Entanglement-Assisted Quantum Error-Correcting Codes

Todd Brun [1] and Min-Hsiu Hsieh [2]

[1] *University of Southern California, USA*
[2] *Centre for Quantum Software and Information, University of Technology Sydney, Australia*

Contents

1 Introduction

Quantum error-correcting codes (QECCs) have turned out to have many applications; but their primary purpose, of course, is to protect quantum information from noise. This need manifests itself in two rather different situations. The first is in quantum computing. The bits in a quantum computer are subject to noise due both to imprecision in the operations (or gates) and to interactions with the external environment. They undergo errors when they are acted upon, and when they are just being stored (though hopefully not at the same rate). All the qubits in the computer must be kept error-free long enough for the computation to be completed. This is the principle of fault-tolerance.

A rather different situation occurs in quantum *communication*. Here, the sender and receiver (Alice and Bob) are assumed to be physically separated, and the qubits travel from the sender to the receiver over a (presumably noisy) channel. This channel is assumed to be the dominant source of errors. While the qubits may undergo processing before and after transmission, errors during this processing are considered negligible compared to the channel errors. This picture of transmission through a channel is very close to the picture underlying classical information theory.

The types of codes that are of interest in quantum computation and quantum communication may be quite different. In computation, reliability is key. Therefore, codes with a high distance (such as concatenated codes) are used. Also, the choice of code may be highly constrained in other ways: for instance, it is very important that the codes allow efficient circuits for encoded logic gates (such as the transversal gates allowed by the Steane code). The more efficient these circuits, the better the fault-tolerant threshold. (See [LB13, chapter 5] for a fuller discussion of this.)

By contrast, in communication one often imagines sending a very large number of quantum bits, perhaps even a continuous stream of qubits (as in the convolutional codes of [LB13, chapter 9]). Here, one is generally trying to maximize the rate of transmission, often by encoding many qubits into a large block, within the constraint of a low error probability for the block. In the asymptotic limit, one would like to achieve the actual capacity of the channel—the maximum rate of communication possible.

Because the sender and receiver are separated, joint unitary transformations are impossible. However, they might be able to draw on other resources, such as extra classical communication, pre-shared randomness, or pre-shared entanglement. In this chapter we study the use of pre-shared entanglement in quantum communication, and how we can design QECCs that use entanglement to boost either the rate of communication or the number of errors that can be corrected. Codes that use pre-shared entanglement are called *entanglement-assisted quantum error-correcting codes* (EAQECCs). We will study a large class of these codes that generalizes the usual stabilizer formalism from [LB13, chapter 2]. We will also see how these codes can readily be constructed from classical linear codes, and how they can be useful tools in building standard QECCs.

1.1 Entanglement-assisted codes

In a standard QECC, the encoding operation proceeds in two steps. First, to the quantum state $|\psi\rangle$ of k qubits that one wishes to encode, one appends some number of ancilla qubits in a standard state (usually $|0\rangle$), and then applies an encoding unitary U_{enc}:

$$|\psi\rangle \rightarrow |\psi\rangle \otimes |0\rangle^{\otimes n-k} \rightarrow |\Psi_L\rangle = U_{\text{enc}}|\psi\rangle \otimes |0\rangle^{\otimes n-k}, \tag{1}$$

where $|\Psi_L\rangle$ is the encoded or logical state. The systems in states $|\psi\rangle$ and $|0\rangle$ are initially in the possession of Alice, the sender. The encoding unitary acts on the space of the input qubits and

ancillas together.

In an EAQECC, one can append not only ancillas, but also ebits, before doing the encoding unitary:

$$|\psi\rangle \rightarrow |\psi\rangle \otimes |0\rangle^{\otimes n-k-c} \otimes |\Phi_+\rangle_{AB}^{\otimes c} \rightarrow (U_{\text{enc}} \otimes \hat{I}_B)|\psi\rangle \otimes |0\rangle^{\otimes n-k} \otimes |\Phi_+\rangle_{AB}^{\otimes c}. \tag{2}$$

The states $|\Phi_+\rangle_{AB}$ are EPR pairs shared between the sender (Alice) and the receiver (Bob). The encoding operation U_{enc} acts on the qubits in Alice's possession; we write $(U_{\text{enc}} \otimes \hat{I}_B)$ above to indicate that the encoding acts as the identity on Bob's halves of the ebits. Obviously, in order to append c ebits to the information qubits, Alice and Bob must have c ebits of pre-shared entanglement. After Alice does the encoding, all of her qubits are sent through the channel, including the c halves of ebits. So this procedure consumes c ebits of preshared entanglement.
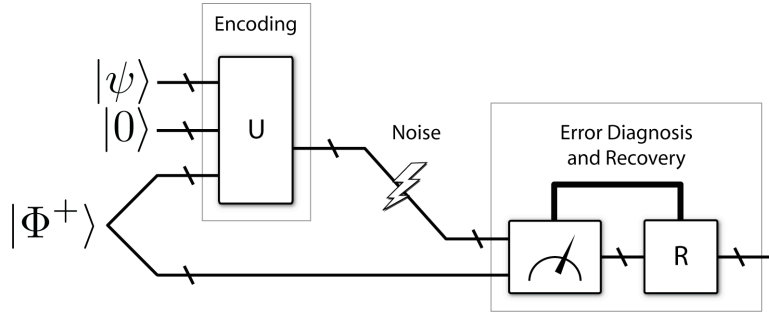

Figure 1: Schematic structure of an entanglement-assisted quantum error-correcting code. Alice and Bob share ebits that Alice uses in encoding her quantum information; these qubits all pass through the noisy channel, but Bob's halves of the initial ebits do not.

However, note that Bob's halves of the c ebits do not pass through the channel. These ebits were prepared ahead of time, by entanglement distillation or a similar procedure (see [BBP$^+$96, BDSW96] and [LB13, chapter 22]). Because Bob's qubits do not pass through the channel, they are assumed to be error-free.

Why should we wish to use shared entanglement in our encoding? There are two ways to see how this could enhance the power of the code. First, we can compare EAQECCs to *teleportation*. In teleportation, an ebit can be used (in combination with classical communication) to transmit one qubit perfectly from the sender to the receiver. We might therefore expect that making use of ebits in error-correction could boost the rate of transmission.

Second, we can compare EAQECCs to *superdense coding*. In superdense coding, by using an ebit the sender can send two classical bits of information to the receiver by means of a single qubit. Each ancilla bit in a standard QECC can be thought of as holding one classical bit of information about any errors that have occurred. Replacing an ancilla with one half of an ebit could, in principle, allow the receiver to extract *two* bits of classical information about the errors; with more information, more errors can be corrected. We shall see, in the constructions that follow, that we can think of the enhanced power of EAQECCs in both of these ways, but the superdense coding interpretation is more often the best fit. EAQECCs can achieve some communication tasks with fewer resources (or less chance of error) than would be needed by a standard QECC plus teleportation [HW10b].

The possibility of constructing entanglement-assisted codes was suggested in [BDSW96], though no practical codes were constructed. From asymptotic results in quantum information theory, we know that having pre-shared entanglement allows an enhanced rate of quantum communication

between a sender and receiver [DHW04, DHW08]. EAQECCs are the finite-length realization of this idea. By constructing larger and larger code blocks, in principle one can approach the entanglement-assisted quantum capacity of the channel. An example of such a code was constructed by Bowen in [Bow02], starting from a standard stabilizer QECC (see [LB13, chapter 2]). In [BDH06, BDH14, HDB07] an essentially complete theory of stabilizer EAQECCs was constructed. That is the theory presented in this chapter.

Note, however, that in general shared entanglement does not come for free: entanglement must be established between the sender and receiver, and this in general requires the use of quantum channels, and perhaps additional entanglement purification. Therefore, EAQECCs do not outperform standard quantum codes under all circumstances. The enhanced communication rates that come from the use of entanglement must be paid for in establishing the entanglement in the first place. For many purposes, what is important is the *net rate* of the code: the number of qubits transmitted minus the number of ebits consumed. With this measure, EAQECCs tend to perform about the same as standard codes. Even in this case, though, EAQECCs sometimes have advantages: as we shall see, there are many fewer restrictions on the construction of EAQECCs than standard codes.

2 Constructing EAQECCs

2.1 Noncommuting "stabilizers"

First, let us recall the stabilizer formalism for conventional quantum error-correcting codes, as presented in [LB13, chapter 2]. Let G_n be the n-qubit Pauli group [NC00]. Every operator in G_n either has eigenvalues ± 1 or $\pm i$. Let $S \subset G_n$ be an Abelian subgroup that does not contain $-I$. Then this subgroup has a common eigenspace $C(S)$ of $+1$ eigenvectors, which we call the *code space* determined by the stabilizer S. Later on, we will just use C to denote the code space. Typically, the stabilizer is represented by a minimal generating set $\{g_1, \ldots, g_m\}$, which makes this a compact way to specify a code (analogous to specifying a classical linear code by its parity-check matrix). We write $S = \langle g_1, \ldots, g_m \rangle$ to denote that S is generated by $\{g_1, \ldots, g_m\}$.

Let $\mathcal{E} \subset \mathcal{G}_n$ be a set of possible errors. If a particular error $E_1 \in \mathcal{E}$ anticommutes with any of the generators of S, then the action of that error can be detected by measuring the generators; if the measurement returns -1 instead of 1, we know an error has occurred. On the other hand, if the error is actually *in* the stabilizer S, then it leaves all the states in C unchanged. The code C can correct any error in $\mathcal{E}$ if either $E_2^\dagger E_1 \notin Z(S)$ or $E_2^\dagger E_1 \in S$ for all pairs of errors E_1 and E_2 in $\mathcal{E}$, where $Z(S)$ is the *centralizer* of S.

We can now generalize this description to the entanglement-assisted case. Given a *nonabelian* subgroup $\mathcal{S} \subset \mathcal{G}_n$ of size 2^m, there exists a set of generators $\{\bar{Z}_1, \cdots, \bar{Z}_{s+c}, \bar{X}_1, \cdots, \bar{X}_c\}$ for $\mathcal{S}$ with the following commutation relations:

$$\begin{aligned}
[\bar{Z}_i, \bar{Z}_j] &= 0 & \forall i, j \\
[\bar{X}_i, \bar{X}_j] &= 0 & \forall i, j \\
[\bar{X}_i, \bar{Z}_j] &= 0 & \forall i \neq j \\
\{\bar{X}_i, \bar{Z}_i\} &= 0 & \forall i.
\end{aligned} \tag{3}$$

(We will see shortly how to find this set of generators for any subgroup.) Here, $[A, B]$ is the commutator and $\{A, B\}$ the anti-commutator of A with B. The parameters s and c satisfy $s + 2c = m$. Let S_I be the *isotropic* subgroup generated by $\{\bar{Z}_{c+1}, \cdots, \bar{Z}_{c+s}\}$ and S_E be the *entanglement* subgroup generated by $\{\bar{Z}_1, \cdots, \bar{Z}_c, \bar{X}_1, \cdots, \bar{X}_c\}$. The numbers of generators of S_I and pairs of

generators of S_E describe the number of ancillas and the number of ebits, respectively, needed by the corresponding EAQECC. The pair of subgroups (S_I, S_E) defines an $[[n, k; c]]$ EAQECC $\mathcal{C}(S)$. We use the notation $[[n, k; c]]$ to denote an EAQECC that encodes $k = n - s - c$ logical qubits into n physical qubits, with the help of s ancillas, and c ebits shared between sender and receiver. These n qubits are transmitted from Alice (the sender) to Bob (the receiver), who measures them together with his half of the c ebits to correct any errors and decode the k logical qubits. We define k/n as the *rate* and $(k - c)/n$ as the *net rate* of the code. (Sometimes we will write $[[n, k, d; c]]$ to indicate that the "distance" of the code is d, meaning it can correct at least $\lfloor \frac{d-1}{2} \rfloor$ single-qubit errors.)

2.2 The Gram-Schmidt Procedure

We now illustrate the idea of the entanglement-assisted stabilizer formalism by an example. Let S be the group generated by the following non-commuting set of operators:

$$
\begin{array}{rcccc}
M_1 = & Z & X & Z & I \\
M_2 = & Z & Z & I & Z \\
M_3 = & X & Y & X & I \\
M_4 = & X & X & I & X
\end{array}
\tag{4}
$$

It is easy to check the commutation relations of this set of generators: M_1 anti-commutes with the other three generators, M_2 commutes with M_3 and anti-commutes with M_4, and M_3 and M_4 anti-commute. We will begin by finding a different set of generators for S with a particular class of commutation relations. We then relate S to a group B with a particularly simple form, and discuss the error-correcting conditions using B. Finally, we relate these results back to the group S.

To see how this works, we need two lemmas. (We only sketch the proofs—these results are well known.) The first lemma shows that there exists a new set of generators for S such that S can be decomposed into an "isotropic" subgroup S_I, generated by a set of commuting generators, and a "symplectic" subgroup S_S, generated by a set of anti-commuting generator pairs [FCY$^+$04].

Lemma 1. Given any arbitrary subgroup V in G_n that has 2^m distinct elements up to overall phase, there exists a set S of m independent generators for V of the forms $\{\bar{Z}_1, \bar{Z}_2, \cdots, \bar{Z}_\ell, \bar{X}_1, \cdots, \bar{X}_{m-\ell}\}$ where $m/2 \leq \ell \leq m$, such that $[\bar{Z}_i, \bar{Z}_j] = [\bar{X}_i, \bar{X}_j] = 0$, for all i, j; $[\bar{Z}_i, \bar{X}_j] = 0$, for all $i \neq j$; and $\{\bar{Z}_i, \bar{X}_i\} = 0$, for all i. Let $V_I = \langle \bar{Z}_{m-\ell+1}, \cdots, \bar{Z}_\ell \rangle$ denote the isotropic subgroup generated by the set S_I of commuting generators, and let $V_S = \langle \bar{Z}_1, \cdots, \bar{Z}_{m-\ell}, \bar{X}_1, \cdots, \bar{X}_{m-\ell} \rangle$ denote the symplectic subgroup generated by the set S_E of anti-commuting generator pairs. Then, with slight abuse of the notation, $V = \langle V_I, V_S \rangle$ denotes that V is generated by subgroups V_I and V_S.

Proof. We sketch a constructive procedure for finding the sets of generators, which is analogous to the Gram-Schmidt procedure for finding an orthonormal basis for a vector space. Suppose one has any set of generators $\{g_1, \ldots, g_m\}$ for the subgroup $\mathcal{V}$. We will successively find new sets of generators for $\mathcal{V}$ while assigning the generators we find either to the set S_I of isotropic generators or the set S_E of symplectic generators. The generators of S_E come in anticommuting pairs.

We start by assigning the anticommuting pairs. Suppose that one has so far assigned k pairs of generators to S_E. Go through the list of as-yet-unassigned generators, and find any two generators g and h that anticommute. These will be our next symplectic pair. First, however, we must make them commute with all the remaining generators. Go through the whole list of unassigned generators (except g and h). If a generator anticommutes with g, then multiply that generator by h and replace it. The new generator will commute with g, and the new set generates the same group. Similarly, if a generator anticommutes with h, multiply it by g and replace it. (Note that

4

some generators may be multiplied by both g and h, and some by neither.) Once we have gone through the entire list, we will have found a new set of generators such that g and h anticommute and all other generators commute with them both. Relabel g as $\bar{Z}_{k+1}$ and h as $\bar{X}_{k+1}$, assign the pair to S_E, and let $k \to k+1$.

Continue this procedure until either all the generators have been assigned to S_E, or all of the remaining unassigned generators commute. If there are a total of $m - \ell$ pairs of generators in S_E, we relabel the remaining generators $\bar{Z}_{m-\ell+1}, \ldots, \bar{Z}_\ell$ and assign them all to S_I. This procedure is optimal, in the sense that it produces the minimum number of symplectic pairs (and hence the minimum number of ebits for the code). $\qquad\square$

For the group S that we are considering, generated by (4), we can follow the steps of the procedure outlined in the lemma to construct such a set of independent generators. We start by taking the first generator M_1 and labeling it $\bar{Z}_1$. We then find the first anticommuting generator—which happens to be M_2—and label it $\bar{X}_1$. We must then eliminate any anticommutation with the remaining generators. $\bar{Z}_1$ anticommutes with M_3 and M_4, so we multiply each of them by $\bar{X}_1$. M_4 anticommutes with $\bar{X}_1$, so we multiply it by $\bar{Z}_1$. The two new generators that result commute with $\bar{Z}_1$ and $\bar{X}_1$, and also commute with each other; so we label them $\bar{Z}_2$ and $\bar{Z}_3$. The resulting set of generators is:

$$
\begin{array}{cccccc}
\bar{Z}_1 = & Z & X & Z & I \\
\bar{X}_1 = & Z & Z & I & Z \\
\bar{Z}_2 = & Y & X & X & Z \\
\bar{Z}_3 = & Z & Y & Y & X
\end{array}
\tag{5}
$$

so that $\mathcal{S}_S = \langle \bar{Z}_1, \bar{X}_1 \rangle$, $\mathcal{S}_I = \langle \bar{Z}_2, \bar{Z}_3 \rangle$, and $\mathcal{S} = \langle \mathcal{S}_I, \mathcal{S}_S \rangle$.

The choice of the notation $\bar{Z}_i$ and $\bar{X}_i$ is not accidental: these generators have exactly the same commutation relations as Pauli operators Z_i and X_i on a set of qubits labeled by i. We now see that the subgroup they generate matches one-to-one with this simpler subgroup. Let B be the group generated by the following set of Pauli operators:

$$
\begin{array}{cccccc}
Z_1 = & Z & I & I & I \\
X_1 = & X & I & I & I \\
Z_2 = & I & Z & I & I \\
Z_3 = & I & I & Z & I
\end{array}
\tag{6}
$$

From the previous lemma, $B = \langle B_I, B_S \rangle$, where $B_S = \langle Z_1, X_1 \rangle$ and $B_I = \langle Z_2, Z_3 \rangle$. Therefore, groups B and S are *isomorphic*, which is denoted $\mathcal{B} \cong \mathcal{S}$. We can relate S to the group B by the following lemma [BFG06]:

Lemma 2. If $\mathcal{B}$ and $\mathcal{S}$ are both subgroups of $\mathcal{G}_n$, and $\mathcal{B} \cong \mathcal{S}$, then there exists a unitary U such that for all $B \in \mathcal{B}$ there exists an $S \in \mathcal{S}$ such that $B = USU^{-1}$ up to an overall phase.

Proof. We only sketch the proof here. First, apply Lemma 1 to both B and S to find a set of generators for each group in the standard form. We call the generators of B $\{\bar{Z}_1, \ldots, \bar{Z}_r, \bar{X}_1, \ldots, \bar{X}_c\}$, and the generators of S $\{\hat{Z}_1, \ldots, \hat{Z}_r, \hat{X}_1, \ldots, \hat{X}_c\}$, where $r = s + c$. (Because $\mathcal{B} \cong \mathcal{S}$ the parameters r and c must be the same for both groups.) The symplectic subgroup of B is generated by $\bar{Z}_1, \ldots, \bar{Z}_c, \bar{X}_1, \ldots, \bar{X}_c$, the isotropic subgroup of B is generated by $\bar{Z}_{c+1}, \ldots, \bar{Z}_r$, and similarly for S.

Starting from the set of generators for B, we add generators to get a complete set of generators for G_n: first we add symplectic partners $\bar{X}_{c+1}, \ldots, \bar{X}_r$ for the isotropic generators $\bar{Z}_{c+1}, \ldots, \bar{Z}_r$, and then we add $n - r$ additional symplectic pairs $(\bar{Z}_{r+1}, \bar{X}_{r+1}), \ldots, (\bar{Z}_n, \bar{X}_n)$ to get a complete set of

generators. In exactly the same way, we add new generators $\hat{X}_{c+1}, \ldots, \hat{X}_r$ and $(\hat{Z}_{r+1}, \hat{X}_{r+1}), \ldots, (\hat{Z}_n, \hat{X}_n)$ to the generators for S to get a different complete set of generators for G_n.

If any of the generators have eigenvalues $\pm i$, multiply them by i, so that all generators have eigenvalues ± 1. (This is allowed because we are ignoring the overall phases.) Now define the following two states: $|\bar{\mathbf{0}}\rangle$ is the simultaneous $+1$ eigenstate of the generators $\bar{Z}_1, \ldots, \bar{Z}_n$, and $|\hat{\mathbf{0}}\rangle$ is the simultaneous $+1$ eigenstate of the generators $\hat{Z}_1, \ldots, \hat{Z}_n$. Starting from these two states, we construct two orthonormal bases. Let $\mathbf{b} = (b_1, \ldots, b_n)$ be a string of n bits. Define the basis states $|\bar{\mathbf{b}}\rangle = \bar{X}_1^{b_1} \cdots \bar{X}_n^{b_n} |\bar{\mathbf{0}}\rangle$ and $|\hat{\mathbf{b}}\rangle = \hat{X}_1^{b_1} \cdots \hat{X}_n^{b_n} |\hat{\mathbf{0}}\rangle$. It is easy to show that these form two orthonormal bases. Therefore we can define a unitary operator

$$U = \sum_{\mathbf{b}} |\bar{\mathbf{b}}\rangle \langle \hat{\mathbf{b}}|.$$

One can now show that $\bar{Z}_j = U \hat{Z}_j U^\dagger$ and $\bar{X}_j = U \hat{X}_j U^\dagger$ for all j. This implies that any operator in $\mathcal{S}$ can be mapped to a corresponding operator in $\mathcal{B}$, by writing the two operators as products of corresponding generators.

<div align="right">□</div>

As a consequence of this lemma, the error-correcting codes $\mathcal{C}(B)$ and $\mathcal{C}(S)$ are also related by a unitary transformation. In what follows, we will use the straightforward group B to discuss the error-correcting conditions for an EAQECC, and then translate the results back to the code $\mathcal{C}(S)$. As we will see, the unitary U constructed in the lemma can be thought of as the *encoding* operator for the code $\mathcal{C}(S)$. Note that while this unitary has been expressed in abstract terms, there are efficient techniques to directly find a quantum circuit for U in terms of CNOTs, Hadamards, and phase gates [CG97, GRB03, WB10a, WB10b].

2.3 Anticommuting pairs and entanglement

What is the code space $\mathcal{C}(B)$ described by B in (6)? Because B is not a commuting group, the usual definition of a QECC $\mathcal{C}(B)$ does not apply, since the generators do not have a common $+1$ eigenspace. However, by *extending* the generators, we can find a new group that *is* Abelian, and for which the usual definition of code space *does* apply. The qubits of the codewords will be embedded in a larger space. Notice that we can append a Z operator at the end of Z_1, a X operator at the end of X_1, and an identity at the end of Z_2 and Z_3, to make B into a new Abelian group B_e:

$$
\begin{array}{rcccc|c}
Z_1' = & Z & I & I & I & Z \\
X_1' = & X & I & I & I & X \\
Z_2' = & I & Z & I & I & I \\
Z_3' = & I & I & Z & I & I
\end{array}
\tag{7}
$$

The four original qubits are possessed by Alice (the sender), but the additional qubit is possessed by Bob (the receiver) and is not subject to errors. Let B_e be the extended group generated by $\{Z_1', X_1', Z_2', Z_3'\}$. We define the code space $\mathcal{C}(B)$ to be the simultaneous $+1$ eigenspace of all elements of B_e, and we can write it down explicitly in this case:

$$\mathcal{C}(B) = \{|\Phi\rangle^{AB} |0\rangle |0\rangle |\psi\rangle\},\tag{8}$$

where $|\Phi\rangle^{AB}$ is the maximally entangled state $(|00\rangle + |11\rangle)/\sqrt{2}$ shared between Alice and Bob, and $|\psi\rangle$ is an arbitrary single-qubit pure state. (Bob's qubit corresponds to the fifth column in (7).) Because entanglement is used, this is an EAQECC. The number of ebits c needed for the encoding

is equal to the number of anti-commuting pairs of generators in B_S. The number of ancilla bits s equals the number of independent generators in B_I. The number k of encoded qubits equals to $n - c - s$. Therefore, $\mathcal{C}(B)$ is a $[[4, 1; 1]]$ EAQECC with zero net rate: $n = 4$, $c = 1$, $s = 2$ and $k = 1$. Note that zero net rate does not mean that no qubits are transmitted by this code! Rather, it implies that the number of ebits needed is equal to the number of qubits transmitted. In general, $k - c$ can be positive, negative, or zero.

Now we will see how the error-correcting conditions are related to the generators of B. If an error $E_a \otimes I^B$ anticommutes with one or more of the operators in $\{Z'_1, X'_1, Z'_2, Z'_3\}$, it can be detected by measuring these operators. This will only happen if the error E_a on Alice's qubits anticommutes with one of the operators in the original set of generators $\{Z_1, X_1, Z_2, Z_3\}$, since the entangled bit held by Bob is assumed to be error-free. Alternatively, if $E_a \otimes I^B \in B_e$, or equivalently $E_a \in B_I$, then E_a does not corrupt the encoded state. (In this case we call the code *degenerate*.) Altogether, $\mathcal{C}(B)$ can correct a set of errors $\mathcal{E}_0$ if and only if $E_a^\dagger E_b \in B_I \cup (G_4 - Z(B))$ for all $E_a, E_b \in \mathcal{E}_0$.

With this analysis of B, we can go back to determine the error-correcting properties of our original stabilizer S. We can construct a QECC from a nonabelian group S if entanglement is available, just as we did for the group B. We add extra operators Z and X on Bob's side to make S abelian as follows:

$$
\begin{array}{rccccc|c}
\bar{Z}'_1 = & Z & X & Z & I & \big| & Z \\
\bar{X}'_1 = & Z & Z & I & Z & \big| & X \\
\bar{Z}'_2 = & Y & X & X & Z & \big| & I \\
\bar{Z}'_3 = & Z & Y & Y & X & \big| & I
\end{array}
\tag{9}
$$

where the extra qubit is once again assumed to be possessed by Bob and to be error-free. Let S_e be the group generated by the above operators. Since $B \cong S$, let U^{-1} be the unitary from Lemma 2. Define the code space $\mathcal{C}(S)$ by $\mathcal{C}(S) = U^{-1}(\mathcal{C}(B))$, where the unitary U^{-1} acts only on Alice's side. This unitary U^{-1} can be interpreted as the encoding operation of the EAQECC defined by S. Observe that the code space $\mathcal{C}(S)$ is a simultaneous eigenspace of all elements of S_e. As in the case of $\mathcal{C}(B)$, the code $\mathcal{C}(S)$ can correct a set of errors $\mathcal{E}$ if and only if

$$
E_a^\dagger E_b \in S_I \cup (G_4 - Z(S)),
\tag{10}
$$

for all $E_a, E_b \in \mathcal{E}$.

The algebraic description is somewhat abstract, so let us translate this into a physical picture. Alice wishes to encode a single ($k = 1$) qubit state $|\psi\rangle$ into four ($n = 4$) qubits, and transmit them through a noisy channel to Bob. Initially, Alice and Bob share a single ($c = 1$) maximally entangled pair of qubits—one ebit. Alice performs the encoding operation U^{-1} on her bit $|\psi\rangle$, her half of the entangled pair, and two ($s = 2$) ancilla bits. She then sends the four qubits through the channel to Bob. Bob measures the extended generators $\bar{Z}'_1$, $\bar{X}'_1$, $\bar{Z}'_2$, and $\bar{Z}'_3$ on the four received qubits plus his half of the entangled pair. The outcome of these four measurements gives the error syndrome; as long as the error set satisfies the error-correcting condition (10), Bob can correct the error and decode the transmitted qubit $|\psi\rangle$. We can see schematically how this procedure works in Fig. 1.

In fact, this particular example is a $[[4, 1, 3; 1]]$ EAQECC: it can correct any single-qubit error. No standard QECC with $n < 5$ can correct an arbitrary single-qubit error. Clearly, the use of an ebit enhances the error-correcting power of the code. However, the need for a shared ebit is also a limitation, since the ebit must presumably have been shared originally by extra channel uses—this is reflected in the fact that the net rate of the code is zero.

We have worked out the procedure for a particular example, but any entanglement-assisted quantum error correction code will function in the same way. If we have c anticommuting pairs of generators $\bar{Z}_j$ and $\bar{X}_j$ in S_E, we resolve their anticommutation by adding a Z and X operator,

respectively, acting on an extra qubit on Bob's side. We will need one such extra qubit for each pair, or c in all; these correspond to c ebits initially shared between Alice and Bob. The particular parameters n, k, c, s will vary depending on the code. It should be noted that the first example of entanglement-assisted error correction produced a $[[3, 1, 3; 2]]$ EAQECC based on the $[[5, 1, 3]]$ standard QECC [Bow02]. We have now produced a completely general description, that also eschews the need for a commuting stabilizer group.

2.4 The symplectic representation of EAQECCs

There is another very useful representation for stabilizer QECCs where tensor products of Pauli operators are represented by pairs of bit strings: $e^{i\phi} X^{\mathbf{a}} Z^{\mathbf{b}} \to (\mathbf{a}|\mathbf{b})$, where $\mathbf{a}$ and $\mathbf{b}$ are strings of bits $\mathbf{a} \equiv a_1 a_2 \cdots a_n$ and $\mathbf{b} \equiv b_1 b_2 \cdots b_n$, and we use the power notation:

$$X^{\mathbf{a}} \equiv X^{a_1} \otimes X^{a_2} \otimes \cdots \otimes X^{a_n}, \quad Z^{\mathbf{b}} \equiv Z^{b_1} \otimes Z^{b_2} \otimes \cdots \otimes Z^{b_n}. \tag{11}$$

The overall phase is lost in this representation, but for our present purposes this is not important. We also define a map in the other direction, from a pair of bit strings to a Pauli operator: $\mathbf{w} \to N_{\mathbf{w}} \equiv X^{\mathbf{a}} Z^{\mathbf{b}}$, where $\mathbf{w} = (\mathbf{a}|\mathbf{b})$.

This symplectic representation has many advantages. Up to a phase, multiplication of Pauli operators is given by binary vector addition. A set of generators $g_1, \ldots, g_{n-k}$ for the stabilizer is given by a set of $n - k$ binary strings of length $2n$, which we write as the rows of a matrix. We can thus represent a quantum stabilizer code by a quantum check matrix, quite analogous to a classical linear code:

$$g_1, \ldots, g_{n-k} \longrightarrow \hat{H} = (H_X | H_Z), \tag{12}$$

where H_X and H_Z are $(n - k) \times n$ binary matrices. The rows of $\hat{H}$ represent the stabilizer generators, and the rowspace of $\hat{H}$ represents the full set of stabilizer operators.

While this symplectic representation does not include the overall phase of Pauli group elements, it does capture the commutation relations between different elements of the Pauli group. If two group elements g and g' are represented by the strings $(\mathbf{a}|\mathbf{b})$ and $(\mathbf{a}'|\mathbf{b}')$, respectively, then their commutation relation is given by the *symplectic inner product* of the two strings:

$$(\mathbf{a}|\mathbf{b}) \odot (\mathbf{a}'|\mathbf{b}') = \mathbf{a} \cdot \mathbf{b}' + \mathbf{a}' \cdot \mathbf{b}, \tag{13}$$

where $\mathbf{a} \cdot \mathbf{b}'$ denotes the usual Boolean inner product. The symplectic inner product between two strings is either 0 or 1; if it is 0, then the operators they represent commute, otherwise they anticommute:

$$N_{\mathbf{w}} N_{\mathbf{w}'} = (-1)^{\mathbf{w} \odot \mathbf{w}'} N_{\mathbf{w}'} N_{\mathbf{w}}.$$

As we will see, this enables us to derive a compact formula for the amount of entanglement needed by a given code. For a standard stabilizer QECC, the symplectic inner product between any two rows of the quantum check matrix $\hat{H}$ must be 0.

For EAQECCs, we typically use the symplectic representation in two slightly different but related ways. First, we can represent the (in general noncommuting) generators on Alice's qubits by an $(n - k) \times 2n$ check matrix, whose rows can have nonzero symplectic inner product. So, for example, the generators in Eq. (5) are represented by the following quantum check matrix:

$$
\begin{array}{l}
\bar{Z}_1 = \\
\bar{X}_1 = \\
\bar{Z}_2 = \\
\bar{Z}_3 =
\end{array}
\begin{array}{cccc}
Z & X & Z & I \\
Z & Z & I & Z \\
Y & X & X & Z \\
Z & Y & Y & X
\end{array}
\longrightarrow
\left(
\begin{array}{cccc|cccc}
0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\
1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\
0 & 1 & 1 & 1 & 1 & 1 & 1 & 0
\end{array}
\right). \tag{14}
$$

It is also useful to represent the augmented operators, in which c extra qubits have been added on Bob's side in order to resolve the anticommutativity of the stabilizer group. In this case, we include the operators on Bob's side as well, with the convention that the bits representing operators on Bob's side will be listed in the c rightmost columns of H_X and H_Z. Taking the augmented generators from (9) we get the symplectic representation

$$
\begin{array}{rcccccc}
\bar{Z}'_1 = & Z & X & Z & I & Z \\
\bar{X}'_1 = & Z & Z & I & Z & X \\
\bar{Z}'_2 = & Y & X & X & Z & I \\
\bar{Z}'_3 = & Z & Y & Y & X & I
\end{array}
\longrightarrow
\left(
\begin{array}{ccccc|ccccc}
0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\
1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\
0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0
\end{array}
\right)
\tag{15}
$$

where the fifth and tenth columns represent the X and Z parts, respectively, of the operators on Bob's qubit.

2.5 The Canonical Code

Consider the following trivial encoding operation defined by

$$
|\varphi\rangle \mapsto |\Phi\rangle^{\otimes c}|0\rangle|\varphi\rangle.
\tag{16}
$$

In other words, registers containing $|0\rangle$ (of size $s = n - k - c$ qubits) and $|\Phi\rangle^{\otimes c}$ (c ebits shared beween Alice and Bob) are appended to the register containing the "encoded" information $|\varphi\rangle$ (of size k qubits). To make the comparison to other codes, we can think of Alice appending the ancillas and her halves of the entangled pairs, and then applying an encoding unitary that is simply the identity I. What errors can we correct with such a simple-minded encoding?

Proposition 1. The code given by (16) and a suitably defined decoding map $\mathcal{D}_0$ can correct an error set (represented in symplectic form)

$$
\begin{aligned}
\mathcal{S}_0 \ = \ & \{(\mathbf{a}_1, \mathbf{b}_1, \mathbf{ff}(\mathbf{a}_1, \mathbf{a}_2, \mathbf{b}_1)|\mathbf{a}_2, \mathbf{b}_2, \mathbf{fi}(\mathbf{a}_1, \mathbf{a}_2, \mathbf{b}_1)) : \\
& \mathbf{a}_1, \mathbf{a}_2 \in (\mathbb{Z}_2)^c, \mathbf{b}_1, \mathbf{b}_2 \in (\mathbb{Z}_2)^s\},
\end{aligned}
\tag{17}
$$

for any pair of known functions $\mathbf{ff}, \mathbf{fi} : (\mathbb{Z}_2)^c \times (\mathbb{Z}_2)^c \times (\mathbb{Z}_2)^s \to (\mathbb{Z}_2)^k$. To put this more intuitively: an error set is correctable by the canonical code if every error in the error set leaves a record in the appended registers that determines exactly what has been done to the information qubits and how to undo it. (This is why the functions $\mathbf{ff}$ and $\mathbf{fi}$ must be known functions.)

Proof. The protocol is shown in figure 2. After applying an error $N_\mathbf{u}$ with

$$
\mathbf{u} = (\mathbf{a}_1, \mathbf{b}_1, \mathbf{ff}(\mathbf{a}_1, \mathbf{a}_2, \mathbf{b}_1)|\mathbf{a}_2, \mathbf{b}_2, \mathbf{fi}(\mathbf{a}_1, \mathbf{a}_2, \mathbf{b}_1))
\tag{18}
$$

the encoded state $|\Phi\rangle^{\otimes c}|0\rangle|\varphi\rangle$ becomes (up to a phase factor)

$$
\begin{aligned}
& (X^{\mathbf{a}_1} Z^{\mathbf{a}_2} \otimes I)|\Phi\rangle^{\otimes c} \otimes X^{\mathbf{b}_1} Z^{\mathbf{b}_2}|0\rangle \otimes X^{\mathbf{ff}(\mathbf{a}_1, \mathbf{a}_2, \mathbf{b}_1)} Z^{\mathbf{fi}(\mathbf{a}_1, \mathbf{a}_2, \mathbf{b}_1)}|\varphi\rangle \\
= \ & |\mathbf{a}_1, \mathbf{a}_2\rangle \otimes |\mathbf{b}_1\rangle \otimes X^{\mathbf{ff}(\mathbf{a}_1, \mathbf{a}_2, \mathbf{b}_1)} Z^{\mathbf{fi}(\mathbf{a}_1, \mathbf{a}_2, \mathbf{b}_1)}|\varphi\rangle,
\end{aligned}
\tag{19}
$$

where $|\mathbf{b}_1\rangle = X^{\mathbf{b}_1}|0\rangle$ and $|\mathbf{a}_1, \mathbf{a}_2\rangle = (X^{\mathbf{a}_1} Z^{\mathbf{a}_2} \otimes I)|\Phi\rangle^{\otimes c}$. As the vector $(\mathbf{a}_1, \mathbf{a}_2, \mathbf{b}_1, \mathbf{b}_2)^\mathsf{T}$ completely specifies the error $\mathbf{u}$, it is called the *error syndrome*. The state (19) only depends on the *reduced syndrome* $\mathbf{r} = (\mathbf{a}_1, \mathbf{a}_2, \mathbf{b}_1)^\mathsf{T}$. In effect, $\mathbf{b}_1$ and $(\mathbf{a}_1, \mathbf{a}_2)$ have been encoded into the ancillas and ebits using plain and superdense coding, respectively. Bob, who holds the entire state (19) at the end, may identify the reduced syndrome by a simple projective measurement. Bob simultaneously
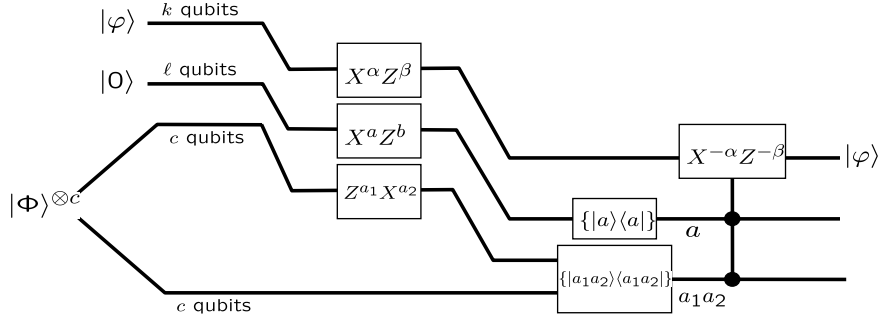
9

Figure 2: The canonical code. Note that this code differs from a general EAQECC only in that the encoding unitary used is the identity, and we assume a particular form for the errors.

measures the $Z^{\mathbf{e}_1} \otimes Z^{\mathbf{e}_1}, \ldots, Z^{\mathbf{e}_c} \otimes Z^{\mathbf{e}_c}$ observables to decode $\mathbf{a}_1$ (where the operators to the right of the $\otimes$ symbol act on Bob's halves of the ebits), the $X^{\mathbf{e}_1} \otimes X^{\mathbf{e}_1}, \ldots, X^{\mathbf{e}_c} \otimes X^{\mathbf{e}_c}$ observables to decode $\mathbf{a}_2$, and the $Z^{\mathbf{e}_{c+1}}, \ldots, Z^{\mathbf{e}_{c+s}}$ observables to decode $\mathbf{b}_1$, where $\{\mathbf{e}_j\}$ are the standard basis vectors. He then performs $Z^{\mathbf{fi}(\mathbf{a},\mathbf{a}_1,\mathbf{a}_2)} X^{\mathbf{ff}(\mathbf{a},\mathbf{a}_1,\mathbf{a}_2)}$ on the remaining k qubit system, leaving it in the original state $|\varphi\rangle$.

Since the goal is the transmission of quantum information, no actual measurement is necessary. Instead, Bob can perform the CPTP map $\mathcal{D}_0$ consisting of the controlled unitary

$$U_{0\,\mathrm{dec}} = \sum_{\mathbf{a}_1, \mathbf{a}_2, \mathbf{b}_1} \hat{\mathcal{P}}_{\mathbf{a}_1, \mathbf{a}_2} \otimes \hat{\mathcal{P}}_{\mathbf{b}_1} \otimes Z^{\mathbf{fi}(\mathbf{a}_1, \mathbf{a}_2, \mathbf{b}_1)} X^{\mathbf{ff}(\mathbf{a}_1, \mathbf{a}_2, \mathbf{b}_1)}, \tag{20}$$

followed by discarding the first two subsystems. Here we use projection operators $\hat{\mathcal{P}}_{\mathbf{a}_1, \mathbf{a}_2} = |\mathbf{a}_1, \mathbf{a}_2\rangle\langle\mathbf{a}_1, \mathbf{a}_2|$ and $\hat{\mathcal{P}}_{\mathbf{b}_1} = |\mathbf{b}_1\rangle\langle\mathbf{b}_1|$. (Note that equations very similar to (18–20) arise in the theory of continuous variable error-correcting codes, as described in [LB13, Chapter 22]. $\square$

The above code is *degenerate* with respect to the error set S, which means that the error can be corrected without knowing the full error syndrome. This is because the $Z^{\mathbf{b}}$ operator acting on the ancillas has no effect beyond a trivial global phase. It is our assumption that the errors acting on the information qubits did not depend on $\mathbf{b}$ that makes this error set correctable. This kind of restriction will always be necessary—no error-correcting code can correct every possible error. Part of the art of error correction is to choose a code whose correctable error set matches, as closely as possible, the actual physical errors that are likely to happen.

We can characterize our code in terms of the check matrix $\hat{H}$ given by

$$\hat{H} = \begin{pmatrix} \hat{H}_I \\ \hat{H}_S \end{pmatrix}, \tag{21}$$

$$\hat{H}_I = \begin{pmatrix} \mathbf{0}_{s \times c} & \mathbf{0}_{s \times s} & \mathbf{0}_{s \times k} \,\big|\, \mathbf{0}_{s \times c} & \mathbf{I}_{s \times s} & \mathbf{0}_{s \times k} \end{pmatrix}, \tag{22}$$

$$\hat{H}_S = \begin{pmatrix} \mathbf{I}_{c \times c} & \mathbf{0}_{c \times s} & \mathbf{0}_{c \times k} & \mathbf{0}_{c \times c} & \mathbf{0}_{c \times s} & \mathbf{0}_{c \times k} \\ \mathbf{0}_{c \times c} & \mathbf{0}_{c \times s} & \mathbf{0}_{c \times k} & \mathbf{I}_{c \times c} & \mathbf{0}_{c \times s} & \mathbf{0}_{c \times k} \end{pmatrix}, \tag{23}$$

with $s = n - k - c$.

The row space of $\hat{H}$ decomposes into a direct sum of the *isotropic* subspace, given by the row space of $\hat{H}_I$ and the *symplectic* subspace, given by the row space of $\hat{H}_S$. Define the *symplectic code* corresponding to $\hat{H}$ by

$$\mathcal{C}_0 = \text{rowspace}(\hat{H})^{\perp}$$

where

$$V^{\perp} = \{\mathbf{w} : \mathbf{w} \odot \mathbf{u}^{\mathsf{T}} = 0, \ \forall \mathbf{u} \in V\}.$$

Note that $(V^{\perp})^{\perp} = V$. Then $\mathcal{C}_0^{\perp} = \text{rowspace}(\hat{H})$, $\text{iso}(\mathcal{C}_0^{\perp}) = \text{rowspace}(\hat{H}_I)$ and $\text{symp}(\mathcal{C}_0^{\perp}) = \text{rowspace}(\hat{H}_S)$.

Just as with standard stabilizer codes (as described earlier in this chapter), each row of the check matrix $\hat{H}$ maps to a stabilizer generator, and the complete set of vectors in the row space of $\hat{H}$ maps onto the complete set of stabilizer operators (up to phases). The rows of $\hat{H}_S$ come in nonorthogonal pairs under the symplectic inner product; these nonorthogonal pairs of rows map into anticommuting pairs of generators in the stabilizer. Each such pair requires one extra qubit on Bob's side to resolve the noncommutativity of the stabilizer, and hence requires one shared ebit between Alice and Bob at the encoding step. The number of ebits used in the code is therefore

$$c = \frac{1}{2} \dim \text{rowspace}(\hat{H}_S)$$

and the number of encoded qubits is

$$
\begin{aligned}
k &= n - \dim \text{rowspace}(\hat{H}_I) - \frac{1}{2} \dim \text{rowspace}(\hat{H}_S) \\
&= n - \dim \text{rowspace}(\hat{H}) + c.
\end{aligned}
\tag{24}
$$

The code parameter $\hat{k} := k - c$, which is the number of encoded qubits minus the number of ebits used, is independent of the symplectic structure of $\hat{H}$:

$$\hat{k} = n - \dim \text{rowspace}(\hat{H}). \tag{25}$$

We see that the quantity k/n is the rate of the code, and $\hat{k}/n$ is the net rate.

The correctable error set S_0 of this symplectic code can be described in terms of $\hat{H}$:

Proposition 2. The set S_0 of errors correctable by the canonical code (in symplectic form) is such that, if $\mathbf{u}, \mathbf{u}' \in S_0$ and $\mathbf{u} \neq \mathbf{u}'$, then either

1. $\mathbf{u} - \mathbf{u}' \notin \mathcal{C}_0$ (equivalently: $\hat{H} \odot (\mathbf{u} - \mathbf{u}')^{\mathsf{T}} \neq \mathbf{0}^{\mathsf{T}}$), or

2. $\mathbf{u} - \mathbf{u}' \in \text{iso}(\mathcal{C}_0^{\perp})$ (equivalently: $\mathbf{u} - \mathbf{u}' \in \text{rowspace}(\hat{H}_I)$).

Proof. If $\mathbf{u}$ is given by (18) then $\hat{H} \odot \mathbf{u}^{\mathsf{T}} = \mathbf{r} = (\mathbf{a}_1, \mathbf{a}_2, \mathbf{b}_1)^{\mathsf{T}}$, the reduced error syndrome. By definition (17), two distinct elements of S_0 either have different reduced syndromes $(\mathbf{a}_1, \mathbf{a}_2, \mathbf{b}_1)$ (condition 1) or they differ by a vector of the form $(\mathbf{0}, \mathbf{0}, \mathbf{0}|\mathbf{0}, \mathbf{b}, \mathbf{0})$ (condition 2). In the first case, the two errors can be distinguished, and appropriate different corrections applied. In the second case, the two errors have the same reduced syndrome and hence cannot be distinguished, but they are both correctable by the same operation. Observe that condition 1 is analogous to the usual error correcting condition for classical codes [MS77], while condition 2 is an example of the quantum phenomenon of degeneracy. □

The parity check matrix $\hat{H}$ also specifies the encoding and decoding operations. The space $\mathcal{H}_2^{\otimes k}$ is encoded into the *codespace* defined by

$$\mathcal{C}_0 = \{|\Phi\rangle^{\otimes c}|\mathbf{0}\rangle|\varphi\rangle : |\varphi\rangle \in \mathcal{H}_2^{\otimes k}\}.$$

It is not hard to see that the codespace is the simultaneous $+1$ eigenspace of the commuting operators:

1. $I \otimes Z^{\mathbf{e}_i} \otimes I \otimes I$, $i = 1, \ldots, s$;

2. $Z^{\mathbf{e}_j} \otimes I \otimes I \otimes Z^{\mathbf{e}_j}$, $j = 1, \ldots, c$;

3. $X^{\mathbf{e}_j} \otimes I \otimes I \otimes X^{\mathbf{e}_j}$, $j = 1, \ldots, c$.

Above, in operators written $A_1 \otimes A_2 \otimes A_3 \otimes B$ the first three operators A_1, A_2, A_3 act on Alice's qubits, and the fourth operator B on Bob's. Define the check matrix

$$H_B = \begin{pmatrix} \mathbf{0}_{s\times c} & \mathbf{0}_{s\times c} \\ \mathbf{I}_{c\times c} & \mathbf{0}_{c\times c} \\ \mathbf{0}_{c\times c} & \mathbf{I}_{c\times c} \end{pmatrix}. \tag{26}$$

Define the *augmented* parity check matrix $\hat{H}_{\mathrm{aug}} = (\hat{H}, H_B)$

$$\hat{H}_{\mathrm{aug}} = \begin{pmatrix} \mathbf{0}_{s\times c} & \mathbf{0}_{s\times s} & \mathbf{0}_{s\times k} & \mathbf{0}_{s\times c} & \mathbf{0}_{s\times c} & \mathbf{I}_{s\times s} & \mathbf{0}_{s\times k} & \mathbf{0}_{s\times c} \\ \mathbf{I}_{c\times c} & \mathbf{0}_{c\times s} & \mathbf{0}_{c\times k} & \mathbf{I}_{c\times c} & \mathbf{0}_{c\times c} & \mathbf{0}_{c\times s} & \mathbf{0}_{c\times k} & \mathbf{0}_{c\times c} \\ \mathbf{0}_{c\times c} & \mathbf{0}_{c\times s} & \mathbf{0}_{c\times k} & \mathbf{0}_{c\times c} & \mathbf{I}_{c\times c} & \mathbf{0}_{c\times s} & \mathbf{0}_{c\times k} & \mathbf{I}_{c\times c} \end{pmatrix}. \tag{27}$$

Observe that rowspace$(\hat{H}_{\mathrm{aug}})$ is purely isotropic. The codespace is now described as the simultaneous $+1$ eigenspace of

$$\{N_{\mathbf{w}} : \mathbf{w} \in \mathrm{rowspace}(\hat{H}_{\mathrm{aug}})\},$$

or, equivalently that of

$$G_0 = \langle N_{\mathbf{w}} : \mathbf{w} \text{ is a row of } \hat{H}_{\mathrm{aug}}\rangle.$$

The decoding operation $\mathcal{D}_0$ is also described in terms of $\hat{H}$. The reduced syndrome $\mathbf{r} = \hat{H} \odot \mathbf{u}^{\mathsf{T}}$ is obtained by simultaneously measuring the observables in $\mathcal{G}_0$. The reduced error syndrome corresponds to an equivalence class of possible errors $\mathbf{u} \in S_0$ that all have an identical effect on the codespace, and can all be corrected by the same operation. Bob performs $\hat{N}_{\mathbf{u}} = \hat{N}_{-\mathbf{u}}$ to undo the error.

2.6 General codes and code parameters

The canonical code described above is not very useful for actual error correction. It corrects a set of highly nonlocal errors, of a type that are not likely to occur in reality. However, it is very helpful in clarifying how the process of error correction works, and the role of the resources used in encoding. Each added ancilla can contain, in principle, one bit of information about any errors that occur. Each shared ebit between the sender and receiver can contain two bits of information about errors. This information about the errors is retrieved by measuring the stabilizer generators. After Alice's qubits have passed through the channel, Bob has all the qubits and can perform these measurements.

A general EAQECC has exactly the same algebraic structure as the canonical code. The encoded state, before going through the channel, has c ebits of entanglement with the c qubits on

Bob's side. Corresponding to this entanglement are c pairs of stabilizer generators. Each pair of generators has the following structure: a pair of anticommuting operators on Alice's qubits, tensored with a Z and an X, respectively, on one of the qubits on Bob's side to "resolve" the anticommutativity. These pairs of generators are the symplectic pairs of the EAQECC. In addition to these pairs, there can be single generators that act as the identity on all of Bob's qubits. These are the isotropic generators. For an $[[n,k,d;c]]$ EAQECC, each of the stabilizer elements acts on $n+c$ qubits, n on Alice's side and c on Bob's side; there is a set of $n-k$ generators for the stabilizer chosen in a standard form, with $2c$ symplectic pairs of generators and $s = n - k - 2c$ isotropic generators.

By Lemma 2, if two sets of Pauli operators can be put in one-to-one correspondence so that they have the same commutation relations, there is a unitary transformation that turns one set into the other (up to a phase). The $n-k$ generators of an $[[n,k;c]]$ EAQECC can be mapped by a unitary operator U_E to the generators of a canonical code.

It is important to realize is that the canonical code describes the *input* of the EAQECC *before encoding*. It describes a set of k information qubits, c ebits, and s ancillas. The unitary U_E transforms the stabilizer for this input into the stabilizer for the EAQECC. Therefore U_E is a unitary encoding operator for this code. It transforms the localized states of the input qubits into nonlocal, entangled states spread over all the qubits. By this transformation, the correctable error set is transformed from the strange, nonlocal errors of the canonical code to a more physically realistic set of errors (most commonly, localized errors on the individual qubits).

3 Constructing EAQECCs from classical linear codes

3.1 Mapping $GF(4)$ to Pauli Operators

We will now examine the $[[4,1;1]]$ EAQECC used as an example in Section 2 above, and show that it can be derived from a classical non-dual-containing quaternary $[4,2]$ code. This is a generalization of the well-known CRSS construction for standard QECCs [CRSS98].

First, note that this $[[4,1;1]]$ code is non-degenerate, and can correct an arbitrary one-qubit error. (The distance d of the code $\mathcal{C}(S)$ is 3.) This is because the 12 errors X_i, Y_i and Z_i, $i = 1, \ldots, 4$, have distinct non-zero error syndromes. X_i denotes the bit flip error on i-th qubit, Z_i denotes the phase error on i-th qubit, and Y_i means that both a bit flip and phase flip error occur on the i-th qubit. It suffices to consider only these three standard one-qubit errors, since any other one-qubit error can be written as a linear combination of these three errors and the identity.

Next, we define the following map between the Pauli operators, symplectic strings, and elements of $GF(4)$, the field with 4 elements:

G	I	X	Y	Z
Symplectic	00	10	11	01
$GF(4)$	0	$\overline{\omega}$	1	ω

Addition of symplectic strings is by bitwise XOR. The elements of $GF(4)$ obey a simple addition rule: $0 + x = x + 0 = x$ and $x + x = 0$ for all $x \in GF(4)$, $\overline{\omega} + 1 = \omega$, $1 + \omega = \overline{\omega}$, and $\omega + \overline{\omega} = 1$, where addition is commutative. Multiplication is also commutative: $0 \cdot x = 0$ and $1 \cdot x = x$ for all $x \in GF(4)$, $\omega \cdot \omega = \overline{\omega}$, $\overline{\omega} \cdot \overline{\omega} = \omega$, and $\omega \cdot \overline{\omega} = 1$. Note that under this map, addition in $GF(4)$ (or of symplectic strings) corresponds to multiplication of the Pauli operators, up to an overall phase. So multiplication of two elements of G_n corresponds to addition of two n-vectors over

13

$GF(4)$, up to an overall phase. The same correspondence applies between multiplication of the Pauli operators and bitwise addition (or XOR) of the symplectic strings.

We can define a linear code over $GF(4)$ in exactly the same way we do for a binary or symplectic code, by writing down a check matrix. Consider the matrix

$$H_4 = \begin{pmatrix} 1 & \omega & 1 & 0 \\ 1 & 1 & 0 & 1 \end{pmatrix}. \tag{28}$$

The code is the null space of H_4 over $GF(4)$—the set of vectors orthogonal to all the rows of the check matrix. H_4 is the check matrix of a classical $[4, 2, 3]$ quaternary code whose rows are not orthogonal, and 3 is the minimum distance between codewords.

From this starting point we can define a new matrix

$$\tilde{H}_4 = \begin{pmatrix} \omega H_4 \\ \bar{\omega} H_4 \end{pmatrix} = \begin{pmatrix} \omega & \bar{\omega} & \omega & 0 \\ \omega & \omega & 0 & \omega \\ \bar{\omega} & 1 & \bar{\omega} & 0 \\ \bar{\omega} & \bar{\omega} & 0 & \bar{\omega} \end{pmatrix}. \tag{29}$$

As a quaternary code, this defines exactly the same code space as the original H_4, since the last two rows are multiples of the first two rows. However, we now apply our map from $GF(4)$ to G_4, where each row of $\tilde{H}_4$ is mapped to a Pauli operator in G_4. The resulting set of operators is exactly the set of generators $\{M_i\}$ given in (4). If we map the matrix $\tilde{H}_4$ to a symplectic matrix, using the correspondence given above, we see that we get exactly the same symplectic representation for the code as in section 2.4. We can carry out this procedure with any linear quaternary code.

The generators corresponding to the rows of the matrix $\tilde{H}_4$ will not commute, in general. In that case, we must apply Lemma 1 to find a new set of generators comprising isotropic generators (that commute with all other generators) and symplectic pairs of generators (that anticommute with each other but commute with all other generators). Extra operators can be added on Bob's side to resolve the anticommutativity, as described above. This produces an EAQECC using c ebits, where c is the number of symplectic pairs.

In the example above, we get a $[[4, 1, 3; 1]]$ EAQECC from a classical $[4, 2, 3]$ quaternary code. This outperforms the best 4-bit self-dual QECC currently known, which is $[[4, 0, 2]]$, both in rate and distance [CRSS98]. This connection between EAQECCs and quaternary classical codes is quite general [BDH14]. Given an arbitrary classical $[n, k, d]$ quaternary code, we can use (29) to construct a non-degenerate $[[n, 2k - n + c, d; c]]$ EAQECC. The rate becomes $(2k - n)/n$ because the $n - k$ classical parity checks give rise to $2(n - k)$ quantum stabilizer generators. This gives us a tremendous advantage in constructing quantum codes, because the parameters of the quantum code depend exactly on those for the classical code: the minimum distance is the same, and high-rate classical codes produce high-rate (though not as high) quantum codes.

The one parameter that is not simply determined by the parameters of the classical code is c, the number of ebits used. For a standard QECC, which uses no entanglement and hence has $c = 0$, the rows of the classical check matrix are all orthogonal. Intuitively, we might expect the amount of entanglement needed to grow the further the matrix is from being self-orthogonal. In fact, for a CRSS code of the type described above, the amount of entanglement needed is

$$c = \text{rank}\left(H_4 H_4^\dagger\right), \tag{30}$$

where the conjugate $H_4^\dagger$ is defined by taking the transpose of H_4 and interchanging $\omega \leftrightarrow \bar{\omega}$ [WB08, Wil09]. For a standard QECC, in which the rows are all orthogonal, $H_4 H_4^\dagger$ will be the zero

14

$n\backslash\hat{k}$	0	1	2	3	4	5	6	7	8	9	10
3	2	2*	1	1							
4	3*	2	2	1	1						
5	3	3	2	2*	1	1					
6	4	3	2	2	2	1	1				
7	3	3	2	2	2	2*	1	1			
8	4	3	3	3	2	2	2	1	1		
9	4	4*	3	3	2	2	2	2*	1	1	
10	5*	4	4	3	3	2	2	2	2	1	1

Table 1: Table of codes up to length 10. The best minimum distance is given for a code with length n and net transmission $\hat{k} = k - c$, where k is the number of information qubits and c the number of ebits used. See [BDH14].

matrix. Since this matrix is $(n - k) \times (n - k)$, the maximum rank (and hence the maximum possible entanglement used by the code) is $n - k$. Such a quantum code will transmit $2k - n + c = k$ qubits—the same as the original classical code.

A special case of this construction is where the original classical code is *binary*—that is, when the check matrix H uses only 0s and 1s, and has no elements ω or $\bar{\omega}$. In that case, the procedure for constructing a quantum code is to create a symplectic check matrix of the form

$$\left(\begin{array}{c|c} H & 0 \\ 0 & H \end{array} \right).$$

The stabilizer generators corresponding to the rows of this matrix either contain only Zs and Is or only Xs and Is, and the two sets of generators are identical in form. In a sense, such a code can be thought of as two classical codes superimposed, one to correct bit flips (X errors) and the other to correct phase flips (Z errors). This is the CSS construction, analogous to the way it is defined in [LB13, Chapter 2] for standard QECCs.

Somewhat more generally, we could define a QECC using two different classical binary codes with check matrices H_1 and H_2:

$$\left(\begin{array}{c|c} H_1 & 0 \\ 0 & H_2 \end{array} \right).$$

Once again the generators would use either Xs or Zs, and not both; but the generators will no longer share the same structure. This approach may be useful when the noise is *asymmetric*; for instance, dominated by phase errors with occasional rare bit flips. The simpler CSS code is obviously the special case where $H_1 = H_2 = H$. It is not necessary that the check matrices H_1 and H_2 be the same size; H_1 can be $(n - k_1) \times n$ and H_2 can be $(n - k_2) \times n$. Together they produce a quantum code that encodes $k_1 + k_2 - n + c$ information qubits in n physical qubits with the use of c ebits. For a CSS code of this type the entanglement needed [BDH14] is

$$c = \mathrm{rank}\left(H_1 H_2^T \right). \tag{31}$$

3.2 Performance

In [CRSS98] a table of the best known QECCs was given. In table 1 we show an updated table which includes EAQEC codes. The entries with an asterisk mark improvements over the table

15

from [CRSS98]. All these codes were constructed from classical quaternary codes by the construction in section 3.1, and were found by numerical search. The corresponding classical quaternary code is available online at:

http://www.win.tue.nl/~aeb/voorlincod.html.

4 Catalytic QECCs

EAQECCs make use of an additional resource—prior shared entanglement—to boost the power of an error-correcting code. This paradigm raises several theoretical and practical questions, which address how these codes might be useful in practice, and how they fit into different families of codes that draw on different resources.

The first practical question is: how was this entanglement shared in the first place? If the sender and receiver communicate by a noisy quantum channel, then the entanglement would have to be shared through this channel. Alice locally prepares EPR pairs, and sends half of each pair through the channel, either protected by a standard QECC, or making use of entanglement purification techniques [BBP+96, BDSW96] to produce a smaller number of perfect EPR pairs. So we see that in this context the enhanced rate of an EAQECC must be paid for by additional channel communication to establish the entanglement in the first place. This means that for many purposes it is the *net rate* $(k - c)/n$ of the EAQECC that is the appropriate figure of merit. This net rate need not be better than that achieved by a standard QECC; indeed, it is quite possible for the net rate to be zero or negative.

Even a code with a negative net rate may actually be useful in practice. One great advantage of shared entanglement as a resource is that it is independent of the message being sent, and can in principle be prepared well ahead of time. One natural application of EAQECCs would therefore be to a quantum network where usage varies at different times. During periods of low usage, shared entanglement could be built up between the sender and receiver. During periods of high usage, this shared entanglement could be drawn on to increase the rate of transmission without sacrificing error-correcting power.

EAQECCs with positive net rates can be used in other ways to improve the power and flexibility of quantum communications. To illustrate this, let us consider the idea of a *catalytic* code. Here, in addition to n uses of a noisy channel, the sender may send c qubits noise-free. One obvious approach is to send c information qubits using the noise-free bits, and then send k more qubits through the channel by means of an $[[n, k]]$ standard QECC. However, another way of proceeding would be to use the c noise-free bits to establish c ebits between Alice and Bob, and then use an $[[n, k'; c]]$ EAQECC. So long as $k' \geq k + c$ (and both codes are assumed to correct the errors in the channel), one does as well or better with the second approach as with the straightforward one. In this case, the noise-free qubits are used to boost the rate of the noisy qubits. We call a procedure of this type a *catalytic code*—the c noise-free qubits enhance transmission by an amount equal to at least c.

In practice, noise-free transmission is not really possible. However, it is possible to simulate it by using a standard QECC to protect the c noise-free qubits. Suppose that we start with an $[[n, k; c]]$ EAQECC. To establish c ebits with Bob, Alice would need to send c qubits through the channel. Suppose that she protects them with an $[[n', c]]$ standard QECC. Then we can think of the full block of $n + n'$ qubits that go through the channel as a single codeword; by putting these two codes together we have constructed an $[[n + n', k]]$ standard QECC.

Of course, the second code could also be an EAQECC. If we have an $[[n, k; c]]$ EAQECC and an $[[n', k'; c']]$ EAQECC, we can send Bob's halves of the c ebits in the second code block, so long as

$k' \geq c$. By putting these two code blocks together, we have constructed an $[[n + n', k + k' - c; c']]$ EAQECC.

Nor need we combine only two codes in this way. One rather simple construction is as follows: take an $[[n, k; c]]$ EAQECC with positive net rate, and encode Bob's halves of the c ebits in another copy of the same code. Repeat this M times, and one quickly builds up a much larger block code: an $[[Mn, M(k - c) + c; c]]$ EAQECC. This procedure is called *bootstrapping*. Two observations of this new code are quickly evident: first, as M becomes large, the rate of the code approaches the net rate $(k - c)/n$ of the original code. Second, as M becomes large, the rate of entanglement usage c/Mn becomes small. By bootstrapping we have built a large code whose rate is the net rate of the original code, and which uses relatively little entanglement.

Practically speaking there is a limit to how big M can be; if the block size becomes too large, the probability that one of the sub-blocks of n qubits will have an uncorrectable error becomes high, and the code will no longer be useful. At this point, little is known about the practical performance of catalytic codes—more research is needed.

By this approach we see another use of EAQECCs: rather than being used themselves directly, they can be used as building blocks for the construction of standard codes. Because EAQECCs can be constructed from classical codes without the need to satisfy the dual-containing constraint, they in some cases will outperform standard codes with otherwise similar parameters. Combining EAQECCs with each other, or with standard codes, may allow the relatively easy construction of standard codes with good error-correcting properties. Moreover, this may lead to applications of EAQECCs beyond the quantum communication paradigm, using them to construct standard codes that might be useful in other applications (such as quantum computing). However, at present this is pure speculation: such applications have yet to be studied.

5   Conclusions

EAQECCs expand the usual paradigm of quantum error correction by allowing the sender and receiver to make use of pre-shared entanglement. This entanglement can increase either the rate of communication or the number of correctable errors. While entanglement can be used to improve quantum communication in other ways—for example, by sending classical information through the channel and teleporting extra qubits—in many cases EAQECCs have better performance [HW10a, HW10b].

In this chapter we have presented an extension of the usual stabilizer formalism for QECCs to include entanglement-assistance. This extension includes the usual theory of stabilizers as a special case, and has a number of advantages; in particular, it allows one to construct quantum codes from classical linear codes without having to impose the dual-containing constraint of standard codes, while retaining the usual relation between the rate and distance of the classical and quantum codes. Moreover, if entanglement-assistance is allowed, one can in some cases outdo the performance of the best standard QECC by optimizing without the dual-containing constraint.

In quantum communication protocols involving entanglement, the applications of EAQECCs are obvious. However, the construction may also be useful in other cases, by using EAQECCs as *catalytic* codes. In this case, we are essentially using EAQECCs as building blocks for standard codes, or for codes that use only a small amount of pre-existing entanglement. In these constructions the net rate of the EAQECCs is the most important parameter; this net rate can be directly calculated from the classical linear code, for codes constructed from classical precursors.

Because these constructions do not require dual-containing codes, it is possible to directly derive quantum versions of highly-efficient modern codes, such as Turbo codes and LDPC codes

[HYH11, WHB14]. Standard quantum versions of these codes have a number of difficulties, since the dual-containing constraint generally implies that such codes have problems in using iterative decoding algorithms. Some examples have been studied, comparing specific EAQECCs to standard QECCs, where the EAQECCs show very good decoding behavior [HBD09, HYH11]. Much more work remains to be done here, however.

This chapter has considered only block codes. Some work has already been done on entanglement-assisted quantum convolutional codes as well [WB10a, WB10b]—standard convolutional QECCs are described in [LB13, Chapter 9]. Many questions remain open; in particular, the minimum entanglement needed to produce a quantum version of a classical convolutional code is unknown, though some answers have been conjectured [WB08]. The field of entanglement-assisted codes is relatively new, and active research is ongoing to find more applications of these codes.

References

[BBP+96]   Charles H. Bennett, Giles Brassard, Sandu Popescu, Benjamin Schumacher, John A. Smolin, and William K. Wootters. Purification of noisy entanglement and faithful teleportation via noisy channels. *Phys. Rev. Lett.*, 76:722–725, 1996.

[BDH06]   Todd A. Brun, Igor Devetak, and Min-Hsiu Hsieh. Correcting quantum errors with entanglement. *Science*, 314:436–439, 2006.

[BDH14]   Todd A. Brun, Igor Devetak, and Min-Hsiu Hsieh. Catalytic quantum error correction. *IEEE Transactions on Information Theory*, 60(6):3073–3089, 2014.

[BDSW96]   Charles H. Bennett, David P. DiVincenzo, John A. Smolin, and William K. Wootters. Mixed state entanglement and quantum error correction. *Phys. Rev. A*, 54:3824–3851, 1996.

[BFG06]   Sergey Bravyi, David Fattal, and Daniel Gottesman. GHZ extraction yield for multipartite stabilizer states. *J. Math. Phys.*, 47:062106, 2006.

[Bow02]   Garry Bowen. Entanglement required in achieving entanglement-assisted channel capacities. *Phys. Rev. A*, 66:052313, 2002.

[CG97]   Richard Cleve and Daniel Gottesman. Efficient computations of encodings for quantum error correction. *Phys. Rev. A*, 56:76–82, 1997.

[CRSS98]   A. Robert Calderbank, Eric M. Rains, Peter W. Shor, and N. J. A. Sloane. Quantum error correction via codes over GF(4). *IEEE Transactions on Information Theory*, 44:1369–1387, 1998.

[DHW04]   Igor Devetak, Aram W. Harrow, and Andreas Winter. A family of quantum protocols. *Phys. Rev. Lett.*, 93:230504, 2004.

[DHW08]   Igor Devetak, Aram W. Harrow, and Andreas Winter. A resource framework for quantum shannon theory. *IEEE Trans. Inf. Theory*, 54:4587–4618, 2008.

[FCY+04]   David Fattal, Toby S. Cubitt, Yoshihisa Yamamoto, Sergey Bravyi, and Isaac L. Chuang. Entanglement in the stabilizer formalism, 2004. eprint quant-ph/0406168.

[GRB03] Markus Grassl, Martin Rötteler, and Thomas Beth. Efficient quantum circuits for non-qubit quantum error-correcting codes. *International Journal of Foundations of Computer Science (IJFCS)*, 14:757–775, 2003.

[HBD09] Min-Hsiu Hsieh, Todd A. Brun, and Igor Devetak. Quantum quasi-cyclic low-density parity-check codes. *Phys. Rev. A*, 79:032340, 2009.

[HDB07] Min-Hsiu Hsieh, Igor Devetak, and Todd Brun. General entanglement-assisted quantum error-correcting codes. *Phys. Rev. A*, 76:062313, Dec 2007.

[HW10a] Min-Hsiu Hsieh and Mark M. Wilde. Entanglement-assisted communication of classical and quantum information. *IEEE Transactions on Information Theory*, 56(9):4682–4704, 2010.

[HW10b] Min-Hsiu Hsieh and Mark M. Wilde. Trading classical communication, quantum communication, and entanglement in quantum shannon theory. *IEEE Transactions on Information Theory*, 56(9):4705–4730, 2010.

[HYH11] Min-Hsiu Hsieh, Wen-Tai Yen, and Li-Yi Hsu. High performance entanglement-assisted quantum ldpc codes need little entanglement. *IEEE Transactions on Information Theory*, 57(3):1761–1769, 2011.

[LB13] Daniel A. Lidar and Todd A. Brun. *Quantum Error Correction*. Cambridge University Press, New York, 2013.

[MS77] F. J. MacWilliams and N. J. A. Sloane. *The Theory of Error-Correcting Codes*. Elsevier, Amsterdam, 1977.

[NC00] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, New York, 2000.

[WB08] Mark M. Wilde and Todd A. Brun. Optimal entanglement formulae for entanglement-assisted quantum coding. *Phys. Rev. A*, 77:064302, 2008.

[WB10a] Mark M. Wilde and Todd A. Brun. Entanglement-assisted quantum convolutional coding. *Phys. Rev. A*, 81:042333, Apr 2010.

[WB10b] Mark M. Wilde and Todd A. Brun. Quantum convolutional coding with shared entanglement: general structure. *Quantum Information Processing*, 9(5):509–540, 2010.

[WHB14] Mark M. Wilde, Min-Hsiu Hsieh, and Zunaira Babar. Entanglement-assisted quantum turbo codes. *IEEE Transactions on Information Theory*, 60(2):1203–1222, 2014.

[Wil09] Mark M. Wilde. Logical operators of quantum codes. *Phys. Rev. A*, 79:062322, 2009.