Decoherence-free subspaces for multiple-qubit errors. II. Universal, fault-tolerant quantum computation

Daniel A. Lidar,[1,*] Dave Bacon,[1,2] Julia Kempe,[1,3,4] and K. B. Whaley[1]
[1]*Department of Chemistry, University of California, Berkeley, California 94720*
[2]*Department of Physics, University of California, Berkeley, California 94720*
[3]*Department of Mathematics, University of California, Berkeley, California 94720*
[4]*École Nationale Superieure des Télécommunications, Paris, France*
(Received 6 July 2000; published 17 January 2001)

Decoherence-free subspaces (DFSs) shield quantum information from errors induced by the interaction with an uncontrollable environment. Here we study a model of correlated errors forming an Abelian subgroup (stabilizer) of the Pauli group (the group of tensor products of Pauli matrices). Unlike previous studies of DFSs, this type of error does not involve any spatial symmetry assumptions on the system-environment interaction. We solve the problem of universal, fault-tolerant quantum computation on the associated class of DFSs. We do so by introducing a hybrid DFS quantum error-correcting-code approach, where errors that arise due to departure of the codewords from the DFS are corrected actively.

DOI: 10.1103/PhysRevA.63.022307 PACS number(s): 03.67.Lx, 03.65.Ta, 03.65.Fd, 89.70.+c

I. INTRODUCTION

Methods to protect fragile quantum superpositions are of paramount importance in the quest to construct devices that can reliably process quantum information [1,2]. Compared to their classical counterparts, such devices feature spectacular advantages in both computation and communication, as discussed in a number of recent reviews [3–5]. The dominant source of the fragility of a quantum information processor (QIP) is the inevitable interaction with its environment. This coupling leads to *decoherence*, a process whereby coherence of the QIP wave function is gradually destroyed. Formally, the evolution of an open system (coupled to an environment) such as a QIP can be described by a completely positive map [6], which can always be written in the explicit form known as the Kraus operator sum representation [7]:

$$\rho(t)=\sum_d A_d(t)\rho(0)A_d^\dagger(t). \tag{1.1}$$

Here ρ is the system density matrix, and the ''Kraus operators'' $\{A_d\}$ are time-dependent operators acting on the system Hilbert space, constrained only to sum to the identity operator: $\Sigma_d A_d^\dagger A_d = I$ (to preserve $\mathrm{Tr}[\rho]$).[1] Decoherence is the situation in which there are at least two Kraus operators that are inequivalent under scalar multiplication. The Kraus operators are in that case related to the different ways in which errors can afflict the quantum information contained in ρ [9]. Conversely, if there is only one Kraus operator, then from the normalization condition it must be unitary: $A=\exp(-iHt)$ with H Hermitian, so that ρ satisfies the *closed*-system Liouville equation $\dot{\rho}=-i[H,\rho]$, H being the system Hamiltonian. In this case there is no decoherence.

Two principal *encoding* methods have been proposed to solve the decoherence problem: (i) Quantum error-correcting codes (QECCs) [10–16] (for a recent review see [17]), (ii) decoherence-free subspaces (DFSs) [18–24], also known as ''noiseless'' or ''error-avoiding quantum codes.'' In both methods, quantum information is protected against decoherence by encoding it into ''codewords'' (entangled superpositions of multiple-qubit states) with special symmetry properties. To exhibit these, it is useful to expand the Kraus operators over a fixed operator basis. For qubits, a particularly useful basis is formed by the elements of the Pauli group, which is the group of tensor products of all Pauli matrices $\{\sigma_k^{\alpha_k}\}$, where $\alpha=0,x,y,z$ (σ^0 is the 2×2 identity matrix) and $k=1\cdots K$ is the qubit index. An element of the Pauli group can be written as $E_a=\otimes_{k=1}^K \sigma_k^{\alpha_k}$, where $a=(\alpha_1,\ldots,\alpha_K)$. The 4^{K+1} elements $\{E_a\}$ of the Pauli group (we include factors of $\pm, \pm i$ in this count) square to identity, are both unitary and Hermitian, either commute or anticommute, and satisfy $\mathrm{Tr}[E_a^\dagger E_b]=\delta_{ab}/2^K$. When the Kraus operators are expanded as

$$A_d(t)=\sum c_{ad}(t)E_a, \tag{1.2}$$

the operators $\{E_a\}$ acquire the significance of representing the different physical errors that can corrupt the quantum information. The weight $w(E_a)$ is the number of nonzero α_k in a. Let us now assume a short-time expansion of the $c_{ad}(t)$ (relative to the bath-correlation time). The situation where only those E_a with $w(E_a)=1$ have nonvanishing $c_{ad}(t)$ is called the ''independent errors'' model (assuming the c_{ad}, which are essentially bath correlation functions [8], are statistically independent). Correlated errors correspond to the situation in which some E_a with $w(E_a)>1$ have nonvanish-

*Present address: Chemistry Department, University of Toronto, 80 St. George Street, Toronto, Ontario, Canada M5S 3H6.

[1]As shown, e.g., in [8], the operator sum representation can be derived from a Hamiltonian model by considering the reduced dynamics of a system coupled to a bath B: $\rho(t)=\mathrm{Tr}_B(U(t)[\rho(0)\otimes\rho_B(0)]U^\dagger(t))$. Here the trace is over the bath degrees of freedom, $U=\exp(-iH_{SB}t)$ is the unitary evolution operator of the combined system bath, and H_{SB} is their interaction Hamiltonian. One finds $A_{d=(\mu,\nu)}=\sqrt{\mu}\langle\mu|U|\nu\rangle$, where $|\mu\rangle,|\nu\rangle$ are bath states in the spectral decomposition of the bath density matrix: $\rho_B=\Sigma\mu|\mu\rangle\langle\mu|$.

ing $c_{ad}(t)$: two or more qubits are acted upon nontrivially with the same coefficient c_{ad}. QECCs can be classified according to the maximum weight of the errors they can still correct (this is related to the notion of a ''distance'' of a code [17]). QECCs can generally deal at least with errors of weight 1. Barring accidental degeneracies, nontrivial DFSs, on the other hand, generally do not exist if there are errors with weight 1 [22]. To make these ideas more precise, let us briefly recall the definitions of QECCs and DFSs.

A QECC is a subspace $\mathcal{C}=\mathrm{span}[\{|i\rangle\}]$ of the system Hilbert space with the symmetry property that different errors take orthogonal codewords $|i\rangle$ and $|j\rangle$ to orthogonal states [16]:

$$\langle i|E_a^\dagger E_b|j\rangle=\gamma_{ab}\delta_{ij}. \tag{1.3}$$

Here γ_{ab} are the elements of a Hermitian matrix γ and δ_{ij} is the Kronecker delta. This property ensures that if an error E_a occurs, it can be detected and subsequently reversed [16]. A large variety of QECCs have been found [17]. A particularly useful and large class, one which will occupy our attention in this paper, arises when one considers Abelian subgroups Q of the Pauli group. Given such an Abelian Pauli subgroup, or *stabilizer* Q (we will use both terms interchangably in this paper), its $+1$ eigenspace is a QECC known as a *stabilizer code* [15]. The set of errors $\{E_a\}$ is correctable by this code if for every two errors E_a, E_b there exists some $q\in Q$ such that

$$\{E_a^\dagger E_b, q\}=0. \tag{1.4}$$

This is because under the stipulated condition $\langle i|E_a^\dagger E_b|j\rangle = \langle i|E_a^\dagger E_b q|j\rangle = -\langle i|qE_a^\dagger E_b|j\rangle = -\langle i|E_a^\dagger E_b|j\rangle$ so that $\langle i|E_a^\dagger E_b|j\rangle \propto \delta_{ij}$ [15]: the QECC condition [Eq. (1.3)] is satisfied. To correct an error E_a one simply applies the unitary operator $E_a^\dagger$ to the code. Note that this involves active intervention, namely measurements to diagnose the error and error reversal.

DFSs can be viewed as highly ''degenerate'' QECCs, where degeneracy refers to the rank of γ: DFSs are rank-1 QECCs (i.e., $\gamma_{ab}=\gamma_a\gamma_b$) [23,25]. Equivalently, a DFS can be defined as the simultaneous eigenspace $\tilde{\mathcal{H}}=\mathrm{span}[\{|\tilde{j}\rangle\}]$ of all Kraus operators [23]:

$$A_d|\tilde{j}\rangle=a_d|\tilde{j}\rangle \tag{1.5}$$

($\{a_d\}$ are the eigenvalues). Viewed in this way, DFSs have the remarkable property that they offer complete protection for quantum information without the need for any active intervention: $\tilde{\rho}(t)=\Sigma_d A_d(t)\tilde{\rho}(0)A_d^\dagger(t)=\tilde{\rho}(0)\Sigma_d|a_d|^2=\tilde{\rho}(0)$, for $\tilde{\rho}$ with support exclusively on $\tilde{\mathcal{H}}$. Thus a DFS is a ''quiet corner'' of the system Hilbert space, which is completely immune to decoherence. Like stabilizer QECCs, DFSs can also be characterized as the $+1$ eigenspace of a stabilizer, which, however, is generally *non-Abelian* over the Pauli group [26,27] (i.e., a DFS is generally a nonadditive code [28]). Most work on DFSs to date has focused on a model of highly correlated errors, known as ''collective decoherence.'' In this model, the (non-Abelian) stabilizer is composed of tensor products of identical SU(2) rotations + contractions on all qubits. Here we will not concern ourselves with the collective decoherence model, and the term ''stabilizer'' will be reserved for the Abelian subgroups of the Pauli group.

In a companion paper [29] (referred to from here on as ''paper I''), we began a study of DFSs for noncollective errors. We derived a necessary and sufficient condition for a subspace to be decoherence-free when the Kraus operators are expanded as linear combinations over the elements of an arbitrary group. The decoherence-free states were shown to be those states that transform according to the one-dimensional irreducible representations (irreps) of this group. As above, it is natural to focus on the case where this group is the Pauli group. This is so not only because of the connection to stabilizer QECCs, but also because the Pauli group arises in the context of many-qubit systems, where it is often natural to expand the Hamiltonians in terms of tensor products of Pauli matrices. To find DFSs, therefore, we focus here on subgroups of the Pauli group. Note that the non-Abelian subgroups of the Pauli group do not have one-dimensional irreps [29], and hence in this case a DFS can be associated only with the Abelian subgroups (which of course have only one-dimensional irreducible representations).

We can now define the error model that will concern us in this paper. Unlike the stabilizer-QECCs case, where the errors that the code can correct are those that anticommute with the stabilizer, *in the DFS case the errors are the elements of the stabilizer itself*. We shall refer to these errors as ''stabilizer errors.'' The Abelian subgroups of the Pauli group cannot contain single-qubit operators, since these would generally generate the whole Pauli group.[2] Hence as errors the elements of the subgroup represent *multiple-qubit* couplings to the bath. As explained above, this is therefore a correlated-errors model, which is distinguished from previous work on DFSs in that it does not involve any spatial-symmetry assumptions. The physical relevance of this error model was discussed in paper I, and will be embellished here. The DFS is not affected by these stabilizer-errors, but the rest of the Hilbert space is and may decohere under their influence. Several examples of DFSs corresponding to Abelian subgroups were given in paper I. Our purpose in this sequel paper is to complete our study of this class of DFSs by showing how to perform universal fault-tolerant quantum computation on them.

The central challenge in demonstrating universal fault-tolerant quantum computation on DFSs is to show how this can be done using only one- and two-body Hamiltonians, and a small number of measurements.[3] Several previous pub-

[2]The exceptions are (i) the subgroup operators have constant $\alpha=x$, y, or z, which is the Pauli matrix index; (ii) the single-qubit operators act only on those qubits where all other operators act as identity.

[3]By ''small'' we mean that the measurements do not have to be fast compared to the bath correlation time. If they are, then the decoherence is avoided essentially by use of the quantum Zeno effect.

lications have addressed the issue of universal quantum computation on DFSs, but left this challenge unanswered [22,30,31]. In Refs. [26,27], we accomplished this task for the first time in the collective decoherence model. Collective decoherence is the situation in which all qubits are coupled in an identical manner to the bath, i.e., there is a strong *spatial symmetry*: qubit permutation invariance. In this case, by using exchange operations, it is possible to implement universal quantum computation without ever leaving the DFS. The procedure is therefore naturally fault-tolerant. In the present paper, we will show how to implement universal fault-tolerant quantum computation on DFSs that arise from the Pauli subgroup error model, without requiring any spatial symmetry assumption. However, *it will not be possible to do so without leaving the DFS, thus exposing the states to the subgroup errors*. As will be shown here, fault tolerance is obtained by using the encoded states twice, in a dual DFS-QECC mode. This duality arises from the fact that the DFS remains a perfectly valid QECC for the errors with which the stabilizer anticommutes.

There are several ways to achieve universal fault-tolerant quantum computation on stabilizer-QECCs; e.g., use of the sets of gates {Hadamard, $\sigma_z^{1/2}$, Toffoli} [32,14] or {Hadamard, $\sigma_z^{1/4}$, controlled-NOT} [33]. Additional methods were provided in [34,35]. Our construction reverts to the early ideas on the implementation of universal quantum computing: we use single-qubit SU(2) operations and a controlled-NOT (CNOT) gate [36–39], except that these are *encoded* operations, acting on codewords (not on physical qubits). In general, such encoded operations involve multiple qubits, and are not naturally available. The key to our construction is a method to generate many-qubit Hamiltonians by composing operations on (at most) pairs of physical qubits. This is done by selectively turning certain interactions on and off. A difficulty is that the very first such step can transform the encoded states and take them outside of the DFS. However, by carefully choosing the interactions that we turn on/off and their order, we show that the transformed states become a QECC with respect to the stabilizer errors to which the DFS was immune. This fact is responsible for the fault tolerance of our procedure. After the final interaction is turned off, the states return to the DFS, and are once again immune to the stabilizer errors.

The structure of the paper is as follows. In Sec. II, we briefly review the main result of paper I and the connection between the DFSs considered here and stabilizer QECCs. We then discuss in Sec. III the meaning of fault tolerance in light of the error model considered in this paper. In the following two sections we present the main ideas and results of this paper: in Sec. IV, we show how to generate many-qubit Hamiltonians by composing two- and single-qubit Hamiltonians, and in Sec. V, we prove the fault tolerance of this procedure. We use it to generate encoded SU(2) operations on the DFS qubits. Section VI shows how, by using similar methods, we can fault tolerantly perform encoded CNOT operations on the encoded qubits, thus coupling blocks of qubits and completing the set of operations needed for universal computation. The final ingredient is presented in Sec. VII, where we show how to fault tolerantly measure the error syndrome throughout our gate construction. While our main motivation in this paper is to study computation on DFSs in the presence of stabilizer errors, it is also interesting to consider the implications of the techniques we develop here to the usual model of errors that anticommute with the stabilizer. We consider this question briefly in Sec. VIII, and show that our methods provide another way to implement universal quantum computation that is fault tolerant with respect to error *detection*. We conclude and summarize in Sec. IX.

II. CONNECTION BETWEEN PAULI SUBGROUP DFSs AND STABILIZER CODES

In paper I we proved the following result.

Theorem 1. Suppose that the Kraus operators belong to the group algebra of some group $\mathcal{G}=\{G_n\}$, i.e., $\mathbf{A}_d=\sum_{n=1}^{N} a_{d,n} G_n$. If a set of states $\{|\tilde{j}\rangle\}$ belong to a given *one-dimensional* irrep of $\mathcal{G}$, then the DFS condition $\mathbf{A}_d|\tilde{j}\rangle = c_d|\tilde{j}\rangle$ holds. If no assumptions are made on the bath coefficients $\{a_{d,n}\}$, then the DFS condition $\mathbf{A}_d|\tilde{j}\rangle = c_d|\tilde{j}\rangle$ implies that $|\tilde{j}\rangle$ belongs to a *one-dimensional* irrep of $\mathcal{G}$.

This theorem provides a characterization of DFSs in terms of the group-representation properties of the basis set used to expand the Kraus operators. There are good physical reasons to choose the Pauli group as this basis set: as argued in paper I, the Pauli group naturally appears as a basis in Hamiltonians involving qubits. Furthermore, using the Pauli group allows us to make a connection to the theory of stabilizer QECCs. To see this, consider the identity irrep, for which each element G_n in the group $\mathcal{G}$ acts on a decoherence-free state $|\psi\rangle$ as

$$G_n|\psi\rangle = |\psi\rangle. \tag{2.1}$$

Choosing $\mathcal{G}$ from now on as a Pauli subgroup Q, the DFS fixed by the identity irrep is a stabilizer code, where Q is the stabilizer group. As mentioned above, a stabilizer code is defined as the $+1$ eigenspace of the Abelian group Q.[4] It is thus clear that the states fixed by Q play a dual role: *they are at once a DFS with respect to the stabilizer errors and a QECC with respect to the errors that anticommute with some element of Q.*

It is simple to verify that basic properties of stabilizer codes hold, e.g., that if the stabilizer group has $K-l$ generators, then the code space (in this case the DFS) has dimension 2^l (i.e., there are l encoded qubits) [15]. Indeed, the dimension of an Abelian group with $K-l$ generators is $N=2^{K-l}$, and we showed in paper I that the dimension of the DFS is $2^K/N=2^l$.

[4]The DFSs corresponding to the other 1D irreps can also be turned into stabilizer codes by a redefinition of the subgroup, taking into account the minus signs appearing in the irrep in question. This kind of freedom is well known in the stabilizer theory of QECCs [15].

III. THE MEANING OF FAULT TOLERANCE

The observation that the Pauli subgroup DFSs are stabilizer codes allows us to employ some results from stabilizer theory, and aids in the analysis of when it is possible to perform universal fault-tolerant computation on these DFSs.[5] Before delving into the analysis, however, we should clarify what we mean by fault tolerance in the present context. The usual meaning of fault tolerance, as it is used in the theory of QECC, is the following: an operation (gate U) is *not* fault tolerant if an error E that the code could fix before application of the gate has become an unfixable error ($UEU^\dagger$) after application of the gate. For example, a single-qubit phase error ($I\otimes Z$) becomes a two-qubit phase error ($Z\otimes Z$) due to the application of a CNOT gate [34]; if the code used could only correct single-qubit errors, then as a result of the CNOT gate (unless it is applied transversally, i.e., not coupling physical qubits involved in representing the same encoded qubit) this code can no longer offer protection. In this scenario, therefore, the CNOT gate was not a fault-tolerant operation. Conversely, an operation *is* fault-tolerant if the code offers the same protection against the errors that appear after application of the operation ($UEU^\dagger$) as it does against the errors before the operation (E).

A complementary (''Heisenberg'' [40]) picture to the (''Schrödinger'') description above is to consider the errors as unchanged and the code $\mathcal{C}$, as well as the stabilizer Q, as transformed after the application of each gate: $\mathcal{C}\mapsto U\mathcal{C}$ and $Q\mapsto UQU^\dagger$. Then fault tolerance can be viewed as the requirement that the new code is capable of correcting the original errors. This point of view will be particularly useful for our purposes. In our case, the original errors are the elements of the Pauli subgroup Q (the stabilizer), and the gates U will turn out not to preserve the original code. Nevertheless, we will show that to the new stabilizer $Q'=UQU^\dagger$ corresponds a QECC (the transformed code $\mathcal{C}'=U\mathcal{C}$) that can correct the original errors. In this way, the fault-tolerance criterion is satisfied.

IV. ENCODED SU(2) FROM HAMILTONIANS

We now begin in earnest our discussion of how to implement universal, fault-tolerant quantum computation on the Pauli-subgroup DFSs. In this section, we show how arbitrary single encoded-qubit operations can be implemented fault tolerantly. We will do so by generating the entire encoded SU(2) group from at most two-qubit Hamiltonians. We assume that the system Hamiltonian is of the general two-qubit form

$$H_S=\sum_{i=1}^{K}\sum_{\alpha=\{x,y,z\}}\omega_i^{\alpha_i}\sigma_i^{\alpha_i}+\sum_{i>j=1}^{K}\sum_{\alpha,\beta=\{x,y,z\}}J_{ij}^{\alpha_i\beta_j}\sigma_i^{\alpha_i}\otimes\sigma_j^{\beta_j}, \tag{4.1}$$

with controllable parameters $\{\omega_i^{\alpha_i}\}$, $\{J_{ij}^{\alpha_i\beta_j}\}$.

A. Background

Suppose we are given an error subgroup Q generated by the elements $\{q_i\}_{i=1}^{|Q|}$. From the results of paper I we know how to identify the corresponding DFS, which is also a stabilizer code with respect to the errors that anticommute with Q. This QECC aspect will not be needed as long as we are only interested in *storing* information in this DFS: then the Q errors will have no effect. However, here we are interested in the more ambitious goal of *computing* in the presence of the Q errors, which means that we must be able to implement logic gates. As discussed above, these gates will take the states out of the DFS and expose them to the Q errors. To be able to compute, we will need some basic results from the theory of fault-tolerant quantum computation using stabilizer codes, as developed primarily in Ref. [34]. Let us briefly review these results.

The set of operators that commute with the stabilizer themselves form a group called the *normalizer* of the code, $N(Q)$. These elements are of interest because they are operations that preserve the DFS. Let $q\in Q$, $|\psi\rangle\in\mathrm{DFS}(Q)$; if $n\in N(Q)$, then

$$q(n|\psi\rangle)=nq|\psi\rangle=n|\psi\rangle, \tag{4.2}$$

so that $n|\psi\rangle$ is in the DFS as well. Clearly, the stabilizer Q is in the normalizer $N(Q)$ and so the only operations that act nontrivially on the subspace are those that are in the normalizer but not in the stabilizer: $N(Q)/Q$. While this means that these operations can be used to perform useful manipulations on the DFS, it also means that if they act uncontrollably, then they appear as errors that the code *cannot* detect. As will be seen later on, these are both crucial aspects in our construction.

For any Pauli-subgroup stabilizer code, the normalizer is generated by the single-qubit $\overline{X_i}$ and $\overline{Z_i}$ operations, where $i=1,\ldots,l$ labels the *encoded* qubits [34]. The bar superscript denotes that these are ''encoded operations'': they perform a bit flip and a phase flip on the encoded qubits. The gates $\overline{X_i}$ and $\overline{Z_i}$, however, are by themselves insufficient for universal quantum computation. The usual stabilizer-QECC construction deals with (typically *un*correlated) errors that anticommute with the stabilizer. In this case, in addition to generating the normalizer of the Pauli group $N(P_K)$, one other operation is needed, such as the Toffoli gate [32]. Such constructions have been covered in several recent publications [32–35,41]. However, as emphasized above, the errors here are qualitatively different: not only are they always correlated, rather than anticommuting with the stabilizer, *the errors are the stabilizer itself*. Thus the usual construction does not apply, and we introduce a different approach. We show how to perform universal fault-tolerant quantum computa-

[5]The reader may wonder whether it should not be possible to simply take over the results about universal fault-tolerant computation from stabilizer theory and apply them directly in the present case. However, a problem is encountered when that construction is applied to the error model considered here, because multiple-qubit errors may propagate back as (nonperturbative) single-qubit errors due to interaction with a ''bare'' (non-DFS) ancilla. We are indebted to Dr. Daniel Gottesman for pointing out this problem to us.

tion using the early SU(2)+controlled NOT (CNOT) construction [37,39,42], but applied to encoded (DFS) qubits.

B. A useful formula: Conjugation by $\pi/4$

Instead of treating $\overline{X}$ and $\overline{Z}$ as gates, as in the usual stabilizer-QECC construction, we employ them as *Hamiltonians*. Since $\overline{X}$ and $\overline{Z}$ are in the normalizer, so are $\exp(i\theta\overline{X})$ and $\exp(i\theta\overline{Z})$, and so are any other encoded SU(2) group [denoted $\overline{\mathrm{SU}(2)}$] operations obtained from them. By applying operations from $\overline{\mathrm{SU}(2)}$ alone, we ensure that the code is preserved. To obtain other $\overline{\mathrm{SU}(2)}$ operations from $\overline{X}$ and $\overline{Z}$, we use the Euler angle construction [43], which shows that any rotation can be composed out of rotations about only two orthogonal axes:

$$\exp[-i\omega(\mathbf{n}\cdot\boldsymbol{\sigma})/2]=\exp(-i\beta\sigma_z/2)\exp(-i\theta\sigma_y/2)\times\exp(-i\alpha\sigma_z/2). \tag{4.3}$$

Here the resulting rotation is by an angle ω about the direction specified by the unit vector $\mathbf{n}$, both of which are functions of α, β, and θ. Using Eq. (4.3) and the mapping $\{\sigma_x,\sigma_y,\sigma_z\}\mapsto\{\overline{X},\overline{Y},\overline{Z}\}$, we can construct any element of $\overline{\mathrm{SU}(2)}$. To do so, we now derive a form of the Euler angle construction that is particularly relevant to operations with Pauli matrices. Assume that A and B are both tensor products of Pauli matrices (and thus square to identity). Then

$$\begin{aligned}&\exp(-i\varphi A)B\exp(+i\varphi A)\\&\quad=(I\cos\varphi-Ai\sin\varphi)B(I\cos\varphi+Ai\sin\varphi)\\&\quad=B\cos^2\varphi+ABA\sin^2\varphi-i\sin\varphi\cos\varphi[A,B]\\&\quad=\begin{cases}B & \text{if } [A,B]=0\\ B\cos 2\varphi+iBA\sin 2\varphi & \text{if } \{A,B\}=0.\end{cases}\end{aligned} \tag{4.4}$$

For the special case of $\varphi=\pi/4$, we define the conjugation with A by

$$\begin{aligned}T_A\circ\exp(i\theta B)&\equiv\exp\left(-i\frac{\pi}{4}A\right)\exp(i\theta B)\exp\left(+i\frac{\pi}{4}A\right)\\&=\begin{cases}\exp(i\theta B) & \text{if } [A,B]=0\\ \exp[i\theta(iAB)] & \text{if } \{A,B\}=0.\end{cases}\end{aligned} \tag{4.5}$$

This can be understood geometrically as a rotation by $\varphi=\pi/4$ about the ''axis'' A, followed by a rotation by θ about B, followed finally by a $\beta=-\pi/4$ rotation about A, resulting overall in rotation by θ about the ''axis'' iAB. All $\varphi=\pi/4$ rotations about a Pauli group member are elements of the normalizer of the Pauli group: they take elements in the Pauli group under conjugation to other elements of the Pauli group.

Note that the ''conjugation-by-$(\pi/4)A$'' operation $T_A\circ\exp(i\theta B)$ is equivalent to multiplication of B to the left by iA inside the exponent. This is very useful, since the elements of the normalizer of any stabilizer can always be written as a tensor product of single-qubit Pauli matrices, i.e., as a tensor product of single-body gates. This is exactly the structure that is suggested by Eq. (4.5), and thus it should allow us to construct $\exp(i\theta\overline{X})$ and $\exp(i\theta\overline{Z})$ for any Pauli subgroup using at most two-body interactions. The caveat, however, is that while $\exp(i\theta\overline{X})$ and $\exp(i\theta\overline{Z})$ always preserve the code (since they are in the normalizer), the operations that generate them from Hamiltonians involving at most two-body interactions may corrupt the code, as explained in Sec. III above.

Let us then state the challenges ahead. We need to show how the Hamiltonians $\overline{X}$ and $\overline{Z}$ can be generated using (i) at most two-body interactions, (ii) fault tolerantly.

C. Simple example: The subgroup Q_4

Let us pause by introducing a simple example illustrating the notion of universal computation using normalizer elements which are two-body Hamiltonians. Our example uses a group whose natural structure is such that the two-body restriction is automatically satisfied. To this end, consider the subgroup $Q_4=\{I^{\otimes 4},X^{\otimes 4},Y^{\otimes 4},Z^{\otimes 4}\}$, which we studied in detail in paper I. It is generated by $K-l=4-l=2$ elements $(X^{\otimes 4},Z^{\otimes 4})$, and therefore encodes $l=2$ qubits, with states given by

$$\begin{aligned}|00\rangle_L&=\frac{1}{\sqrt{2}}(|0000\rangle+|1111\rangle),\\|01\rangle_L&=\frac{1}{\sqrt{2}}(|1001\rangle+|0110\rangle),\\|10\rangle_L&=\frac{1}{\sqrt{2}}(|1100\rangle+|0011\rangle),\\|11\rangle_L&=\frac{1}{\sqrt{2}}(|0101\rangle+|1010\rangle).\end{aligned} \tag{4.6}$$

These states are easily seen to be $+1$ eigenstates of Q_4. The normalizer in this case contains two $\overline{X}_i$ and $\overline{Z}_i$ operations, one for each encoded qubit:

$$\begin{aligned}\overline{X}_1&=XXII,\quad \overline{Z}_1=IZZI,\\\overline{X}_2&=IXXI,\quad \overline{Z}_2=ZZII.\end{aligned} \tag{4.7}$$

Indeed, we have, for example, $\overline{X}_1|a,b\rangle_L=|1-a,b\rangle_L$ and $\overline{Z}_1|a,b\rangle_L=(-1)^a|a,b\rangle_L$, so $\overline{X}_1$ and $\overline{Z}_1$ act, respectively, as a bit flip and a phase flip on the first encoded qubit. As easily checked, $\overline{X}_i$ and $\overline{Z}_i$ commute with Q_4, so that they keep states within the DFS, as should be the case for normalizer elements. As *Hamiltonians*, $\overline{X}_i$ and $\overline{Z}_i$ are valid two-body interactions and hence can be used directly to generate the encoded SU(2) group on each encoded qubit. That is, $\exp(i\alpha\overline{X}_i)$ and $\exp(i\beta\overline{Z}_i)$ can be combined directly, with arbitrary values for the angles α and β, to produce any operation in $\overline{\mathrm{SU}(2)}$ by using the Euler angle formula. For example, we

can construct a rotation about the encoded Y_i axis by conjugation: $\exp(i\theta\overline{Y_i})=\exp(-i(\pi/4)\overline{X_i})\exp(-i\theta\overline{Z_i})\exp(+i(\pi/4)\overline{X_i})$. We have, therefore, two independent encoded qubits that can be operated upon separately by encoded SU(2) operations.

What about coupling between the encoded qubits so that the full $\overline{\mathrm{SU}(4)}$ can be used to do computation? Note that Hamiltonians like $\overline{Z}_1\otimes\overline{Z}_2=ZIZI$, which are two-body on the encoded qubits, can be implemented directly since they are also two-body on the physical qubits (this is not a generic feature, however, as discussed in Sec. V D below). It is a fundamental theorem of universal quantum computation [37,39,42] that the ability to perform SU(2) on two qubits plus the ability to perform *any* nontrivial two-body *Hamiltonian* between these qubits is universal over the combined SU(4) of these two qubits. Thus we can perform universal computation on the Q_4 DFS. In this case, the normalizer elements that perform the $\overline{\mathrm{SU}(4)}$ are all two-body Hamiltonians, and there is no need to apply any new methods in order to perform fault-tolerant computation, which preserves this DFS.

Anticipating the discussion in Sec. VI, note that while we have demonstrated universal computation on a single DFS block, we have not yet addressed how to accomplish this when we have clusters of the Q_4 DFSs. This, of course, is necessary to scale up the quantum computer under the Q_4 model of decoherence. In order to perform universal fault-tolerant computation with clusters, we must show that these can be coupled in a nontrivial manner. Methods for performing nontrivial couplings between clusters exist for any stabilizer code [34]. In particular, the Q_4 DFS is a Calderbank-Shor-Steane (CSS) code, whose clusters can be coupled by performing bitwise parallel controlled-NOT gates between two clusters of qubits. This implements as desired an encoded controlled-NOT between these clusters. In Sec. VI, we will discuss what is needed to make this procedure fault-tolerant

Q_4 is a special case because of the fact that the normalizer elements are all two-body interactions. In general, the normalizer elements will be many-body interactions and more general techniques are needed, to which we turn next.

D. Generating $\overline{X}$ and $\overline{Z}$ using at most two-body interactions

We now move on to the general case in which the normalizer elements are possibly many-body Pauli operators. Our first task is to show that the ''conjugation-by-$(\pi/4)A$'' operation $T_A\circ\exp(i\theta B)$ can be used to generate any many-body Hamiltonian inside the exponent using at most two-qubit Hamiltonians. In Sec. V, we show that this is a fault-tolerant procedure if applied correctly to a DFS.

Suppose the many-body Pauli Hamiltonian H we want to generate is of the following general form:

$$H=\sigma_b^\beta\bigotimes_{j\in\mathcal{J}}\sigma_j^{\alpha_j}, \tag{4.8}$$

where $\mathcal{J}$ is some index set and $b\notin\mathcal{J}$. From Eq. (4.1), we have at our disposal a single-qubit Hamiltonian σ_b^β and a set of two-qubit Hamiltonians $A_j=\sigma_b^{\gamma_j}\otimes\sigma_j^{\alpha_j}$ with $j\in\mathcal{J}$ and $\gamma_j\neq\beta$. We call the bth qubit the ''base qubit.'' A_j and σ_b^β agree on one qubit index but differ on the Pauli matrix applied to that qubit, so they anticommute: $\{A_j,\sigma_b^\beta\}=0$. Let $\mathcal{J}(i)$ denote the ith element in the index set $\mathcal{J}$. If we use the ''conjugation-by-$(\pi/4)A_{\mathcal{J}(1)}$'' operation about $\exp(i\theta\sigma_b^\beta)$ [recall Eq. (4.5)], we obtain

$$\begin{aligned}T_{A_{\mathcal{J}(1)}}\circ\exp(i\theta\sigma_b^\beta)&=\exp[i\theta(i\sigma_b^{\gamma_{\mathcal{J}(1)}}\otimes\sigma_{\mathcal{J}(1)}^{\alpha_{\mathcal{J}(1)}})\sigma_b^\beta]\\&=\exp[\pm i\theta\sigma_b^{\eta_1}\otimes\sigma_{\mathcal{J}(1)}^{\alpha_{\mathcal{J}(1)}}],\end{aligned} \tag{4.9}$$

where the sign is determined by that of $\varepsilon_{\gamma_{\mathcal{J}(1)}\beta\eta_1}$, according to the usual rule of multiplication Pauli matrices:

$$\sigma^\alpha\sigma^\beta=-i\varepsilon_{\alpha\beta\gamma}\sigma^\gamma. \tag{4.10}$$

Applying all other ''conjugation-by-$(\pi/4)A_{\mathcal{J}(i)}$'' operations, $i=1\cdots|\mathcal{J}|$, we obtain

$$\begin{aligned}&T_{A_{\mathcal{J}(|\mathcal{J}|)}}\circ\cdots\circ T_{A_{\mathcal{J}(i)}}\circ\exp(i\theta\sigma_b^\gamma)\\&\quad=\exp(\pm i\theta\bigotimes_{j\in\mathcal{J}}\sigma_b^\eta\otimes\sigma_j^{\alpha_j}).\end{aligned} \tag{4.11}$$

It is clear that by appropriately choosing the sequence of Pauli matrices, i.e., the $\alpha_{\mathcal{J}(i)}$, we can obtain $\eta=\beta$. Further, conjugating by $-\pi/4$ (instead of $+\pi/4$) allows us to always adjust the sign in the exponent to $+$. Thus the action of this gate sequence is to generate the Hamiltonian H, as desired:

$$T_{A_{\mathcal{J}(|\mathcal{J}|)}}\circ\cdots\circ T_{A_{\mathcal{J}(i)}}\circ\exp(i\theta\sigma_b^\gamma)=\exp(i\theta H). \tag{4.12}$$

An example of this type of gate network (analyzed in detail in Sec. V C) is shown in Fig. 1. Since the elements of the normalizer of any stabilizer can always be written as a tensor product of single-qubit Pauli matrices, Eq. (4.12) gives a constructive way of generating these normalizer elements as *Hamiltonians* (i.e., appearing as arguments in the exponent). We have thus met the first challenge mentioned above: we have shown how to generate the Hamiltonians $\overline{X}$ and $\overline{Z}$ using at most two-body interactions. More generally, Eq. (4.12) can be considered as a constructive procedure for generating desired many-body Hamiltonians from given two-body interactions.

Finally, we note that it is perfectly possible to replace the central single-qubit Hamiltonian with a two-qubit one, specifically by $A_{\mathcal{J}(1)}\sigma_b^\gamma$. This may be more convenient for practical applications, where control of two-body interactions may be more easily achievable (as in the case of exchange interactions in quantum dots [44]). This change would not affect our fault-tolerance analysis in the next sections.

V. GENERATING ENCODED SU(2) FAULT TOLERANTLY FOR ANY ABELIAN PAULI SUBGROUP

We are now ready to show how to generate encoded SU(2) operations fault tolerantly for any Pauli error subgroup. Let Q be such a subgroup, generated by the elements

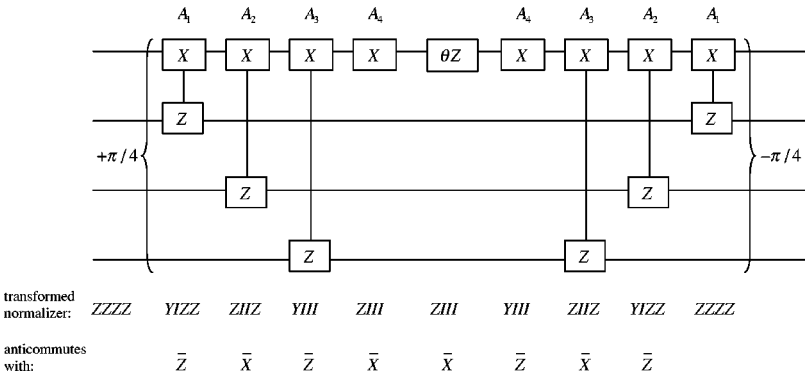

FIG. 1. Fault-tolerant circuit implementing $\exp(i\theta\overline{Z})$ for the Q_{2X} subgroup. The transformed $\overline{Z}$ is shown at each gate, and directly below the original normalizer element with which it anticommutes.

$\{q_i\}_{i=1}^n$, $|Q|=2^n$. Recall that here these elements play the dual role of errors and of defining the DFS by fixing its elements. A new (transformed) stabilizer is obtained after each application of a gate $\exp(i\varphi_j A_j)$. To this sequence of stabilizers corresponds a sequence of stabilizer QECCs $\mathcal{C}_j$. Our strategy will be to find conditions on the Hamiltonians $\{A_j\}$ such that after each gate application, the current QECC is able to correct the original Q errors.

Let Q_j $[N(Q_j)]$ denote the stabilizer (normalizer) obtained after application of the gate $U_j=\exp(i\varphi_j A_j)$. If φ_j is an integer multiple of $\pi/4$ (as we will always assume), then there are only three mutually exclusive possibilities for the errors $e\in Q$ (we use the notations e and q for members of Q to emphasize the error and stabilizer element aspects, respectively), as follows.

(i) $e\in Q_j$. The error is part of the transformed stabilizer. In this case, the transformed code is immune to e (i.e., the transformed code is a DFS with respect to e), and there is no problem.

(ii) e anticommutes with some element of Q_j. The error is detectable by the transformed code.

(iii) $e\in N(Q_j)/Q_j$ (i.e., e commutes with Q_j but is not in it). The error infiltrated the transformed normalizer. This is a problem since the error is *un*detectable by the transformed code, and acts on it in a nontrivial manner.

Suppose the errors $e\in Q$ are exclusively of type (i) or (ii). Then those that are of type (ii) are not only detectable but also correctable. This is so because they form a group (Q), and therefore any product of two errors is again either of type (i) or (ii), which is exactly the error correction criterion.[6] Thus the problematic case is (iii), and this is the case we focus on in order to make a prudent choice of Hamiltonians A_j. To simplify the notation, from now on we shall denote $N(Q_j)/Q_j$ simply by N_j (and by N when $Q_j=Q$), and refer to this as the normalizer (without risk of confusion).

Is there a simple criterion to check whether $e\in N_j$? The answer is contained in the theorem following this lemma.

Lemma 1. Let Q be a stabilizer over the Pauli group and let N be its normalizer, i.e., the set of all operations that commute with Q: $N=Q'$ (N is the *commutant* of Q). Then $N'=Q$ (i.e., the stabilizer is the commutant of the normalizer).

Proof. The Pauli group splits into three sets of operators: (i) the stabilizer Q, (ii) the normalizer N, (iii) the errors $\{E\}$, which anticommute with Q. The normalizer itself splits into two parts: the elements that are in the stabilizer, N_Q, and the elements that are not, N_P. Now, clearly the errors $\{E\}$ are not in the commutant of N, because they anticommute with N_Q. The elements of N_P are not in the commutant of N because as is well known from the theory of stabilizer codes, N_P forms a representation of a Pauli group (i.e., if we encode l qubits, then N_P is a representation of the Pauli group on l qubits). But, for every member of a representation of a Pauli group there is another element with which it anticommutes. Thus N_P cannot be in the commutant of N either. Finally, N_Q is clearly a member of the commutant of N, by definition of the normalizer.

Theorem 2. Given are a Pauli subgroup of errors Q, its normalizer N, and a sequence of their images $\{Q_j\}$ and $\{N_j\}$ under conjugation by unitaries $\{U_j\}$. Corresponding to Q is a DFS (code) $\mathcal{C}$. A sufficient condition so that no $e\in Q$ is ever in N_j is that either (i) each $n_j\in N_j$ equals its source in N, or (ii) for each $n_j\in N_j$ there exists $m\in N$ such that $\{n_j,m\}=0$. Then the transformed codes $\mathcal{C}_j=U_j\mathcal{C}_{j-1}$ $(\mathcal{C}_1=U_1\mathcal{C})$ can always correct the original Q errors.

Proof. From lemma 1 we know that $(Q')'=Q$. In other words, the only operations that commute with the normalizer are those in the stabilizer. Now let n_j be the image of $n\in N$ after the jth transformation. The observation $N'=Q$ allows us to exclude case (iii) by checking if, for every $n_j\in N_j$ (where $n_j\neq n$), there exists $m\in N$ with which n_j anticommutes. To see this, note first that if $n_j=n$, then by definition n_j cannot be in Q. Second, for an $n_j\in N_j$ that differs from its source in N, assume that it anticommutes with some $m\in N$. This implies that n_j is not in the commutant of N, and is therefore not in Q. If this is true for all $n_j\in N_j$, then we have covered the entire new normalizer N_j and not found an element of Q in it. This guarantees that no element of the original stabilizer Q becomes a member of the new normalizer N_j. QED.

Note that if the conditions of the theorem are satisfied, then *all* elements of the original stabilizer are excluded from the transformed normalizer. Therefore, also all products of stabilizer elements are excluded (since the stabilizer is a

[6]Note that this is not true for errors in the usual stabilizer-QECC case, where the errors do not close as a group under multiplication.

group), so that all stabilizer errors are both detectable and *correctable*.

Below we make repeated use of the result of theorem 2. The first application is to show how to construct two-body Hamiltonians $\{A_j\}$ that can be applied in succession to produce arbitrary normalizer elements, such that at every point the theorem is satisfied. To this end, we need a basic result from the theory of stabilizer codes, regarding a standard form for the normalizer. We then illustrate the general construction with the relatively simple case of CSS codes, and finally move on to general stabilizer errors.

A. Standard form of the normalizer for stabilizer codes

It is shown in [45] that, due to the fact that the normalizer is invariant under multiplication by stabilizer elements, the normalizer of every stabilizer code can be brought into the following standard form:

$$\bar{Z}_j=(\underbrace{I\otimes\cdots\otimes I\otimes Z_j\otimes I\otimes\cdots\otimes I}_{l})\otimes(\underbrace{M_Z^j}_{r})\otimes(\underbrace{I\otimes\cdots\otimes I}_{K-l-r}), \tag{5.1}$$

$$\bar{X}_j=(\underbrace{I\otimes\cdots\otimes I\otimes X_j\otimes I\otimes\cdots\otimes I}_{l})\otimes(\underbrace{N_Z^j}_{r})\otimes(\underbrace{M_X^j}_{K-l-r}). \tag{5.2}$$

Here $M_Z^j=\otimes_{n\in\mathcal{Z}_j}Z_n$, $N_Z^j=\otimes_{n'\in\mathcal{Z}'_j}Z_{n'}$, and $M_X^j=\otimes_{i\in\mathcal{X}_j}X_i$, where $\mathcal{Z}_j$, $\mathcal{Z}'_j$, and $\mathcal{X}_j$ are (possibly empty) index sets of lengths r, r, and $K-l-r$, respectively (i.e., M_Z^j, N_Z^j, and M_X^j are tensor products of I's and single-qubit Pauli Z and X matrices, respectively). Recall that K is the number of physical qubits; l is the number of encoded qubits. The exact form of M_Z^j, N_Z^j, and M_X^j, as well as the value of the integer r, can be found from the stabilizer [45], but is unimportant for our purposes. We only need the result that for every pair of encoded Z and X operations, acting on the jth encoded qubit, it is possible to express the operations in the blockwise product shown in Eqs. (5.1) and (5.2).

B. CSS stabilizer errors on one encoded qubit

For simplicity, let us now restrict attention to the case of a single encoded qubit in CSS codes, i.e., those codes where every $\overline{Z}$ and $\overline{X}$ can be written as a product of only Z's and only X's, respectively. Then, from Eqs. (5.1) and (5.2) the standard form is (dropping the j index)

$$\overline{Z}=Z_1\otimes M_Z\otimes I^{\otimes K-l-r}, \tag{5.3}$$

$$\overline{X}=X_1\otimes I^{\otimes r}\otimes M_X, \tag{5.4}$$

i.e., $N_Z=I^{\otimes r}$. Our goal is to construct such $\overline{Z}$ and $\overline{X}$ from single- and two-body Hamiltonians. We shall do this by starting from the single-body Hamiltonians Z_1 and X_1, and conjugating by certain two-body Hamiltonians. The idea is to successively construct the Z's in M_Z and the X's in M_X. We claim that the required two-body Hamiltonians have the natural form

$$A_n=X_1Z_{z_n},\quad z_n\in\mathcal{Z}, \tag{5.5}$$

$$B_i=Z_1X_{x_i},\quad x_i\in\mathcal{X}, \tag{5.6}$$

where $n=1\cdots|\mathcal{Z}|$ and $i=1\cdots|\mathcal{X}|$, i.e., A_n (B_i) has a Z (X) in the nth (ith) position of the index set $\mathcal{Z}$ ($\mathcal{X}$). If there is an even number of Z's in $\overline{Z}$, then the last Hamiltonian should be taken as $A_{|\mathcal{Z}|}=X_1$ (since, as we show below, in that case in the penultimate step we have $Y_1\otimes M_Z\otimes I^{\otimes K-l-r}$ for $\overline{Z}$), and similarly for the last B_i.[7] Note that $[A_n,\overline{X}]=[B_i,\overline{Z}]=0$, so that transforming $\overline{Z}$ does not affect $\overline{X}$, and vice versa. There are now two ways to construct $\overline{Z}$ and $\overline{X}$ fault tolerantly: in parallel or in series. The parallel implementation has the advantage that it requires only three basic steps and thus is very efficient. Its disadvantage is that it may be hard to implement in practice because it requires simultaneous control over many qubits.

1. Series Construction

We assume throughout this discussion that we wish to generate $\exp(i\theta\overline{Z})$. The symmetry between $\overline{Z}$ and $\overline{X}$ in the CSS case implies that our arguments hold for $\exp(i\theta\overline{X})$ as well, with obvious modifications.

The series construction consists of applying first the sequence of gates $\{\exp(i(\pi/4)A_n)\}_{n=1}^{|\mathcal{Z}|}$, then the gate $\exp(i\theta Z_1)$, and then the reverse sequence of gates $\{\exp(-i(\pi/4)A_n)\}_{n=|\mathcal{Z}|}^{1}$. An example is shown in Fig. 1. First, as an application of the general Eq. (4.12), let us prove that this procedure indeed generates $\exp(i\theta\overline{Z})$:

$$\begin{aligned}
&T_{A_1}\circ T_{A_2}\circ\cdots T_{A_{|\mathcal{Z}|}}\circ\exp(i\theta Z_1)\\
&\quad=\left[\bigotimes_{n=1}^{|\mathcal{Z}|}\exp\left(-i\frac{\pi}{4}A_n\right)\right]\exp(i\theta Z_1)\\
&\qquad\times\left[\bigotimes_{n=|\mathcal{Z}|}^{1}\exp\left(+i\frac{\pi}{4}A_n\right)\right]\\
&\quad=\exp\left[i\theta\left(i^{|\mathcal{Z}|}\prod_{n=1}^{|\mathcal{Z}|}A_nZ_1\right)\right]\\
&\quad=\exp[(-)^{|\mathcal{Z}|}i\theta\overline{Z}],
\end{aligned} \tag{5.7}$$

where in the first line we used the definition of the ''conjugation-by-$(\pi/4)A_n$'' operation, in the second the result that this operation corresponds to multiplication inside the exponent, and in the third the form in Eq. (5.3) for $\overline{Z}$. Note that the reason we have a series of conjugation-by-$(\pi/4)A_n$ operations (as opposed to trivial identity operations)

[7] For example, suppose $\overline{Z}=Z(ZZZ)(II)$ and $\overline{X}=X(III)(XX)$; then $A_1=X_1Z_2$, $A_2=X_1Z_3$, $A_3=X_1Z_4$, $A_4=X_1$, $B_1=Z_1X_5$, and $B_2=Z_1X_6$. Then we have $Z(III)(II)\overset{A_1}{\mapsto}Y(ZII)(II)\overset{A_2}{\mapsto}Z(ZZI)(II)\overset{A_3}{\mapsto}Y(ZZZ)(II)\overset{A_4}{\mapsto}\overline{Z}$, and $X(III)(II)\overset{B_1}{\mapsto}Y(III)(XI)\overset{B_2}{\mapsto}\overline{X}$.

is that $\{\Pi_{n=1}^{k-1}A_nZ_1,A_k\}=0\,\forall k\leqslant|\mathcal{Z}|$. Finally, we can eliminate the minus sign (if $|\mathcal{Z}|$ is odd) by changing one of the $\pi/4$'s to $-\pi/4$.

Next we must demonstrate that the conditions of theorem 2 are satisfied at each point in the corresponding circuit in order to guarantee the fault tolerance of this implementation. Let us divide the proof into three parts, by following the transformations of the normalizer elements before and after the central $\exp(i\theta Z_1)$ gate, and showing that either $\overline{Z}$ or $\overline{X}$ anticommutes with the transformed normalizer at each step along the way.

Errors before the central gate. After application of the first k gates $\{\exp(i(\pi/4)A_n)\}_{n=1}^{k}$, $\overline{Z}$ is transformed to $\overline{Z}^{(k)}\equiv\Pi_{n=1}^{k}A_n\overline{Z}$ (we neglect the unimportant factors of i from now on). From Eq. (5.5) the product is

$$\prod_{n=1}^{k}A_n=\begin{cases}\displaystyle\prod_{n=1}^{k}Z_{z_n} & \text{if } k=2l\\ \displaystyle X_1\prod_{n=1}^{k}Z_{z_n} & \text{if } k=2l+1.\end{cases}\tag{5.8}$$

Therefore, using the standard form,

$$\begin{aligned}\left(\prod_{n=1}^{2l}A_n\overline{Z}\right)\overline{X}&=\prod_{n=1}^{2l}Z_{z_n}\overline{ZX}\\&=-\overline{X}\prod_{n=1}^{2l}Z_{z_n}\overline{Z}\\&=-\overline{X}\left(\prod_{n=1}^{2l}A_n\overline{Z}\right),\end{aligned}\tag{5.9}$$

so that $\{\overline{Z}^{(2l)},\overline{X}\}=0$. On the other hand,

$$\begin{aligned}\left(\prod_{n=1}^{2l+1}A_n\overline{Z}\right)\overline{Z}&=X_1\prod_{n=1}^{2l+1}Z_{z_n}\overline{ZZ}\\&=-\overline{Z}X_1\prod_{n=1}^{2l+1}Z_{z_n}\overline{Z}\\&=-\overline{Z}\left(\prod_{n=1}^{2l+1}A_n\overline{Z}\right),\end{aligned}\tag{5.10}$$

so that $\{\overline{Z}^{(2l+1)},\overline{Z}\}=0$. Thus theorem 2 is satisfied after each gate application, with $\overline{Z}$ and $\overline{X}$ alternating in the role of the anticommuting original-normalizer element.

Error immediately after the central gate. At the end of step (i), $\overline{Z}$ has been transformed to Z_1. Since the central gate (θ rotation) uses only Z_1, the transformed $\overline{Z}$ does not change. Therefore, is still anticommutes with the original $\overline{X}$ and satisfies the criterion of theorem 2. For the same reason, however, $\overline{X}$ is transformed by the central gate:

$$\overline{X}\mapsto\overline{X}_\theta=\overline{X}\cos(2\theta)+i\overline{X}Z_1\sin(2\theta).\tag{5.11}$$

Thus it suffices to show that $\overline{X}_\theta$ anticommutes with $\overline{Z}$, which is true since $[\overline{Z},Z_1]=0$:

$$\begin{aligned}\overline{X}_\theta\overline{Z}&=\overline{XZ}\cos(2\theta)+i\overline{X}Z_1\overline{Z}\sin(2\theta)\\&=-\overline{ZX}\cos(2\theta)-i\overline{ZX}Z_1\sin(2\theta)\\&=-\overline{ZX}_\theta.\end{aligned}\tag{5.12}$$

Errors after the central gate. After application of the first k' inverse gates $\{\exp(-i(\pi/4)A_n)\}_{n=|\mathcal{Z}|}^{|\mathcal{Z}|-k'+1}$, Z_1 is transformed to $\overline{Z}'^{(k)}\equiv\Pi_{n=1}^{k'}A_n\overline{Z}$. Therefore, the same reasoning as in (i) applies to $\overline{Z}'^{(k)}$. As for $\overline{X}$ (which is now $\overline{X}_\theta$), the $\overline{X}\cos(2\theta)$ component commutes with the inverse gates $\exp(-i(\pi/4)A_n)$ so that it does not change. The $i\overline{X}Z_1\sin(2\theta)$ component, however, anticommutes with the inverse gates $\exp(-i(\pi/4)A_n)$. Therefore, it flips back and forth between $i\overline{X}Z_1\sin(2\theta)$ and $i\overline{X}Y_1\sin(2\theta)$. These terms anticommute with the original $\overline{Z}$ and $\overline{Y}$, respectively. But so does the $\overline{X}\cos(2\theta)$ component, so their sum anticommutes alternately with the original $\overline{Z}$ and $\overline{Y}$.

We conclude that theorem 2 is satisfied at each stage of the circuit. Therefore, the series construction is indeed fault tolerant. Of course, this fault tolerance is achieved in practice by supplementing the circuit with error-detection and -correction procedures after each gate (the parallel construction discussed next is much more economical for this reason). We discuss this issue in Sec. VII.

2. Parallel construction

Since the A_n (B_i) all commute, the corresponding gates can also be implemented *in parallel*. That is,

$$\begin{aligned}U_A&\equiv\bigotimes_{n\in\mathcal{Z}}\exp\left(i\frac{\pi}{4}A_n\right)=\exp\left(i\frac{\pi}{4}\sum_{n\in\mathcal{Z}}A_n\right),\\U_B&\equiv\bigotimes_{i\in\mathcal{X}}\exp\left(i\frac{\pi}{4}B_i\right)=\exp\left(i\frac{\pi}{4}\sum_{i\in\mathcal{X}}B_i\right),\end{aligned}\tag{5.13}$$

can be used as parallel gates in our circuit (see Fig. 2 for an example). To see directly that this circuit really does implement the normalizer gate $\exp(i\theta\overline{Z})$ [or $\exp(i\theta\overline{X})$], observe that, by definition $\{A_n,Z_1\}=\{B_i,X_1\}=0$ for all n and i. This means that conjugation of Z_1 by U_A will act as multiplication by $\Pi_{n\in\mathcal{Z}}A_n$ and thus transform Z_1 to $\overline{Z}$ (without changing X_1). The same is true for X_1 by changing Z's to X's and U_A to U_B. Therefore, $U_AZ_1U_A^\dagger=\overline{Z}$ and $U_BX_1U_B^\dagger=\overline{X}$, from which it follows immediately by Taylor expansion that

$$\begin{aligned}U_A\exp(i\theta Z_1)U_A^\dagger&=\exp(i\theta\overline{Z}),\\U_B\exp(i\theta X_1)U_B^\dagger&=\exp(i\theta\overline{X}).\end{aligned}\tag{5.14}$$

This too is a fault-tolerant construction. The reason is that it corresponds to looking at the series construction just at the

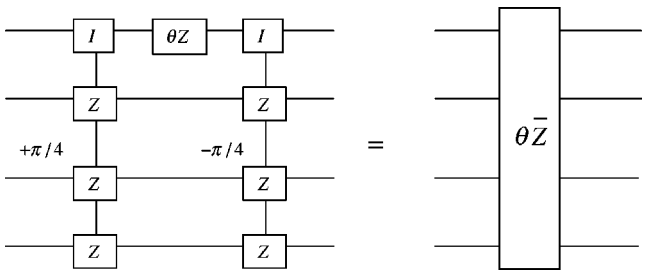

FIG. 2. Parallel implementation of $\theta\overline{Z}$ for the Q_{2X} subgroup.

following three points: right before the central gate, right after the central gate, and the end.

C. Example: The subgroup Q_{2X}

As an example with a many-body normalizer element, consider the Pauli subgroup/stabilizer generated by the errors $XXII$, $IXXI$, $IIXX$:

$$Q_{2X}=\{IIII,XXII,XIIX,IIXX,XIXI,IXXI,IXIX,XXXX\}. \tag{5.15}$$

It describes a physically interesting error model, of bit-flip errors that act on all pairs of nearest-neighbor qubits. This situation is of interest, e.g., when decoherence results from spin-rotation coupling in a dipolar Hamiltonian, typical in NMR [46]:

$$H_I=\sum_{j,k}\frac{\gamma_j\gamma_k}{r_{jk}^3}[\boldsymbol{\sigma}_j\cdot\boldsymbol{\sigma}_k-3(\boldsymbol{\sigma}_j\cdot\mathbf{r}_{jk})(\boldsymbol{\sigma}_k\cdot\mathbf{r}_{jk})]. \tag{5.16}$$

Here γ_j is the gyromagnetic ratio of spin j, and r_{jk} is the distance between spins j and k. In the anistropic case (e.g., a liquid crystal) this can be rewritten as

$$H_I=\sum_{j,k}\frac{\gamma_j\gamma_k}{r_{jk}^3}\sum_{\alpha,\beta=-1}^{1}g_{jk}^{\alpha\beta}(\sigma_j^{\alpha}\otimes\sigma_k^{\beta})Y_2^{-\alpha-\beta}, \tag{5.17}$$

where Y_l^m are the spherical harmonics and $g_{jk}^{\alpha\beta}$ is the anisotropy tensor. When $g_{jk}^{\alpha\beta}=\delta_{\alpha 0}\delta_{\beta 0}g_{jk}$, only the $\sigma_j^z\otimes\sigma_k^z$ terms remain (coupled to Y_2^0), which leads to decoherence described by the subgroup Q_{2Z} (defined similarly to Q_{2X}), analyzed in paper I.

To find the DFS under Q_{2X}, we construct in accordance with the techniques of paper I the projector $P=\frac{1}{8}\Sigma_i q_i$ (corresponding to the identity irrep of Q_{2X}), where the sum is over all $q_i\in Q_{2X}$. Applying this projector to an arbitrary initial state, we find a two-dimensional DFS, spanned by the states

$$\begin{aligned}|0_L\rangle&=(|0000\rangle+|0011\rangle+|0101\rangle+|0110\rangle+|1001\rangle\\&\quad+|1010\rangle+|1100\rangle+|1111\rangle)/\sqrt{8},\\|1_L\rangle&=(|0001\rangle+|0010\rangle+|0100\rangle+|0111\rangle+|1000\rangle\\&\quad+|1011\rangle+|1101\rangle+|1110\rangle)/\sqrt{8}.\end{aligned} \tag{5.18}$$

This DFS thus encodes a full qubit.

Since for Q_{2X} there is just one encoded qubit, we expect to find just one $\overline{X}$ and one $\overline{Z}$. In the case of Q_{2X} it is easily verified that the normalizer is generated by

$$\begin{aligned}\overline{X}&=XIII,\\\overline{Z}&=ZZZZ.\end{aligned} \tag{5.19}$$

$\overline{X}$ is already a single-body Hamiltonian and therefore can be implemented directly. Let us show how $\overline{Z}$ can be implemented as a Hamiltonian using at most two-body interactions.

Note that Q_{2X} supports a CSS code. Comparing the above expressions for $\overline{Z}$ to the standard form for CSS normalizer elements [Eq. (5.3)], we have $M_Z=Z_2Z_3Z_4$ and $M_X=\emptyset$. Therefore, from the recipe of Eq. (5.5), $A_n=X_1Z_{n+1}$ for $n=1\cdots 3$ while $A_4=X_1$. The series-circuit implementing $\exp(i\theta\overline{Z})$ thus has the form depicted in Fig. 1. The parallel version of the same circuit is shown in Fig. 2. To verify directly that these circuits indeed implement $\exp(i\theta\overline{Z})$, use Eq. (4.12) and choose the base qubit to be the first qubit. Then

$$T_{XZII}\circ T_{XIZI}\circ T_{XIIZ}\circ T_{XIII}\circ\exp(i\theta ZIII)=\exp(i\theta ZZZZ). \tag{5.20}$$

As required, this is an implementation that uses at most two-body interactions.

Figure 1 also shows the transformed $\overline{Z}$ at each point, and directly below the original normalizer element ($\overline{X}$ or $\overline{Z}$) with which this transformed normalizer element anticommutes. This verifies that the circuit is indeed a fault-tolerant implementation of $\exp(i\theta\overline{Z})$ for Q_{2X}.

D. CSS stabilizer errors on multiple encoded qubits

The CSS case of more than one encoded qubit is a simple extension of the single encoded qubit case discussed above. From Eqs. (5.1) and (5.2) the standard form for a CSS code is now

$$\overline{Z_j}=Z_j\otimes M_Z^j\otimes I^{\otimes K-l-r}, \tag{5.21}$$

$$\overline{X_j}=X_j\otimes I^{\otimes r}\otimes M_X^j. \tag{5.22}$$

Operations on different encoded qubits j,j' commute. Therefore, the single encoded qubit constructions still holds when the Hamiltonians are modified to read

$$A_n^{(j)}=X_jZ_{z_n},\quad z_n\in\mathcal{Z}_j \tag{5.23}$$

$$B_i^{(j)}=Z_jX_{x_i},\quad x_i\in\mathcal{X}_j. \tag{5.24}$$

As is easily checked, the entire proof for the single encoded qubit case carries through when the base qubit becomes physical qubit number j instead of number 1. This thus allows us to fault tolerantly implement $\overline{\mathrm{SU}(2)}^{\otimes l}$ on all l encoded qubits.

To couple encoded qubits within the same block [thus generating $\overline{\mathrm{SU}(2^l)}$], one could use a standard trick from stabilizer theory [34], using an auxiliary block to swap information into and out of. This transversal operation involves applying encoded controlled-NOT operations, which we treat in Sec. VI below. In that section, we also show how coupling multiple encoded qubits can be achieved directly, without resorting to an auxiliary block.

E. General stabilizer errors

The entire analysis for the CSS case carries through in the general stabilizer case for the implementation of $\exp(i\theta\overline{Z})$, since $\overline{Z}$ remains unchanged [recall Eq. (5.1)]. However, the encoded X operation now includes the additional block N_Z: $\overline{X}=X_1\otimes N_Z\otimes M_X$ [Eq. (5.2)]. Therefore, to generate this operation we must include a new set of Hamiltonians:

$$C_{n'}=Z_1Z_{n'}, \quad n'\in\mathcal{Z}'. \tag{5.25}$$

If there is an even number of Z's in $\overline{Z}$, then the last Hamiltonian should be taken as $C_{|\mathcal{Z}'|}=Z_1$. We now need to repeat the analysis for the generation of $\exp(i\theta\overline{X})$. Again, there is a series and a parallel construction. Since the $C_{n'}$ and B_i all commute, the gate

$$\begin{aligned} U_{BC}&\equiv U_B\otimes U_C \\ &=\left[\bigotimes_{i\in\mathcal{X}}\exp\left(i\frac{\pi}{4}B_i\right)\right]\otimes\left[\bigotimes_{n'\in\mathcal{Z}'}\exp\left(i\frac{\pi}{4}C_{n'}\right)\right] \\ &=\exp\left[i\frac{\pi}{4}\left(\sum_{i\in\mathcal{X}}B_i+\sum_{n'\in\mathcal{Z}'}C_{n'}\right)\right] \end{aligned} \tag{5.26}$$

can be implemented in parallel. Conjugation of $\exp(i\theta X_1)$ by U_{BC} will yield $\exp(i\theta\overline{X})$ by Eq. (4.12), since $\{X_1,B_i\}=\{X_1,C_{n'}\}=0$. It is further straightforward to check that this is a fault-tolerant implementation, since the arguments used in the case of a single encoded CSS qubit are still valid here.

We are thus left to check only the series construction. Here the only new element is that we must make sure that the application of the $C_{n'}$ Hamiltonians does not allow for undetectable errors to take place. Apart from this, everything is the same as in the CSS case. Now, after application of the first k gates $\{\exp(i(\pi/4)C_{n'})\}_{n'=1}^{k}$, $\overline{X}$ is transformed to $\overline{X}^{(k)}\equiv\Pi_{n'=1}^{k}C_{n'}\overline{X}$. This product is

$$\prod_{n'=1}^{k}C_{n'}=\begin{cases}\displaystyle\prod_{n'\in\mathcal{Z}'_k}Z_{n'} & \text{if } k=2l \\ \displaystyle Z_1\prod_{n'\in\mathcal{Z}'_k}Z_{n'} & \text{if } k=2l+1,\end{cases} \tag{5.27}$$

where $\mathcal{Z}'_k$ are the first k elements of the index set $\mathcal{Z}'$. Therefore,

$$\begin{aligned} \{\overline{X}^{(k)},\overline{Z}\}&=\left[(Z_1)^k\prod_{n'\in\mathcal{Z}'_k}Z_{n'}\overline{X}\right]\overline{Z}+\overline{Z}\left[(Z_1)^k\prod_{n'\in\mathcal{Z}'_k}Z_{n'}\overline{X}\right] \\ &=\left[(Z_1)^k\prod_{n'\in\mathcal{Z}'_k}Z_{n'}\right]\{\overline{X},\overline{Z}\}=0. \end{aligned} \tag{5.28}$$

Thus theorem 2 is satisfied after each $C_{n'}$-gate application, with $\overline{Z}$ playing the role of the anticommuting original-normalizer element. This means that use of the Hamiltonians $C_{n'}$ does not spoil the fault tolerance of the circuit. We know from the calculations in the single encoded qubit case that the rest of the circuit is also fault tolerant. Hence we can conclude at this point that our method of constructing normalizer elements is fault-tolerant for any stabilizer code.

F. Summary

Let us recapitulate the main result of this section. Given a set of errors corresponding to some Abelian subgroup of the Pauli group (i.e., a stabilizer), there is a DFS that is immune to these errors. We have shown how to implement arbitrary encoded SU(2) operations on this class of DFSs. To do so, we gave an explicit construction of encoded σ_x and σ_z operations, which together span encoded SU(2)'s for each DFS qubit. The construction involves turning on and off a series of one- and two-body Hamiltonians for a specific duration. Each such operation takes the encoded states outside of the DFS. However, our construction guarantees that the errors always remain correctable by the code formed by the transformed states. That is, these states form a QECC with respect to the Pauli subgroup errors. Therefore, our construction works by supplementing the unitary gates executing the encoded σ_x and σ_z operations by appropriate error-correction procedures. To complete the construction, we still need to show how to execute encoded two-body gates, and how to fault tolerantly measure the error syndrome. This is the subject of the next two sections.

VI. ENCODED CONTROLLED-NOT

The unitary CNOT operation from the first qubit (''control qubit) to the second qubit (''target qubit'') can be written in the basis of σ_z eigenstates as

$$U_{\mathrm{CNOT}}=\begin{pmatrix}I & 0\\ 0 & X\end{pmatrix} \tag{6.1}$$

(where each entry is a 2×2 matrix). Since we are working in the Heisenberg picture, it is useful to consider how two-qubit operators transform under CNOT. For example,

$$\begin{aligned} X\otimes I\mapsto U_{\mathrm{CNOT}}(X\otimes I)U_{\mathrm{CNOT}}^{\dagger}&=\begin{pmatrix}I & 0\\ 0 & X\end{pmatrix}\begin{pmatrix}0 & I\\ I & 0\end{pmatrix}\begin{pmatrix}I & 0\\ 0 & X\end{pmatrix} \\ &=\begin{pmatrix}0 & X\\ X & 0\end{pmatrix}. \end{aligned} \tag{6.2}$$

As is simple to verify, the full transformation table is

$$\begin{aligned} X\otimes I &\mapsto X\otimes X,\\ I\otimes X &\mapsto I\otimes X,\\ Z\otimes I &\mapsto Z\otimes I,\\ I\otimes Z &\mapsto Z\otimes Z. \end{aligned} \tag{6.3}$$

Since $U(A\otimes B)U^\dagger = U(A\otimes I)U^\dagger U(I\otimes B)U^\dagger$, the rest of the transformations under CNOT follow simply by taking appropriate products of the above, e.g., $X\otimes Z=(X\otimes I)(I\otimes Z)\mapsto(X\otimes X)(Z\otimes Z)=-Y\otimes Y$. We need to show how to fault tolerantly construct an encoded CNOT operation for the DFS corresponding to a given Pauli subgroup of errors.

A. CSS stabilizer errors

It is well known that a bitwise CNOT gate between physical qubits in different blocks is an operation that preserves any CSS code, and acts as the encoded CNOT gate between the blocks encoding different qubits [34]. However, this is true only at the *conclusion* of the operation, i.e., after all the bitwise operations have been applied. During the execution of the bitwise operations, the codewords are exposed to errors. To demonstrate this, consider the transformation of the normalizer elements of a CSS code. Let CNOT_{j_A,j_B} denote the CNOT operation from control qubit j (in the first block A) to target qubit j (in the second block B). For definiteness, let us consider the transformation of $\overline{X}_j\otimes I^{\otimes K}$ under bitwise CNOT's. Then, because of the standard form for $\overline{X}_j$, the first CNOT operation is applied from control qubit j, and subsequent CNOT's from control qubits are determined by the index set $\mathcal{X}$, i.e., acting on pairs of physical qubits at positions $\{(i_A,i_B)\}_{i\in\mathcal{X}}$. Using Eqs. (5.21) and (5.22) for $\overline{Z}_j$ and $\overline{X}_j$, and the transformation table of Eq. (6.3), we find

$$\begin{aligned} \overline{X}_j\otimes I^{\otimes K} &= [X_j\otimes I^{\otimes r}\otimes M_X]\otimes[I^{\otimes l}\otimes I^{\otimes r}\otimes I^{\otimes K-l-r}]\\ &\overset{\mathrm{CNOT}_{j_A,j_B}}{\mapsto} [X_j\otimes I^{\otimes r}\otimes M_X]\otimes[X_j\otimes I^{\otimes r}\otimes I^{\otimes K-l-r}]\\ &\overset{\mathrm{CNOT}_{i_{1A},i_{1B}}}{\mapsto} [X_j\otimes I^{\otimes r}\otimes M_X]\otimes[X_j\otimes I^{\otimes r}\otimes X_{i_1}]\\ &\mapsto\cdots\mapsto [X_j\otimes I^{\otimes r}\otimes M_X]\otimes[X_j\otimes I^{\otimes r}\otimes M_X]\\ &= \overline{X}_j\otimes\overline{X}_j. \end{aligned} \tag{6.4}$$

Similarly, one can check that the rest of the transformations of Eq. (6.3) are satisfied at the encoded level. Therefore, this calculation demonstrates that the full bitwise CNOT gate indeed acts as an *encoded* CNOT operation, since it transforms encoded normalizer operations according to the transformation rules of CNOT, as per Eq. (6.3). In our context, this implies that given a certain Pauli subgroup of errors, application of the full bitwise CNOT gate will implement the $\overline{\mathrm{CNOT}}$ gate on the DFS in a way that keeps the codewords inside the DFS at the end of the operation. However, as in the $\overline{\mathrm{SU}(2)}$ case, this is not true at intermediate steps, meaning that the code leaves the DFS.[8] As in the $\overline{\mathrm{SU}(2)}$ case, we must check that the original errors are still correctable at each intermediate step. Theorem 2 will still apply if error-correction procedures are implemented on each block separately, after each bitwise CNOT operation (since the blocks are only coupled during the execution of the CNOT). Therefore, we need to check that for each block in which the normalizer changed, there exists an element in the original normalizer that anticommutes with the transformed normalizer. It is easy to see from Eq. (6.4) that $\overline{X}_j$ does not change in the first block, and the sequence of transformed $\overline{X}_j$'s in the second block anticommutes with $\overline{Z}_j$ at every step. Therefore, error correction is possible at each intermediate step.

To complete the construction, it is necessary to check that the remaining normalizer elements are appropriately transformed. Repeating the calculation of Eq. (6.4), it is straightforward to check that this is true, namely

$$\begin{aligned} I^{\otimes K}\otimes\overline{X}_j &\mapsto I^{\otimes K}\otimes\overline{X}_j,\\ \overline{Z}_j\otimes I^{\otimes K} &\mapsto \overline{Z}_j\otimes I^{\otimes K},\\ I^{\otimes K}\otimes\overline{Z}_j &\mapsto \overline{Z}_j\otimes\overline{Z}_j, \end{aligned} \tag{6.5}$$

with $I^{\otimes K}\otimes\overline{X}_j$ and $\overline{Z}_j\otimes I^{\otimes K}$ invariant under the bitwise CNOT's (thus requiring no error correction), and the transformed $I^{\otimes K}\otimes\overline{Z}_j$ anticommuting at each step with the original $\overline{X}_j$.

This completes our demonstration that a $\overline{\mathrm{CNOT}}$ gate can be implemented fault tolerantly using bitwise CNOT's in the CSS case.

B. General stabilizer errors

In the non-CSS case, the bitwise CNOT does not act as a $\overline{\mathrm{CNOT}}$. One quick way to realize this is to note that since $X\otimes I\mapsto X\otimes X$, by unitarity $X\otimes X\mapsto X\otimes I$, but this is not the case at the encoded level:

$$\begin{aligned} \overline{X}\otimes\overline{X} &= [X_1\otimes N_Z\otimes M_X]\otimes[X_{K+1}\otimes N_Z\otimes M_X]\\ &\mapsto [X_1\otimes I^{\otimes r}\otimes M_X]\otimes[I_{K+1}\otimes N_Z\otimes I^{\otimes K-l-r}]\\ &\neq \overline{X}\otimes I^{\otimes K}. \end{aligned}$$

Thus a different implementation of the $\overline{\mathrm{CNOT}}$ is needed. Now, it is clear that if the product of stabilizers for different blocks (each encoding one qubit or more) is mapped to itself at the end of the $\overline{\mathrm{CNOT}}$ implementation, then the stabilizer errors will not have changed, the DFS qubits will not have changed, and thus the DFS code still offers protection against the stabilizer errors. Gottesman [34] has given such an implementation of the $\overline{\mathrm{CNOT}}$ for arbitrary stabilizer codes. It uses transformations involving four blocks at a time where

[8] Note that this is equally true for stabilizer QECCs, which are thus exposed to errors during gate execution.

two blocks serve as ancillas and are discarded after a measurement at the end of the implementation. We will not repeat this analysis here—the interested reader is referred to p. 133 of [34] for details. The faster the gate sequence implementing this $\overline{\text{CNOT}}$ is executed compared to the time scale for the errors to appear, the higher the probability that the code will not be taken outside of the DFS. However, as shown in Appendix A, the gate sequence (Fig. 2 of [34]) does not have the property we have been able to demonstrate above for all our constructions, i.e., it allows for errors to become part of the transformed normalizer. Therefore, we cannot use this construction. Instead, we now introduce a different construction for the $\overline{\text{CNOT}}$, in the spirit of what we have done above for the $\overline{\text{SU}(2)}$ operations.

Consider two blocks A and B encoding one DFS qubit each. We already know how to implement $\exp(i\theta I_A\otimes\overline{X}_B)$. Suppose one can also implement $\exp(i\theta\overline{Z}_A\otimes\overline{X}_B)$. Then by use of the Trotter formula $\exp[i(t_1O_1+t_2O_2)/n]=\lim_{n\to\infty}[\exp(i(t_1/n)O_1)\exp(i(t_2/n)O_2)]^n$ [47], or its short-time approximation

$$\exp[it(O_1+O_2)/n]=\exp[itO_1/n]\exp[itO_2/n]+O(n^{-2}) \tag{6.6}$$

valid for arbitrary operators O_1 and O_2, we can form, to any desired accuracy,

$$\exp[i\theta(I_A\otimes\overline{X}_B-\overline{Z}_A\otimes\overline{X}_B)/2]=\begin{pmatrix} I & 0 \\ 0 & \exp(i\theta\overline{X}_B)\end{pmatrix}. \tag{6.7}$$

For $\theta=\pi/2$ this is the $\overline{\text{CNOT}}$ operation between the two blocks. Thus our problem reduces to showing how $\exp(i\theta\overline{Z}_A\otimes\overline{X}_B)$ can be implemented fault tolerantly for arbitrary stabilizer DFSs.

Consider the circuit shown in Fig. 3. It describes the implementation of $\overline{Z}$ and $\overline{X}$ operations, as in the $\overline{\text{SU}(2)}$ case, with the difference that the single-body central gates have been replaced with a two-body gate, generated by the Hamiltonian $H_{AB}=Z_1^A\otimes X_1^B$ (here A and B are the two blocks and the subscript 1 indicates the first physical qubit in each block). By the $\overline{\text{SU}(2)}$ construction, we have that $U_AZ_1^AU_A^\dagger=\overline{Z}_A$ and $U_BX_1^BU_B^\dagger=\overline{X}_B$ (recall Sec. V B 2). Therefore, using the fact that for any nonsingular matrix M the equality $M\exp(H)M^{-1}=\exp(MHM^{-1})$ holds, the gates in Fig. 3 yield

$$\begin{aligned}(U_A\otimes U_B)&\exp(i\theta H_{AB})(U_A^\dagger\otimes U_B^\dagger)\\ &=\exp[i\theta(U_A\otimes U_B)H_{AB}(U_A^\dagger\otimes U_B^\dagger)]\\ &=\exp[i\theta(U_AZ_1^AU_A^\dagger)\otimes(U_BX_1^BU_B^\dagger)]\\ &=\exp(i\theta\overline{Z}_A\otimes\overline{X}_B),\end{aligned} \tag{6.8}$$

as desired.

It remains to verify that this is a fault-tolerant construction. The only difference compared to the $\overline{\text{SU}(2)}$ construction above is the fact that we are now using a *two*-body central Hamiltonian. It is reasonable to assume that if the system can couple the two blocks connected by this Hamiltonian, then so can the environment. Therefore, instead of considering the error subgroups Q_A and Q_B separately, we must now consider the new error subgroup $Q_A\times Q_B$. But then the appropriate normalizer is $N_{AB}=N_A\times N_B$, and the sequence of transformed normalizers satisfies $N_{AB,j}=N_{A,j}\times N_{B,j}$. This makes the fault-tolerance verification task very simple: We already checked in our $\overline{\text{SU}(2)}$ discussion that theorem 2 is satisfied for each block separately. Now, clearly both $N_A\otimes I_B, I_A\otimes N_B\in N_{AB}$. Therefore, since for every transformed normalizer element in $N_{A,j}$ $[N_{B,j}]$ there is an anticommuting element in the original normalizer N_A $[N_B]$, it follows that $N_A\otimes I_B$ $[I_A\otimes N_B]$ will correspondingly anticommute with the elements of $N_{AB,j}$. This means that theorem 2 is satisfied also for the combination of blocks A and B, and fault tolerance is guaranteed as in the $\overline{\text{SU}(2)}$ case.

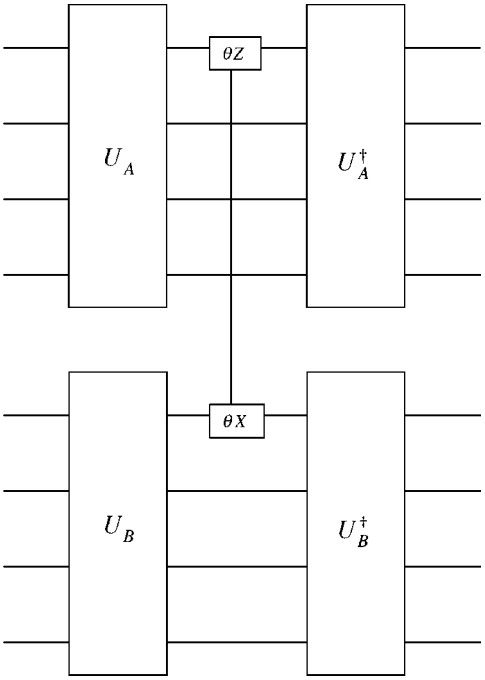

FIG. 3. Fault-tolerant implementation of $\exp(i\theta\overline{ZX})$ needed to generate $\overline{\text{CNOT}}$.

As promised in Sec. V D, the construction presented here also applies to multiple qubits encoded into a single block. To see this, consider the case of two encoded qubits in the same block, and let us show that we can generate $\exp(i\theta\overline{Z}_1\otimes\overline{Z}_2)$ between them. This coupling, together with single encoded-qubit operations, suffices to generate $\overline{\text{SU}(2^l)}$ (for l encoded qubits in a block). Now, from the standard form we have

$$\overline{Z}_1=Z_1\otimes M_Z^1\otimes I^{\otimes K-l-r}, \tag{6.9}$$

$$\overline{Z}_2=Z_2\otimes M_Z^2\otimes I^{\otimes K-l-r}. \tag{6.10}$$

Let $\overline{Z}_1=U_1Z_1U_1^\dagger$ and $\overline{Z}_2=U_2Z_1U_2^\dagger$. Note that $[U_1,Z_2]=[U_2,Z_1]=0$ since $U_{1(2)}$ contains $X_{2(1)}$. For the same reason also $[U_1,U_2]=0$. Therefore,

$$
\begin{aligned}
&(U_1\otimes U_2)\exp(i\theta Z_1\otimes Z_2)(U_1^\dagger\otimes U_2^\dagger)\\
&\quad=\exp[i\theta(U_1\otimes U_2)Z_1\otimes Z_2(U_1^\dagger\otimes U_2^\dagger)]\\
&\quad=\exp[i\theta(U_1Z_1U_1^\dagger)\otimes(U_2Z_2U_2^\dagger)]\\
&\quad=\exp(i\theta\overline{Z}_1\otimes\overline{Z}_2). \qquad (6.11)
\end{aligned}
$$

The same idea can be used to implement $\overline{\text{CNOT}}$ between multiple qubits encoded into a single block. We have thus provided a fault-tolerant implementation of $\overline{\text{CNOT}}$ for any stabilizer DFS.

VII. FAULT-TOLERANT MEASUREMENT OF THE ERROR SYNDROME

So far we have taken for granted that error detection and correction is possible in between gate applications. We now complete our discussion by showing that it is indeed possible to do so fault tolerantly. This requires the ability to measure the sequence of transformed stabilizer generators in a manner that does not introduce new errors in a catastrophic way. To accomplish this fault-tolerant measurement, we follow, with some modifications, the usual stabilizer construction [45].

Let us recall the basics of measurement within stabilizer theory. A DFS state $|\psi\rangle$ in the stabilizer Q is a $+1$ eigenstate of all elements of Q. An error e is an operator that anticommutes with at least one element of the stabilizer Q, say q. If $|\psi\rangle\in Q$, then $qe|\psi\rangle=-eq|\psi\rangle=-e|\psi\rangle$, so that $e|\psi\rangle$ is an eigenstate of q with eigenvalue -1. Therefore, each generator measurement that returns the eigenvalue $+1$ indicates that no error has occurred, while each -1 result indicates an error, which can be fixed by applying the error e to the state. The sequence of ±1's that results from measuring all stabilizer generators is called the ''error syndrome.'' The identity of e is uniquely determined by this ''syndrome,'' since the measurement process projects any linear combination of errors to an error in the Pauli group.

A. CSS stabilizer errors

In this case, the stabilizer generators contain either products only of Z's (''Z-type'') or products only of X's (''X-type''). Suppose we wish to measure a Z-type stabilizer generator. The $+1$ eigenstates of such a generator are the ''even-parity states,'' i.e., those states containing an even number of $|1\rangle$'s. Prepare an ancilla in the encoded $|0_L\rangle$ state (below we discuss how). Then for each data qubit where the given stabilizer generator has a Z (not an I), apply a controlled $\overline{X}$ from this qubit to the ancilla. The ancilla will flip every time the data qubit was a $|1\rangle$, so measuring the ancilla at the end and finding it in $|0_L\rangle$ will indicate no error (even number of flips), whereas $|1_L\rangle$ will indicate an error (odd number of flips). Distinguishing between $|0_L\rangle$ and $|1_L\rangle$ amounts to measuring $\overline{Z}$ on the ancilla, which we can do directly by measuring Z on all those ancilla qubits whose $\overline{Z}$ has a Z.

Now suppose we wish to measure an X-type stabilizer element. The same procedure as for Z-type generators can be applied, with one modification: a Hadamard transform

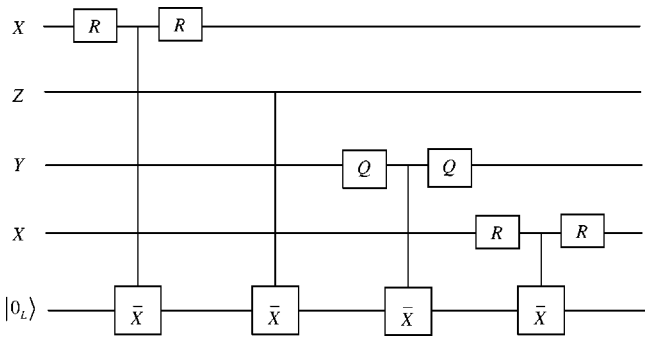

FIG. 4. Measurement of the stabilizer element $XZYX$.

$$
R=\frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix} \qquad (7.1)
$$

must be applied before and after the controlled-$\overline{X}$ operation. The effect of the Hadamard transform before the controlled-$\overline{X}$ operation is to change the corresponding qubit into the Z eigenbasis, after which the Z-type construction applies. The second Hadamard transform returns the qubit to the original basis. This construction is shown schematically in Fig. 4.

Note that since $\overline{X}$ is in the normalizer, it commutes with all stabilizer errors. This means that any such error occurring on the ancilla before the $\overline{X}$ is equivalent to the same error after the $\overline{X}$, and therefore the error has no effect. In other words, the ancilla does not ever leave the DFS under the application of $\overline{X}$, nor can an error on the ancilla propagate back to the data qubits.[9] Note further that since the ancilla is at all times unentangled from the data qubits, the measurement is nondestructive on the data qubits.

What if a stabilizer error occurs on the data qubits right after the application of the Hadamard gate? This can clearly present a problem, since it may, for example, flip the data qubit controlling the $\overline{X}$ applied to the ancilla. One (standard) way of dealing with such errors is to repeat the measurement several times in order to improve our confidence in the result. An alternative is to use concatenated codes [35,48–50]. This will be of use if the stabilizer error is correctable by the transformed code, i.e., if we can verify that the conditions of theorem 2 are satisfied. Then we can use the DFS at the lowest level, and concatenate it with the QECC it transforms into under the stabilizer errors (see Ref. [23] for concatenated DFS-QECC in the collective decoherence model). Now, recall the CSS form of the normalizer elements, Eq. (5.3). For every Hadamard transform in the first set (i.e.,

[9]This DFS construction is different from the usual QECC-stabilizer construction, where multiple control operations to the same ancilla qubit are not fault-tolerant because they are not transversal. There multiple CNOT's from different data qubits to the same ancilla qubit can cause errors to spread catastrophically if the ancilla qubit undergoes a phase error (recall that under CNOT, $I\otimes Z\mapsto Z\otimes Z$).

before the controlled-$\overline{X}$ operations) on a qubit in a position corresponding to an X in an X-type stabilizer generator, the normalizer elements transform by having X and Z interchange in this position. In the standard form of Eq. (5.3), if this happens to be the first qubit, then $\overline{Z}\mapsto X\otimes M_Z\otimes I$, which anticommutes with the original $\overline{Z}$, and $\overline{X}\mapsto Z\otimes I\otimes M_X$, which in turn anticommutes with the original $\overline{X}$. If the position of the X in the X-type stabilizer generator is where M_Z has a Z, then $\overline{Z}\mapsto Z\otimes M_Z'\otimes I$, where M_Z' has that Z changed into an X. This transformed $\overline{Z}$ anticommutes with the original $\overline{X}$. Similarly, $\overline{X}\mapsto X\otimes I\otimes M_X'$ with an X changed into a Z, and this transformed $\overline{X}$ anticommutes with the original $\overline{Z}$. Thus the conditions of theorem 2 are again satisfied.

The second set of Hadamard transforms restores the original normalizer. One then proceeds to measure the next stabilizer generator. We thus see that this measurement procedure is fault-tolerant of stabilizer errors.

B. General stabilizer errors

In the non-CSS case, the stabilizer generators may contain Y's as well, so our analysis above requires some modifications. The unitary operation that transforms Y to Z is

$$Q=\frac{1}{\sqrt{2}}\begin{pmatrix}1 & -i\\ 1 & i\end{pmatrix}. \tag{7.2}$$

It also maps $Z\mapsto X\mapsto Y$. When this operation is applied immediately before the controlled-$\overline{X}$ to the ancilla and immediately after it for every Y in the stabilizer, the Z-type construction applies again. However, for the purpose of concatenation we need to check that the procedure is still fault-tolerant of stabilizer errors. The normalizer generators now have the form of Eqs. (5.1) and (5.2). Every time a Hadamard or Q operation is applied, $Z\mapsto X$ in a single position in $\overline{Z}$. Similarly, $Z\mapsto X$, or $X\mapsto Z$ (if Hadamard) or Y (if Q) in a single position in $\overline{X}$.

The case of the transformed $\overline{Z}$ is trivial: if $Z\mapsto X$ anywhere, then the transformed $\overline{Z}$ anticommutes with the original $\overline{Z}$. Consider the transformed $\overline{X}$. The possibilities are (i) $X_1\mapsto Z_1$ or Y_1, (ii) $Z\mapsto X$ in the N_Z part, (iii) $X\mapsto Z$ or Y in the M_X part. In all these cases, it is easily verified that the transformed $\overline{X}$ anticommutes with the original $\overline{X}$. Therefore, the measurement procedure is fault-tolerant also in the non-CSS case.

VIII. OUTLOOK: IMPLICATIONS FOR THE INDEPENDENT-ERRORS MODEL

The methods we have introduced in this paper need not be restricted to stabilizer errors. In this section, we briefly touch upon the implications of our construction for universal quantum computation in the independent-errors model, when stabilizer errors are taken into account as well. We thus generalize the standard treatment of stabilizer codes [34], where stabilizer errors that may occur during the course of gate implementation are ignored. However, here we are only able to consider independent single-qubit errors, so that the inclusion of the special type of correlated many-body errors represented by the stabilizer errors is a rather unrealistic error model. The main importance of the result presented here is that it suggests an alternative route to universal quantum computation that is fault-tolerant with respect to error *detection*, and is highly parallelizable. We believe that this may lead to an improved threshold for fault-tolerant computation in the setting of concatenated codes [50].

Let us recall the error-detection and -correction criteria for a stabilizer code $Q=\{q_k\}$ to be able to deal with all single-qubit errors:

$$\forall i,j,\alpha,\beta\exists k \text{ s.t. } \{q_k,\sigma_i^\alpha\otimes\sigma_j^\beta\}=0. \tag{8.1}$$

Can we implement encoded SU(2) operations in the independent-errors model similarly to what we did above for stabilizer errors? To do so, we need to make sure that the errors do not become part of the sequence of transformed normalizers. The important difference compared to the stabilizer-errors case is that now the errors are ''small'' (single-body), which means that we must avoid using a single-qubit Hamiltonian as a central gate (for it is a normalizer element that will not be distinguishable from an error). If we restrict ourselves to using two-body Hamiltonians as central gates (which we can always do—recall the comment at the end of Sec. IV D), then we run into a similar problem regarding the two-body form of Eq. (8.1), i.e., if the central gate uses the Hamiltonian $\sigma_i^\alpha\otimes\sigma_j^\beta$, then we will not be able to correct the two errors σ_i^α and σ_j^β. However, as we now show, as long as we use a two-body central gate, it is nearly always possible to satisfy the error-*detection* criterion, $\forall i,\alpha\exists k$ s.t. $\{q_k,\sigma_i^\alpha\}=0$.

Let us demonstrate this explicitly for Steane's seven-qubit code [12]. This is a CSS code encoding one qubit into seven, and in standard form has the normalizer

$$\begin{aligned}\overline{X}&=X_1X_5X_6,\\ \overline{Z}&=Z_1Z_3Z_4.\end{aligned} \tag{8.2}$$

Consider the gate construction [derived from Eq. (5.5)]

$$\exp(i\theta\overline{Z})=T_{X_1Z_3}\circ\exp(i\theta Y_1Z_4). \tag{8.3}$$

The normalizer transforms as

$$\begin{aligned}\overline{X}&\overset{X_1Z_3}{\mapsto}\overline{X}\overset{Y_1Z_4}{\mapsto}\cos(2\theta)\overline{X}+i\sin(2\theta)Z_1Z_4X_5X_6\\ &\overset{X_1Z_3}{\mapsto}\cos(2\theta)\overline{X}+\sin(2\theta)\overline{Y}\\ &=\overline{X}\exp(2i\theta\overline{Z})\overline{Z}\overset{X_1Z_3}{\mapsto}Y_1Z_4\overset{Y_1Z_4}{\mapsto}Y_1Z_4\overset{X_1Z_3}{\mapsto}\overline{Z}.\end{aligned} \tag{8.4}$$

We see that at no point does a single-qubit error become part of the transformed normalizer, so that all single-qubit errors are detectable. On the other hand, while we can always de-

tect the occurrence of both the Y_1 and Z_4 errors, we cannot distinguish between them after the first gate has been applied (since our normalizer is $Y_1 Z_4$ at that point). Since we might accidentally try to reverse the error Y_1 when in fact the error Z_4 has taken place, this means that our construction is fault-tolerant only for error detection. Similarly, the gate construction

$$\exp(i\theta\overline{X}) = T_{Z_1 X_5} \circ \exp(i\theta Y_1 X_6) \tag{8.5}$$

yields

$$\begin{aligned} &\overline{X} \overset{Z_1X_5}{\mapsto} Y_1X_6 \overset{Y_1X_6}{\mapsto} Y_1X_6 \overset{Z_1X_5}{\mapsto} \overline{XZ} \overset{Z_1X_5}{\mapsto} \overline{Z} \\ &\overset{Y_1X_6}{\mapsto} \cos(2\theta)\overline{Z} + i\sin(2\theta)X_1Z_3Z_4X_6 \\ &\overset{Z_1X_5}{\mapsto} \cos(2\theta)\overline{Z} + \sin(2\theta)\overline{Y} = \overline{Z}\exp(-2i\theta\overline{X}), \end{aligned} \tag{8.6}$$

which also satisfies the error-detection (but not correction) condition for single-qubit errors, in that no single-qubit error becomes part of the transformed stabilizer.

Let us now consider the general stabilizer case. Recall once more the standard form of the normalizer, Eqs. (5.1) and (5.2). Our gate construction acts by transforming one of the normalizer elements to two-body form, where it is applied as the central θ gate, and then is transformed back to its standard form. All other normalizer elements are left unchanged until the application of the central gate, with which they anticommute. At this point each $\overline{Z}$ $[\overline{X}]$ is multiplied by $\exp(-2i\theta\overline{X})$ $[\exp(2i\theta\overline{Z})]$. The final sequence of gates flips these normalizer elements back and forth between $\exp(-2i\theta\overline{X})$ and $\exp(-2i\theta\overline{Y})$ $[\exp(2i\theta\overline{Z})$ and $\exp(2i\theta\overline{Y})]$ (recall the analysis in Sec. V B). All these operations have the effect of expanding, rather than shrinking, the normalizer elements, as seen in the example of the seven-qubit code above.

The ability to error-detect at each point thus translates to the question of whether any normalizer element ever becomes a single-body Hamiltonian under this sequence of transformations. It is not hard to see from the above description of the orbit of the normalizer that this can only be the case if in the standard form the normalizer contains a single-body element to begin with. This is certainly possible, as indeed shown in our Q_{2X} example (Sec. V C), where $\overline{X} = XIII$. However, it is not the case for most interesting stabilizer codes, i.e., those offering protection against arbitrary single-qubit errors. Such codes must have ''large'' normalizer elements since they may not contain any single-qubit errors to begin with. We conclude that our $\overline{\mathrm{SU}(2)}$ construction using just two-qubit Hamiltonians works for all stabilizer codes of interest, in the sense that it is fault-tolerant with respect to error detection.

To complete the repertoire of universal operations, the $\overline{\mathrm{CNOT}}$ gate is still needed. The discussion given in Sec. VI applies here as well, with the modification that for non-CSS stabilizer codes it is once again necessary to apply two-body central gates. Fault-tolerant measurement of the error syndrome can be done using the standard techniques available for stabilizer codes [34].

IX. SUMMARY AND CONCLUSIONS

In a previous paper [29], we derived conditions for the existence of a class of decoherence-free subspaces (DFSs) defined by having Abelian stabilizers over the Pauli group. In this sequel paper, we addressed the problem of universal, fault-tolerant quantum computation on this class of DFSs. The errors in this model are the elements of the stabilizer, and thus are necessarily correlated. This model is complementary to the standard model of quantum computation using stabilizer quantum error-correcting codes (QECCs), where the errors that are correctable by the code anticommute with the stabilizer (rather than being part of it). The correlation between errors in the present model implies no spatial symmetry in the system-bath interaction, unlike in most previous studies of computation on DFSs (which considered the ''collective decoherence'' model, and where the stabilizer is non-Abelian). Therefore, our present results significantly enlarge the scope of the theory of DFSs.

It turns out that even though the class of DFSs we considered are Pauli-group stabilizer codes, the usual universality constructions do not apply, because of the different error model we assume. Our alternative construction of a set of universal quantum gates resorts to the early ideas about universal quantum computation, except that our operations all act on *encoded* (DFS) qubits: we showed how to implement arbitrary single-encoded-qubit operations [the $\overline{\mathrm{SU}(2)}$ group] and $\overline{\mathrm{CNOT}}$ gates between pairs of encoded qubits. The challenge here was to show how to accomplish this implementation using only physically reasonable Hamiltonians, i.e., those involving no more than two-body interactions. To do so, we switched from the usual point of view of treating the normalizer elements (i.e., the operations that preserve the DFS) as gates to one in which these elements are considered as many-body Hamiltonians. We then introduced a procedure whereby these Hamiltonians could be simulated using at most two-body interactions. Unlike our previous work concerning universal computation in the collective decoherence model [26,27], the gate sequence implementing this simulation does not preserve the DFS except at the beginning and end. Throughout the execution of the gates, the DFS states are exposed to the stabilizer errors. However, we showed that in fact the DFS is transformed into a sequence of stabilizer codes, each of which is capable of detecting and correcting the original stabilizer errors. Moreover, we showed that these errors can be diagnosed in a fault-tolerant manner, i.e., without introducing new errors as a result of the associated measurements. In all, we showed how by using this type of hybrid DFS-QECC approach, universal, fault-tolerant quantum computation can be implemented.

Our results have implications beyond computation on DFSs. We briefly considered here also the question of whether our techniques can be used to compute fault tolerantly in the standard stabilizer error model. We found the

answer to be affirmative for the purpose of single-qubit error detection, but not correction. While this is interesting in its own right because of the new universality construction we introduced, it may also have important implications for the question of quantum computation using concatenated codes. The reason is that our construction is highly parallelizable, meaning that it requires a very small number of operations during which the encoded information is exposed to errors. We speculate that this can significantly reduce the threshold for fault-tolerant quantum computation.

Finally, an interesting open question is whether the methods developed here are applicable to the problem of universal quantum computation on other classes of DFSs.

ACKNOWLEDGMENTS

This material is based upon work supported by the U.S. Army Research Office under Contract/Grant No. DAAG55-98-1-0371, and in part by NSF CHE-9616615. We would like to thank Dr. Daniel Gottesman for very useful correspondence.

APPENDIX: WHY THE FOUR-BLOCK IMPLEMENTATION OF $\overline{\text{CNOT}}$ IS NOT FAULT-TOLERANT FOR NON-CSS STABILIZERS

The construction of the $\overline{\text{CNOT}}$ in Ref. [34] uses a series of bitwise CNOT's (along with some other operations) acting between pairs of qubits in four different blocks. Let us calculate the result of applying bitwise CNOT's on $I^{\otimes K}\otimes\overline{X}$ (i.e., on two out of the four blocks). Recall that for a non-CSS code $\overline{X}=X\otimes N_Z\otimes M_X$ [Eq. (5.2)]. Therefore, it follows from Eq. (6.3) that

$$I^{\otimes K}\otimes\overline{X}\mapsto[I\otimes N_Z\otimes I^{\otimes K-1-r}]\otimes\overline{X}, \tag{A1}$$

i.e., the Z's are copied backwards into the first block. Therefore, the normalizer on the first block now contains $I\otimes N_Z\otimes I^{\otimes K-1-r}$. This element obviously commutes with both the original $\overline{X}$ and $\overline{Z}$ [Eq. (5.1)], but does not equal either. Therefore, it must be in the original stabilizer Q. Turning this around, we see that an error $e\in Q$ has become part of the new normalizer $N_j(Q_j)/Q_j$, which is catastrophic since this error is now undetectable.

[1] H. K. Lo, S. Popescu, and T. P. Spiller, *Introduction to Quantum Computation and Information* (World Scientific, Singapore, 1999).
[2] C. Williams and S. Clearwater, *Explorations in Quantum Computing* (Springer-Verlag, New York, 1998).
[3] A. M. Steane, Rep. Prog. Phys. **61**, 117 (1998); e-print quant-ph/9708022.
[4] D. Aharonov, e-print quant-ph/9812037.
[5] R. Cleve, e-print quant-ph/9906111.
[6] R. Alicki and K. Lendi, in *Quantum Dynamical Semigroups and Applications*, Lecture Notes in Physics (Springer-Verlag, Berlin, 1987).
[7] K. Kraus, *States, Effects and Operations, Fundamental Notions of Quantum Theory* (Academic, Berlin, 1983).
[8] D. Bacon, D. A. Lidar, and K. B. Whaley, Phys. Rev. A **60**, 1944 (1999); e-print quant-ph/9902041.
[9] M. A. Nielsen, C. M. Caves, B. Schumacher, and H. Barnum, Proc. R. Soc. London, Ser. A **454**, 277 (1998); e-print quant-ph/9706064.
[10] P. W. Shor, Phys. Rev. A **52**, 2493 (1995).
[11] A. R. Calderbank and P. W. Shor, Phys. Rev. A **54**, 1098 (1996).
[12] A. M. Steane, Phys. Rev. Lett. **77**, 793 (1996).
[13] C. H. Bennett, D. P. DiVincenzo, J. A. Smolin, and W. K. Wootters, Phys. Rev. A **54**, 3824 (1996).
[14] A. Yu. Kitaev, Russ. Math. Surveys **52**, 1191 (1996).
[15] D. Gottesman, Phys. Rev. A **54**, 1862 (1996); e-print quant-ph/9604038.
[16] E. Knill and R. Laflamme, Phys. Rev. A **55**, 900 (1997).
[17] A. M. Steane, in *Introduction to Quantum Computation and Information*, edited by H. K. Lo, S. Popescu, and T. P. Spiller (World Scientific, Singapore, 1999), p. 184.
[18] P. Zanardi and M. Rasetti, Phys. Rev. Lett. **79**, 3306 (1997); e-print quant-ph/9705044.
[19] P. Zanardi and M. Rasetti, Mod. Phys. Lett. B **11**, 1085 (1997); e-print quant-ph/9710041.
[20] L.-M. Duan and G.-C. Guo, Phys. Rev. A **57**, 737 (1998).
[21] L.-M. Duan and G.-C. Guo, Phys. Lett. A **243**, 265 (1998).
[22] D. A. Lidar, I. L. Chuang, and K. B. Whaley, Phys. Rev. Lett. **81**, 2594 (1998); e-print quant-ph/9807004.
[23] D. A. Lidar, D. Bacon, and K. B. Whaley, Phys. Rev. Lett. **82**, 4556 (1999); e-print quant-ph/9809081.
[24] E. Knill, R. Laflamme, and L. Viola, Phys. Rev. Lett. **84**, 2525 (2000); e-print quant-ph/9908066.
[25] L. Duan and G. Guo, Phys. Lett. A **255**, 209 (1999); e-print quant-ph/9809057.
[26] D. Bacon, J. Kempe, D. A. Lidar, and K. B. Whaley, Phys. Rev. Lett. **85**, 1758 (2000); e-print quant-ph/9909058.
[27] J. Kempe, D. Bacon, D. A. Lidar, and K. B. Whaley, Phys. Rev. A (to be published); available as e-print quant-ph/0004064.
[28] E. M. Rains, R. H. Hardin, P. W. Shor, and N. J. A. Sloane, Phys. Rev. Lett. **79**, 953 (1997).
[29] D. A. Lidar, D. Bacon, J. Kempe, and K. B. Whaley, preceding paper, Phys. Rev. A **63**, 022306 (2000).
[30] P. Zanardi, Phys. Rev. A **60**, R729 (1999), e-print quant-ph/9901047.
[31] A. Beige, D. Braun, B. Tregenna, and P. L. Knight, Phys. Rev. Lett. **85**, 1762 (2000).
[32] P. W. Shor, in *Proceedings of the 37th Symposium on Foundations of Computing* (IEEE Computer Society Press, Los Alamitos, CA, 1996), p. 56; e-print quant-ph/9605011.
[33] P. Boykin, T. Mor, M. Pulver, V. Roychowdhury, and F. Vatan, in *40th Annual Symposium on Foundations of Computer Science* (IEEE Comput. Soc., Los Alamitos, CA, 1999), p. 486; e-print quant-ph/9906054.
[34] D. Gottesman, Phys. Rev. A **57**, 127 (1997); e-print quant-ph/9702029.

[35] E. Knill, R. Laflamme, and W. Zurek, Proc. R. Soc. London, Ser. A **454**, 365 (1998); e-print quant-ph/9702058.
[36] D. Deutsch, A. Barenco, and A. Ekert, Proc. R. Soc. London, Ser. A **449**, 669 (1995).
[37] D. P. DiVincenzo, Phys. Rev. A **51**, 1015 (1995).
[38] T. Sleator and H. Weinfurter, Phys. Rev. Lett. **74**, 4087 (1995).
[39] S. Lloyd, Phys. Rev. Lett. **75**, 346 (1995).
[40] D. Gottesman, e-print quant-ph/9807006.
[41] A. R. Calderbank, E. M. Rains, P. W. Shor, and N. J. A. Sloane, IEEE Trans. Inf. Theory **44**, 1369 (1998); e-print quant-ph/9608006.
[42] A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. Smolin, and H. Weinfurter, Phys. Rev. A **52**, 3457 (1995).
[43] M.E. Rose, *Elementary Theory of Angular Momentum* (Dover, New York, 1995).
[44] D. Loss and D. P. DiVincenzo, Phys. Rev. A **57**, 120 (1998); e-print quant-ph/9701055.
[45] D. Gottesman, Ph.D. thesis, California Institute of Technology, Pasadena, CA, 1997; e-print quant-ph/9705052.
[46] C. Slichter, in *Principles of Magnetic Resonance*, Springer Series in Solid-State Sciences Vol. 1 (Springer, Berlin, 1996).
[47] R. Bhatia, in *Matrix Analysis*, Graduate Texts in Mathematics Vol. 169 (Springer-Verlag, New York, 1997).
[48] D. Aharonov and M. Ben-Or, in *Proceedings of the 29th Annual ACM Symposium on Theory of Computing (STOC)* (ACM, New York, 1997), p. 46; e-print quant-ph/9611025.
[49] C. Zalka, e-print quant-ph/9612028.
[50] E. Knill, R. Laflamme, and W. Zurek, Science **279**, 342 (1998).