Entanglement-assisted quantum convolutional coding

Mark M. Wilde* and Todd A. Brun†
Communication Sciences Institute, Department of Electrical Engineering, University of Southern California, Los Angeles, California 90089, USA
(Received 10 February 2010; published 30 April 2010)

We show how to protect a stream of quantum information from decoherence induced by a noisy quantum communication channel. We exploit preshared entanglement and a convolutional coding structure to develop a theory of entanglement-assisted quantum convolutional coding. Our construction produces a Calderbank-Shor-Steane (CSS) entanglement-assisted quantum convolutional code from two arbitrary classical binary convolutional codes. The rate and error-correcting properties of the classical convolutional codes directly determine the corresponding properties of the resulting entanglement-assisted quantum convolutional code. We explain how to encode our CSS entanglement-assisted quantum convolutional codes starting from a stream of information qubits, ancilla qubits, and shared entangled bits.

DOI: 10.1103/PhysRevA.81.042333 PACS number(s): 03.67.Pp, 03.67.Hk

I. INTRODUCTION

Quantum error correction theory [1–6] stands as the pivotal theoretical tool that will make reliable quantum computing and quantum communication possible. Any future quantum information processing device will operate faithfully only if it employs an error correction scheme. This scheme can be an active scheme [4], a passive scheme [7–9], or a combination of both techniques [10–14].

Mermin proclaims it a "miracle" that quantum error correction is even possible [15]. Various obstacles such as the no-cloning theorem [16], measurement destroying a quantum state, and continuous quantum errors seem to pose an insurmountable barrier to a protocol for quantum error correction. Despite these obstacles, Shor demonstrated the first quantum error-correcting code that reduces the negative effects of decoherence on a quantum bit [1]. Shor's code overcame all of the above difficulties and established the basic principles for constructing a general theory of quantum error correction [4–6].

Gottesman formalized the theory of quantum block coding by establishing the stabilizer formalism [4]. The stabilizer formalism allows one to import self-orthogonal classical block codes for use in quantum error correction [6]. This technique has the benefit of exploiting the large body of research on classical coding theory [17] for use in quantum error correction, but the self-orthogonality constraint limits the classical block codes that we can import.

Bowen was the first to extend the stabilizer formalism by providing an example of a code that exploits entanglement shared between a sender and a receiver [18]. The underlying assumption of Bowen's code is that the sender and receiver share a set of noiseless ebits (entangled qubits) before quantum communication begins. Many quantum protocols such as teleportation [19] and superdense coding [20] are "entanglement-assisted" protocols because they assume that noiseless ebits are available.

Brun, Devetak, and Hsieh generalized Bowen's example by constructing a theory of stabilizer codes that employs ancilla qubits and shared ebits for encoding a quantum error-correcting code [21,22]. The so-called entanglement-assisted stabilizer formalism subsumes the stabilizer formalism as the theory of active quantum error correction.

The major benefit of the entanglement-assisted stabilizer formalism is that we can construct an entanglement-assisted quantum code from two arbitrary classical binary block codes or from an arbitrary classical quaternary block code. The rates and error-correcting properties of the classical codes translate to the resulting quantum codes. The entanglement-assisted stabilizer formalism may be able to reduce the problem of finding high-performance quantum codes approaching the quantum capacity [23–27] to the problem of finding good classical linear codes approaching the classical capacity [28].

Another extension of the theory of quantum error correction protects a potentially infinite stream of quantum information against the corruption induced by a noisy quantum communication channel [29–35]. These quantum convolutional codes possess several advantages over quantum block codes. A quantum convolutional code typically has lower encoding and decoding complexity and superior code rate when compared to a block code that protects the same number of information qubits [35].

Forney *et al.* have determined a method for importing an arbitrary classical self-orthogonal quaternary code for use as a quantum convolutional code [34,35]. The technique is similar to that for importing a classical block code as a quantum block code [6]. One limitation of this technique is that the self-orthogonality constraint is more restrictive in the convolutional setting. Each generator for the quantum convolutional code must commute not only with the other generators, but it must commute also with any arbitrary shift of itself and any arbitrary shift of the other generators. Forney *et al.* performed specialized searches to determine classical quaternary codes that satisfy the restrictive self-orthogonality constraint [35].

In this paper, we develop a theory of entanglement-assisted quantum convolutional coding for a broad class of codes. Our major result is that we can produce an

*mwilde@gmail.com
†tbrun@usc.edu

entanglement-assisted quantum convolutional code from two *arbitrary* classical binary convolutional codes. The resulting quantum convolutional codes admit a Calderbank-Shor-Steane (CSS) structure [2,3,36]. The rates and error-correcting properties of the two binary classical convolutional codes directly determine the corresponding properties of the entanglement-assisted quantum convolutional code.

Our techniques for encoding and decoding are also an expansion of previous techniques from quantum convolutional coding theory. Previous techniques for encoding and decoding include finite-depth operations only. A finite-depth operation propagates errors to a finite number of neighboring qubits in the qubit stream. We introduce an infinite-depth operation to the set of shift-invariant Clifford operations and explain it in detail in Sec. VI. We must be delicate when using infinite-depth operations because they can propagate errors to an infinite number of neighboring qubits in the qubit stream. We explain our assumptions in detail in Sec. VII for including infinite-depth operations in our entanglement-assisted quantum convolutional codes. An infinite-depth operation gives more flexibility when designing encoding circuits—similar to the way in which an infinite-impulse response filter gives more flexibility in the design of classical convolutional circuits. It also is the key operation enabling us to import arbitrary classical convolutional codes for entanglement-assisted quantum coding.

Our CSS entanglement-assisted quantum convolutional codes divide into two classes based on certain properties of the classical codes from which we produce them. These properties of the classical codes determine the structure of the encoding and decoding circuit for the code, and the structure of the encoding and decoding circuit in turn determines the class of the entanglement-assisted quantum convolutional code.

1. Codes in the first class admit both a finite-depth encoding and decoding circuit.

2. Codes in the second class have an encoding circuit that employs both finite-depth and infinite-depth operations. Their decoding circuits have finite-depth operations only.

We structure our work as follows. Sec. II reviews the stabilizer formalism for quantum block codes, entanglement-assisted quantum codes, and convolutional stabilizer codes. We review the important isomorphism that allows us to work with matrices of binary polynomials rather than infinite tensor products of Pauli matrices. Sec. III reviews finite-depth Clifford operations for use in encoding and decoding [31–33]. We outline the operation of an entanglement-assisted quantum convolutional code and present our main theorem in Sec. IV. This theorem shows how to produce a CSS entanglement-assisted quantum convolutional code from two arbitrary classical binary convolutional codes. The theorem gives the rate and error-correcting properties of a CSS entanglement-assisted quantum convolutional code as a function of the parameters of the classical convolutional codes. Sec. V completes the proof of the theorem for our first class of entanglement-assisted quantum convolutional codes. In Sec. VI, we introduce an infinite-depth encoding operation to the set of shift-invariant Clifford operations and discuss its effect on both the stabilizer and the logical operators for the information qubits. Sec. VII completes the proof of our theorem for the second class of entanglement-assisted quantum convolutional codes. We discuss the implications of the assumptions for the different classes of entanglement-assisted quantum convolutional codes while developing the constructions. Our hope is that our theory will produce high-performance quantum convolutional codes by importing high-performance classical convolutional codes.

II. REVIEW OF THE STABILIZER FORMALISM

The stabilizer formalism is a mathematical framework for quantum error correction [4,37]. This framework has many similarities with classical coding theory, and it is even possible to import a classical code for use in quantum error correction by employing the CSS construction [2,3,36]. We briefly review the stabilizer theory for quantum block codes, entanglement-assisted quantum block codes, and quantum convolutional codes (see Refs. [35,38] for a more detailed review).

A. Stabilizer formalism for quantum block codes

The following four matrices,

$$I \equiv \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad X \equiv \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix},$$
$$Y \equiv \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad Z \equiv \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix},$$

in the Pauli group $\Pi = \{I, X, Y, Z\}$ are the most important in formulating a quantum error-correcting code. Two crucial properties of these matrices are useful: Each matrix in Π has eigenvalues equal to $+1$ or -1, and any two matrices in Π either commute or anticommute. Matrices in Π act on a two-dimensional complex vector, or equivalently, a single qubit.

In general, a quantum error-correcting code uses n physical qubits to protect a smaller set of information qubits against decoherence or quantum noise. An n-qubit quantum error-correcting code employs elements of the Pauli group Π^n. The Pauli group Π^n consists of n-fold tensor products of Pauli matrices:

$$\Pi^n = \left\{ \begin{array}{c} e^{i\phi} A_1 \otimes \cdots \otimes A_n : \forall j \in \{1, \ldots, n\}, \\ A_j \in \Pi, \quad \phi \in \{0, \pi/2, \pi, 3\pi/2\} \end{array} \right\}. \tag{1}$$

We liberally omit the tensor product symbol in what follows so that $A_1 \cdots A_n \equiv A_1 \otimes \cdots \otimes A_n$. The above two crucial properties for the single-qubit Pauli group Π still hold for the Pauli group Π^n (up to an irrelevant phase for the eigenvalue property). Matrices in Π^n act on a 2^n-dimensional complex vector, or equivalently, an n-qubit quantum register.

We can phrase the theory of quantum error correction in purely mathematical terms using elements of Π^n. Consider a matrix $g_1 \in \Pi^n$ that is not equal to $\pm I$. Matrix g_1 then has two eigenspaces each of size 2^{n-1}. We can identify one eigenspace with the eigenvalue $+1$ and the other eigenspace with eigenvalue -1. Consider a matrix $g_2 \in \Pi^n$ different from g_1 that commutes with g_1. Matrix g_2 also has two eigenspaces each of size 2^{n-1} and identified similarly by its eigenvalues ± 1. Both g_1 and g_2 have simultaneous eigenspaces because they commute. These matrices together have four different

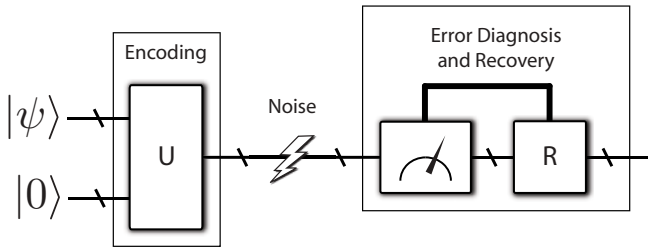

FIG. 1. The operation of a general stabilizer code. Thin lines denote quantum information and thick lines denote classical information. Slanted bars denote multiple qubits. A sender employs a unitary encoding operation U to encode a set of information qubits in the state $|\psi\rangle$ with the help of ancilla qubits each in the state $|0\rangle$. The sender transmits the encoded qubits over the noisy quantum communication channel. The receiver performs quantum measurements to diagnose which error has occurred. He finally performs a recovery operation R to reverse the error from the channel.

eigenspaces, each of size 2^{n-2} and identified by the eigenvalues $\pm 1, \pm 1$ of g_1 and g_2, respectively. We can continue this process of adding more commuting and independent matrices to a set $\mathcal{S}$. The matrices in $\mathcal{S}$ are independent in the sense that no matrix in $\mathcal{S}$ is a product of two or more other matrices in $\mathcal{S}$. Adding more matrices from Π^n to $\mathcal{S}$ continues to divide the eigenspaces of matrices in $\mathcal{S}$. In general, suppose $\mathcal{S}$ consists of $n-k$ independent and commuting matrices $g_1, \ldots, g_{n-k} \in \Pi^n$. These $n-k$ matrices then have 2^{n-k} different eigenspaces each of size 2^k and identified by the eigenvalues $\pm 1, \ldots, \pm 1$ of $g_1, \ldots, g_{n-k}$, respectively. Consider that the Hilbert space of k qubits has size 2^k. A dimension count immediately suggests that we can encode k qubits into one of the eigenspaces of $\mathcal{S}$. We typically encode these k qubits into the simultaneous $+1$ eigenspace of $g_1, \ldots, g_{n-k}$. This eigenspace is the *codespace*. An $[n,k]$ quantum error-correcting code encodes k information qubits into the simultaneous $+1$ eigenspace of $n-k$ matrices $g_1, \ldots, g_{n-k} \in \Pi^n$. The rate of an $[n,k]$ code is the ratio of information qubits to physical qubits: k/n.

The operation of an $[n,k]$ quantum error-correcting code consists of four steps. Figure 1 highlights these steps. First, a unitary operation U encodes k qubits and $n-k$ ancilla qubits into the simultaneous $+1$ eigenspace of the matrices $g_1, \ldots, g_{n-k}$. The sender transmits the n encoded qubits by using the noisy quantum communication channel n times. The receiver performs quantum measurements of the $n-k$ matrices $g_1, \ldots, g_{n-k}$. These measurements learn only about errors that may occur and do not disturb the encoded quantum information. Each measurement gives a bit result equal to $+1$ or -1, and the result of all the measurements is to project the n-qubit quantum register into one of the 2^{n-k} different eigenspaces of $g_1, \ldots, g_{n-k}$. Suppose that no error occurs. Then the measurements project the n qubits into the simultaneous $+1$ eigenspace and return a bit vector consisting of $n-k$ ones. Now suppose that a quantum error in an error set $\mathcal{E}$ occurs. The error takes the encoded quantum state out of the codespace and into one of the other $2^{n-k}-1$ orthogonal eigenspaces. The measurements can detect that an error has occurred because the result of the measurements is a bit vector differing from the all ones vector. The receiver may be able to identify uniquely which error has occurred if it satisfies the following quantum error correction conditions:

$$\forall E_a, E_b \in \mathcal{E} \;\; \exists\, g_i \in \mathcal{S} : \{g_i, E_a^\dagger E_b\} = 0 \;\text{ or }\; E_a^\dagger E_b \in \mathcal{S}.$$

The first condition states that errors are detectable if they anticommute with one of the generators in $\mathcal{S}$, and the second condition states that errors have no effect on the encoded state if they are in $\mathcal{S}$. If the receiver can identify which error occurs, he can then apply unitary operation R that is the inverse of the error. He finally performs a decoding unitary that decodes the k information qubits.

We comment briefly on the encoding operation U. The encoding operation U is a special type of unitary matrix called a Clifford operation. A Clifford operation U is one that preserves elements of the Pauli group under conjugation: $A \in \Pi^n \Rightarrow UAU^\dagger \in \Pi^n$. The controlled-NOT (C-NOT) gate, the Hadamard gate H, and the phase gate P suffice to implement any unitary matrix in the Clifford group [4]. A quantum code with the CSS structure needs only the C-NOT and Hadamard gates for encoding and decoding. The matrix for the C-NOT gate acting on two qubits is

$$\text{C-NOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \tag{2}$$

the matrix for the Hadamard gate H acting on a single qubit is

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \tag{3}$$

and the matrix for the phase gate P acting on a single qubit is

$$P = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}. \tag{4}$$

For the C-NOT gate, the first qubit is the "control" qubit and the second qubit is the "target" qubit. The standard basis for elements of the two-qubit Pauli group Π^2 is as follows:

$$\begin{matrix} Z & I \\ I & Z \\ X & I \\ I & X \end{matrix}, \tag{5}$$

because any element of Π^2 is a product of the above four matrices up to an irrelevant phase. The standard basis for Π^1 is X and Z for the same reasons. The C-NOT gate transforms the standard basis of Π^2 under conjugation as follows:

$$\begin{matrix} Z & I \\ I & Z \\ X & I \\ I & X \end{matrix} \rightarrow \begin{matrix} Z & I \\ Z & Z \\ X & X \\ I & X \end{matrix}, \tag{6}$$

where the first qubit is the control and the second qubit is the target. The Hadamard gate H transforms the standard basis of Π^1 under conjugation as follows:

$$\begin{matrix} Z \\ X \end{matrix} \rightarrow \begin{matrix} X \\ Z \end{matrix}, \tag{7}$$

and the phase gate P transforms the standard basis as follows:

$$\begin{matrix} Z \\ X \end{matrix} \rightarrow \begin{matrix} Z \\ Y \end{matrix}. \tag{8}$$

The appendix of Ref. [38] details an algorithm that determines an encoding circuit consisting of C-NOT, H, and P gates for any stabilizer code or any entanglement-assisted stabilizer code (we review entanglement-assisted codes in the next section).

Another aspect of the theory of quantum error correction is later useful for our purposes in quantum convolutional coding. This aspect concerns the information qubits and the operators that change them. Consider that the initial unencoded state of a quantum error-correcting code is a simultaneous $+1$ eigenstate of the matrices $Z_{k+1},\ldots,Z_n$, where Z_i has a Z matrix operating on qubit i and the identity I on all other qubits. Therefore, the matrices $Z_{k+1},\ldots,Z_n$ constitute a stabilizer for the unencoded state. The initial unencoded logical operators for the information qubits are $Z_1,X_1,\ldots,Z_k,X_k$. The encoding operation U rotates the unencoded stabilizer matrices $Z_{k+1},\ldots,Z_n$ and the unencoded logical operators $Z_1,X_1,\ldots,Z_k,X_k$ to the encoded stabilizer $\bar{Z}_{k+1},\ldots,\bar{Z}_n$ and the encoded logical operators $\bar{Z}_1,\bar{X}_1,\ldots,\bar{Z}_k,\bar{X}_k$, respectively. The encoded matrices $\bar{Z}_{k+1},\ldots,\bar{Z}_n$ are, respectively, equivalent to the matrices $g_1,\ldots,g_{n-k}$ in the above discussion. The encoded operators obey the same commutation relations as their unencoded counterparts. We would violate the uncertainty principle if this invariance does not hold. Therefore, each of the encoded logical operators commutes with elements of the stabilizer $\mathcal{S}$. Let A denote an arbitrary logical operator from the above set and let $\bar{Z}_i$ denote an arbitrary element of the stabilizer $\mathcal{S}$. The operator $A\bar{Z}_i$ (or equivalently $\bar{Z}_iA$) is an equivalent logical operator because $A\bar{Z}_i$ and A have the same effect on an encoded state $|\bar{\psi}\rangle$:

$$\bar{Z}_iA|\bar{\psi}\rangle = A\bar{Z}_i|\bar{\psi}\rangle = A|\bar{\psi}\rangle. \tag{9}$$

We make extensive use of the above fact in our work.

The logical operators also provide a useful way to characterize the information qubits. Gottesman showed that the logical operators for the information qubits provide a straightforward way to characterize the information qubits as they progress through a quantum circuit [4]. As an example of this technique, he develops quantum teleportation in the stabilizer formalism. The logical operators at the beginning of the protocol are X_1 and Z_1 and become X_3 and Z_3 at the end of the protocol. The quantum information in qubit one teleports to qubit three because the logical operators act on only qubit three at the end of the protocol. We use the same idea throughout this paper to determine if our decoding circuits have truly decoded the information qubits.

It is possible to produce a stabilizer code from two classical binary block codes by employing the CSS construction. The elements of the stabilizer group of a CSS stabilizer code commute if and only if the codewords of one classical code are orthogonal to the codewords of the other classical code with respect to the binary inner product. The codes that we can import must satisfy this condition because the commuting condition is essential in formulating a quantum code. The entanglement-assisted stabilizer formalism finds a clever way around this restriction by exploiting entanglement shared between sender and receiver.

B. Entanglement-assisted stabilizer formalism for quantum block codes

The entanglement-assisted stabilizer formalism is a significant extension of the standard stabilizer formalism that incorporates shared entanglement as a resource for encoding [21,22]. Several references provide a review of this technique and generalizations of the basic theory to block [39] and convolutional [38] entanglement distillation protocols, continuous-variable codes [40], and entanglement-assisted operator codes for discrete-variable [13,14] and continuous-variable systems [41].

An entanglement-assisted code employs ebits or Bell states in addition to ancilla qubits for quantum redundancy. We express the state $|\Phi^+\rangle$ of an ebit shared between a sender Alice and a receiver Bob as follows:

$$|\Phi^+\rangle \equiv \frac{|00\rangle^{AB} + |11\rangle^{AB}}{\sqrt{2}}. \tag{10}$$

The advantage of the entanglement-assisted stabilizer formalism is that it allows us to exploit the error-correcting properties of an arbitrary set of Pauli matrices. They do not necessarily have to form a commuting set. In particular, this construction allows us to produce a quantum block code from two arbitrary classical binary block codes by employing the CSS construction. Two high-performance classical block codes lead to a high-performance entanglement-assisted quantum code. The entanglement-assisted method allows us to exploit the full error-correcting power of classical coding theory.

An $[n,k;c]$ entanglement-assisted code uses c ebits and $n-k-c$ ancilla qubits to encode k information qubits. It operates as follows. The sender and receiver share c ebits before quantum communication begins. The sender encodes her k information qubits with the help of $n-k-c$ ancilla qubits and her half of the c ebits. She performs an encoding operation U on her n qubits and sends them over a noisy quantum communication channel. The noisy channel affects these n qubits only and does not affect the receiver's half of the c ebits. The receiver combines his half of the c ebits with those he receives from the noisy quantum channel. He performs measurements on all $n+c$ qubits to diagnose an error that may occur on the n qubits. He learns which error occurs and performs a recovery operation that eliminates the error. Figure 2 illustrates the operation of an entanglement-assisted stabilizer code.

Suppose we have an arbitrary set of Pauli matrices in Π^n whose error-correcting properties we would like to exploit. We do not necessarily know beforehand how many ebits we require for the Pauli matrices to form a commuting set, and we would like a method to determine the minimum number of ebits. Several methods exist [13,14,21,22,38], but the algorithm in the appendix of Ref. [38] determines the minimum number of ebits required for the code, the encoding and decoding circuit for the code, and the measurements the receiver performs to diagnose errors. It essentially "kills three birds with one stone." The algorithms we employ in this work are similar to the

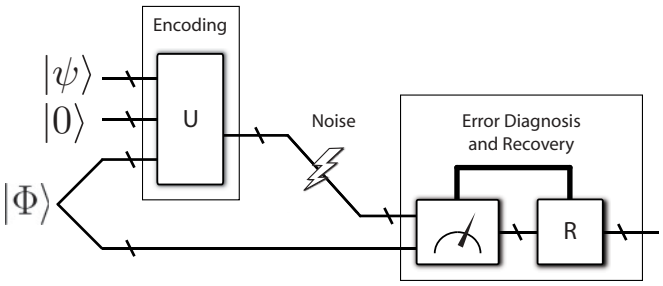

FIG. 2. The operation of a general entanglement-assisted stabilizer code. The sender encodes a set of information qubits with the help of ancilla qubits and her half of a set of shared ebits. She sends her encoded qubits over a noisy quantum communication channel. The entanglement-assisted communication paradigm assumes that the receiver's half of the shared ebits remain noiseless throughout this process. The receiver combines the noisy qubits with his half of the shared ebits. He performs measurements on all of the qubits to diagnose which error occurs and reverses the effect of this error by performing a recovery operation.

algorithm in Ref. [38], but they are quite a bit more complicated because of the convolutional nature of our codes.

1. Rate of an entanglement-assisted quantum code

We can interpret the rate of an entanglement-assisted quantum convolutional code in three different ways [21,22,38]. Suppose that an entanglement-assisted quantum code encodes k qubits in n qubits with the help of c ebits.

1. The "entanglement-assisted" rate assumes that entanglement shared between sender and receiver is free. Bennett *et al.* make this assumption when deriving the entanglement-assisted capacity of a quantum channel for sending quantum information [26,27]. The entanglement-assisted rate for the above example is k/n.

2. The "trade-off" rate assumes that entanglement is not free and a rate pair determines performance. The first number in the pair is the number of noiseless qubits generated per channel use, and the second number in the pair is the number of ebits consumed per channel use. The rate pair for the above example is $(k/n,c/n)$. Quantum information theorists have computed asymptotic trade-off curves that bound the rate region in which achievable rate pairs lie [42]. Brun *et al.*'s construction for an entanglement-assisted quantum block code minimizes the number c of ebits given a fixed number k and n of information qubits and encoded qubits, respectively [21,22].

3. The "catalytic rate" assumes that bits of entanglement are built up at the expense of transmitted qubits [21,22]. A noiseless quantum channel or the encoded use of noisy quantum channel are two different ways to build up entanglement between a sender and receiver. The catalytic rate for the above code is $(k-c)/n$.

Which interpretation is most reasonable depends on the context in which we use the code. In any case, the parameters n, k, and c ultimately govern performance, regardless of which definition of the rate we use to interpret that performance.

C. Stabilizer formalism for quantum convolutional codes

We review the theory of convolutional stabilizer codes by considering a set of Pauli matrices that stabilize a stream of encoded qubits. We follow with the most important part of this review—the isomorphism from the set of Pauli sequences to the module over the ring of binary polynomials [30,31,35]. We name it the Pauli-to-binary (P2B) isomorphism. The P2B isomorphism is important because it is easier to perform manipulations with vectors of binary polynomials than with Pauli sequences.

We review the notation and basic definitions first. A Pauli sequence $\mathbf{A}$ is a countably infinite tensor product of Pauli matrices A_i:

$$\mathbf{A} = \bigotimes_{i=0}^{\infty} A_i.$$

The weight of a Pauli sequence is the number of Pauli matrices in the countably-infinite tensor product that are not equal to the identity matrix. A Pauli sequence has finite support if its weight is finite. Let $\Pi^{\mathbb{Z}^+}$ denote the set of all Pauli sequences and let $F(\Pi^{\mathbb{Z}^+})$ denote the set of Pauli sequences with finite support.

Definition 1. A rate-k/n quantum convolutional code consists of a basic set $\mathcal{G}_0$ of $n-k$ generators and all of their n-qubit shifts [29,30,35]. The generators in $\mathcal{G}_0$ commute with each other and with all of their n-qubit shifts. The parameters k and n satisfy $0 \leqslant k \leqslant n$ and the basic set $\mathcal{G}_0$ is as follows:

$$\mathcal{G}_0 = \{\mathbf{G}_i \in F(\Pi^{\mathbb{Z}^+}) : 1 \leqslant i \leqslant n-k\}.$$

A frame of the code consists of n qubits.

The operation of a rate-k/n quantum convolutional code begins with the sender encoding a stream of information qubits. Figure 3 of Ref. [38] illustrates the basic operation of a quantum convolutional code. The sender encodes $n-k$ ancilla qubits and k information qubits per frame [31,33] and transmits the encoded qubits over a noisy quantum channel. The above stabilizer $\mathcal{G}_0$ and all of its shifts act like a parity check matrix for the quantum convolutional code. The receiver measures the generators in the stabilizer to determine an error syndrome. It is important that the generators in $\mathcal{G}_0$ have finite weight so that the receiver can perform the measurements and produce an error syndrome. It is also important that the generators have a block-band form so that the receiver can perform the measurements online as the noisy encoded qubits arrive. The receiver processes the error syndrome with a method such as the Viterbi algorithm [43] or any other decoding algorithm [44] to determine the most likely error for each frame of quantum data. The receiver performs a unitary that reverses the errors. He finally processes the encoded qubits with a decoding circuit to recover the original stream of information qubits.

1. The P2B isomorphism

We now review the P2B isomorphism from the set of phase-free Pauli sequences to the module over the ring of binary polynomials [30,35,38]. We illustrate it by example (see Ref. [38] for a more rigorous development).

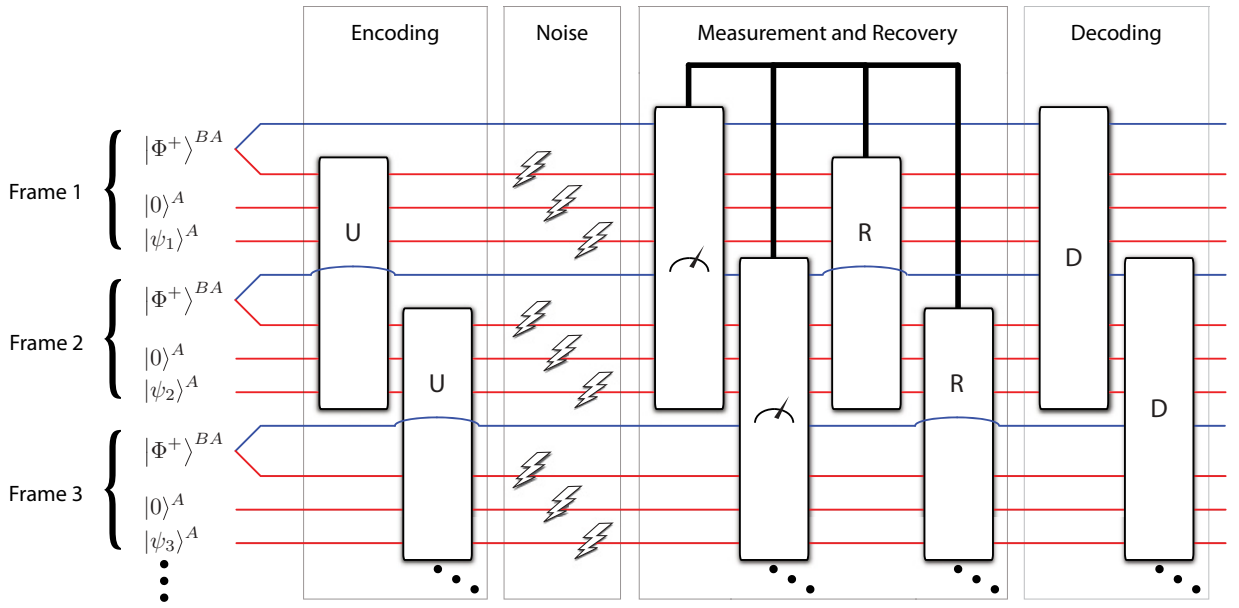

FIG. 3. (Color online) An entanglement-assisted quantum convolutional code operates on a stream of qubits partitioned into a countable number of frames. The sender encodes the frames of information qubits, ancilla qubits, and half of shared ebits with a repeated, overlapping encoding circuit U. The noisy channel affects the sender's encoded qubits but does not affect the receiver's half of the shared ebits. The receiver performs overlapping measurements on both the encoded qubits and his half of the shared ebits. These measurements produce an error syndrome which the receiver can process to determine the most likely error. The receiver reverses the errors on the noisy qubits from the sender. The final decoding circuit operates on all qubits in a frame and recovers the original stream of information qubits.

Suppose the following two basic generators specify a rate-1/3 quantum convolutional code [34,35],

$$\cdots \left| \begin{matrix} III \\ III \end{matrix} \right| \begin{matrix} XXX \\ ZZZ \end{matrix} \left| \begin{matrix} XZY \\ ZYX \end{matrix} \right| \begin{matrix} III \\ III \end{matrix} \left| \cdots \right. \tag{11}$$

The vertical bars indicate that we shift by multiples of three to obtain the other generators in the quantum convolutional code. Observe that the above two generators commute with all of their three-qubit shifts.

The P2B isomorphism is a mapping from the above stabilizer generators to a matrix whose entries are binary polynomials. The left side of the matrix is the "Z" matrix and the right side of the matrix is the "X" matrix. We consider the entries in the first frame of the stabilizer generators in (11) for now and map these entries to a matrix with binary entries. The first frame of the first generator in (11) has "X" entries only and the first frame of the second generator in (11) has "Z" entries only. The binary matrix corresponding to the entries in the first frame is as follows:

$$H_0 = \left[\begin{array}{ccc|ccc} 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 \end{array} \right].$$

The vertical bar now indicates the separation of the "Z" matrix on the left and the "X" matrix on the right. A "Y" entry maps to a "1" in both the "Z" and "X" matrix. Let us consider the entries in the second frame of (11). They map to the following binary matrix:

$$H_1 = \left[\begin{array}{ccc|ccc} 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 \end{array} \right].$$

We form a matrix of binary polynomials by incorporating the delay transform or D transform. The following binary polynomial matrix $H(D)$ fully specifies the quantum convolutional code:

$$\begin{aligned} H(D) &= H_0 + H_1 \cdot D \\ &= \left[\begin{array}{ccc|ccc} 0 & D & D & 1+D & 1 & 1+D \\ 1+D & 1+D & 1 & 0 & D & D \end{array} \right]. \end{aligned}$$

The above description of a quantum convolutional code with a binary polynomial matrix is powerful because it allows us to perform manipulations with finite polynomials rather than with countably infinite sequences of Pauli matrices (classical convolutional coding theory exploits the same idea [44]). The first and second rows of $H(D)$ capture all of the information about the first and second generators in (11) and all of their three-qubit shifts. We obtain the nl shift of either of the above generators by multiplying the corresponding row in $H(D)$ by D^l.

2. *The shifted symplectic product*

The shifted symplectic product $\odot$ provides a way to determine the commutative properties of any convolutional stabilizer code [30,38] (see Ref. [38] for a detailed discussion of the shifted symplectic product with examples). Let $z_1(D)$ and $z_2(D)$ denote the first and second respective rows of the "Z" matrix of $H(D)$. Let $x_1(D)$ and $x_2(D)$ be the first and second respective rows of the "X" matrix of $H(D)$. Let

$$\begin{aligned} h_1(D) &= (z_1(D)|x_1(D)), \\ h_2(D) &= (z_2(D)|x_2(D)), \end{aligned}$$

denote the first and second respective rows of $H(D)$. The vectors $h_1(D)$ and $h_2(D)$ specify the first and second respective

generators in (11). We define the shifted symplectic product of $h_1(D)$ and $h_2(D)$ as follows:

$$(h_1 \odot h_2)(D) = z_1(D^{-1}) \cdot x_2(D) + x_1(D^{-1}) \cdot z_2(D),$$

where $\cdot$ denotes the binary inner product and addition is binary.

The shifted symplectic product $(h_1 \odot h_2)(D)$ vanishes in the above case. The shifted symplectic products $(h_1 \odot h_1)(D)$ and $(h_2 \odot h_2)(D)$ also vanish. The shifted symplectic product between two vectors of binary polynomials vanishes if and only if their corresponding Pauli sequences commute [30,38]. Time reversal (substituting D^{-1} for D) ensures that the shifted symplectic product checks commutativity for every shift of the two Pauli sequences being compared. The cases where the shifted symplectic product does not vanish (where the two Pauli sequences anticommute for one or more shifts) are important for constructing entanglement-assisted quantum convolutional codes.

3. Row and column operations

We can perform row operations on binary polynomial matrices for quantum convolutional codes. A row operation is merely a "mental" operation that has no effect on the states in the codespace or on the error-correcting properties of the code. We have three types of row operations:

1. An elementary row operation multiplies a row times an arbitrary binary polynomial and adds the result to another row. This additive invariance holds for any code that admits a description within the stabilizer formalism. Additive codes are invariant under multiplication of the stabilizer generators in the "Pauli picture" or under row addition in the "binary-polynomial picture."

2. Another type of row operation is to multiply a row by an arbitrary power of D. Ollivier and Tillich discuss such row operations as "multiplication of a line by D" and use them to find encoding operations for their quantum convolutional codes [30]. Grassl and Rötteler use this type of operation to find a subcode of a given quantum convolutional code with an equivalent asymptotic rate and equivalent error-correcting properties [31]. We use this type of row operation in each of our two classes of entanglement-assisted quantum convolutional codes.

3. We also employ row operations that multiply a row by an arbitrary polynomial (not necessarily a power of D). We only use these operations when the receiver performs a measurement to diagnose an error. This type of row operation occurs when we have generators with infinite weight that we would like to reduce to finite weight so that the receiver can perform measurements in an online fashion as qubits arrive from the noisy channel. We use this type of row operation in our second class of entanglement-assisted quantum convolutional codes.

A row operation does not change the shifted symplectic product when all generators commute. A row operation *does* change the shifted symplectic product of a set of generators that do not commute. It is a convenient tool for constructing our entanglement-assisted quantum convolutional codes.

We can also perform column operations on binary polynomial matrices for quantum convolutional codes. Column operations change the error-correcting properties of the code and are important for realizing a periodic encoding circuit for the code. We have two types of column operations:

1. An elementary column operation multiplies one column by an arbitrary binary polynomial and adds the result to another column. We implement elementary column operations with gates from the shift-invariant Clifford group [31,33].

2. Another column operation is to multiply column i in both the "X" and "Z" matrix by D^l where $l \in \mathbb{Z}$. We perform this operation by delaying or advancing the processing of qubit i by l frames relative to the original frame.

A column operation implemented on the "X" side of the binary polynomial matrix has a corresponding effect on the "Z" side of the binary polynomial matrix. This corresponding effect is a manifestation of the Heisenberg uncertainty principle because commutation relations remain invariant with respect to the action of quantum gates. The shifted symplectic product is therefore invariant with respect to column operations from the shift-invariant Clifford group. We describe possible column operations for implementing encoding circuits in the next section.

III. FINITE-DEPTH CLIFFORD OPERATIONS

One of the main advantages of a quantum convolutional code is that its encoding circuit has a periodic form. We can encode a stream of quantum information with the same physical routines or devices and therefore reduce encoding and decoding complexity.

Ollivier and Tillich were the first to discuss encoding circuits for quantum convolutional codes [29,30]. They provided a set of necessary and sufficient conditions to determine when an encoding circuit is noncatastrophic. A noncatastrophic encoding circuit does not propagate uncorrected errors infinitely through the decoded information qubit stream. Classical convolutional coding theory has a well-developed theory of noncatastrophic encoding circuits [44].

Grassl and Rötteler later showed that Ollivier and Tillich's conditions for a circuit to be noncatastrophic are too restrictive [31–33]. They found subcodes of quantum convolutional codes that admit noncatastrophic encoders. The noncatastrophic encoders are a sequence of Clifford circuits with finite depth. They developed a formalism for encapsulating the periodic structure of an encoding circuit by decomposing the encoding circuit as a set of elementary column operations. Their decoding circuits are exact inverses of their encoding circuits because their decoding circuits perform the encoding operations in reverse order.

Definition 2. A finite-depth operation transforms every finite-weight stabilizer generator to one with finite weight.

We review the finite-depth operations in the shift-invariant Clifford group [31–33]. The shift-invariant Clifford group is an extension of the Clifford group operations mentioned in Sec. II A. We describe how finite-depth operations in the shift-invariant Clifford group affect the binary polynomial matrix for a code. Each of the following operations acts on every frame of a quantum convolutional code.

1. The sender performs a C-NOT from qubit i to qubit j in every frame where qubit j is in a frame delayed by k. The effect on the binary polynomial matrix is to multiply column i by D^k and add the result to column j in the "X" matrix and

to multiply column j by D^{-k} and add the result to column i in the "Z" matrix.

2. A Hadamard on qubit i swaps column i in the "X" matrix with column i in the "Z" matrix.

3. A phase gate on qubit i adds column i from the "X" matrix to column i in the "Z" matrix.

4. A controlled-phase gate from qubit i to qubit j in a frame delayed by k multiplies column i in the "X" matrix by D^k and adds the result to column j in the "Z" matrix. It also multiples column j in the "X" matrix by D^{-k} and adds the result to column i in the "Z" matrix.

5. A controlled-phase gate from qubit i to qubit i in a frame delayed by k multiplies column i in the "X" matrix by $D^k + D^{-k}$ and adds the result to column i in the "Z" matrix.

We use finite-depth operations extensively in this work, but we employ only the above Hadamard and C-NOT gates because our entanglement-assisted quantum convolutional codes have the CSS structure. Figure 4 gives an example of an entanglement-assisted quantum convolutional code that employs several finite-depth operations. The circuit encodes a stream of information qubits with the help of ebits shared between sender and receiver.

Multiple C-NOT gates can realize an elementary column operation as described at the end of Sec. II. Suppose the elementary column operation multiplies column i in the "X" matrix by $f(D)$ and adds the result to column j. Polynomial $f(D)$ is a summation of some finite set $\{l_1, \ldots, l_n\}$ of powers of D:

$$f(D) = D^{l_1} + \cdots + D^{l_n}.$$

We can realize $f(D)$ by performing a C-NOT gate from qubit i to qubit j in a frame delayed by l_i for each $i \in \{1, \ldots, n\}$.

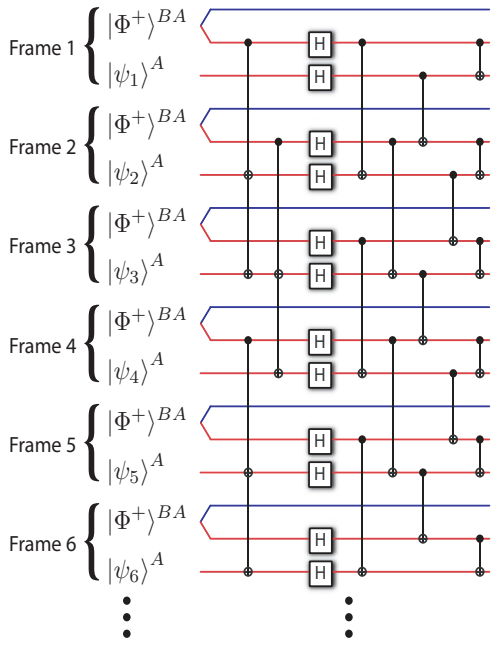

FIG. 4. (Color online) The finite-depth encoding circuit for the entanglement-assisted quantum convolutional code in Example 1. The above operations in reverse order give a valid decoding circuit.

IV. CSS ENTANGLEMENT-ASSISTED QUANTUM CONVOLUTIONAL CODES

An entanglement-assisted quantum convolutional code operates similarly to a standard quantum convolutional code. The main difference is that the sender and receiver share entanglement in the form of ebits. An $[[n,k;c]]$ entanglement-assisted quantum convolutional code encodes k information qubits per frame with the help of c ebits and $n-k-c$ ancilla qubits per frame. Figure 3 highlights the main features of the operation of an entanglement-assisted quantum convolutional code. The sender encodes a stream of quantum information using both additional ancillas and ebits. The sender performs the encoding operations on her qubits only (i.e., not including the halves of the ebits in possession of the receiver). The encoding operations have a periodic structure so that the same operations act on qubits in different frames and give the code a memory structure. The sender can perform these encoding operations in an online manner as she places more qubits in the unencoded qubit stream. The sender transmits her encoded qubits over the noisy quantum communication channel. The noisy channel does not affect the receiver's half of the shared ebits. The receiver combines the received noisy qubits with his half of the ebits and performs measurements to diagnose errors that may occur. These measurements may overlap on some of the same qubits. The receiver then diagnoses errors using a classical technique such as Viterbi error estimation [43], reverses the errors that the channel introduces, and finally performs an online decoding circuit that outputs the original information qubit stream.

Our main theorem below allows us to import two arbitrary classical convolutional codes for use as a CSS entanglement-assisted quantum convolutional code. Grassl and Rötteler were the first to construct CSS quantum convolutional codes from two classical binary convolutional codes that satisfy an orthogonality constraint—the polynomial parity check matrices $H_1(D)$ and $H_2(D)$ of the two classical codes are orthogonal with respect to the shifted symplectic product [33]:

$$H_1(D)H_2^T(D^{-1}) = 0. \tag{12}$$

The resulting symplectic code has a self-orthogonal parity-check matrix when we join them together using the CSS construction. Our theorem generalizes the work of Grassl and Rötteler because we can import two *arbitrary* classical binary convolutional codes—the codes do not necessarily have to obey the self-orthogonality constraint.

The theorem gives a direct way to compute the amount of entanglement that the code requires. The number of ebits required is equal to the rank of a particular matrix derived from the check matrices of the two classical codes. It generalizes an earlier theorem that determines the amount of entanglement required for an entanglement-assisted quantum block code [13].

Theorem 1 also provides a formula to compute the performance parameters of the entanglement-assisted quantum convolutional code from the performance parameters of the two classical codes. This formula ensures that high-rate classical convolutional codes produce high-rate entanglement-assisted quantum convolutional codes. Our constructions also

ensure high performance for the "trade-off" and "catalytic" rates by minimizing the number of ebits that the codes require.

We begin the proof of the theorem in this section and complete it in different ways for each of our two classes of entanglement-assisted quantum convolutional codes in Secs. V and VII. The proofs detail how to encode a stream of information qubits, ancilla qubits, and shared ebits into a code that has the CSS structure.

Theorem 1. Let $H_1(D)$ and $H_2(D)$ be the respective check matrices corresponding to noncatastrophic, delay-free encoders for classical binary convolutional codes C_1 and C_2. Suppose that classical code C_i encodes k_i information bits with n bits per frame where $i = 1,2$. The respective dimensions of $H_1(D)$ and $H_2(D)$ are thus $(n-k_1)\times n$ and $(n-k_2)\times n$. Then the resulting entanglement-assisted quantum convolutional code encodes k_1+k_2-n+c information qubits per frame and is an $[[n,k_1+k_2-n+c;c]]$ entanglement-assisted quantum convolutional code. The code requires c ebits per frame where c is equal to the rank of $H_1(D)H_2^T(D^{-1})$.

Let us begin the proof of the above theorem by constructing an entanglement-assisted quantum convolutional code. Consider the following quantum check matrix in CSS form:

$$\left[\begin{array}{c|c} H_1(D) & 0 \\ 0 & H_2(D) \end{array}\right]. \tag{13}$$

We label the above matrix as a "quantum check matrix" for now because it does not necessarily correspond to a commuting stabilizer. The quantum check matrix corresponds to a set of Pauli sequences whose error-correcting properties are desirable.

The following lemma begins the proof of the above theorem. It details an initial decomposition of the above quantum check matrix for each of our two classes of entanglement-assisted quantum convolutional codes.

Lemma 1. Elementary row and column operations relate the quantum check matrix in (13) to the following matrix:

$$\left[\begin{array}{cc|cc} E(D) & F(D) & 0 & 0 \\ 0 & 0 & I & 0 \end{array}\right],$$

where $E(D)$ is dimension $(n-k_1)\times(n-k_2)$, $F(D)$ is $(n-k_1)\times k_2$, the identity matrix is $(n-k_2)\times(n-k_2)$, and the null matrix on the right is $(n-k_2)\times k_2$. We give a definition of $E(D)$ and $F(D)$ in the following proof.

Proof. The Smith form [44] of $H_i(D)$ for each $i = 1,2$ is

$$H_i(D) = A_i(D)[\,I \quad 0\,]B_i(D), \tag{14}$$

where $A_i(D)$ is $(n-k_i)\times(n-k_i)$, the matrix in brackets is $(n-k_i)\times n$, and $B_i(D)$ is $n\times n$ [44]. Let $B_{ia}(D)$ be the first $n-k_i$ rows of $B_i(D)$ and let $B_{ib}(D)$ be the last k_i rows of $B_i(D)$:

$$B_i(D) = \left[\begin{array}{c} B_{ia}(D) \\ B_{ib}(D) \end{array}\right].$$

The $(n-k_i)\times(n-k_i)$ identity matrix in brackets in (14) indicates that the invariant factors of $H_i(D)$ for each $i = 1,2$ are all equal to one [44]. The invariant factors are all unity for both check matrices because the check matrices correspond to noncatastrophic, delay-free encoders [44]. The matrices $A_i(D)$ and $B_i(D)$ are a product of a sequence of elementary row and column operations, respectively [44].

Premultiplying $H_i(D)$ by $A_i^{-1}(D)$ gives a check matrix $H_i'(D)$ for each $i = 1,2$. Matrix $H_i'(D)$ is a check matrix for code C_i with equivalent error-correcting properties as $H_i(D)$ because row operations relate the two matrices. This new check matrix $H_i'(D)$ is equal to the first $n-k_i$ rows of matrix $B_i(D)$:

$$H_i'(D) = B_{ia}(D).$$

The invariant factors of $H_1(D)H_2^T(D^{-1})$ are equivalent to those of $H_1'(D)H_2'^T(D^{-1})$ because they are related by row and column operations [44]:

$$H_1(D)H_2^T(D^{-1}) = A_1(D)H_1'(D)H_2'^T(D^{-1})A_2^T(D^{-1}). \tag{15}$$

We now decompose the above quantum check matrix into a basic form using elementary row and column operations. The row operations have no effect on the error-correcting properties of the code, and the column operations correspond to elements of an encoding circuit. We later show how to incorporate ebits so that the quantum check matrix forms a valid commuting stabilizer.

Perform the row operations in matrices $A_i^{-1}(D)$ for both check matrices $H_i(D)$. The quantum check matrix becomes

$$\left[\begin{array}{c|c} B_{1a}(D) & 0 \\ 0 & B_{2a}(D) \end{array}\right]. \tag{16}$$

The error-correcting properties of the above generators are equivalent to those of the generators in (13) because row operations relate the two sets of generators. The matrix $B_2(D)$ corresponds to a sequence of elementary column operations $B_{2,i}(D)$:

$$B_2(D) = B_{2,1}(D)\cdots B_{2,l}(D) = \prod_{i=1}^{l} B_{2,i}(D).$$

The inverse matrix $B_2^{-1}(D)$ is therefore equal to the above sequence of operations in reverse order:

$$B_2^{-1}(D) = B_{2,l}(D)\cdots B_{2,1}(D) = \prod_{i=l}^{1} B_{2,i}(D).$$

Perform the elementary column operations in $B_2^{-1}(D)$ with C-NOT and SWAP gates [31]. The effect of each elementary column operation $B_{2,i}(D)$ is to postmultiply the "X" matrix by $B_{2,i}(D)$ and to postmultiply the "Z" matrix by $B_{2,i}^T(D^{-1})$. Therefore, the effect of all elementary operations is to postmultiply the "Z" matrix by $B_2^T(D^{-1})$ because

$$\prod_{i=l}^{1} B_{2,i}^T(D^{-1}) = \left(\prod_{i=1}^{l} B_{2,i}(D^{-1})\right)^T = B_2^T(D^{-1}).$$

The quantum check matrix in (16) becomes

$$\left[\begin{array}{cc|cc} B_{1a}(D)B_2^T(D^{-1}) & & 0 & 0 \\ 0 & & I & 0 \end{array}\right]. \tag{17}$$

Let $E(D)$ be equal to the first $n-k_1$ rows and $n-k_2$ columns of the "Z" matrix:

$$E(D) \equiv B_{1,a}(D)B_{2,a}^T(D^{-1}),$$

and let $F(D)$ be equal to the first $n-k_1$ rows and last k_2 columns of the "Z" matrix:

$$F(D) \equiv B_{1,a}(D)B_{2,b}^T(D^{-1}).$$

The quantum check matrix in (17) is then equivalent to the following matrix:

$$\left[\begin{array}{cc|cc} E(D) & F(D) & 0 & 0 \\ 0 & 0 & I & 0 \end{array}\right], \tag{18}$$

where each matrix above has the dimensions stated in the theorem above. ∎

The above operations end the initial set of operations that each of our two classes of entanglement-assisted quantum convolutional codes employs. We outline the remaining operations for each class of codes in what follows.

V. ENTANGLEMENT-ASSISTED QUANTUM CONVOLUTIONAL CODES WITH FINITE-DEPTH ENCODING AND DECODING CIRCUITS

This section details entanglement-assisted quantum convolutional codes in our first class. Codes in the first class admit an encoding and decoding circuit that employ finite-depth operations only. The check matrices for codes in this class have a property that allows this type of encoding and decoding. The following lemma gives the details of this property, and the proof outlines how to encode and decode this class of entanglement-assisted quantum convolutional codes.

Lemma 2. Suppose the Smith form of $H_1(D)H_2^T(D^{-1})$ is

$$H_1(D)H_2^T(D^{-1}) = A(D)\left[\begin{array}{cc} \Gamma(D) & 0 \\ 0 & 0 \end{array}\right]B(D),$$

where $A(D)$ is an $(n-k_1)\times(n-k_1)$ matrix, $B(D)$ is an $(n-k_2)\times(n-k_2)$ matrix, $\Gamma(D)$ is a diagonal $c\times c$ matrix whose entries are powers of D, and the matrix in brackets has dimension $(n-k_1)\times(n-k_2)$. Then the resulting entanglement-assisted quantum convolutional code has both a finite-depth encoding and decoding circuit.

Proof. We begin the proof of this theorem by continuing where the proof of Lemma IV ends. The crucial assumption for the above lemma is that the invariant factors of $H_1(D)H_2^T(D^{-1})$ are all powers of D. The Smith form of $E(D)$ in (18) therefore becomes

$$A_1^{-1}(D)A(D)\left[\begin{array}{cc} \Gamma(D) & 0 \\ 0 & 0 \end{array}\right]B(D)A_2^{-1}(D),$$

by employing the hypothesis of Lemma V and (15). The rank of both $H_1(D)H_2^T(D^{-1})$ and $E(D)$ is equal to c.

Perform the inverse of the row operations in $A_1^{-1}(D)A(D)$ on the first $n-k_1$ rows of the quantum check matrix in (18). Perform the inverse of the column operations in matrix $B(D)A_2^{-1}(D)$ on the first $n-k_2$ columns of the quantum check matrix in (18). We execute these column operations with Hadamard, C-NOT, and SWAP gates. These column operations have a corresponding effect on columns in the "X" matrix, but we can exploit the identity matrix in the last $n-k_2$ rows of the "X" matrix to counteract this effect. We perform row operations on the last $n-k_2$ rows of the matrix that act as the inverse of the column operations, and therefore the quantum check matrix in (18) becomes

$$\left[\begin{array}{ccc|ccc} \Gamma(D) & 0 & F_1(D) & 0 & 0 & 0 \\ 0 & 0 & F_2(D) & 0 & 0 & 0 \\ 0 & 0 & 0 & I & 0 & 0 \\ 0 & 0 & 0 & 0 & I & 0 \end{array}\right],$$

where $F_1(D)$ and $F_2(D)$ are the first c and $n-k_1-c$ respective rows of $A^{-1}(D)A_1(D)F(D)$. We perform Hadamard and C-NOT gates to clear the entries in $F_1(D)$ in the "Z" matrix above. The quantum check matrix becomes

$$\left[\begin{array}{ccc|ccc} \Gamma(D) & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & F_2(D) & 0 & 0 & 0 \\ 0 & 0 & 0 & I & 0 & 0 \\ 0 & I & 0 & 0 & 0 & 0 \end{array}\right]. \tag{19}$$

The Smith form of $F_2(D)$ is

$$F_2(D) = A_F(D)[\Gamma_F(D)\ 0]B_F(D),$$

where $\Gamma_F(D)$ is a diagonal matrix whose entries are powers of D, $A_F(D)$ is $(n-k_1-c)\times(n-k_1-c)$, and $B_F(D)$ is $k_2\times k_2$. The Smith form of $F_2(D)$ takes this particular form because the original check matrix $H_2(D)$ is noncatastrophic and column operations with Laurent polynomials change the invariant factors only up to powers of D.

Perform row operations corresponding to $A_F^{-1}(D)$ on the second set of $n-k_1-c$ rows with $F_2(D)$ in (19). Perform column operations corresponding to $B_F^{-1}(D)$ on columns $n-k_2+1,\ldots,n$ with Hadamard, C-NOT, and SWAP gates. The resulting quantum check matrix has the following form:

$$\left[\begin{array}{cccc|cccc} \Gamma(D) & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \Gamma_F(D) & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & I & 0 & 0 & 0 \\ 0 & I & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right]. \tag{20}$$

We have now completed the decomposition of the original quantum check matrix in (13) for this class of entanglement-assisted quantum convolutional codes. It is not possible to perform row or column operations to decompose the above matrix any further. The problem with the above quantum check matrix is that it does not form a valid quantum convolutional code. The first set of rows with matrix $\Gamma(D)$ are not orthogonal under the shifted symplectic product to the third set of rows with the identity matrix on the "X" side. Equivalently, the set of Pauli sequences corresponding to the above quantum check matrix do not form a commuting stabilizer. We can use entanglement shared between sender and receiver to solve this problem. Entanglement adds columns to the above quantum check matrix to resolve the issue. The additional columns correspond to qubits on the receiver's side. We next show in detail how to incorporate ancilla qubits, ebits, and information qubits to obtain a valid stabilizer code. The result is that we can exploit the error-correcting properties of the original code to protect the sender's qubits.

Consider the following check matrix corresponding to a commuting stabilizer:

$$\left[\begin{array}{ccccc|ccccc} I & I & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & I & I & 0 & 0 & 0 \\ 0 & 0 & I & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right], \tag{21}$$

where the identity matrices in the first and third sets of rows each have dimension $c \times c$, the identity matrix in the second set of rows has dimension $(n-k_1-c)\times(n-k_1-c)$, and the identity matrix in the fourth set of rows has dimension $(n-k_2-c)\times(n-k_2-c)$. The first and third sets of c rows stabilize a set of c ebits shared between Alice and Bob. Bob possesses the "left" c qubits and Alice possesses the "right" n qubits. The second and fourth sets of rows stabilize a set of $2(n-c)-k_1-k_2$ ancilla qubits that Alice possesses. The stabilizer, therefore, stabilizes a set of c ebits, $2(n-c)-k_1-k_2$ ancilla qubits, and k_1+k_2-n+c information qubits.

Observe that the last n columns of the "Z" and "X" matrices in the above stabilizer are similar in their layout to the entries in (20). We can delay the rows of the above stabilizer by an arbitrary amount to obtain the desired stabilizer. So the above stabilizer is a subcode of the following stabilizer in the sense of Ref. [31]:

$$\left[\begin{array}{ccccc|ccccc} \Gamma(D) & \Gamma(D) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \Gamma_F(D) & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & I & I & 0 & 0 & 0 \\ 0 & 0 & I & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right].$$

The stabilizer in (21) has equivalent error-correcting properties to and the same asymptotic rate as the above desired stabilizer. The above stabilizer matrix is an augmented version of the quantum check matrix in (20) that includes entanglement. The sender performs all of the encoding column operations detailed in the proofs of this lemma and Lemma 1 in reverse order. The result of these operations is an $\mathbf{[}[n,k_1+k_2-n+c;c]\mathbf{]}$ entanglement-assisted quantum convolutional code with the same error-correcting properties as the quantum check matrix in (13). The receiver decodes the original information-qubit stream by performing the column operations in the order presented. The information qubits appear as the last k_1+k_2-n+c in each frame of the stream (corresponding to the k_1+k_2-n+c columns of zeros in both the "Z" and "X" matrices above). ■

Example 1. Consider a classical convolutional code with the following check matrix:

$$H(D) = [1+D^2 \; 1+D+D^2].$$

We can use $H(D)$ in an entanglement-assisted quantum convolutional code to correct for both bit-flip errors and phase-flip errors. We form the following quantum check matrix:

$$\left[\begin{array}{cc|cc} 1+D^2 & 1+D+D^2 & 0 & 0 \\ 0 & 0 & 1+D^2 & 1+D+D^2 \end{array}\right]. \tag{22}$$

This code falls in the first class of entanglement-assisted quantum convolutional codes because $H(D)H^T(D^{-1})=1$. We do not show the decomposition of the above check matrix as outlined in Lemma 2, but instead show how to encode it starting from a stream of information qubits and ebits. Each frame has one ebit and one information qubit. Let us begin with a polynomial matrix that stabilizes the unencoded state:

$$\left[\begin{array}{ccc|ccc} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{array}\right].$$

Alice possesses the two qubits on the "right" and Bob possesses the qubit on the "left." We label the middle qubit as "qubit one" and the rightmost qubit as "qubit two." Alice performs a C-NOT from qubit one to qubit two in a delayed frame and a C-NOT from qubit one to qubit two in a frame delayed by two. The stabilizer becomes

$$\left[\begin{array}{ccc|ccc} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & D+D^2 \end{array}\right].$$

Alice performs Hadamard gates on both of her qubits. The stabilizer becomes

$$\left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & D+D^2 & 1 & 0 & 0 \end{array}\right].$$

Alice performs a C-NOT from qubit one to qubit two in a delayed frame. The stabilizer becomes

$$\left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 0 & 1 & D \\ 0 & D & D+D^2 & 1 & 0 & 0 \end{array}\right].$$

Alice performs a C-NOT from qubit two to qubit one in a delayed frame. The stabilizer becomes

$$\left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 0 & 1+D^2 & D \\ 0 & D & 1+D+D^2 & 1 & 0 & 0 \end{array}\right].$$

Alice performs a C-NOT from qubit one to qubit two. The stabilizer becomes

$$\left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 0 & 1+D^2 & 1+D+D^2 \\ 0 & 1+D^2 & 1+D+D^2 & 1 & 0 & 0 \end{array}\right].$$

A row operation that switches the first row with the second row gives the following stabilizer:

$$\left[\begin{array}{ccc|ccc} 0 & 1+D^2 & 1+D+D^2 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1+D^2 & 1+D+D^2 \end{array}\right].$$

The entries on Alice's side of the above stabilizer have equivalent error-correcting properties to the quantum check matrix in (22). Figure 4 illustrates how the above operations encode a stream of ebits and information qubits for our example.

4. Discussion

Codes in the first class are more useful in practice than those in the second because their encoding and decoding circuits are finite depth. An uncorrected error propagates only to a finite number of information qubits in the decoded qubit stream. Codes in the first class therefore do not require any assumptions about noiseless encoding or decoding.

The assumption about the invariant factors in the Smith form of $H_1(D)H_2^T(D^{-1})$ holds only for some classical check matrices. Only a subclass of classical codes satisfy this

assumption, but it still expands the set of available quantum codes beyond those whose check matrices $H_1(D)$ and $H_2(D)$ are orthogonal. We need further techniques to handle the classical codes for which this assumption does not hold. The following sections provide these further techniques to handle a larger class of entanglement-assisted quantum convolutional codes.

VI. INFINITE-DEPTH CLIFFORD OPERATIONS

We now introduce a new type of operation, an infinite-depth operation, to the set of operations in the shift-invariant Clifford group available for encoding and decoding quantum convolutional codes. We require infinite-depth operations to expand the set of classical convolutional codes that we can import for quantum convolutional coding.

Definition 3. An infinite-depth operation can transform a finite-weight stabilizer generator to one with infinite weight (but does not necessarily do so to every finite-weight generator).

A decoding circuit with infinite-depth operations on qubits sent over the noisy channel is undesirable because it spreads uncorrected errors infinitely into the decoded information qubit stream. But an encoding circuit with infinite-depth operations is acceptable if we assume a communication paradigm in which the only noisy process is the noisy quantum channel.

We later show several examples of circuits that include infinite-depth operations. Infinite-depth operations expand the possibilities for quantum convolutional circuits in much the same way that incorporating feedback expands the possibilities for classical convolutional circuits.

We illustrate the details of several infinite-depth operations for use in an entanglement-assisted quantum convolutional code. We first provide some specific examples of infinite-depth operations and then show how to realize an arbitrary infinite-depth operation.

We consider both the stabilizer and the logical operators for the information qubits in our analysis. Tracking both of these sets of generators is necessary for determining the proper decoding circuit when including infinite-depth operations.

A. Examples of infinite-depth operations

Our first example of an infinite-depth operation involves a stream of information qubits and ancilla qubits. We divide the stream into frames of three qubits where each frame has two ancilla qubits and one information qubit. The following two generators and each of their three-qubit shifts stabilize the qubit stream:

$$\cdots \left| \begin{matrix} I & I & I \\ I & I & I \end{matrix} \right| \left| \begin{matrix} Z & I & I \\ I & Z & I \end{matrix} \right| \left| \begin{matrix} I & I & I \\ I & I & I \end{matrix} \right| \cdots \tag{23}$$

The binary polynomial matrix corresponding to this stabilizer is as follows:

$$\left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{array} \right]. \tag{24}$$

We obtain any Pauli sequence in the stabilizer by multiplying the above rows by a power of D and applying the inverse of the P2B isomorphism. The logical operators for the information qubits are as follows:

$$\cdots \left| \begin{matrix} I & I & I \\ I & I & I \end{matrix} \right| \left| \begin{matrix} I & I & X \\ I & I & Z \end{matrix} \right| \left| \begin{matrix} I & I & I \\ I & I & I \end{matrix} \right| \cdots$$

They also admit a description with a binary polynomial matrix:

$$\left[\begin{array}{ccc|ccc} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{array} \right]. \tag{25}$$

We refer to the above matrix as the "information-qubit matrix."

1. Encoding

Suppose we would like to encode the above stream so that the following generators stabilize it:

$$\cdots \left| \begin{matrix} I & I & I \\ I & I & I \end{matrix} \right| \left| \begin{matrix} X & X & X \\ Z & Z & I \end{matrix} \right| \left| \begin{matrix} X & X & I \\ I & I & I \end{matrix} \right| \cdots,$$

or equivalently, the following binary polynomial matrix stabilizes it:

$$\left[\begin{array}{ccc|ccc} 0 & 0 & 0 & D+1 & D+1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 \end{array} \right]. \tag{26}$$

We encode the above stabilizer using a combination of finite-depth operations and an infinite-depth operation. We perform a Hadamard on the first qubit in each frame and follow with a C-NOT from the first qubit to the second and third qubits in each frame. These operations transform the matrix in (24) to the following matrix:

$$\left[\begin{array}{ccc|ccc} 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 \end{array} \right],$$

or equivalently transform the generators in (23) to the following generators:

$$\cdots \left| \begin{matrix} I & I & I \\ I & I & I \end{matrix} \right| \left| \begin{matrix} X & X & X \\ Z & Z & I \end{matrix} \right| \left| \begin{matrix} I & I & I \\ I & I & I \end{matrix} \right| \cdots$$

The information-qubit matrix becomes

$$\left[\begin{array}{ccc|ccc} 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \end{array} \right].$$

We now perform an infinite-depth operation: a C-NOT from the third qubit in one frame to the third qubit in a delayed frame and repeat this operation for all following frames. Figure 5 shows this operation acting on our stream of qubits with three qubits per frame. The effect of this operation is to translate the above stabilizer generators as follows:

$$\cdots \left| \begin{matrix} I & I & I \\ I & I & I \end{matrix} \right| \left| \begin{matrix} X & X & X \\ Z & Z & I \end{matrix} \right| \left| \begin{matrix} I & I & X \\ I & I & I \end{matrix} \right| \left| \begin{matrix} I & I & X \\ I & I & I \end{matrix} \right| \cdots$$

The first generator above and each of its three-qubit shifts is an infinite-weight generator if the above sequence of C-NOTs acts on the entire countably infinite qubit stream. We represent the above stabilizer with the binary *rational* polynomial matrix,

$$\left[\begin{array}{ccc|ccc} 0 & 0 & 0 & 1 & 1 & 1/(1+D) \\ 1 & 1 & 0 & 0 & 0 & 0 \end{array} \right], \tag{27}$$

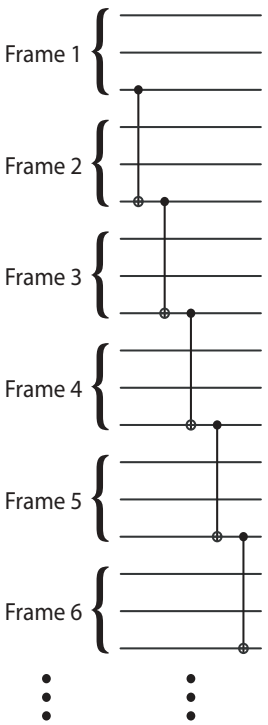

FIG. 5. An example of an infinite-depth operation. A sequence of C-NOT gates acts on the third qubit of every frame. This infinite-depth operation effectively multiplies the third column of the "X" side of the binary polynomial matrix by the rational polynomial $1/(1+D)$ and multiplies the third column of the "Z" side of the binary polynomial matrix by $1+D^{-1}$.

where $1/(1+D) = 1 + D + D^2 + \cdots$ is a repeating fraction. The operation is infinite-depth because it translates the original finite-weight stabilizer generator to one with infinite weight.

It is possible to perform a row operation that multiplies the first row by $D+1$. This operation gives a stabilizer matrix that is equivalent to the desired stabilizer in (26). The receiver of the encoded qubits measures the finite-weight stabilizer generators in (26) to diagnose errors. These measurements do not disturb the information qubits because they also stabilize the encoded stream.

The above encoding operations transform the information-qubit matrix as follows:

$$\left[\begin{array}{ccc|ccc} 0 & 0 & 0 & 0 & 0 & 1/(1+D) \\ 1 & 0 & 1+D^{-1} & 0 & 0 & 0 \end{array}\right]. \tag{28}$$

The infinite-depth operation on the third qubit has an effect on the "Z" or left side of the information-qubit matrix as illustrated in the second row of the above matrix. The effect is to multiply the third column of the "Z" matrix by $f(D^{-1})$ if the operation multiplies the third column of the "X" matrix by $1/f(D)$. This corresponding action on the "Z" side occurs because the commutation relations of the Pauli operators remain invariant under quantum gates, or equivalently, the shifted symplectic product remains invariant under column operations. The original shifted symplectic product for the logical operators is one, and it remains as one because $f(D^{-1})^{-1}/f(D) = 1$.

2. Decoding

We perform finite-depth operations to decode the stream of information qubits. Begin with the stabilizer and information-qubit matrix in (27) and (28), respectively. Perform a C-NOT from the first qubit to the second qubit. The stabilizer becomes

$$\left[\begin{array}{ccc|ccc} 0 & 0 & 0 & 1 & 0 & 1/(1+D) \\ 0 & 1 & 0 & 0 & 0 & 0 \end{array}\right],$$

and the information-qubit matrix does not change. Perform a C-NOT from the third qubit to the first qubit in the same frame and in a delayed frame. These gates multiply column three in the "X" matrix by $1+D$ and add the result to column one. The gates also multiply column one in the "Z" matrix by $1+D^{-1}$ and add the result to column three. The effect is as follows on both the stabilizer:

$$\left[\begin{array}{ccc|ccc} 0 & 0 & 0 & 0 & 0 & 1/(1+D) \\ 0 & 1 & 0 & 0 & 0 & 0 \end{array}\right], \tag{29}$$

and the information-qubit matrix:

$$\left[\begin{array}{ccc|ccc} 0 & 0 & 0 & 1 & 0 & 1/(1+D) \\ 1 & 0 & 0 & 0 & 0 & 0 \end{array}\right]. \tag{30}$$

We can multiply the logical operators by any element of the stabilizer and obtain an equivalent logical operator [4]. We perform this multiplication in the "binary-polynomial picture" by adding the first row of the stabilizer in (29) to the first row of (30). The information-qubit matrix becomes

$$\left[\begin{array}{ccc|ccc} 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{array}\right], \tag{31}$$

so that the resulting logical operators act only on the first qubit of every frame. We have successfully decoded the information qubits with finite-depth operations. The information qubits teleport coherently [45,46] from being the third qubit of each frame as in (25) to being the first qubit of each frame as in (31). We exploit the above method of encoding with infinite-depth operations and decoding with finite-depth operations for the class of entanglement-assisted quantum convolutional codes in Sec. VII.

B. General infinite-depth operations

We discuss the action of a general infinite-depth operation on two weight-one "X" and "Z" Pauli sequences where each frame has one Pauli matrix. Our analysis then determines the effect of an infinite-depth operation on an arbitrary stabilizer or information-qubit matrix. The generators in the "Pauli picture" are as follows:

$$\cdots \left|\begin{array}{c} I \\ I \end{array}\right| \left|\begin{array}{c} X \\ Z \end{array}\right| \left|\begin{array}{c} I \\ I \end{array}\right| \cdots, \tag{32}$$

or as follows in the "binary-polynomial picture":

$$\left[\begin{array}{c|c} 0 & 1 \\ 1 & 0 \end{array}\right].$$

An infinite-depth $1/f(D)$ operation, where $f(D)$ is an arbitrary polynomial, should transform the above matrix to the following one:

$$\left[\begin{array}{c|c} 0 & 1/f(D) \\ f(D^{-1}) & 0 \end{array}\right].$$

A circuit that performs this transformation preserves the shifted symplectic product because $f(D^{-1}) \cdot 1/f(D^{-1}) = 1$. The circuit should operate on a few qubits at a time and should be shift invariant—the same device or physical routines implement it.

First perform the long-division expansion of binary rational polynomial $1/f(D)$. This expansion has a particular repeating pattern with period l. For example, suppose that $f(D) = 1 + D + D^3$. Its long-division expansion is $1 + D + D^2 + D^4 + D^7 + D^8 + D^9 + D^{11} + \cdots$ and exhibits a repeating pattern with period seven. We want a circuit that realizes the following Pauli generators:

$$\cdots \left|\begin{array}{c} I \\ Z \end{array}\right| \left|\begin{array}{c} I \\ I \end{array}\right| \left|\begin{array}{c} I \\ Z \end{array}\right| \left|\begin{array}{c} X \\ Z \end{array}\right| \left|\begin{array}{c} X \\ I \end{array}\right| \left|\begin{array}{c} X \\ I \end{array}\right| \left|\begin{array}{c} I \\ I \end{array}\right| \left|\begin{array}{c} X \\ I \end{array}\right| \left|\begin{array}{c} I \\ I \end{array}\right| \cdots, \tag{33}$$

where the pattern in the X matrices is the same as the repeating polynomial $1/f\,(D)$ and continues infinitely to the right, and the pattern on the Z matrices is the same as that in $f\left(D^{-1}\right)$ and terminates at the left. The above Pauli sequence is equivalent to the following binary rational polynomial matrix:

$$\left[\begin{array}{c|c} 0 & 1/(1 + D + D^3) \\ 1 + D^{-1} + D^{-3} & 0 \end{array}\right].$$

We now discuss a method that realizes an arbitrary rational polynomial $1/f(D)$ as an infinite-depth operation. Our method for encoding the generators in (33) from those in (32) consists of a "sliding-window" technique that determines transformation rules for the circuit. The circuit is an additive, shift-invariant filtering operation. It resembles an infinite-impulse response filter because the sequence it produces extends infinitely. In general, the number N of qubits that the encoding unitary operates on is as follows:

$$N = \deg[f(D)] - \mathrm{del}[f(D)] + 1,$$

where $\deg[f(D)]$ and $\mathrm{del}[f(D)]$ are the respective highest and lowest powers of polynomial $f(D)$. Therefore, our exemplary encoding unitary operates on four qubits at a time. We delay the original sequence in (32) by three frames. These initial frames are "scratch" frames that give the encoding unitary enough "room" to generate the desired Paulis in (33). The first set of transformation rules is as follows:

$$\left|\begin{array}{c} I \\ I \end{array}\right| \left|\begin{array}{c} I \\ I \end{array}\right| \left|\begin{array}{c} I \\ I \end{array}\right| \left|\begin{array}{c} X \\ Z \end{array}\right| \rightarrow \left|\begin{array}{c} I \\ Z \end{array}\right| \left|\begin{array}{c} I \\ I \end{array}\right| \left|\begin{array}{c} I \\ Z \end{array}\right| \left|\begin{array}{c} X \\ Z \end{array}\right|, \tag{34}$$

and generates the first four elements of the pattern in (33). Now that the encoding unitary has acted on the first four frames, we need to shift our eyes to the right by one frame in the sequence in (33) to determine the next set of rules. So we shift the above outputs by one frame to the *left* (assuming that only identity matrices lie to the right) and determine the next set of transformation rules that generate the next elements of the sequence in (33):

$$\left|\begin{array}{c} I \\ I \end{array}\right| \left|\begin{array}{c} I \\ Z \end{array}\right| \left|\begin{array}{c} X \\ Z \end{array}\right| \left|\begin{array}{c} I \\ I \end{array}\right| \rightarrow \left|\begin{array}{c} I \\ I \end{array}\right| \left|\begin{array}{c} I \\ Z \end{array}\right| \left|\begin{array}{c} X \\ Z \end{array}\right| \left|\begin{array}{c} X \\ I \end{array}\right|.$$

Shift the above outputs to the left by one frame to determine the next set of transformation rules:

$$\left|\begin{array}{c} I \\ Z \end{array}\right| \left|\begin{array}{c} X \\ Z \end{array}\right| \left|\begin{array}{c} X \\ I \end{array}\right| \left|\begin{array}{c} I \\ I \end{array}\right| \rightarrow \left|\begin{array}{c} I \\ Z \end{array}\right| \left|\begin{array}{c} X \\ Z \end{array}\right| \left|\begin{array}{c} X \\ I \end{array}\right| \left|\begin{array}{c} X \\ I \end{array}\right|.$$

We obtain the rest of the transformation rules by continuing this sliding process, and we stop when the pattern in the sequence in (33) begins to repeat:

$$\left|\begin{array}{c} X \\ Z \end{array}\right| \left|\begin{array}{c} X \\ I \end{array}\right| \left|\begin{array}{c} X \\ I \end{array}\right| \left|\begin{array}{c} I \\ I \end{array}\right| \rightarrow \left|\begin{array}{c} X \\ Z \end{array}\right| \left|\begin{array}{c} X \\ I \end{array}\right| \left|\begin{array}{c} X \\ I \end{array}\right| \left|\begin{array}{c} I \\ I \end{array}\right|$$
$$\left|\begin{array}{c} X \\ X \end{array}\right| \left|\begin{array}{c} X \\ I \end{array}\right| \left|\begin{array}{c} I \\ I \end{array}\right| \left|\begin{array}{c} I \\ I \end{array}\right| \rightarrow \left|\begin{array}{c} X \\ X \end{array}\right| \left|\begin{array}{c} X \\ I \end{array}\right| \left|\begin{array}{c} I \\ I \end{array}\right| \left|\begin{array}{c} X \\ I \end{array}\right|$$
$$\left|\begin{array}{c} X \\ I \end{array}\right| \left|\begin{array}{c} X \\ X \end{array}\right| \left|\begin{array}{c} I \\ I \end{array}\right| \left|\begin{array}{c} I \\ I \end{array}\right| \rightarrow \left|\begin{array}{c} X \\ I \end{array}\right| \left|\begin{array}{c} X \\ I \end{array}\right| \left|\begin{array}{c} X \\ I \end{array}\right| \left|\begin{array}{c} I \\ I \end{array}\right|$$
$$\left|\begin{array}{c} I \\ X \end{array}\right| \left|\begin{array}{c} I \\ I \end{array}\right| \left|\begin{array}{c} I \\ I \end{array}\right| \left|\begin{array}{c} I \\ I \end{array}\right| \rightarrow \left|\begin{array}{c} I \\ I \end{array}\right| \left|\begin{array}{c} I \\ I \end{array}\right| \left|\begin{array}{c} I \\ I \end{array}\right| \left|\begin{array}{c} I \\ I \end{array}\right|$$
$$\left|\begin{array}{c} X \\ I \end{array}\right| \left|\begin{array}{c} I \\ I \end{array}\right| \left|\begin{array}{c} I \\ I \end{array}\right| \left|\begin{array}{c} I \\ I \end{array}\right| \rightarrow \left|\begin{array}{c} X \\ I \end{array}\right| \left|\begin{array}{c} I \\ I \end{array}\right| \left|\begin{array}{c} I \\ I \end{array}\right| \left|\begin{array}{c} I \\ I \end{array}\right|.$$

The above set of rules determines the encoding unitary and only a few of them are actually necessary. We can multiply the rules together to form equivalent rules because the circuit obeys additivity (in the "binary-polynomial picture"). The rules become as follows after rearranging into a standard form:

$$\left|\begin{array}{c} Z \\ I \end{array}\right| \left|\begin{array}{c} I \\ Z \end{array}\right| \left|\begin{array}{c} I \\ I \end{array}\right| \left|\begin{array}{c} I \\ I \end{array}\right| \rightarrow \left|\begin{array}{c} Z \\ I \end{array}\right| \left|\begin{array}{c} I \\ Z \end{array}\right| \left|\begin{array}{c} I \\ I \end{array}\right| \left|\begin{array}{c} I \\ I \end{array}\right|$$
$$\left|\begin{array}{c} I \\ I \end{array}\right| \left|\begin{array}{c} I \\ I \end{array}\right| \left|\begin{array}{c} Z \\ I \end{array}\right| \left|\begin{array}{c} I \\ Z \end{array}\right| \rightarrow \left|\begin{array}{c} I \\ I \end{array}\right| \left|\begin{array}{c} I \\ I \end{array}\right| \left|\begin{array}{c} Z \\ I \end{array}\right| \left|\begin{array}{c} I \\ Z \end{array}\right|$$
$$\left|\begin{array}{c} I \\ I \end{array}\right| \left|\begin{array}{c} I \\ I \end{array}\right| \left|\begin{array}{c} I \\ Z \end{array}\right| \left|\begin{array}{c} Z \\ I \end{array}\right| \rightarrow \left|\begin{array}{c} I \\ I \end{array}\right| \left|\begin{array}{c} I \\ I \end{array}\right| \left|\begin{array}{c} Z \\ I \end{array}\right| \left|\begin{array}{c} Z \\ Z \end{array}\right|$$
$$\left|\begin{array}{c} X \\ I \end{array}\right| \left|\begin{array}{c} I \\ I \end{array}\right| \left|\begin{array}{c} I \\ I \end{array}\right| \left|\begin{array}{c} I \\ I \end{array}\right| \rightarrow \left|\begin{array}{c} X \\ I \end{array}\right| \left|\begin{array}{c} I \\ I \end{array}\right| \left|\begin{array}{c} I \\ I \end{array}\right| \left|\begin{array}{c} X \\ I \end{array}\right|$$
$$\left|\begin{array}{c} I \\ X \end{array}\right| \left|\begin{array}{c} I \\ I \end{array}\right| \left|\begin{array}{c} I \\ I \end{array}\right| \left|\begin{array}{c} I \\ I \end{array}\right| \rightarrow \left|\begin{array}{c} I \\ X \end{array}\right| \left|\begin{array}{c} I \\ I \end{array}\right| \left|\begin{array}{c} I \\ I \end{array}\right| \left|\begin{array}{c} I \\ I \end{array}\right|$$
$$\left|\begin{array}{c} I \\ I \end{array}\right| \left|\begin{array}{c} I \\ I \end{array}\right| \left|\begin{array}{c} X \\ I \end{array}\right| \left|\begin{array}{c} I \\ I \end{array}\right| \rightarrow \left|\begin{array}{c} I \\ I \end{array}\right| \left|\begin{array}{c} I \\ I \end{array}\right| \left|\begin{array}{c} X \\ I \end{array}\right| \left|\begin{array}{c} X \\ I \end{array}\right|$$
$$\left|\begin{array}{c} I \\ I \end{array}\right| \left|\begin{array}{c} I \\ I \end{array}\right| \left|\begin{array}{c} I \\ I \end{array}\right| \left|\begin{array}{c} X \\ I \end{array}\right| \rightarrow \left|\begin{array}{c} I \\ I \end{array}\right| \left|\begin{array}{c} I \\ I \end{array}\right| \left|\begin{array}{c} I \\ I \end{array}\right| \left|\begin{array}{c} X \\ I \end{array}\right|.$$

A C-NOT from qubit one to qubit four and a C-NOT from qubit three to qubit four suffice to implement this circuit. We repeatedly apply these operations shifting by one frame at a time to implement the infinite-depth operation. We could have observed that these gates suffice to implement the "Z" transformation in the first set of transformation rules in (34), but we wanted to show how this method generates the full periodic "X" sequence in (33). Figure 6 shows how the above encoding unitary acts on a stream of quantum information.

We can determine the encoding unitary for an arbitrary rational polynomial $1/f\,(D)$ using a similar method. Suppose that $\mathrm{del}\,[f\,(D)] = n$ and suppose $n \neq 0$ as in the above case. First delay or advance the frames if $n > 0$ or if $n < 0$, respectively. Determine the C-NOT gates that transform the "Z" Pauli sequence,

$$[\,1\,|\,0\,],$$

to

$$[\,D^n f(D^{-1})\,|\,0\,].$$

These C-NOT gates form the encoding circuit that transform both the "X" and "Z" Pauli sequences. We perform the

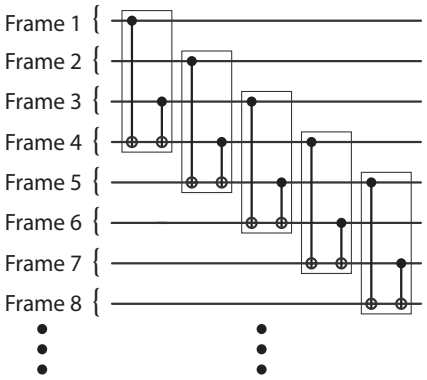

FIG. 6. Another example of an infinite-depth operation. An infinite-depth operation acts on qubit i in every frame. This particular infinite-depth operation multiplies column i on the "X" side of the binary polynomial matrix by $1/\left(1+D+D^3\right)$ and multiplies column i on the "Z" side of the binary polynomial matrix by $1+D^{-1}+D^{-3}$.

encoding unitary, shift by one frame, perform it again, and keep repeating. Our method encodes any arbitrary polynomial $1/f(D)$ on the "X" side and $f(D^{-1})$ on the "Z" side.

We can implement the "time-reversed" polynomial $1/f(D^{-1})$ on the "X" side by first delaying the frames by $m = \deg[f(D)] - \mathrm{del}[f(D)]$ frames and performing the circuit corresponding to $1/D^m[f(D^{-1})]$. These operations implement the circuit $D^m/D^m[f(D^{-1})] = 1/f(D^{-1})$.

C. Infinite-depth operations in practice

We assume above that each of the infinite-depth operations acts on the entire countably infinite stream of qubits. In practice, each infinite-depth operation acts on a finite number of qubits at a time so that the encoding and decoding circuits operate in an "online" manner. Therefore, each infinite-depth operation approximates its corresponding rational polynomial. This approximation does not pose a barrier to implementation. We can implement each of the above infinite-depth operations by padding the initial qubits of the information qubit stream with some "scratch" qubits. We first transmit these "scratch" qubits that contain no useful quantum information so that the later information qubits enjoy the full protection of the code. These scratch qubits do not affect the asymptotic rate of the code and merely serve as a convenience for implementing the infinite-depth operations. From now on, we adhere to describing infinite-depth operations with binary rational polynomials because it is more convenient to do so mathematically.

D. Entanglement-assisted quantum convolutional codes with infinite-depth operations

In the section that follows, our entanglement-assisted quantum convolutional codes have infinite-depth operations in their encoding circuits. This possibility is acceptable because the entanglement-assisted communication paradigm assumes that noiseless encoding is possible and that the receiver's half of the ebits are noiseless. We later briefly discuss the effects of relaxing this assumption in a realistic system.

Our decoding circuits in the second class of codes perform finite-depth operations. Some of our decoding circuits are not the exact inverse of their corresponding encoding circuits, but the decoding circuits invert the effect of the encoding circuits because they produce the original stream of information qubits at their output.

VII. ENTANGLEMENT-ASSISTED QUANTUM CONVOLUTIONAL CODES WITH INFINITE-DEPTH ENCODING AND FINITE-DEPTH DECODING CIRCUITS

This section details codes whose encoding circuits have both infinite-depth and finite-depth operations. We therefore assume that encoding is noiseless to eliminate the possibility of encoding errors spreading infinitely into the encoded qubit stream. Their decoding circuits require finite-depth operations only.

Just as with the previous class, this class of codes is determined by the properties of their corresponding classical check matrices, as described in the following lemma.

Lemma 3. Suppose the Smith form of $E(D)$ does not admit the form from Lemma 2. Then the entanglement-assisted quantum convolutional code has an encoding circuit with both infinite-depth and finite-depth operations. Its decoding circuit has finite-depth operations.

Proof. We perform all of the operations from Lemma 1. The Smith form of $E(D)$ is in general as follows:

$$A_E(D)\begin{bmatrix}\Gamma_1(D) & 0 & 0\\ 0 & \Gamma_2(D) & 0\\ 0 & 0 & 0\end{bmatrix}B_E(D),$$

where $A_E(D)$ is $(n-k_1)\times(n-k_1)$, $\Gamma_1(D)$ is an $s\times s$ diagonal matrix whose entries are powers of D, $\Gamma_2(D)$ is a $(c-s)\times(c-s)$ diagonal matrix whose entries are arbitrary polynomials, and $B_E(D)$ is $(n-k_2)\times(n-k_2)$. Perform the row operations in $A_E^{-1}(D)$ and the column operations in $B_E^{-1}(D)$ on the quantum check matrix in (18). Counteract the effect of the column operations on the identity matrix in the "X" matrix by performing row operations. The quantum check matrix in (18) becomes

$$\left[\begin{array}{cccc|cc}\Gamma_1(D) & 0 & 0 & F_1(D) & 0 & 0\\ 0 & \Gamma_2(D) & 0 & F_2(D) & 0 & 0\\ 0 & 0 & 0 & F_3(D) & 0 & 0\\ 0 & 0 & 0 & 0 & I & 0\end{array}\right],$$

where $F_1(D)$, $F_2(D)$, and $F_3(D)$ are the respective s, $c-s$, and $n-k_1-c$ rows of $A_E^{-1}(D)F(D)$. The Smith form of $F_3(D)$ is as follows:

$$F_3(D) = A_{F_3}(D)\,[\Gamma_{F_3}(D)\;0]\,B_{F_3}(D),$$

where $A_{F_3}(D)$ is $(n-k_1-c)\times(n-k_1-c)$, $\Gamma_{F_3}(D)$ is an $(n-k_1-c)\times(n-k_1-c)$ diagonal matrix whose entries are powers of D, and $B_{F_3}(D)$ is $k_2\times k_2$. The entries of $\Gamma_{F_3}(D)$ are powers of D because the original check matrix $H_2(D)$ is noncatastrophic and column and row operations with Laurent polynomials change the invariant factors only by a power of

D. Perform the row operations in $A_{F_3}^{-1}(D)$ and the column operations in $B_{F_3}^{-1}(D)$. The quantum check matrix becomes

$$\left[\begin{array}{cccccc|cc} \Gamma_1(D) & 0 & 0 & F_{1a}^{'}(D) & F_{1b}^{'}(D) & & 0 & 0 \\ 0 & \Gamma_2(D) & 0 & F_{2a}^{'}(D) & F_{2b}^{'}(D) & & 0 & 0 \\ 0 & 0 & 0 & \Gamma_{F_3}(D) & 0 & & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & & I & 0 \end{array}\right],$$

where $F_{1a}^{'}(D)$, $F_{1b}^{'}(D)$, $F_{2a}^{'}(D)$, $F_{2b}^{'}(D)$ are the matrices resulting from the column operations in $B_{F_3}^{-1}(D)$. Perform row operations from the entries in $\Gamma_{F_3}(D)$ to the rows above it to clear the entries in $F_{1a}^{'}(D)$ and $F_{2a}^{'}(D)$. Use Hadamard and C-NOT gates to clear the entries in $F_{1b}^{'}(D)$. The quantum check matrix becomes

$$\left[\begin{array}{ccccc|cc} \Gamma_1(D) & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \Gamma_2(D) & 0 & 0 & F_{2b}^{'}(D) & 0 & 0 \\ 0 & 0 & 0 & \Gamma_{F_3}(D) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & I & 0 \end{array}\right].$$

We can reduce $F_{2b}^{'}(D)$ to a lower triangular form with an algorithm consisting of column operations only. The algorithm operates on the last $k_2 + k_1 - n + c$ columns. It is similar to the Smith algorithm but does not involve row operations. Consider the first row of $F_{2b}^{'}(D)$. Perform column operations between the different elements of the row to reduce it to one nonzero entry. Swap this nonzero entry to the leftmost position. Perform the same algorithm on elements $2, \ldots, k_2 + k_1 - n + c$ of the second row. Continue on for all rows of $F_{2b}^{'}(D)$ to reduce it to a matrix of the following form:

$$F_{2b}^{'}(D) \rightarrow [\overbrace{L(D)}^{c-s} \quad \overbrace{0}^{k_1+k_2-n+s}],$$

where $L(D)$ is a lower triangular matrix. The above quantum check matrix becomes

$$\left[\begin{array}{cccccc|cc} \Gamma_1(D) & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \Gamma_2(D) & 0 & 0 & L(D) & 0 & 0 & 0 \\ 0 & 0 & 0 & \Gamma_{F_3}(D) & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & I & 0 \end{array}\right].$$

We have completed decomposition of the first set of s rows with $\Gamma_1(D)$, the third set of $n - k_1 - c$ rows with $\Gamma_{F_3}(D)$, and rows $n - k_1 + 1, \ldots, n - k_1 + s$ with the identity matrix on the "X" side.

We now consider an algorithm with infinite-depth operations to encode the following submatrix of the above quantum check matrix:

$$\left[\begin{array}{cc|cc} \Gamma_2(D) & L(D) & 0 & 0 \\ 0 & 0 & I & 0 \end{array}\right]. \tag{35}$$

We begin with a set of $c - s$ ebits and $c - s$ information qubits. The following matrix stabilizes the ebits:

$$\left[\begin{array}{ccc|ccc} I & I & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I & I & 0 \end{array}\right],$$

and the following matrix represents the information qubits:

$$\left[\begin{array}{ccc|ccc} 0 & 0 & I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & I \end{array}\right],$$

where all matrices have dimension $(c - s) \times (c - s)$ and Bob possesses the $c - s$ qubits on the "left" and Alice possesses the $2(c - s)$ qubits on the "right." We track both the stabilizer and the information qubits as they progress through some encoding operations. Alice performs C-NOT and Hadamard gates on her $2(c - s)$ qubits. These gates multiply the middle $c - s$ columns of the "Z" matrix by $L(D)$ and add the result to the last $c - s$ columns and multiply the last $c - s$ columns of the "X" matrix by $L^T(D^{-1})$ and add the result to the middle $c - s$ columns. The stabilizer becomes

$$\left[\begin{array}{ccc|ccc} I & I & L(D) & 0 & 0 & 0 \\ 0 & 0 & 0 & I & I & 0 \end{array}\right],$$

and the information-qubit matrix becomes

$$\left[\begin{array}{ccc|ccc} 0 & 0 & I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & L^T(D^{-1}) & I \end{array}\right].$$

Alice performs infinite-depth operations on her first $c - s$ qubits corresponding to the rational polynomials $\gamma_{2,1}^{-1}(D^{-1})$, $\ldots$, $\gamma_{2,c-s}^{-1}(D^{-1})$ in $\Gamma_2^{-1}(D^{-1})$. The stabilizer matrix becomes

$$\left[\begin{array}{ccc|ccc} I & \Gamma_2(D) & L(D) & 0 & 0 & 0 \\ 0 & 0 & 0 & I & \Gamma_2^{-1}(D^{-1}) & 0 \end{array}\right],$$

and the information-qubit matrix becomes

$$\left[\begin{array}{ccc|ccc} 0 & 0 & I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & L^T(D^{-1})\Gamma_2^{-1}(D^{-1}) & I \end{array}\right].$$

Alice's part of the above stabilizer matrix is equivalent to the quantum check matrix in (35) by row operations [premultiplying the second set of rows in the stabilizer by $\Gamma_2(D)$]. Bob can therefore make stabilizer measurements that have finite weight and that are equivalent to the desired stabilizer.

We now describe a method to decode the above encoded stabilizer and information-qubit matrix so that the information qubits appear at the output of the decoding circuit. Bob performs Hadamard gates on his first and third sets of $c - s$ qubits, performs C-NOT gates from the first set of qubits to the third set of qubits corresponding to the entries in $L(D)$, and performs the Hadamard gates again. The stabilizer becomes

$$\left[\begin{array}{ccc|ccc} I & \Gamma_2(D) & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I & \Gamma_2^{-1}(D^{-1}) & 0 \end{array}\right], \tag{36}$$

and the information-qubit matrix becomes

$$\left[\begin{array}{ccc|ccc} 0 & 0 & I & 0 & 0 & 0 \\ 0 & 0 & 0 & L^T(D^{-1}) & L^T(D^{-1})\Gamma_2^{-1}(D^{-1}) & I \end{array}\right].$$

Bob finishes decoding at this point because we can equivalently express the information-qubit matrix as follows:

$$\left[\begin{array}{ccc|ccc} 0 & 0 & I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & I \end{array}\right],$$

by multiplying the last $c-s$ rows of the stabilizer by $L^T(D^{-1})$ and adding to the last $c-s$ rows of the information-qubit matrix.

The overall procedure for encoding is to begin with a set of c ebits, $2(n-c)-k_1-k_2$ ancilla qubits, and k_1+k_2-n+c information qubits. We perform the infinite-depth operations detailed in the paragraph with (35) for $c-s$ of the ebits. We then perform the finite-depth operations detailed in the proofs of this lemma and Lemma 1 in reverse order. The resulting stabilizer has equivalent error-correcting properties to the quantum check matrix in (13).

The receiver decodes by first performing all of the finite-depth operations in the encoding circuit in reverse order. The receiver then decodes the infinite-depth operations by the procedure listed in the paragraph with (36) so that the original k_1+k_2-n+c information qubits per frame are available for processing at the receiving end. ■

A. Special case of entanglement-assisted codes with infinite-depth encoding circuits and finite-depth decoding circuits

We now detail a special case of the above codes in this final section. These codes are interesting because the information qubits teleport coherently to other physical qubits when encoding and decoding is complete.

Lemma 4. Suppose that the Smith form of $F(D)$ in (18) is

$$F(D) = A_F(D)\,\Gamma_F(D)\,0\,B_F(D),$$

where $A_F(D)$ is $(n-k_1)\times(n-k_1)$, $\Gamma_F(D)$ is an $(n-k_1)\times(n-k_1)$ diagonal matrix whose entries are powers of D, and $B_F(D)$ is $k_2\times k_2$. Then the resulting entanglement-assisted code admits an encoding circuit with both infinite-depth and finite-depth operations and admits a decoding circuit with finite-depth operations only. The information qubits also teleport coherently to other physical qubits for this special case of codes.

Proof. We perform all the operations in Lemma 1 to obtain the quantum check matrix in (18). Then perform the row operations in $A_F^{-1}(D)$ and the column operations in $B_F^{-1}(D)$. The quantum check matrix becomes

$$\left[\begin{array}{ccc|ccc} E'(D) & \Gamma_F(D) & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I & 0 & 0 \end{array}\right],$$

where $E'(D) = A_F^{-1}(D)\,E(D)$. The Smith form of $E'(D)$ is

$$E'(D) = A_{E'}(D)\begin{bmatrix} \Gamma_1(D) & 0 & 0 \\ 0 & \Gamma_2(D) & 0 \\ 0 & 0 & 0 \end{bmatrix} B_{E'}(D),$$

where $A_{E'}(D)$ is $(n-k_1)\times(n-k_1)$, $\Gamma_1(D)$ is an $s\times s$ diagonal matrix whose entries are powers of D, $\Gamma_2(D)$ is a $(c-s)\times(c-s)$ diagonal matrix whose entries are arbitrary polynomials, and $B_{E'}(D)$ is $(n-k_2)\times(n-k_2)$.

Now perform the row operations in $A_{E'}^{-1}(D)$ and the column operations in $B_{E'}^{-1}(D)$. It is possible to counteract the effect of the row operations on $\Gamma_F(D)$ by performing column operations, and it is possible to counteract the effect of the column operations on the identity matrix in the "X" matrix by performing row operations. The quantum check matrix becomes

$$\left[\begin{array}{ccccccc|ccc} \Gamma_1 & 0 & 0 & \Gamma_1' & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \Gamma_2 & 0 & 0 & \Gamma_2' & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \Gamma_3' & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & I & 0 & 0 \end{array}\right],$$

where Γ_1', Γ_2', and Γ_3' represent the respective $s\times s$, $(c-s)\times(c-s)$, and $(n-k_1-c)\times(n-k_1-c)$ diagonal matrices resulting from counteracting the effect of row operations $A_{E'}^{-1}(D)$ on $\Gamma_F(D)$. (We suppress the D argument in all of the matrices in the above equation.) We use Hadamard and C-NOT gates to clear the entries in $\Gamma_1'(D)$. The quantum check matrix becomes

$$\left[\begin{array}{ccccccc|ccc} \Gamma_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \Gamma_2 & 0 & 0 & \Gamma_2' & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \Gamma_3' & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & I & 0 & 0 \end{array}\right].$$

The first s rows with Γ_1 and rows $n-k_1-c+1,\ldots,n-k_1-c+s$ with the identity matrix on the "X" side stabilize a set of s ebits. The $n-k_1-c$ rows with Γ_3' and the $n-k_2-c$ rows with identity in the "X" matrix stabilize a set of $2(n-c)-k_1-k_2$ ancilla qubits (up to Hadamard gates). The s and k_2-n+k_1 columns with zeros in both the "Z" and "X" matrices correspond to information qubits. The decomposition of these rows is now complete.

We need to finish processing the $c-s$ rows with $\Gamma_2(D)$ and $\Gamma_2'(D)$ as entries and the $c-s$ rows of the identity in the "X" matrix. We construct a submatrix of the above quantum check matrix:

$$\left[\begin{array}{cc|cc} \Gamma_2(D) & \Gamma_2'(D) & 0 & 0 \\ 0 & 0 & I & 0 \end{array}\right]. \tag{37}$$

We describe a procedure to encode the above entries with $c-s$ ebits and $c-s$ information qubits using infinite-depth operations. Consider the following stabilizer matrix:

$$\left[\begin{array}{ccc|ccc} I & I & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I & I & 0 \end{array}\right], \tag{38}$$

where all identity and null matrices are $(c-s)\times(c-s)$. The above matrix stabilizes a set of $c-s$ ebits and $c-s$ information qubits. Bob's half of the ebits are the $c-s$ columns on the left in both the "Z" and "X" matrices and Alice's half are the next $c-s$ columns. We also track the logical operators for the information qubits to verify that the circuit encodes and decodes properly. The information-qubit matrix is as follows:

$$\left[\begin{array}{ccc|ccc} 0 & 0 & I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & I \end{array}\right], \tag{39}$$

where all matrices are again $(c-s)\times(c-s)$. Alice performs Hadamard gates on her first $c-s$ qubits and then performs

C-NOT gates from her first $c-s$ qubits to her last $c-s$ qubits to transform (38) to the following stabilizer:

$$\left[\begin{array}{ccc|ccc} I & 0 & 0 & 0 & I & \Gamma_2'(D) \\ 0 & I & 0 & I & 0 & 0 \end{array}\right].$$

The information-qubit matrix in (39) becomes

$$\left[\begin{array}{ccc|ccc} 0 & \Gamma_2'(D^{-1}) & I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & I \end{array}\right].$$

Alice then performs infinite-depth operations on her last $c-s$ qubits. These infinite-depth operations correspond to the elements of $\Gamma_2^{-1}(D)$. She finally performs Hadamard gates on her $2(c-s)$ qubits. The stabilizer becomes

$$\left[\begin{array}{ccc|ccc} I & I & \Gamma_2^{-1}(D)\Gamma_2'(D) & 0 & 0 & 0 \\ 0 & 0 & 0 & I & I & 0 \end{array}\right], \tag{40}$$

and the information-qubit matrix becomes

$$\left[\begin{array}{ccc|ccc} 0 & 0 & 0 & 0 & \Gamma_2'(D^{-1}) & \Gamma_2(D^{-1}) \\ 0 & 0 & \Gamma_2^{-1}(D) & 0 & 0 & 0 \end{array}\right]. \tag{41}$$

The stabilizer in (40) is equivalent to the following stabilizer by row operations [premultiplying the first $c-s$ rows by $\Gamma_2(D)$]:

$$\left[\begin{array}{ccc|ccc} \Gamma_2(D) & \Gamma_2(D) & \Gamma_2'(D) & 0 & 0 & 0 \\ 0 & 0 & 0 & I & I & 0 \end{array}\right]. \tag{42}$$

The measurements that Bob performs have finite weight because the row operations are multiplications of the rows by the arbitrary polynomials in $\Gamma_2(D)$. Alice thus encodes a code equivalent to the desired quantum check matrix in (37) using $c-s$ ebits and $c-s$ information qubits.

We now discuss decoding the stabilizer in (40) and information qubits. Bob performs C-NOTs from the first $c-s$ qubits to the next $c-s$ qubits. The stabilizer becomes

$$\left[\begin{array}{ccc|ccc} 0 & I & \Gamma_2^{-1}(D)\Gamma_2'(D) & 0 & 0 & 0 \\ 0 & 0 & 0 & I & 0 & 0 \end{array}\right], \tag{43}$$

and the information-qubit matrix does not change. Bob uses Hadamard and finite-depth C-NOT gates to multiply the last $c-s$ columns in the "Z" matrix by $\Gamma_2'(D^{-1})\Gamma_2(D)$ and add the result to the middle $c-s$ columns. It is possible to use finite-depth operations because the entries of $\Gamma_2'(D)$ are all powers of D so that $\Gamma_2'(D^{-1}) = \Gamma_2'^{-1}(D)$. The stabilizer in (43) becomes

$$\left[\begin{array}{ccc|ccc} 0 & 0 & \Gamma_2^{-1}(D)\Gamma_2'(D) & 0 & 0 & 0 \\ 0 & 0 & 0 & I & 0 & 0 \end{array}\right],$$

and the information-qubit matrix in (41) becomes

$$\left[\begin{array}{ccc|ccc} 0 & 0 & 0 & 0 & \Gamma_2'(D^{-1}) & 0 \\ 0 & \Gamma_2'(D^{-1}) & \Gamma_2^{-1}(D) & 0 & 0 & 0 \end{array}\right].$$

We premultiply the first $c-s$ rows of the stabilizer by $\Gamma_2'(D^{-1})$ and add the result to the second $c-s$ rows of the information-qubit matrix. These row operations from the stabilizer to the information-qubit matrix result in the information-qubit matrix having pure logical operators for the middle $c-s$ qubits. Perform Hadamard gates on the second set of $c-s$ qubits. The resulting information-qubit matrix is as follows:

$$\left[\begin{array}{ccc|ccc} 0 & \Gamma_2'(D^{-1}) & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \Gamma_2'(D^{-1}) & 0 \end{array}\right], \tag{44}$$

so that the information qubits are available at the end of decoding. Processing may delay or advance them with respect to their initial locations because the matrix $\Gamma_2'(D^{-1})$ is diagonal with powers of D. We can determine that the information qubits teleport coherently from the last set of $c-s$ qubits to the second set of $c-s$ qubits in every frame by comparing (44) to (39).

The overall procedure for encoding is to begin with a set of c ebits, $2(n-c)-k_1-k_2$ ancilla qubits, and k_1+k_2-n+c information qubits. We perform the infinite-depth operations detailed in (37)–(42) for $c-s$ of the ebits. We then perform the finite-depth operations detailed in the proofs of this lemma and Lemma 1 in reverse order. The resulting stabilizer has equivalent error-correcting properties to the quantum check matrix in (13).

The receiver decodes by first performing all of the finite-depth operations in reverse order. The receiver then decodes the infinite-depth operations by the procedure listed in (43) and (44) so that the original k_1+k_2-n+c information qubits per frame are available for processing at the receiving end. ■

Example 2. Consider a classical convolutional code with the following check matrix:

$$H(D) = [1\ 1+D].$$

We can use the above check matrix in an entanglement-assisted quantum convolutional code to correct for both bit flips and phase flips. We form the following quantum check matrix:

$$\left[\begin{array}{cc|cc} 1 & 1+D & 0 & 0 \\ 0 & 0 & 1 & 1+D \end{array}\right]. \tag{45}$$

We first perform some manipulations to put the above quantum check matrix into a standard form. Perform a C-NOT from qubit one to qubit two in the same frame and in the next frame. The above matrix becomes

$$\left[\begin{array}{cc|cc} D^{-1}+1+D & 1+D & 0 & 0 \\ 0 & 0 & 1 & 0 \end{array}\right].$$

Perform a Hadamard gate on qubits one and two. The matrix becomes

$$\left[\begin{array}{cc|cc} 0 & 0 & D^{-1}+1+D & 1+D \\ 1 & 0 & 0 & 0 \end{array}\right].$$

Perform a C-NOT from qubit one to qubit two. The matrix becomes

$$\left[\begin{array}{cc|cc} 0 & 0 & D^{-1}+1+D & D^{-1} \\ 1 & 0 & 0 & 0 \end{array}\right].$$

Perform a row operation that delays the first row by D. Perform a Hadamard on both qubits. The stabilizer becomes

$$\left[\begin{array}{cc|cc} 1+D+D^2 & 1 & 0 & 0 \\ \text{`0} & 0 & 1 & 0 \end{array}\right].$$

The above matrix is now in standard form. The matrix $F(D) = 1$ as in (18) so that its only invariant factor is equal to one. The code falls into the second class of entanglement-assisted quantum convolutional codes. We begin encoding with one ebit and one information qubit per frame. The stabilizer matrix for the unencoded stream is as follows:

$$\left[\begin{array}{ccc|ccc} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{array}\right],$$

and the information-qubit matrix is as follows:

$$\left[\begin{array}{ccc|ccc} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{array}\right].$$

Perform a Hadamard on qubit two and a C-NOT from qubit two to qubit three so that the above stabilizer becomes

$$\left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \end{array}\right],$$

and the information-qubit matrix becomes

$$\left[\begin{array}{ccc|ccc} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 \end{array}\right].$$

Perform an infinite-depth operation corresponding to the rational polynomial $1/\left(1+D+D^2\right)$ on qubit three. Follow with a Hadamard gate on qubits two and three. The stabilizer matrix becomes

$$\left[\begin{array}{ccc|ccc} 1 & 1 & 1/\left(1+D+D^2\right) & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{array}\right], \tag{46}$$

and the information-qubit matrix becomes

$$\left[\begin{array}{ccc|ccc} 0 & 0 & 1/\left(1+D+D^2\right) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1+D^{-1}+D^{-2} \end{array}\right]. \tag{47}$$

Perform the finite-depth operations above in reverse order so that the stabilizer becomes

$$\left[\begin{array}{ccc|ccc} D^{-1} & \frac{1}{1+D+D^2} & \frac{1+D}{1+D+D^2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1+D \end{array}\right],$$

and the information-qubit matrix becomes

$$\left[\begin{array}{ccc|ccc} 0 & \frac{D^{-1}+D^{-2}}{1+D+D^2} & \frac{1}{1+D+D^2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & D^{-1}+D^{-2} & D^{-1} \end{array}\right].$$

The above stabilizer is equivalent to the desired quantum check matrix in (45) by a row operation that multiplies its first row by $1+D+D^2$. The receiver decodes by performing the finite-depth encoding operations in reverse order and gets the stabilizer in (46) and the information-qubit matrix in (47). The receiver performs a C-NOT from qubit one to qubit two and follows with a C-NOT from qubit two to qubit three in the same frame, in an advanced frame, and in a twice-advanced frame. Finally, perform a Hadamard gate on qubits two and three. The stabilizer becomes

$$\left[\begin{array}{ccc|ccc} 0 & 0 & 0 & 0 & 0 & 1/\left(1+D+D^2\right) \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right],$$

and the information-qubit matrix becomes

$$\left[\begin{array}{ccc|ccc} 0 & 0 & 0 & 0 & 1 & 1/\left(1+D+D^2\right) \\ 0 & 1 & 0 & 0 & 0 & 0 \end{array}\right].$$

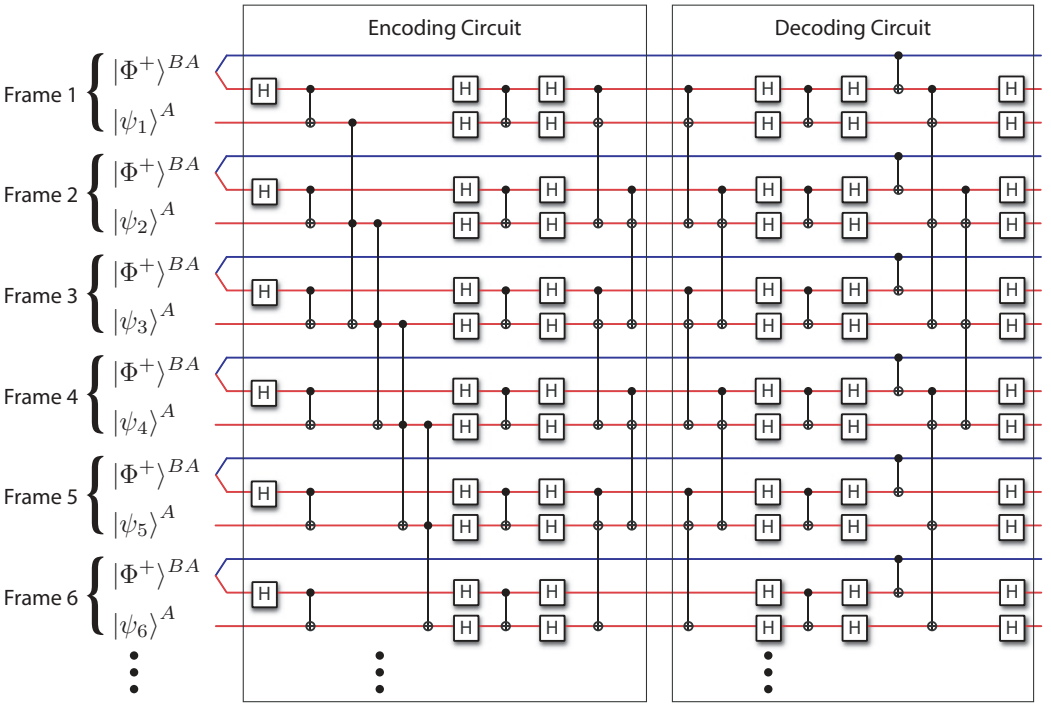

FIG. 7. (Color online) The encoding and decoding circuits for the entanglement-assisted quantum convolutional code in Example 2. The third series of gates in the above encoding circuit is an infinite-depth operation. The other operations in the encoding circuit are finite-depth operations. The decoding circuit has finite-depth operations only.

The receiver decodes the information qubits successfully because a row operation from the first row of the stabilizer to the first row of the information-qubit matrix gives the proper logical operators for the information qubits. Figure 7 details the above encoding and decoding operations for this entanglement-assisted quantum convolutional code.

B. Discussion

This second class of codes assumes that noiseless encoding is available. We require this assumption because the encoding circuit employs infinite-depth encoding operations.

If an error does occur during the encoding process, it can propagate infinitely through the encoded qubit stream. The result of a single encoding error can distort both the encoded quantum information, the syndromes that result from measurements, and the final recovery operations based on the syndromes.

We may be able to relax the noiseless encoding assumption if nearly noiseless encoding is available. The probability of an error would have to be negligible in order to ensure that the probability for a catastrophic failure is negligible. One way to lower the probability of an encoding error is to encode first with a quantum block code and then further encode with our quantum convolutional coding method. Many classical coding systems exploit this technique, the most popular of which is a Reed-Solomon encoder followed by a convolutional encoder.

VIII. CONCLUSION AND CURRENT WORK

This work develops the theory of entanglement-assisted quantum convolutional coding. We show several methods for importing two arbitrary classical binary convolutional codes for use in an entanglement-assisted quantum convolutional code. Our methods outline different ways for encoding and decoding our entanglement-assisted quantum convolutional codes.

We introduce the notion of an infinite-depth operation for encoding circuits. We use these infinite-depth operations in both encoding and decoding. These operations are acceptable if we assume that noiseless processing is available both at the sender's end and on the receiver's half of shared ebits.

Our first class of codes employs only finite-depth operations in their encoding and decoding procedures. These codes are the most useful in practice because they do not have the risk of catastrophic error propagation. An error that occurs during encoding, measurement, recovery, or decoding propagates only to a finite number of neighboring qubits.

Our second class of codes uses infinite-depth operations during encoding. This assumption is reasonable only if noiseless encoding is available. The method of concatenated coding is one way to approach nearly noiseless encoding in practice.

We suggest several lines of inquiry from here. Our codes are not only useful for quantum communication, but should also be useful for private classical communication because of the well-known connection between a quantum channel and private classical channel [25]. It may make sense from a practical standpoint to begin investigating the performance of our codes for encoding secret classical messages. The commercial success of quantum key distribution for the generation of a private shared secret key motivates this investigation. It is also interesting to determine which entanglement-assisted codes can correct for errors on the receiver's side. Codes that possess this property will be more useful in practice.

ACKNOWLEDGMENTS

The authors thank Hari Krovi and Markus Grassl for useful discussions. They thank Shesha Raghunathan and Markus Grassl for useful comments on the manuscript. M.M.W. acknowledges support from National Science Foundation (NSF) Grant No. CCF-0545845, and T.A.B. acknowledges support from NSF Grants No. CCF-0448658 and No. CCF-0830801.

[1] P. W. Shor, Phys. Rev. A **5**2, R2493 (1995).
[2] A. M. Steane, Phys. Rev. Lett. **77**, 793 (1996).
[3] A. R. Calderbank and P. W. Shor, Phys. Rev. A **54**, 1098 (1996).
[4] D. Gottesman, Ph.D. dissertation, California Institute of Technology, 1997.
[5] A. R. Calderbank, E. M. Rains, P. W. Shor, and N. J. A. Sloane, Phys. Rev. Lett. **78**, 405 (1997).
[6] A. Calderbank, E. Rains, P. Shor, and N. Sloane, IEEE Trans. Inf. Theory **44**, 1369 (1998).
[7] P. Zanardi and M. Rasetti, Phys. Rev. Lett. **79**, 3306 (1997).
[8] P. Zanardi and M. Rasetti, Mod. Phys. Lett. B **11**, 1085 (1997).
[9] D. A. Lidar, I. L. Chuang, and K. B. Whaley, Phys. Rev. Lett. **81**, 2594 (1998).
[10] D. Kribs, R. Laflamme, and D. Poulin, Phys. Rev. Lett. **94**, 180501 (2005).
[11] D. W. Kribs, R. Laflamme, D. Poulin, and M. Lesosky, Quantum Inf. Comput. **6**, 383 (2006).
[12] D. Poulin, Phys. Rev. Lett. **95**, 230504 (2005).
[13] T. Brun, I. Devetak, and M.-H. Hsieh, in *IEEE International Symposium on Information Theory, 2007* [online].
[14] M.-H. Hsieh, I. Devetak, and T. A. Brun, Phys. Rev. A **76**, 062313 (2007).
[15] N. D. Mermin, *Quantum Computer Science* (Cambridge University Press, Cambridge, 2007).
[16] W. K. Wootters and W. H. Zurek, Nature **299**, 802 (1982).
[17] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes* (North Holland, Amsterdam, 1983).
[18] G. Bowen, Phys. Rev. A **66**, 052313 (2002).
[19] C. H. Bennett, G. Brassard, C. Crépeau, R. Jozsa, A. Peres, and W. K. Wootters, Phys. Rev. Lett. **70**, 1895 (1993).
[20] C. H. Bennett and S. J. Wiesner, Phys. Rev. Lett. **69**, 2881 (1992).
[21] T. A. Brun, I. Devetak, and M.-H. Hsieh, e-print arXiv:quant-ph/0608027 (2006).

[22] T. A. Brun, I. Devetak, and M.-H. Hsieh, Science **314**, 436 (2006).
[23] S. Lloyd, Phys. Rev. A **55**, 1613 (1997).
[24] P. W. Shor, in *Lecture Notes, MSRI Workshop on Quantum Computation*, 2002 [online].
[25] I. Devetak, IEEE Trans. Inf. Theory **51**, 44 (2005).
[26] C. H. Bennett, P. W. Shor, J. A. Smolin, and A. V. Thapliyal, Phys. Rev. Lett. **83**, 3081 (1999).
[27] C. H. Bennett, P. W. Shor, J. A. Smolin, and A. V. Thapliyal, IEEE Trans. Inf. Theory **48**, 2637 (2002).
[28] T. M. Cover and J. A. Thomas, *Elements of Information Theory* (Wiley-Interscience, Malden, 1991).
[29] H. Ollivier and J.-P. Tillich, Phys. Rev. Lett. **91**, 177902 (2003).
[30] H. Ollivier and J.-P. Tillich, e-print arXiv:quant-ph/0401134 (2004).
[31] M. Grassl and M. Rötteler, in *IEEE International Symposium on Information Theory*, e-print arXiv:quant-ph/0602129 (2006).
[32] M. Grassl and M. Rötteler, in *Proceedings 44th Allerton Conference on Communication, Control, and Computing*, pp. 510–519, 2006 [online].
[33] M. Grassl and M. Rötteler, in *IEEE Int. Symposium on Information Theory*, 2007, e-print arXiv:quant-ph/0703182.
[34] G. D. Forney and S. Guha, in *IEEE International Symposium on Information Theory*, 2005, e-print arXiv:quant-ph/0501099 (2005).
[35] G. D. Forney, M. Grassl, and S. Guha, IEEE Trans. Inf. Theory **53**, 865 (2007).
[36] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, 2000).
[37] D. Gottesman, Phys. Rev. A **54**, 1862 (1996).
[38] M. M. Wilde, H. Krovi, and T. A. Brun, e-print arXiv:0708.3699 (2007).
[39] Z. Luo and I. Devetak, Phys. Rev. A **75**, 010303(R) (2007).
[40] M. M. Wilde, H. Krovi, and T. A. Brun, Phys. Rev. A **76**, 052308 (2007).
[41] M. M. Wilde and T. A. Brun, Quant. Info. Proc. **8**, 401 (2009).
[42] I. Devetak, A. W. Harrow, and A. Winter, IEEE Trans. Inf. Th. **54**, 4587 (2008).
[43] A. J. Viterbi, IEEE Trans. Inf. Theory **13**, 260 (1967).
[44] R. Johannesson and K. S. Zigangirov, *Fundamentals of Convolutional Coding* (Wiley IEEE Press, New York, NY, 1999).
[45] A. Harrow, Phys. Rev. Lett. **92**, 097902 (2004).
[46] M. M. Wilde, H. Krovi, and T. A. Brun, Phys. Rev. A **75**, 060303(R) (2007).