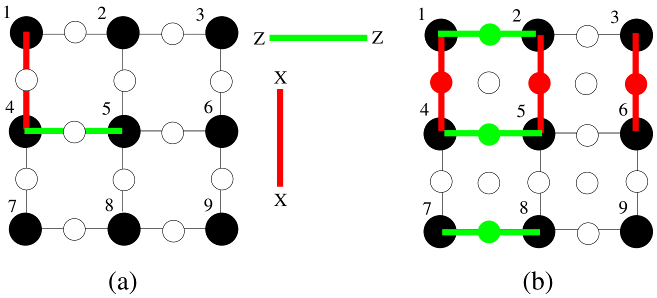

FIG. 15 (color online). (a) In order to measure the XX and ZZ operators one can place ancilla qubits (open dots) in between the data qubits. Such an ancilla qubit interacts with the two adjacent data qubits to collect the syndrome. (b) Alternatively, to measure $\mathbf{Z}_{\|,i}$ one can prepare a three-qubit entangled cat state $(1/\sqrt{2})(|000\rangle+|111\rangle)$ (vertical line of dots) which interacts locally with the adjacent system qubits. $\mathbf{X}_{=,1}$ could be measured by preparing a cat state for the ancilla qubits placed at, say, the horizontal lines of dots. The ancilla qubits at the open dots can be used to prepare the cat states.

m-qubit cat state (Shor error correction); see, e.g., Brooks and Preskill (2013). We could measure $\mathbf{Z}_{\parallel,1}$ using the circuit in Fig. 1(a) with a single ancilla qubit in the $|+\rangle$ state and controlled-phase gates (CZ). However, a single X error on the ancilla qubit can feed back to the code qubits and cause multiple Z errors, making the procedure non-fault-tolerant. In addition, the interaction between the ancilla qubit and the code qubits is nonlocal. Instead, we encode the ancilla qubit $|+\rangle$ using the repetition code, i.e., we prepare the m-qubit cat state $(1/\sqrt{2})(|00\cdots0\rangle+|11\cdots1\rangle)$ such that a CZ gate acts between one cat qubit and one code qubit. The m-qubit cat state, which itself is stabilized by Z_iZ_{i+1} and $X_1\cdots X_m$, can be made by preparing $|+\rangle^{\otimes m}$ and measuring Z_iZ_{i+1} using local ancilla qubits. The Z_iZ_{i+1} eigenvalues are recorded to provide the Pauli frame. Brooks and Preskill (2013) have given further details of this scheme, including estimates of the noise threshold for asymmetric Bacon-Shor codes which shows that the Bacon-Shor codes may be competitive, depending on further detailed numerical analysis, with the 2D surface code.

Consider now the first method of directly measuring XX and ZZ: what happens when the local XX and ZZ checks are measured inaccurately? The good news is that this causes only local errors on the system qubits. The bad news is that if the measurement outcome of, say, XX has some probability of error q, then the error probability for a nonlocal stabilizer check $\mathbf{X}_{=,i}$ will approximately be mq. This is a disadvantage of the Bacon-Shor code. Researchers (Aliferis and Cross, 2007; Brooks and Preskill, 2013) have sought to improve the fault tolerance of the parity check measurements by replacing the preparation of simple single-qubit ancillas by fault-tolerant ones (methods by Steane and Knill). Aliferis and Cross (2007) numerically obtained a best noise threshold of $p_c\approx 0.02\%$ for the concatenated [[25,1,5]] code. Napp and Preskill (2013) considered an alternative way of making the syndrome more robust, namely, by simple repetition of the XX and ZZ measurements and a collective processing of the information (as is done for the surface code). We view the effect of repetition as extending the 1D line of defects to a 2D lattice of defects, as in Fig. 8, so that minimum-weight decoding corresponds to finding a minimum-weight matching of defect end points. The error rate for vertical (black) links representing the parity check errors scales with m while the error rate for horizontal links (when one column has an even and the other column has an odd number of errors) scales, for low p, also with m.

Napp and Preskill (2013) estimated that the optimal size for the Bacon-Shor code is then $m\approx 0.014/p$ and that for this choice, the logical error rate $\overline{p}(p)\lesssim\exp(-0.0068/p)$. Hence for an error rate of $p=5\times10^{-4}$, we can choose $m=28$ giving a logical X (or Z) error rate of $\overline{p}\approx1.25\times10^{-6}$. This does not compare favorably with the logical error rate for the surface code with $L=28$, which, using the empirical formula $\overline{p}\approx0.03(p/p_c)^{L/2}$ for even L in Fowler, Mariantoni *et al.* (2012), is much lower than 10^{-6}.

2. Surface code with harmonic oscillators

In this section we discuss whether it is possible to encode quantum information in a 2D lattice of coupled harmonic oscillators. We start by defining a continuous-variable version of the surface code which encodes an oscillator in a 2D array of oscillators. Then we discuss how to modify this construction so that we concatenate the qubit-into-oscillator code described in Sec. II.D.3 with the regular surface code and express the checks of the surface code in terms of operators on the local oscillators. This scheme may be of interest if the qubit encoded in the oscillator has a sufficiently low error rate which we want to improve upon by further surface code encoding. For example, we can imagine a set of 2D or 3D microwave cavities each of which by itself encodes a qubit which we couple in a 2D array.

It is possible to define a qudit stabilizer surface code [see, e.g., Bullock and Brennen (2007)], where the elementary constituents on the edges of the lattice are qudits with internal dimension d and the code encodes one or several qudits. Here we focus on the special case when we take $d\to\infty$ and each edge is represented by a harmonic oscillator with conjugate variables $\hat{p}$, $\hat{q}$. The goal of such a continuous-variable surface code is to encode a nonlocal oscillator into a 2D array of oscillators such that the code states are protected against local shifts in $\hat{p}$ and $\hat{q}$. In addition, one can imagine using continuous-variable graph states to prepare such encoded states and observe anyonic statistics (Zhang *et al.*, 2008).

To get a surface code, we replace Pauli X by $X(b)=\exp(2\pi ib\hat{p})$ and Pauli Z by $Z(a)=\exp(2\pi ia\hat{q})$ with real parameters such that $Z^\dagger(a)=Z^{-1}(a)=Z(-a)$, etc. It follows that for any two oscillators 1 and 2, we have

$$\forall\ a,b,\qquad [Z_1(a)Z_2(-a),X_1(b)X_2(b)]=0. \tag{10}$$

In the bulk of the surface code lattice, a plaquette operator centered at site u can be chosen as $B_u(a)=Z_{u-\hat{x}}(a)Z_{u+\hat{x}}(-a)Z_{u-\hat{y}}(a)Z_{u+\hat{y}}(-a)$ while a star operator at site s is equal to $A_s(b)=X_{s-\hat{x}}(-b)X_{s+\hat{x}}(b)X_{s-\hat{y}}(b)X_{s+\hat{y}}(-b)$; see Fig. 16.

Here $Z_{u-\hat{x}}(a)=\exp(2\pi ia\hat{q}_{u-\hat{x}})$, where $\hat{q}_{u-\hat{x}}$ is the position variable of the oscillator at site $u-\hat{x}$ ($\hat{x}$ and $\hat{y}$ are orthogonal unit vectors on the lattice). One can observe from Fig. 16 and Eq. (10) that $B_u(a)$ and $B_u^\dagger(a)$ commute with $A_s(b)$ and $A_s^\dagger(b)$ for all a, b in the bulk and at the boundary.

We define Hermitian operators with real eigenvalues in the interval $[-1,1]$ as

$$\begin{aligned}H_u(a)&=\tfrac{1}{2}[B_u(a)+B_u^\dagger(a)]\\&=\cos[2\pi a(q_{u-\hat{x}}-q_{u+\hat{x}}+q_{u-\hat{y}}-q_{u+\hat{y}})]\end{aligned}$$

and

$$\begin{aligned}H_s(b)&=\tfrac{1}{2}[A_s(b)+A_s^\dagger(b)]\\&=\cos[2\pi b(-p_{s-\hat{x}}+p_{s+\hat{x}}+p_{s-\hat{y}}-p_{s+\hat{y}})].\end{aligned}$$

We now define the code space as the $+1$ eigenspace of all $H_p(a)$ and $H_s(b)$ for all a and b. It follows that a state in the code space is a delta function in the positions of the oscillators around all plaquettes u; that is, $\delta(q_{u-\hat{x}}-q_{u+\hat{x}}+q_{u-\hat{y}}-q_{u+\hat{y}})$ for every u, while it is a delta function in the momenta of the oscillators $\delta(-p_{s-\hat{x}}+p_{s+\hat{x}}+p_{s-\hat{y}}-p_{s+\hat{y}})$ located at all stars s.

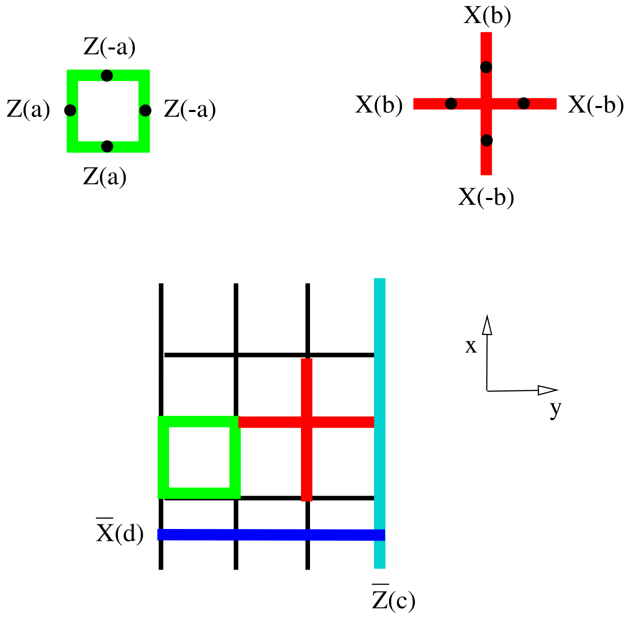

FIG. 16 (color online). Small example of the oscillator surface code where oscillators on the edges are locally coupled with plaquette and star operators so as to define an encoded oscillator with logical, nonlocal displacements $\bar{X}(d)$ and $\bar{Z}(c)$. The realization of the four-oscillator interaction will require strong four-mode squeezing in either position (at plaquettes) or momenta (at stars).

One can compare such a highly entangled code state with its simpler cousin, the two-mode Einstein-Podolsky-Rosen (EPR) state. In the two-mode case we have two commuting operators, namely, $Z_1(a)Z_2(-a)$ and $X_1(b)X_1(b)$ on oscillators 1 and 2. The single state which is the $+1$ eigenstate of $\cos[2\pi a(q_1 - q_2)]$ and $\cos[2\pi b(p_1 + p_2)]$ for all a and b is the two-mode infinitely squeezed EPR state $\delta(p_1 + p_2)\delta(q_1 - q_2)$.

In contrast to the two-mode case, the oscillator surface code space is not one dimensional, but infinite dimensional as it encodes a nonlocal oscillator. The operators $\overline{Z}(c) = \exp(2\pi i c\sum_{i\in\gamma_1}\hat{q}_i)$ where the path γ_1 runs straight from north to south commute with all $H_p(a)$, $H_s(b)$; see Fig. 16. Similarly, we have $\overline{X}(d) = \exp(2\pi i d\sum_{j\in\gamma_2}\hat{p}_j)$, where γ_2 runs straight from east to west. As $\overline{Z}(c)\overline{X}(d) = e^{-i(2\pi c)(2\pi d)}\overline{X}(d)\overline{Z}(c)$, we can interpret $\overline{Z}(c)$ and $\overline{X}(d)$ as phase-space displacements of the encoded oscillator with logical position and momentum $\overline{p} = \sum_{i\in\gamma_2}p_i$ and $\overline{q} = \sum_{i\in\gamma_1}q_i$. We deform these nonunique logical operators to follow deformed paths, e.g., we multiply $\overline{Z}(c)$ by $B_p(c)$ plaquettes [note that if we multiply by $B_p(c')$ with $c' \neq c$ we get an operator with the union of supports].

How would one use such a code to encode quantum information and what protection would it offer? As its qubit incarnation, a sufficiently low density of independent errors on the lattice can be corrected. For the array of oscillators or bosonic modes, one expects that each oscillator i independently suffers from small dephasing, photon loss, etc., that is, errors which can be expanded into small shifts $Z_i(e)X_i(e')$ with $|e|,|e'| \ll 1$; see Sec. II.D.3. This means that the likelihood for logical errors of the form $\overline{Z}(c)\overline{X}(d)$ for small c and d will be high, which relates of course to the fact that we are attempting to encode a continuous variable rather than a discrete amount of information.

However, one can imagine using only a two-dimensional subspace, in particular, the code words of the GKP qubit-into-oscillator code; see Sec. II.D.3 for each oscillator in the array. One can also view this as a concatenation of the GKP code and the surface code in which we express the surface code plaquette and star operators in terms of the operators on the elementary oscillators in the array. One could prepare the encoded states $|\overline{0}\rangle, |\overline{1}\rangle, |\overline{+}\rangle, |\overline{-}\rangle$ of the surface code by preparing each local oscillator in the qubit-into-oscillator logical states $|0\rangle, |1\rangle, |+\rangle, |-\rangle$ and subsequently projecting onto the perfectly correlated momenta and position subspace. For example, the state $|0\rangle$ of a local oscillator i is an eigenstate of $S_q(\alpha) = e^{2\pi i\hat{q}_i/\alpha}$, $S_p = e^{-2i\hat{p}_i\alpha}$, and the local $\overline{Z}_i = e^{i\pi\hat{q}_i/\alpha}$. This implies that after projecting onto the space with $H_p(a) = 1$, $H_s(b) = 1$ for all a, b, it will be an eigenstate of $\overline{Z}(1/(2\alpha)) = e^{i\pi\sum_{i\in\gamma_1}\hat{q}_i/\alpha}$, i.e., the encoded $|\overline{0}\rangle$.

3. Subsystem surface code

It is clear that codes for which one has to measure high-weight parity checks are disadvantageous: it requires that an ancilla qubit couples to many data qubits through noisy gates, leading to a large error rate on the syndrome and in turn to a lower noise threshold. Can we have a 2D stabilizer code which has weight-2 or weight-3 parity checks? The answer is no: one can prove that 2D qubit codes defined as eigenspaces of at most 3-local (involving at most three qubits), mutually commuting terms are trivial [with $O(1)$ distance] as quantum codes (Aharonov and Eldar, 2011). In addition any stabilizer code with only weight-2 checks can be shown to be trivial.

Such results do not hold for subsystem codes: the Bacon-Shor code shows that it is possible to have only two-qubit noncommuting parity checks. However, the Bacon-Shor code is not a topological subsystem code as the stabilizer checks are nonlocal on the 2D lattice and its asymptotic noise threshold is vanishing. Several topological subsystem codes have been proposed in which weight-2 parity checks are measured (Bombin, 2010b), but the asymptotic noise threshold for such codes is typically quite a bit lower than for the surface code; see, e.g., Suchara, Bravyi, and Terhal (2011). The question is whether it is possible to find a 2D subsystem code with checks of weight less than 4 that has a noise threshold that is similar to the surface code. Such a subsystem code may be of high interest if it is considerably easier to realize a weight-3 check in the physical hardware than a weight-4 check.

Bravyi *et al.* (2013) proposed a topological subsystem code—a subsystem surface code—in which the noncommuting parity checks are of weight 3 and the stabilizer generators are of weight 6; see Fig. 17. More precisely, the gauge group $\mathcal{G}$ is generated by the triangle operators XXX and ZZZ, including cutoff weight-2 operators at the boundary. The stabilizer group $\mathcal{S} = \mathcal{G}\cap\mathcal{C}(\mathcal{G})$ is generated by weight-6 plaquette operators (at the boundary → weight-2 operators). By measuring, say, the Z triangles we can deduce the eigenvalues of the Z plaquettes which are used to do error correction.

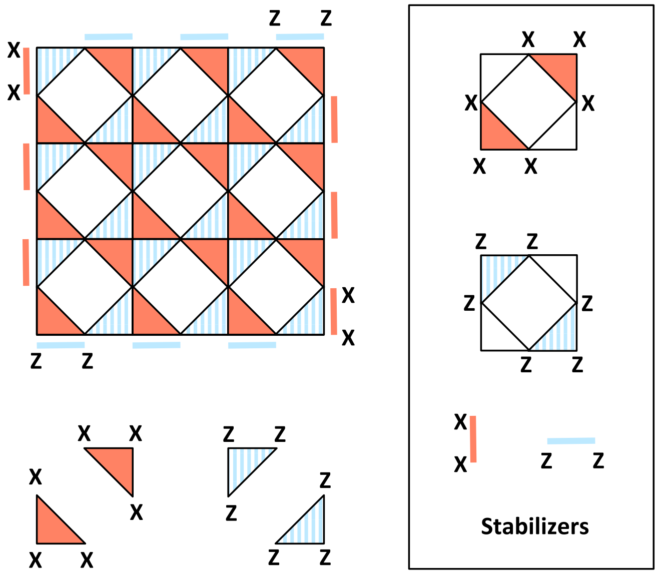

FIG. 17 (color online). Subsystem surface code on a lattice of size $L \times L$ with L^2 square plaquettes (depicted is $L = 3$). The qubits exist on the edges and vertices of the plaquettes and are acted upon by weight-3 X- and Z-triangle operators (which are modified to become weight-2 operators at the boundary). The stabilizer checks are weight 6 except at the boundary. From Bravyi *et al.*, 2013.

For an $L \times L$ lattice one has a total of $3L^2 + 4L + 1$ qubits and $2L^2 + 4L$ independent stabilizer generators, which give $L^2 + 1$ logical qubits. One of these qubits is the logical qubit whose $\overline{Z}$ and $\overline{X}$ commute with all Z and X triangles. As in the surface code, a vertical Z line through $2L$ qubits can realize $\overline{Z}$ while a horizontal X line realizes $\overline{X}$. The logical operators for the L^2 gauge qubits, one for each plaquette, are pairs of triangle operators on a plaquette generating the group $\mathcal{G}$. One can multiply, say, the vertical Z line by Z triangles to become a $\overline{Z}$ which acts only on L qubits: Bravyi *et al.* (2013) indeed proved that the distance of the code is L. Note that such weight-L $\overline{Z}$ acts on the logical qubit and the irrelevant gauge qubits.

For a code with distance $L = 3$ one thus needs 41 elementary qubits, substantially more than for the surface code. Multiple qubits can be encoded in this subsystem code by making holes as for the surface code. One can expect that braiding and lattice surgery methods for this code can be established in the same way as for the surface code. The interesting feature of this code is its relatively high noise threshold obtained by reduced-weight parity checks (at the price of a bit more overhead). Decoding of stabilizer syndrome information is done by interpreting the syndrome as defects on a virtual lattice which can be processed, similar to the surface code, by minimum-weight matching of defects or by RG decoding. For noise-free perfect error correction and independent X,Z noise, they report a maximum threshold of $p_c \approx 7\%$ (compare with 11% for the surface code). For noisy error correction the threshold depends on how single errors with probability p in the parity check circuit affect the error rate on the virtual lattice. Modeling this effective noise rate on the virtual lattice, they found a noise threshold of $p_c \approx 0.6\%$.

It is not surprising that decoding for the subsystem surface codes can be done using the decoding method for the surface code. Bombin, Duclos-Cianci, and Poulin (2012) proved that any 2D topological subsystem or stabilizer code can be locally mapped onto copies of the toric code. The upshot is that for any such code one can find, after removing some errors by local correction, a virtual lattice with toric code parity checks and an underlying effective error model. An example of another 2D topological subsystem code that can be analyzed this way is a concatenation of the [[4,2,2]] code with the surface code. If we use the [[4,2,2]] code as the subsystem code then the concatenated code has weight-2 and weight-8 check operators. The scheme may be of interest if the weight-2 checks can be measured fast and with high accuracy.

D. Decoding and direct parity check measurements

One can ask whether quantum error correction for the surface or other topological codes in $D = 2$ or higher is possible by purely local means. The dissipative correction procedure of stabilizer pumping described in Sec. II.F is an example of a purely local error-correction mechanism which does not use any communication. We can consider what such a mechanism does if we apply it to the toric code discussed in Sec. III.A. Imagine that a single X error occurs. For the toric code, such an X error is heralded by two odd-parity Z checks, two defects. The error can be corrected by applying any small X string that terminates at these defects. However, the mechanism described in Sec. II.F that applies an X correction at a fixed qubit for every defect will have the effect of moving the single X error around or it will create more X errors. It will utterly fail at removing errors. Dengis, König, and Pastawski (2014) showed that it is possible to use such a very local, but assumed to be perfect, dissipative decoder to efficiently encode a quantum state in a toric code quantum memory. In this decoder the corrections are chosen such that the errors are pushed in a certain direction, where they can mutually annihilate and a simple input state determines what state is encoded by the dissipative evolution. In such an extremely local form of dissipative error correction there is absolutely no guarantee that the corrections that are applied are minimum-weight corrections.

It is clear that an engineered dissipative dynamics for quantum error correction should at least correlate the parities of neighboring checks before applying any corrections. As an error string is heralded only by its two end-point defects, longer error strings require correlation of the parity checks in a larger neighborhood, and hence more communication and delay in order to annihilate the error string. Said differently, one needs to dissipatively engineer the action of a nonlocal minimum-weight matching or RG decoder.

One can in fact view the classical nonlocal minimum-weight matching decoder as a source of computational power that jump-starts our quantum memory. Note that the RG decoder is nonlocal (even allowing for parallel processing of clusters on the lattice by local automata) as the maximum number of recursions r scales as $O(\log L)$ leading to a maximum cluster size proportional to the linear size of the lattice. The lowest levels of the RG decoder are of course local and will provide some measure of protection by local means.

Harrington (2004) devised a scheme for doing purely local quantum error correction for the surface code by a 2D cellular automaton, assuming independent local errors; the mere existence of such a scheme is nontrivial as it has to deal with noise and communication delays. Other examples of a local dissipative surface code decoder are the proposals by Fujii *et al.* (2014) and Herold *et al.* (2014), but both these decoders assume noise-free parity check measurements and noise-free classical processing in the cellular automaton.

Fowler (2015) claimed that one can do minimum-weight matching in $O(1)$ parallel time on average taking into account a finite speed of communication between processing cells. Establishing this in full rigor is an important fundamental question since it would be problematic to have a syndrome processing rate r_{proc} which fundamentally depends on L: if this were the case, there would always be a large enough L such that one runs into the backlog problem discussed in Sec. II.G.3.

The advantage of a 2D local on-chip decoder over extracting the syndrome data record to the classical world and using standard classical processing is that such an on-chip decoder (1) can potentially lead to faster decoding (and this is important to avoid a syndrome record backlog), as it uses parallelism in full with dedicated hardware, and (2) avoids a large data stream having to come out of the quantum device. It is of interest to explore whether one can build such a local cellular automaton decoder out of reliable classical, and sufficiently fast [complementary metal-oxide-semiconductor (CMOS)] logic at low [$O(10)$ mK] temperature.

One should contrast the challenge of designing a fast decoder for the 2D surface code with the local decoder for the 4D toric code (Dennis *et al.*, 2002); see the description of the 4D toric code in Sec. III.A.1. The local redundancy of the error syndrome of the 4D toric code has the following consequence. A nontrivial error syndrome will form a closed loop which is the boundary of an error cluster: this is similar to a domain of flipped spins surrounded by a domain wall in a 2D ferromagnetic Ising model. Then the idea of a local decoder, which, in contrast to the 2D case, does not require communication over lengths depending on L is as follows. One removes an error cluster by locally shrinking the length of the nontrivial error syndrome loop. Thus there is a purely locally defined “defect energy” function whose minimization leads to the shrinking of the error cluster and thus to error correction. The local decoder could, for example, be entirely realized as a quantum circuit, requiring no quantum measurement. In this local decoding quantum circuit one should also expect errors to occur at a certain rate, which means that clusters of errors on the data qubits can sometimes grow instead of shrink. For sufficiently low error rates, one may expect that more errors are locally removed rather than added, either by decoherence or by incorrect decoding, such that a logical error is exponentially (exponential in some function of the block size n) rare and the quantum information is protected.

The absence of a local cost function, which a local decoder can minimize, is a generic property of 2D stabilizer codes and is directly related to the stringlike nature of the error excitations which have observable defects only at their zero-dimensional boundary. It has been proven that all 2D stabilizer codes (Bravyi and Terhal, 2009) have stringlike logical operators and this directly ties in with the lack of self-correction for these models; see Sec. III.E.

1. Parity check measurements and their implementation

In a variety of physical systems (parity check) measurements are implemented as weak continuous measurements in time rather than a sequence of strong projective measurements. Examples of weak continuous qubit measurements are the measurement of a spin qubit in a semiconducting quantum dot through a quantum point contact (Elzerman *et al.*, 2003) and the measurement of a superconducting transmon qubit through homodyne measurement of a microwave cavity field with which it interacts.

For short time scales such continuous weak measurements suffer from inevitable shot noise as the current or voltage is carried by low numbers of quanta (electrons or photons); this noise averages out on longer times scales, revealing the signal. The shot noise thus bounds the rate at which parity information can be gathered. The effect of leakage of qubits either appears in such measurement traces as a different output signal (detection of leakage) or, in the worst case, leads to a similar, and thus untrustworthy, output signal which makes the parity check record unreliable for as long as a qubit occupies a leaked state.

The idea of realizing the surface code in superconducting circuit-QED systems using ancilla qubits for measurement, laying out a possible way to couple transmon qubits and resonators, was considered by DiVincenzo (2009). A scalable surface code architecture was proposed and a basic unit implemented by Barends *et al.* (2013) [see also Barends *et al.* (2014)]. The optimal way of using superconducting transmon qubits to realize a surface code architecture is a subject of current ongoing research; see, e.g., Ghosh, Fowler, and Geller (2012) for a direct comparison between three different architectures that differ in how transmon qubits are coupled to microwave resonators. In a transmon-qubit-based architecture it is important to show how one can deal with leakage errors since transmon qubits are weakly anharmonic multilevel systems. An important feature of a physical parity check measurement scheme is whether it allows leakage on data or ancilla qubits to be detected or whether leakage goes undetected. Ghosh and Fowler (2015) and Suchara, Cross, and Gambetta (2014) recently started to consider how to handle leakage in a surface code architecture.

In order to reduce qubit overhead and possibly make better use of the given physical interactions, e.g., cavity-atom (in cavity-QED) or cavity–superconducting qubit (in circuit-QED) interactions, it is worthwhile to consider the idea of a direct parity check measurement instead of a parity measurement that is realized with several two-qubit gates and an ancilla as in Fig. 1.

Any mechanism through which a probe pulse [modeled as a simple coherent state $|\alpha(t)\rangle$] picks up a π phase shift depending on a qubit state being $|0\rangle$ or $|1\rangle$ could function as the basis for such direct parity measurement. As long as the imprint of multiple qubits is through the addition of such phase shifts, we have $\alpha(t) \to \alpha(t)e^{i\pi P}$, where P is the parity of the qubits. Homodyne detection of such a probe pulse in time, that is, the continuous measurement of $\langle a(t) + a^{\dagger}(t)\rangle = (-1)^P 2\langle\alpha(t)\rangle$

(assuming $\alpha = \alpha^*$), could then realize a weak continuous parity measurement.

This kind of setup is natural for strong light-matter interactions in cavity QED and circuit QED, where the state of the qubit can alter the refractive index of the medium (cavity) on which a probe pulse impinges. The challenge is to obtain phase shifts as large as π and ensure that these probe pulses do not contain more information about the qubits than their parity as this would lead to additional dephasing inside the odd- or even-parity subspace.

In the cavity-QED setting, Kerckhoff *et al.* (2009) considered the realization of a continuous weak two-qubit parity measurement on two multilevel atoms each contained in a different cavity [for possible improvements on this scheme, see Nielsen (2010)]. Lalumière, Gambetta, and Blais (2010) considered a direct two-qubit parity measurement of two transmon qubits dispersively coupled to a single microwave cavity (circuit-QED setting). Similarly, DiVincenzo and Solgun (2013) and Nigg and Girvin (2013) considered direct three or more qubit parity check measurements for transmon qubits coupled to 2D or 3D microwave cavities.

Kerckhoff *et al.* (2010, 2011) developed the interesting idea of a fully autonomous quantum memory which runs with fixed, time-independent input driving fields. In this approach, it was imagined that qubits are encoded in multilevel atoms coupled to the standing electromagnetic modes of (optical) cavities. Both Z- and X-parity checks of the qubits are continuously obtained via probe pulses applied to these cavities. These probe pulses are to be subsequently processed via photonic switches to coherently perform continuous quantum error correction.

E. Topological order and self-correction

A different route toward protecting quantum information is based on passive Hamiltonian engineering. In this approach quantum information is encoded in an eigenspace, typically the ground space, of a many-body, topologically ordered Hamiltonian. There is no completely rigorous definition of topological order in the literature. At an intuitive level it means that there does not exist a local order parameter or observable which distinguishes different degenerate ground states. Such a property is immediately obtained when the ground space is the code space of a quantum error-correcting code with macroscopic (meaning scaling as some function of the system size) distance as follows.

The quantum error-correction conditions, Eq. (4), can be slightly reformulated; see Theorem 3 in Gottesman (2009): A code C can correct a set of errors $E \in \mathcal{E}$ if and only if for all states $|\overline{\psi}\rangle$ in the code space C we have

$$\langle\overline{\psi}|E^{\dagger}E|\overline{\psi}\rangle = c(E), \tag{11}$$

where the constant $c(E)$ is independent of $|\overline{\psi}\rangle$. If the set of errors $\mathcal{E}$ is a set of errors which act locally on $O(1)$ qubits not scaling with system size, such that $E^{\dagger}E$ are local observables, then this condition precisely captures the intuitive idea of topological order. Thus if we devise a physical system with a Hamiltonian such that the ground space corresponds to the code space of a code which can correct any set of local errors, such a system would be topologically ordered.

The simplest examples of such systems are D-dimensional stabilizer codes with macroscopic distances scaling with system size. For such codes we can define a many-body qubit Hamiltonian $H_{\text{topo}} = -\Delta\sum_i S_i$, where S_i is a set of (overcomplete) stabilizer generators. A consequence of topological order is that the ground-space degeneracy of the Hamiltonian H is insensitive to weak local perturbations. This feature has been rigorously proven for stabilizer codes by Bravyi, Hastings, and Michalakis (2010) under a slightly sharpened form of the quantum error-correction conditions referred to as local topological order. It has not yet been established whether subsystem stabilizer code Hamiltonians of the form $H = -\Delta\sum_i G_i$ with local generators G_i of the gauge group $\mathcal{G}$ also have an eigenspace degeneracy which is insensitive to weak perturbations.

If we store quantum information passively in a physical system described by some effective Hamiltonian H_{topo}, we assume no physical mechanism which actively removes error excitations. Rather we invoke the argument that the presence of a sufficiently large energy gap above the ground space in the Hamiltonian will exponentially [as $\exp(-\Delta/T)$] suppress error excitations at sufficiently low temperature T. Whether this is practically sufficient depends on the empirical value of Δ/T (and the uniformity of this value across a physical sample).

One may consider how to engineer a physical system such that it has the effective, e.g., four-qubit interactions of the surface code between nearby qubits in a 2D array (Kitaev, 2006). The strength of this approach is that the protection is built into the hardware instead of being imposed dynamically, negating, for example, the need for control lines for time-dependent pulses. The challenge of this approach is that it requires one-, two-, and three-qubit terms in the effective Hamiltonian to be small: the elementary qubits of the many-body system should therefore have approximately degenerate levels $|0\rangle$ and $|1\rangle$. However, in order to encode information in, say, the ground space of such Hamiltonian, one will need to lift this degeneracy to be able to address these levels. Another challenging aspect of such Hamiltonian engineering is that the desired, e.g., four-body, interactions will typically be arrived at perturbatively. This means that their strength and therefore the gap of the topologically ordered Hamiltonian compared with the temperature T may be small, leading to inevitable error excitations. Douçot and Ioffe (2012) reviewed several ideas for the topological protection of quantum information in superconducting systems, while Gladchenko *et al.* (2009) demonstrated their experimental feasibility. Another example is the proposal to realize the parity checks of the surface code through Majorana fermion tunneling between 2D arrays of superconducting islands, each supporting four Majorana bound states with fixed parity (Terhal, Hassler, and DiVincenzo, 2012).

The information stored in such passive, topologically ordered many-body systems is, at sufficiently low temperature, protected by a nonzero energy gap. Research has been devoted to the question of whether the $T = 0$ topological phase can genuinely extend to nonzero temperature $T > 0$ (Dennis *et al.*, 2002). The same question has also been approached from a dynamical perspective with the notion of a self-correcting quantum memory (Bacon, 2006) [see also the notion of thermal fragility discussed by Nussinov and Ortiz (2009)].

A self-correcting quantum memory is a quantum memory in which the accumulation of error excitations over time, which can in turn lead to logical errors, is energetically disfavored due to the presence of macroscopic energy barriers. In this approach it is assumed that the quantum system is in contact with a thermal heat bath which is a source of error excitations as well as error correction depending on the energy of the error excitations and the temperature of the bath. The difference from the active quantum error-correction approach is thus that for active quantum error correction we strive to actively engineer part of the environment which should perform the error correction (via parity check measurements). For a passive thermal memory one expects the rate of logical errors to scale empirically as an Arrhenius law as $A\exp(-E_{\rm barrier}/kT)$, where $E_{\rm barrier}$ is the height of the energy barrier and A is an entropic prefactor. In order to achieve self-correction, we want a logical qubit encoded in such quantum memory to have a coherence time $\tau(T,n)$ which grows with the size n (the elementary qubits of the memory) for some temperature $0<T<T_c$.

One can study the question of self-correction for Hamiltonians $H_{\rm topo}=-\Delta\sum_i S_i$ related to D-dimensional stabilizer (or subsystem) codes. For stabilizer codes, Pauli errors map the ground space onto excited eigenstates with energy at least 2Δ. The energy barrier associated with such a Hamiltonian is defined as the minimum energy that has to be expended in order to perform any logical operator by means of a sequence of local $O(1)$-weight Pauli errors (Bravyi and Terhal, 2009). The application of each local Pauli error maps an energy eigenstate onto a new energy eigenstate: a sequence of such operators describes a path through the energy landscape. One can consider all sequences of local errors which result in overall execution of a logical operator. The energy barrier of the logical operator is then given by the minimum over all paths of the maximum energy barrier on each path.

One important finding concerning self-correcting quantum memories is that a finite-temperature "quantum memory phase" based on macroscopic energy barriers is unlikely to exist for genuinely local 2D quantum systems. One can prove that any 2D stabilizer code has an energy barrier $E_{\rm barrier}=O(1)$: this result is obtained by showing that there always exist stringlike logical operators for a 2D stabilizer code (Bravyi and Terhal, 2009). The surface code with its stringlike logical $\overline{X}$ and $\overline{Z}$ operators that run between boundaries provides a good example of this generic behavior.

The 3D toric code on a lattice of $n=O(L^3)$ qubits, as discussed in Sec. III.A.1, has a surfacelike logical $\overline{X}$ operator [element in $H^1(T_3,\mathbb{Z}_2)$] and thus an energy barrier $E_{\rm barrier}\sim L$ for the logical $\overline{X}$. But the logical $\overline{Z}$ [element in $H_1(T_3,\mathbb{Z}_2)$] is stringlike and has an $O(1)$ energy barrier. One can view the 3D toric code as a model for storing a classical bit passively in a thermal environment (Castelnovo and Chamon, 2008).

The 4D toric code on a cubic lattice with linear dimension L has been shown to be a good example of finite-temperature topological order or a self-correcting memory with a coherence time $\tau(T<T_c,L)\sim\exp[O(L)]$; see Dennis *et al.* (2002) and Alicki *et al.* (2010). The properties of the 4D toric code that make this possible are the facts that both logical operators are surfacelike and error clusters are surrounded by closed nontrivial syndrome loops as discussed in Secs. III.A.1 and III.D.

For three-dimensional stabilizer codes that are translationally invariant and for which the number of encoded qubits does not depend on the lattice size, it has been shown that there always exist stringlike logical operators and thus the energy barrier is again $O(1)$ (Yoshida, 2011). For homological codes defined on three-dimensional manifolds this result can be understood by invoking (Poincaré) duality. For a D-dimensional manifold M the kth cohomology group $H^k(M,\mathbb{Z}_2)$ is isomorphic to the homology group $H_{D-k}(M,\mathbb{Z}_2)$ [as i-simplices are mapped to $(n-i)$-simplices on the dual lattice]. Thus in three dimensions, the presence of a surfacelike logical operator in, say, $H_2(M,\mathbb{Z}_2)$ also implies the presence of a matching stringlike logical operator in $H^2(M,\mathbb{Z}_2)\simeq H_1(M,\mathbb{Z}_2)$.

Given this duality perspective, it should be considered surprising that it is possible to construct three-dimensional stabilizer codes that have an energy barrier that scales as a function of L. Such codes have to avoid Yoshida's no-go result by either being nontranslationally invariant or encoding a number of qubits that depends on the lattice size (or both). The first example of such a 3D code was the Haah code with $E_{\rm barrier}\geq c\log L$ (Bravyi and Haah, 2011; Haah, 2011) for which all logical operators are fractal (instead of string- or surfacelike). For the Haah code the number of encoded qubits nontrivially depends on the lattice size.

Another construction is Michnicki's welded code (Michnicki, 2014) which breaks translational invariance and has an energy barrier $E_{\rm barrier}=O(L^{2/3})$ for an $n=O(L^3)$ system. For the Haah code it was shown by Bravyi and Haah (2013) that the existence of the energy barrier implies that $\tau(T,n)\sim L^{c/kT}$ as long as L is below some critical temperature-dependent length scale and a similar result holds for the welded code.

It is an open question whether topological subsystem codes in 3D behave differently from stabilizer codes in terms of their self-correcting properties. See Wootton (2012) and references therein for another overview of results in this area of research.

IV. DISCUSSION

The current qubit realizations seem perhaps awkwardly suited to constitute the elementary qubits of an error-correcting code. Most elementary qubits are realized as *nondegenerate* eigenlevels (in a higher-dimensional space), approximately described by some $H_0=-(\omega/2)Z$. The presence of H_0 immediately gives a handle on this qubit, i.e., processes which exchange energy with this qubit will drive it from $|1\rangle$ to $|0\rangle$ and vice versa (Rabi oscillations) and coupling of the qubit to other quantum degrees of freedom can be used for qubit readout. Passive time-independent interactions with other quantum systems are intentionally weak and lead to significant multiple-qubit interactions only if we supply energy in the form of time-dependent ac or dc fields meeting resonance conditions. To drive, keep, or project multiple qubits via local parity checks in a code space where they are highly entangled, active control at the elementary qubit level will thus be continuously needed, making the macroscopic coding overhead look daunting.

For such nondegenerate qubits typically all gates and preparation steps are realized in the rotating frame: the frame of reference in which the qubit state is no longer precessing around the z axis on the Bloch sphere due to the presence of H_0. Any code word $|\overline{\psi}\rangle$ is then only a fixed quantum state in this rotating frame while it is dynamically rotating under single-qubit Z rotations in the laboratory frame. As measurements are done only in the laboratory frame, it is only Z measurements that can be done directly while X measurements typically require an active rotation (e.g., Hadamard) followed by a Z measurement.

Once elementary qubits are used for times much longer than their coherence time, i.e., when they are used together in a quantum memory, the question of stability of this laboratory reference frame or the stability of the qubit frequency ω becomes important. Because of $1/f$ noise and aging of materials from which qubits are constructed, the qubit frequency ω can systematically drift over longer times and average to new values which are different from short time averages. This has two consequences: First, one needs to determine the qubit frequency periodically, for example, by taking qubits periodically offline and measuring them. In this manner one can recalibrate gates whose implementation depends on knowing the rotating frame. Second, shifts in qubit frequency also induce shifts in coherence times as these times depend on the noise power spectral density $S(\omega)$ at the qubit frequency. Such fluctuations of coherence times over longer time scales have been observed. As an example we can take the results of Metcalfe *et al.* (2007) which report that the T_1 time of a superconducting "quantronium" qubit is changing every few seconds over a range of 1.4–1.8 μs. It is clear that if elementary qubits are to be successfully used in a quantum memory, then fluctuations of the noise rate have to be such that one remains below the noise threshold of the code that is employed in the memory at all times.

We conclude by listing some issues on which we expect to see more progress from the perspective of coding theory. One question is the issue of minimizing qubit and computational overhead in a fault-tolerant computer. It is not clear that the surface code is the ideal platform for this because of its large overhead. It may be advantageous to consider architectures with nonlocal connects so that one can use a quantum LDPC code which does not make reference to spatial locality and which can escape the no-go results for low-dimensional stabilizer codes in allowing for, say, a transversal T gate. In addition, quantum LDPC codes which are not restricted to D dimensions allow for a constant encoding rate k/n. How much they can reduce overhead also depends on the numerical value of this rate which for various quantum LDPC codes has not yet been determined.

We illustrate the issue of overhead due to the nontransversality of the T gate by the following consideration. An efficient quantum algorithm on N qubits takes $\mathrm{poly}(N)$ gates, where $\mathrm{poly}(N)$ is typically not linear in N. For example, for Shor's factoring algorithm the bulk of the algorithm uses $O(N^3)$ Toffoli gates which are non-Clifford gates. If one uses ancillas to create such Toffoli gates (or T gates for that matter, which can be used to make Toffoli gates), it means that one needs at least $O(N^3) + N$ qubits. The size of the original quantum circuit in non-Clifford gates is thus converted to the number of logical qubits. As a concrete example, Fowler, Mariantoni *et al.* (2012) estimated that, in order to factor a 2000-bit number with the surface code architecture, using magic state distillation, only 6% of the logical qubits are data qubits; all others are logical ancillas for the T gates. For this architecture, each logical qubit is already comprised of 14 500 physical qubits, leading to a total of about 1×10^9 physical qubits.

One possible approach to reduce overhead is to choose the surface code as a bottom code and a code with a transversal T gate and a high rate as a top code, as suggested by Cross, DiVincenzo, and Terhal (2009). In principle the choice of a top code is not restricted by physical locality as one can implement a SWAP gate between the logical qubits of the bottom code using three CNOT gates, so any quantum LDPC code could be used. This SWAP gate will take a time which scales at least with L [as one has to repeat syndrome measurements $O(L)$ times]; hence more nonlocality would lead to a slower computation.

ACKNOWLEDGMENTS

I thank Ben Criger for quickly making some of the figures and Nikolas Breuckmann, David DiVincenzo, and Daniel Gottesman for interesting discussions and feedback on this review. This research was supported in part by the Perimeter Institute for Theoretical Physics. Research at the Perimeter Institute is supported by the Government of Canada through Industry Canada and by the Province of Ontario through the Ministry of Economic Development & Innovation. Funding is acknowledged through the European Union via the program ScaleQIT.

REFERENCES

Aharonov, D., and M. Ben-Or, 1997, "Fault-tolerant quantum computation with constant error," in Proceedings of the 29th STOC, pp. 176–188 [http://arxiv.org/abs/quant-ph/9611025].

Aharonov, D., and L. Eldar, 2011, "On the complexity of commuting local Hamiltonians, and tight conditions for topological order in such systems," in Proceedings of FOCS 2011 (IEEE), pp. 334–343 [http://arxiv.org/abs/1102.0770].

Aharonov, D., A. Kitaev, and J. Preskill, 2006, "Fault-Tolerant Quantum Computation with Long-Range Correlated Noise," Phys. Rev. Lett. **96**, 050504.

Ahn, C., A. C. Doherty, and A. J. Landahl, 2002, "Continuous quantum error correction via quantum feedback control," Phys. Rev. A **65**, 042301.

Alicea, J., 2012, "New directions in the pursuit of Majorana fermions in solid state systems," Rep. Prog. Phys. **75**, 076501.

Alicki, R., M. Horodecki, P. Horodecki, and R. Horodecki, 2010, "On thermal stability of topological qubit in Kitaev's 4D model," Open Syst. Inf. Dyn. **17**, 1.

Aliferis, P., 2007, "Level Reduction and the Quantum Threshold Theorem", Ph.D. thesis (CalTech) [http://arxiv.org/abs/quant-ph/0703230].

Aliferis, P., and A. Cross, 2007, "Subsystem fault-tolerance with the Bacon-Shor code," Phys. Rev. Lett. **98**, 220502.

Aliferis, P., D. Gottesman, and J. Preskill, 2006, "Quantum accuracy threshold for concatenated distance-3 codes," Quantum Inf. Comput. **6**, 97.

Aliferis, P., D. Gottesman, and J. Preskill, 2008, "Accuracy threshold for postselected quantum computation," Quantum Inf. Comput. **8**, 181.

Aliferis, P., and J. Preskill, 2009, "Fibonacci scheme for fault-tolerant quantum computation," Phys. Rev. A **79**, 012332.

Aliferis, P., and B. M. Terhal, 2007, "Fault-tolerant quantum computation for local leakage faults," Quantum Inf. Comput. **7**, 139.

Aoki, T., G. Takahashi, T. Kajiya, J.-i. Yoshikawa, S. L. Braunstein, P. van Loock, and A. Furusawa, 2009, "Quantum error correction beyond qubits," Nat. Phys. **5**, 541.

Bacon, D., 2006, "Operator quantum error correcting subsystems for self-correcting quantum memories," Phys. Rev. A **73**, 012340.

Barends, R., *et al.*, 2013, "Coherent Josephson Qubit Suitable for Scalable Quantum Integrated Circuits," Phys. Rev. Lett. **111**, 080502.

Barends, R., *et al.*, 2014, "Superconducting quantum circuits at the surface code threshold for fault tolerance," Nature (London) **508**, 500.

Barreiro, J. T., M. Müller, P. Schindler, D. Nigg, T. Monz, M. Chwalla, M. Hennrich, C. F. Roos, P. Zoller, and R. Blatt, 2011, "An open-system quantum simulator with trapped ions," Nature (London) **470**, 486.

Bell, B. A., D. A. Herrera-Martí, M. S. Tame, D. Markham, W. J. Wadsworth, and J. G. Rarity, 2014, "Experimental demonstration of a graph state quantum error-correction code," Nat. Commun. **5**, 3658.

Bennett, C. H., D. P. DiVincenzo, J. A. Smolin, and W. K. Wootters, 1996, "Mixed state entanglement and quantum error correction," Phys. Rev. A **54**, 3824.

Bény, C., and O. Oreshkov, 2010, "General Conditions for Approximate Quantum Error Correction and Near-Optimal Recovery Channels," Phys. Rev. Lett. **104**, 120501.

Bombin, H., 2010a, "Topological Order with a Twist: Ising Anyons from an Abelian Model," Phys. Rev. Lett. **105**, 030403.

Bombin, H. 2010b, "Topological subsystem codes," Phys. Rev. A **81**, 032301.

Bombin, H., 2013, "Gauge Color Codes," arXiv:1311.0879.

Bombin, H., R. S. Andrist, M. Ohzeki, H. G. Katzgraber, and M. A. Martin-Delgado, 2012, "Strong Resilience of Topological Codes to Depolarization," Phys. Rev. X **2**, 021004.

Bombin, H., G. Duclos-Cianci, and D. Poulin, 2012, "Universal topological phase of two-dimensional stabilizer codes," New J. Phys. **14**, 073048.

Bombin, H., and M. A. Martin-Delgado, 2006, "Topological Quantum Distillation," Phys. Rev. Lett. **97**, 180501.

Bombin, H., and M. A. Martin-Delgado, 2007, "Topological Computation without Braiding," Phys. Rev. Lett. **98**, 160502.

Bombin, H., and M. A. Martin-Delgado, 2009, "Quantum measurements and gates by code deformation," J. Phys. A **42**, 095302.

Braunstein, S., 1998, "Error correction for continuous quantum variables," Phys. Rev. Lett. **80**, 4084.

Bravyi, S., G. Duclos-Cianci, D. Poulin, and M. Suchara, 2013, "Subsystem surface codes with three-qubit check operators," Quantum Inf. Comput. **13**, 0963.

Bravyi, S., and J. Haah, 2011, "Energy Landscape of 3D Spin Hamiltonians with Topological Order," Phys. Rev. Lett. **107**, 150504.

Bravyi, S., and J. Haah, 2013, "Analytic and numerical demonstration of quantum self-correction in the 3D Cubic Code," Phys. Rev. Lett. **111**, 200501.

Bravyi, S., M. B. Hastings, and S. Michalakis, 2010, "Topological quantum order: Stability under local perturbations," J. Math. Phys. (N.Y.) **51**, 093512.

Bravyi, S., and A. Kitaev, 2005, "Universal quantum computation with ideal Clifford gates and noisy ancillas," Phys. Rev. A **71**, 022316.

Bravyi, S., and R. Koenig, 2013, "Classification of topologically protected gates for local stabilizer codes," Phys. Rev. Lett. **110**, 170503.

Bravyi, S., B. Leemhuis, and B. M. Terhal, 2011, "Topological order in an exactly solvable 3D spin model," Ann. Phys. (Amsterdam) **326**, 839.

Bravyi, S., D. Poulin, and B. M. Terhal, 2010, "Tradeoffs for Reliable Quantum Information Storage in 2D Systems," Phys. Rev. Lett. **104**, 050503.

Bravyi, S., and B. M. Terhal, 2009, "A no-go theorem for a two-dimensional self-correcting quantum memory based on stabilizer codes," New J. Phys. **11**, 043029.

Bravyi, S. B., and A. Yu. Kitaev, 1998, "Quantum codes on a lattice with boundary," arXiv:quant-ph/9811052.

Brell, C. G., S. Burton, G. Dauphinais, S. T. Flammia, and D. Poulin, 2014, "Thermalization, Error Correction, and Memory Lifetime for Ising Anyon Systems," Phys. Rev. X **4**, 031058.

Brooks, P., and J. Preskill, 2013, "Fault-tolerant quantum computation with asymmetric Bacon-Shor codes," Phys. Rev. A **87**, 032310.

Bullock, S. S., and G. K. Brennen, 2007, "Qudit surface codes and gauge theory with finite cyclic groups," J. Phys. A **40**, 3481.

Castelnovo, C., and C. Chamon, 2008, "Topological order in a three-dimensional toric code at finite temperature," Phys. Rev. B **78**, 155120.

Chase, B. A., A. J. Landahl, and J. Geremia, 2008, "Efficient feedback controllers for continuous-time quantum error correction," Phys. Rev. A **77**, 032304.

Cross, A., G. Smith, J. A. Smolin, and Bei Zeng, 2009, "Codeword stabilized quantum codes," IEEE Trans. Inf. Theory **55**, 433.

Cross, A. W., D. P. DiVincenzo, and B. M. Terhal, 2009, "A comparative code study for quantum fault tolerance," Quantum Inf. Comput. **9**, 541.

Dengis, J., R. König, and F. Pastawski, 2014, "An optimal dissipative encoder for the toric code," New J. Phys. **16**, 013023.

Dennis, E., A. Kitaev, A. Landahl, and J. Preskill, 2002, "Topological quantum memory," J. Math. Phys. (N.Y.) **43**, 4452.

DiVincenzo, D. P., 2009, "Fault-tolerant architectures for superconducting qubits," Phys. Scr. **T137**, 014020.

DiVincenzo, D. P., and P. Aliferis, 2007, "Effective fault-tolerant quantum computation with slow measurements," Phys. Rev. Lett. **98**, 020501.

DiVincenzo, D. P., and F. Solgun, 2013, "Multi-qubit parity measurement in circuit quantum electrodynamics," New J. Phys. **15**, 075001.

Douçot, B., and L. Ioffe, 2012, "Physical implementation of protected qubits," Rep. Prog. Phys. **75**, 072001.

Duclos-Cianci, G., and D. Poulin, 2010, "A renormalization group decoding algorithm for topological quantum codes," in *Proceedings of Information Theory Workshop (ITW)* (IEEE, New York), pp. 1–5 [http://arxiv.org/abs/1006.1362].

Duclos-Cianci, G., and D. Poulin, 2014, "Fault-tolerant renormalization group decoder for Abelian topological codes," Quantum Inf. Comput. **14**, 721.

Eastin, B., and E. Knill, 2009, "Restrictions on transversal encoded quantum gate sets," Phys. Rev. Lett. **102**, 110502.

Edmonds, Jack, 1965, "Paths, trees, and flowers," Can. J. Math. **17**, 449.

Elzerman, J. M., R. Hanson, J. S. Greidanus, L. H. Willems van Beveren, S. De Franceschi, L. M. K. Vandersypen, S. Tarucha, and L. P. Kouwenhoven, 2003, "Few-electron quantum dot circuit with integrated charge read out," Phys. Rev. B **67**, 161308.

Fowler, A., 2015, "Minimum weight perfect matching of fault-tolerant topological quantum error correction in average O(1) parallel time," Quantum Inf. Comput. **15**, 0145.

Fowler, A. G., 2013a, "Accurate simulations of planar topological codes cannot use cyclic boundaries," Phys. Rev. A **87**, 062320.

Fowler, A. G., 2013b, "Optimal complexity correction of correlated errors in the surface code," arXiv:1310.0863.

Fowler, A. G., M. Mariantoni, J. M. Martinis, and A. N. Cleland, 2012, "Surface codes: Towards practical large-scale quantum computation," Phys. Rev. A **86**, 032324.

Fowler, A. G., A. M. Stephens, and P. Groszkowski, 2009, "High-threshold universal quantum computation on the surface code," Phys. Rev. A **80**, 052312.

Fowler, A. G., A. C. Whiteside, A. L. McInnes, and A. Rabbani, 2012, "Topological code Autotune," Phys. Rev. X **2**, 041003.

Fowler, Austin G., Adam C. Whiteside, and Lloyd C. L. Hollenberg, 2012, "Towards practical classical processing for the surface code: Timing analysis," Phys. Rev. A **86**, 042313.

Freedman, M. H., and M. B. Hastings, 2013, "Quantum Systems on non-k-hyperfinite complexes: a generalization of classical statistical mechanics on expander graphs," Quantum Inf. Comput. **14**, 144.

Freedman, M. H., and D. A. Meyer, 2001, "Projective plane and planar quantum codes," Found. Comput. Math. **1**, 325.

Fujii, K., M. Negoro, N. Imoto, and M. Kitagawa, 2014, "Measurement-free topological protection using dissipative feedback," Phys. Rev. X **4**, 041039.

Ghosh, J., and A. G. Fowler, 2015, "A leakage-resilient scheme for the measurement of stabilizer operators in superconducting quantum circuits," Phys. Rev. A **91**, 020302(R).

Ghosh, J., A. G. Fowler, and M. R. Geller, 2012, "Surface code with decoherence: An analysis of three superconducting architectures," Phys. Rev. A **86**, 062318.

Gladchenko, S., D. Olaya, E. Dupont-Ferrier, B. Douçot, L. B. Ioffe, and M. E. Gershenson, 2009, "Superconducting nanocircuits for topologically protected qubits," Nat. Phys. **5**, 48.

Glancy, S., and E. Knill, 2006, "Error analysis for encoding a qubit in an oscillator," Phys. Rev. A **73**, 012325.

Gottesman, D., 1997, "Stabilizer Codes and Quantum Error Correction," Ph.D. thesis (CalTech) [http://arxiv.org/abs/quant-ph/9705052].

Gottesman, D., 1999a, "Fault-tolerant quantum computation with higher-dimensional systems," Chaos Solitons Fractals **10**, 1749.

Gottesman, D., 1999b, "The Heisenberg representation of quantum computers," in *Group22: Proceedings of the XXII International Colloquium on Group Theoretical Methods in Physics*, edited by S. P. Corney, R. Delbourgo, and P. D. Jarvis, pp. 32–43 (International Press, Cambridge, MA) [http://arxiv.org/abs/quant-ph/9807006].

Gottesman, D., 2000, "Fault-tolerant quantum computation with local gates," J. Mod. Opt. **47**, 333.

Gottesman, D., 2009, "An Introduction to Quantum Error Correction and Fault-Tolerant Quantum Computation," arXiv:0904.2557.

Gottesman, D., 2014, "Fault-Tolerant Quantum Computation with Constant Overhead," Quantum Inf. Comput. **14**, 1338.

Gottesman, D., and I. L. Chuang, 1999, "Demonstrating the viability of universal quantum computation using teleportation and single-qubit operations," Nature (London) **402**, 390.

Gottesman, D., A. Yu. Kitaev, and J. Preskill, 2001, "Encoding a qubit in an oscillator," Phys. Rev. A **64**, 012310.

Guth, L., and A. Lubotzky, 2014, "Quantum error-correcting codes and 4-dimensional arithmetic hyperbolic manifolds," J. Math. Phys. (N.Y.) **55**, 082202.

Haah, J., 2011, "Local stabilizer codes in three dimensions without string logical operators," Phys. Rev. A **83**, 042330.

Haroche, S., M. Brune, and J.-M. Raimond, 2007, "Measuring the photon number parity in a cavity: from light quantum jumps to the tomography of non-classical field states," J. Mod. Opt. **54**, 2101.

Haroche, S., and J.-M. Raimond, 2006, *Exploring the Quantum: Atoms, Cavities, and Photons* (Oxford University Press, Oxford).

Harrington, J., 2004, "Analysis of quantum error-correcting codes: symplectic lattice codes and toric codes," Ph.D. thesis (CalTech) [http://thesis.library.caltech.edu/1747/].

Hastings, M. B., 2013, "Decoding in hyperbolic spaces: LDPC Codes with linear rate and efficient error correction," arXiv:1312.2546.

Hastings, M. B., and A. Geller, 2014, "Reduced space-time and time costs using dislocation codes and arbitrary ancillas," arXiv:1408.3379.

Herold, M., E. T. Campbell, J. Eisert, and M. J. Kastoryano, 2014, "Cellular-automaton decoders for topological quantum memories," arXiv:1406.2338.

Horsman, C., A. G. Fowler, S. Devitt, and R. Van Meter, 2012, "Surface code quantum computing by lattice surgery," New J. Phys. **14**, 123011.

Jones, C., 2013a, "Multilevel distillation of magic states for quantum computing," Phys. Rev. A **87**, 042305.

Jones, C., 2013b, "Logic Synthesis for Fault-Tolerant Quantum Computers," Ph.D. thesis (Stanford) [http://arxiv.org/abs/1310.7290].

Katzgraber, H. G., and R. S. Andrist, 2013, "Stability of topologically-protected quantum computing proposals as seen through spin glasses," J. Phys. Conf. Ser. **473**, 012019.

Kerckhoff, J., L. Bouten, A. Silberfarb, and H. Mabuchi, 2009, "Physical model of continuous two-qubit parity measurement in a cavity-QED network," Phys. Rev. A **79**, 024305.

Kerckhoff, J., H. I. Nurdin, D. S. Pavlichin, and H. Mabuchi, 2010, "Designing quantum memories with embedded control: photonic circuits for autonomous quantum error correction," Phys. Rev. Lett. **105**, 040502.

Kerckhoff, J., D. S. Pavlichin, H. Chalabi, and H. Mabuchi, 2011, "Design of nanophotonic circuits for autonomous subsystem quantum error correction," New J. Phys. **13**, 055022.

Kitaev, A., 2003, "Fault-tolerant quantum computation by anyons," Ann. Phys. (Amsterdam) **303**, 2.

Kitaev, A., 2006, "Protected qubit based on a superconducting current mirror," arXiv:cond-mat/0609441.

Kitaev, Alexei, 2006, "Anyons in an exactly solved model and beyond," Ann. Phys. (Amsterdam) **321**, 2.

Kitaev, A. Yu., 1997, "Quantum computations: algorithms and error correction," Russ. Math. Surv. **52**, 1191.

Knill, E., 2005, "Quantum computing with realistically noisy devices," Nature (London) **434**, 39.

Knill, E., and R. Laflamme, 1997, "A theory of quantum error-correcting codes," Phys. Rev. A **55**, 900.

Knill, E., R. Laflamme, and W. Zurek, 1998, "Resilient quantum computation," Science **279**, 342.

Koch, J., M. Yu. Terri, J. Gambetta, A. A. Houck, D. I. Schuster, J. Majer, A. Blais, M. H. Devoret, S. M. Girvin, and R. J. Schoelkopf, 2007, “Charge-insensitive qubit design derived from the Cooper pair box,” Phys. Rev. A **76**, 042319.

Koenig, R., G. Kuperberg, and B. W. Reichardt, 2010, “Quantum computation with Turaev-Viro codes,” Ann. Phys. (Amsterdam) **325**, 2707.

Kovalev, A. A., and L. P. Pryadko, 2013, “Fault-tolerance of quantum low-density parity check codes with sublinear distance scaling,” Phys. Rev. A **87**, 020304.

Kribs, D., R. Laflamme, and D. Poulin, 2005, “Unified and generalized approach to quantum error correction,” Phys. Rev. Lett. **94**, 180501.

Lalumière, K., J. M. Gambetta, and A. Blais, 2010, “Tunable joint measurements in the dispersive regime of cavity QED,” Phys. Rev. A **81**, 040301.

Landahl, A. J., J. T. Anderson, and P. R. Rice, 2011, “Fault-tolerant quantum computing with color codes,” arXiv:1108.5738.

Leghtas, Z., G. Kirchmair, B. Vlastakis, R. J. Schoelkopf, M. H. Devoret, and M. Mirrahimi, 2013, “Hardware-efficient autonomous quantum memory protection,” Phys. Rev. Lett. **111**, 120501.

Leung, D. W., M. A. Nielsen, I. L. Chuang, and Y. Yamamoto, 1997, “Approximate quantum error correction can lead to better codes,” Phys. Rev. A **56**, 2567.

Levin, M. A., and X.-G. Wen, 2005, “String-net condensation: A physical mechanism for topological phases,” Phys. Rev. B **71**, 045110.

Levy, J. E., A. Ganti, C. A. Phillips, B. R. Hamlet, A. J. Landahl, T. M. Gurrieri, R. D. Carr, and M. S. Carroll, 2009, “The impact of classical electronics constraints on a solid-state logical qubit memory,” in *Proceedings of the 21st annual symposium on Parallelism in algorithms and architectures* (ACM, New York), pp. 166–168 [http://arxiv.org/abs/0904.0003].

Lidar, D. A., 2014, “Review of decoherence free subspaces, noiseless subsystems, and dynamical decoupling,” Adv. Chem. Phys. **154**, 295.

Lidar, D. A., and T. A. Brun, 2013, Eds., *Quantum Error Correction* (Cambridge University Press, Cambridge, England).

Lloyd, S., and J.-J. Slotine, 1998, “Analog quantum error correction,” Phys. Rev. Lett. **80**, 4088.

Menicucci, N. C., 2014, “Fault-tolerant measurement-based quantum computing with continuous-variable cluster states,” Phys. Rev. Lett. **112**, 120504.

Metcalfe, M., E. Boaknin, V. Manucharyan, R. Vijay, I. Siddiqi, C. Rigetti, L. Frunzio, R. J. Schoelkopf, and M. H. Devoret, 2007, “Measuring the decoherence of a quantronium qubit with the cavity bifurcation amplifier,” Phys. Rev. B **76**, 174516.

Michnicki, K. P., 2014, “3D Topological quantum memory with a power-law energy barrier,” Phys. Rev. Lett. **113**, 130501.

Mirrahimi, M., Z. Leghtas, V. V. Albert, S. Touzard, R. J. Schoelkopf, L. Jiang, and M. H. Devoret, 2013, “Dynamically protected cat-qubits: a new paradigm for universal quantum computation,” arXiv:1312.2017.

Müller, M., K. Hammerer, Y. L. Zhou, C. F. Roos, and P. Zoller, 2011, “Simulating open quantum systems: from many-body interactions to stabilizer pumping,” New J. Phys. **13**, 085007.

Napp, J., and J. Preskill, 2013, “Optimal Bacon-Shor codes,” Quantum Inf. Comput. **13**, 490.

Ng, H. K., and P. Mandayam, 2010, “Simple approach to approximate quantum error correction based on the transpose channel,” Phys. Rev. A **81**, 062342.

Nielsen, A. E. B., 2010, “Fighting decoherence in a continuous two-qubit odd- or even-parity measurement with a closed-loop setup,” Phys. Rev. A **81**, 012307.

Nielsen, M. A., and I. L. Chuang, 2000, *Quantum computation and quantum information* (Cambridge University Press, Cambridge, England).

Nielsen, M. A., and D. Poulin, 2007, “Algebraic and information-theoretic conditions for operator quantum error-correction,” Phys. Rev. A **75**, 064304(R).

Nigg, D., M. Müller, E. A. Martinez, P. Schindler, M. Hennrich, T. Monz, M. A. Martin-Delgado, and R. Blatt, 2014, “Quantum computations on a topologically encoded qubit,” Science **345**, 302.

Nigg, S. E., and S. M. Girvin, 2013, “Stabilizer quantum error correction toolbox for superconducting qubits,” Phys. Rev. Lett. **110**, 243604.

Nussinov, Z., and G. Ortiz, 2009, “Symmetry and Topological Order,” Proc. Natl. Acad. Sci. U.S.A. **106**, 16944.

Paetznick, A., and B. W. Reichardt, 2013, “Universal fault-tolerant quantum computation with only transversal gates and error correction,” Phys. Rev. Lett. **111**, 090505.

Poulin, D., 2005, “Stabilizer formalism for operator quantum error correction,” Phys. Rev. Lett. **95**, 230504.

Poulin, D., 2006, “Optimal and efficient decoding of concatenated quantum block codes,” Phys. Rev. A **74**, 052333.

Preskill, J., 1998, “Fault-tolerant quantum computation,” in *Introduction to Quantum Computation* (World Scientific, Singapore), pp. 213–269.

Raussendorf, R., and J. Harrington, 2007, “Fault-tolerant quantum computation with high threshold in two dimensions,” Phys. Rev. Lett. **98**, 190504.

Raussendorf, R., J. Harrington, and K. Goyal, 2007, “Topological fault-tolerance in cluster state quantum computation,” New J. Phys. **9**, 199.

Reichardt, Ben W., 2005, “Quantum universality by distilling certain one- and two-qubit states with stabilizer operations,” Quantum Inf. Process. **4**, 251.

Shor, P. W., 1996, “Fault-tolerant quantum computation,” in *Proceedings of 37th FOCS* (IEEE Computer Society, Washington, DC), pp. 56–65.

Steane, A., 1999, “Quantum Reed-Muller codes,” IEEE Trans. Inf. Theory **45**, 1701.

Steane, A., 2003, “Overhead and noise threshold of fault-tolerant quantum error correction,” Phys. Rev. A **68**, 042322.

Steane, A. M., 1997, “Active stabilization, quantum computation, and quantum state synthesis,” Phys. Rev. Lett. **78**, 2252.

Suchara, M., S. Bravyi, and B. Terhal, 2011, “Constructions and noise threshold of topological subsystem codes,” J. Phys. A **44**, 155301.

Suchara, M., A. W. Cross, and J. M. Gambetta, 2014, “Leakage suppression in the toric code,” arXiv:1410.8562.

Suchara, M., A. Faruque, C.-Y. Lai, G. Paz, F. T. Chong, and J. Kubiatowicz, 2013, “Comparing the overhead of topological and concatenated quantum error correction,” arXiv:1312.2316.

Sun, L., *et al.*, 2014, “Tracking photon jumps with repeated quantum non-demolition parity measurements,” Nature (London) **511**, 444.

Svore, K. M., A. W. Cross, I. L. Chuang, and A. V. Aho, 2006, “A flow-map model for analyzing pseudothresholds in fault-tolerant quantum computing,” Quantum Inf. Comput. **6**, 193.

Svore, K. M., D. P. DiVincenzo, and B. M. Terhal, 2007, “Noise threshold for a fault-tolerant two-dimensional lattice architecture,” Quantum Inf. Comput. **7**, 297.

Terhal, B. M., F. Hassler, and D. P. DiVincenzo, 2012, “From Majorana fermions to topological order,” Phys. Rev. Lett. **108**, 260504.

Tillich, Jean-Pierre, and Gilles Zémor, 2014, “Quantum LDPC codes with positive rate and minimum distance proportional to $n^{\frac{1}{2}}$,” IEEE Trans. Inf. Theory **60**, 1193.

Vandersypen, L. M. K., and I. L. Chuang, 2005, “NMR techniques for quantum control and computation,” Rev. Mod. Phys. **76**, 1037.

van Handel, R., and H. Mabuchi, 2005, “Optimal error tracking via quantum coding and continuous syndrome measurement,” arXiv: quant-ph/0511221.

Vasconcelos, H., L. Sanz, and S. Glancy, 2010, “All-optical generation of states for “Encoding a qubit in an oscillator”,” Opt. Lett. **35**, 3261.

Wang, C., J. Harrington, and J. Preskill, 2003, “Confinement-Higgs transition in a disordered gauge theory and the accuracy threshold for quantum memory,” Ann. Phys. (Amsterdam) **303**, 31.

Wang, D. S., A. G. Fowler, and L. C. L. Hollenberg, 2011, “Quantum computing with nearest neighbor interactions and error rates over 1%,” Phys. Rev. A **83**, 020302(R).

Weissman, M. B., 1988, “$1/f$ noise and other slow, nonexponential kinetics in condensed matter,” Rev. Mod. Phys. **60**, 537.

Wiseman, H., and G. J. Milburn, 2010, *Quantum Measurement and Control* (Cambridge University Press, Cambridge).

Wootton, J. R., 2012, “Quantum memories and error correction,” J. Mod. Opt. **59**, 1717.

Wootton, J. R., J. Burri, S. Iblisdir, and D. Loss, 2014, “Decoding non-Abelian topological quantum memories,” Phys. Rev. X **4**, 011051.

Yao, X.-C., *et al.*, 2012, “Experimental demonstration of topological error correction,” Nature (London) **482**, 489.

Yoshida, Beni, 2011, “Feasibility of self-correcting quantum memory and thermal stability of topological order,” Ann. Phys. (Amsterdam) **326**, 2566.

Zeng, Bei, Andrew W. Cross, and Isaac L. Chuang, 2011, “Transversality versus universality for additive quantum codes,” IEEE Trans. Inf. Theory **57**, 6272.

Zhang, J., C. Xie, K. Peng, and P. van Loock, 2008, “Anyon statistics with continuous variables,” Phys. Rev. A **78**, 052121.

Zhou, X., D. W. Leung, and I. L. Chuang, 2000, “Methodology for quantum logic gate construction,” Phys. Rev. A **62**, 052316.