Extra shared entanglement reduces memory demand in quantum convolutional coding

Mark M. Wilde[1,2] and Todd A. Brun[2]

[1]*Centre for Quantum Technologies, National University of Singapore, 3 Science Drive 2, Singapore 117543, Singapore*

[2]*Department of Electrical Engineering, Center for Quantum Information Science and Technology, Communication Sciences Institute,
University of Southern California, Los Angeles, California 90089, USA*

We show how extra entanglement shared between sender and receiver reduces the memory requirements for a general entanglement-assisted quantum convolutional code. We construct quantum convolutional codes with good error-correcting properties by exploiting the error-correcting properties of an arbitrary basic set of Pauli generators. The main benefit of this particular construction is that there is no need to increase the frame size of the code when extra shared entanglement is available. Then there is no need to increase the memory requirements or circuit complexity of the code because the frame size of the code is directly related to these two code properties. Another benefit, similar to results of previous work in entanglement-assisted convolutional coding, is that we can import an arbitrary classical quaternary code for use as an entanglement-assisted quantum convolutional code. The rate and error-correcting properties of the imported classical code translate to the quantum code. We provide an example that illustrates how to import a classical quaternary code for use as an entanglement-assisted quantum convolutional code. We finally show how to "piggyback" classical information to make use of the extra shared entanglement in the code.

PACS number(s): 03.67.Hk, 03.67.Pp

I. INTRODUCTION

Quantum convolutional coding is a useful technique for encoding a stream of quantum information before sending it over a noisy quantum communication channel [1–9]. Ollivier and Tillich [1,2] started the investigations into the theory of quantum convolutional coding and many authors have since contributed to a general theory of quantum convolutional coding [3–9].

Quantum convolutional codes have several key benefits. The tradeoff between performance and complexity for a quantum convolutional code is superior to that of a block code that encodes the same number of information qubits [7]. Some benefits of their classical counterparts [10] carry over to quantum convolutional codes—e.g., the encoding circuits for both quantum and classical convolutional codes have a periodic structure.

Entanglement-assisted quantum error-correcting codes exploit entanglement shared between sender and receiver [11]. The entanglement-assisted communication paradigm assumes that a sender and receiver share a set of noiseless ebits; the sender can use her half of these ebits in the encoding procedure and the receiver's half of the shared ebits remain noiseless throughout communication. The benefit of entanglement-assisted stabilizer codes is that the sender and receiver can exploit the error-correcting properties of an arbitrary set of Pauli generators rather than restrict themselves to a commuting set. Another benefit is that entanglement boosts the quantum communication rate of the entanglement-assisted quantum code constructed from either two classical binary codes or one classical quaternary code.

We have recently constructed entanglement-assisted quantum convolutional codes that admit a Calderbank-Shor-Steane (CSS) structure [12] and others that admit a more general structure [13]. These papers developed several ways of encoding and decoding entanglement-assisted quantum convolutional codes with operations that are both finite depth and infinite depth. The upshot of these constructions is that we can import two arbitrary classical binary convolutional codes for use as a CSS entanglement-assisted quantum convolutional code [12] or we can import an arbitrary classical quaternary convolutional code for use as an entanglement-assisted quantum convolutional code [13]. The rates and error-correcting properties of the classical codes translate directly to the quantum code.

In Ref. [13], we showed how to accommodate entangled bits in an entanglement-assisted quantum convolutional code by increasing the code's frame size. (A quantum convolutional code partitions quantum data into uniformly-sized *frames* consisting of ebits, ancilla qubits, and information qubits that a periodic encoding circuit acts on.) Increasing the frame size of the code is undesirable because it increases the complexity of the encoding and decoding circuits and demands a larger memory for the code.

In this paper, we show that there is no need to increase the frame size of the entanglement-assisted quantum convolutional code if one makes use of extra shared entanglement. The result in Ref. [13] and our findings in this paper seem to point to a tradeoff between efficient use of entanglement and that of circuit resources for non-CSS entanglement-assisted quantum convolutional codes. We again focus on more general entanglement-assisted quantum convolutional codes that do not have the CSS structure. The outcome of this work is that we can import an arbitrary classical quaternary convolutional code for use as an entanglement-assisted quantum convolutional code with the benefit that it is not necessary to increase the frame size of the code. The drawback of the technique in this paper is that it does not make the most efficient use of entanglement.

II. MAIN THEOREM

The proof of our main theorem below outlines how to encode a stream of information qubits, ancilla qubits, and

shared ebits so that the encoded qubits have the error-correcting properties of an arbitrary set of Paulis. The receiver may employ an error estimation algorithm such as Viterbi decoding [14] to determine the most likely errors that the noisy quantum communication channel induces on the encoded stream. We then show how to decode the encoded qubit stream so that the information qubits become available at the receiving end of the channel.

The encoding circuits in the proof of our theorem employ both finite-depth and infinite-depth operations. The decoding circuits employ finite-depth operations only. Finite-depth operations take a finite-weight stabilizer to one with finite weight. Infinite-depth operations take *some* finite-weight stabilizer generators to ones with infinite weight.

Infinite-depth operations can lead to catastrophic error propagation [1–3,12]. In our proof below, we restrict infinite-depth operations to act on qubits before sending them over the noisy channel. Catastrophic error propagation does not occur under the ideal circumstance when the operations in the encoding circuit are noiseless.

Theorem 1 below begins with a "quantum check matrix" that consists of a set of Pauli sequences with desirable error-correcting properties [12]. This quantum check matrix represents an arbitrary basic set of Pauli generators and thus does not necessarily correspond to a commuting stabilizer. The proof of the theorem shows how to incorporate ebits so that the sender realizes the same quantum check matrix for her qubits and the sender and receiver's set of generators form a valid commuting stabilizer.

The entries of the quantum check matrix in Theorem 1 are binary polynomials. The usual way to represent a quantum code is with a tensor product of Pauli matrices, but it is more convenient for our purposes to perform all manipulations with matrices of binary polynomials rather than with tensor products of Pauli matrices (see Refs. [2–4,7,12,15] for details of this isomorphism). Consider the quantum check matrix in Theorem 1. The matrix on the left (right) side of the vertical bar includes binary polynomials corresponding to Z (X) Pauli operators. Also, the delay operator D gives a simple way of obtaining all the generators that the quantum check matrix represents. Multiplication of any of the rows in the matrix by any power of D gives a corresponding generator.

We perform two types of manipulations on the below quantum check matrix: row operations and column operations. Row operations do not change the error-correcting properties of the code. Column operations correspond to quantum circuit elements in the shift-invariant Clifford group [3,4] and do not change the shifted symplectic product [12,15] between the rows of the quantum check matrix. Both row and column operations are crucial for arriving at the proper decomposition of the below quantum check matrix.

Theorem 1. Suppose we would like to exploit the error-correcting properties of the generators in the following quantum check matrix:

$$S(D) = [Z(D)|X(D)] \in \mathbb{F}_2[D]^{(n-k)\times 2n},$$

where $S(D)$ is of full rank and does not necessarily form a commuting stabilizer. Then an $[[n,k;c]]$ entanglement-

assisted quantum convolutional code exists that has the same error-correcting properties as the above quantum check matrix $S(D)$ where $c=\text{rank}[X(D)]$ (an $[[n,k;c]]$ code is one that encodes k information qubits per frame into n "channel" qubits per frame while consuming c ebits per frame [11–13]).

Proof. The first part of the proof in Eqs. (1)–(13) involves decomposing the above check matrix using controlled NOT (CNOT) and SWAP gates chosen according to the Smith algorithm [10] (it employs intermediate Hadamard gates as well). The resulting check matrix in Eq. (13) is then no longer decomposable using finite-depth operations only. We then start with a matrix representing unencoded ebits and show how to perform infinite-depth encoding operations in order to obtain part of the check matrix in Eq. (13). The last part of the proof shows how to decode the information qubits properly using finite-depth operations only. The order of the steps in this proof is similar to the order of the steps in Refs. [12,13], but the details of the proof are different. The details of performing column operations with CNOT and Hadamard gates, and infinite-depth operations corresponding to multiplying by rational polynomials, are also in those papers.

Suppose the Smith form [10] of $X(D)$ is as follows:

$$X(D) = A(D)\begin{bmatrix} \Gamma_1(D) & 0 & 0 \\ 0 & \Gamma_2(D) & 0 \\ 0 & 0 & 0 \end{bmatrix}B(D), \quad (1)$$

where $A(D)$ is $(n-k)\times(n-k)$, $B(D)$ is $n\times n$, $\Gamma_1(D)$ is an $s\times s$ diagonal matrix whose entries are powers of D, and $\Gamma_2(D)$ is a $(c-s)\times(c-s)$ diagonal matrix whose entries are arbitrary polynomials. Perform the row operations in $A^{-1}(D)$ and the column operations in $B^{-1}(D)$ on $S(D)$. The quantum check matrix $S(D)$ becomes

$$\begin{bmatrix} E(D) & \begin{matrix} \Gamma_1(D) & 0 & 0 \\ 0 & \Gamma_2(D) & 0 \\ 0 & 0 & 0 \end{matrix} \end{bmatrix}, \quad (2)$$

where $E(D)=A^{-1}(D)Z(D)B^T(D^{-1})$. Suppose $E_1(D)$ is the first c columns of $E(D)$ and $E_2(D)$ is the next $n-c$ columns of $E(D)$ so that the quantum check matrix is as follows:

$$\begin{bmatrix} E_1(D) & E_2(D) & \begin{matrix} \Gamma_1(D) & 0 & 0 \\ 0 & \Gamma_2(D) & 0 \\ 0 & 0 & 0 \end{matrix} \end{bmatrix}. \quad (3)$$

Perform Hadamard gates on the last $n-c$ qubits so that the quantum check matrix becomes

$$\begin{bmatrix} E_1(D) & 0 & \begin{matrix} \Gamma_1(D) & 0 & E_{2,1}(D) \\ 0 & \Gamma_2(D) & E_{2,2}(D) \\ 0 & 0 & E_{2,3}(D) \end{matrix} \end{bmatrix}, \quad (4)$$

where

$$E_2(D) = \begin{bmatrix} E_{2,1}(D) \\ E_{2,2}(D) \\ E_{2,3}(D) \end{bmatrix}. \quad (5)$$

Perform CNOT operations from the first s qubits to the last $n-c$ qubits to clear the entries in $E_{2,1}(D)$. The quantum check matrix becomes

$$\begin{bmatrix} & & \Gamma_1(D) & 0 & 0 \\ E_1(D) & 0 & 0 & \Gamma_2(D) & E_{2,2}(D) \\ & & 0 & 0 & E_{2,3}(D) \end{bmatrix}. \qquad (6)$$

The Smith form of $E_{2,3}(D)$ is as follows:

$$E_{2,3}(D) = A_E(D)[\Gamma(D) \quad 0]B_E(D), \qquad (7)$$

where $A_E(D)$ is $(n-k-c)\times(n-k-c)$, $B_E(D)$ is $(n-c)\times(n-c)$, and $\Gamma(D)$ is an $(n-k-c)\times(n-k-c)$ diagonal matrix whose entries are polynomials. The Smith form of $E_{2,3}(D)$ is full rank because the original quantum check matrix $S(D)$ is full rank. Perform the row operations in $A_E^{-1}(D)$ and the column operations in $B_E^{-1}(D)$. The quantum check matrix becomes

$$\begin{bmatrix} & & \Gamma_1(D) & 0 & 0 & 0 \\ E_1'(D) & 0 & 0 & \Gamma_2(D) & E_{2,2a}'(D) & E_{2,2b}'(D) \\ & & 0 & 0 & \Gamma(D) & 0 \end{bmatrix}, \quad (8)$$

where

$$E_1'(D) = \begin{bmatrix} I & 0 \\ 0 & A_E^{-1}(D) \end{bmatrix} E_1(D), \qquad (9)$$

$$E_{2,2}'(D) = E_{2,2}(D)B_E^{-1}(D) \qquad (10)$$

$$= [E_{2,2a}'(D) \quad E_{2,2b}'(D)]. \qquad (11)$$

Perform a modified version of the Smith algorithm to reduce the $(c-s)\times(n-c)$ matrix $E_{2,2b}'(D)$ to a lower triangular form [12]. This modified algorithm uses only column operations to transform

$$E_{2,2b}'(D) \rightarrow [L(D) \quad 0], \qquad (12)$$

where $L(D)$ is $(c-s)\times(c-s)$ and the null matrix is $(c-s)\times(n+s-2c)$. The quantum check matrix becomes

$$\begin{bmatrix} & & \Gamma_1(D) & 0 & 0 & 0 & 0 \\ E_1'(D) & 0 & 0 & \Gamma_2(D) & E_{2,2a}'(D) & L(D) & 0 \\ & & 0 & 0 & \Gamma(D) & 0 & 0 \end{bmatrix}. \quad (13)$$

We have now completed the decomposition of the quantum check matrix with column and row operations.

We now showing how to encode a certain quantum check matrix that proves to be useful in encoding the above quantum check matrix. Consider the following quantum check matrix:

$$\begin{bmatrix} I & 0 & 0 & 0 \\ 0 & 0 & \Gamma_2(D) & L(D) \end{bmatrix}, \qquad (14)$$

where $\Gamma_2(D)$ and $L(D)$ are from the matrix in Eq. (13) and each of them, the identity matrix and the null matrices, has dimension $(c-s)\times(c-s)$.

We use a method for encoding the quantum check matrix in Eq. (14) similar to the method outlined in Ref. [12] for the

second class of CSS entanglement-assisted quantum convolutional codes. We begin with a set of $c-s$ ebits and $c-s$ information qubits. The following matrix stabilizes the ebits:

$$\begin{bmatrix} I & I & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I & I & 0 \end{bmatrix}, \qquad (15)$$

where Bob has the $c-s$ qubits in the leftmost column of each submatrix, Alice has the $2(c-s)$ qubits in the right two columns, and each block is $(c-s)\times(c-s)$. The following matrix represents the logical operators for the information qubits and gives a useful way of tracking the information qubits while processing them:

$$\begin{bmatrix} 0 & 0 & I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & I \end{bmatrix}. \qquad (16)$$

Tracking the information-qubit matrix helps to confirm that the information qubits decode properly at the receiver's end [12].

We now track both the above stabilizer and the information-qubit matrix as they progress through some encoding operations. Alice performs CNOT gates from her first $c-s$ qubits to her next $c-s$ qubits. These gates multiply the middle $c-s$ columns of the X matrix by $L(D)$ and add the result to the last $c-s$ columns, and multiply the last $c-s$ columns of the Z matrix by $L^T(D^{-1})$ and add the result to the last $c-s$ columns. The stabilizer becomes

$$\begin{bmatrix} I & I & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I & I & L(D) \end{bmatrix}, \qquad (17)$$

and the information-qubit matrix becomes

$$\begin{bmatrix} 0 & L^T(D^{-1}) & I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & I \end{bmatrix}. \qquad (18)$$

Alice performs infinite-depth operations on her first $c-s$ qubits corresponding to the rational polynomials $\gamma_{2,1}^{-1}(D^{-1})$, ..., $\gamma_{2,c-s}^{-1}(D^{-1})$ in $\Gamma_2^{-1}(D^{-1})$. These operations multiply the middle $c-s$ columns of the Z matrix by $\Gamma_2^{-1}(D^{-1})$ and multiply the middle $c-s$ columns of the X matrix by $\Gamma_2(D)$. The stabilizer matrix becomes

$$\begin{bmatrix} I & \Gamma_2^{-1}(D^{-1}) & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I & \Gamma_2(D) & L(D) \end{bmatrix}, \qquad (19)$$

and the information-qubit matrix becomes

$$\begin{bmatrix} 0 & L^T(D^{-1})\Gamma_2^{-1}(D^{-1}) & I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & I \end{bmatrix}. \qquad (20)$$

Alice's part of the above stabilizer matrix is equivalent to the quantum check matrix in Eq. (14) by row operations [premultiplying the first set of rows by $\Gamma_2(D^{-1})$].

We now show how to encode the quantum check matrix in Eq. (13) using ebits, ancilla qubits, and information qubits. We employ the encoding technique for the submatrix listed above in Eqs. (15)–(20) and use some other techniques as well. Suppose that we have the following matrix that stabilizes a set of c ebits per frame, $n-k-c$ ancilla qubits per frame, and k information qubits per frame:

$$\begin{bmatrix} I & 0 & I & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & I & 0 & I & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & I & 0 & I & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & I & 0 & I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & I & 0 & 0 \end{bmatrix}. \quad (21)$$

The first and third sets of rows have s rows and correspond to s ebits per frame, the second and fourth sets of rows have $c-s$ rows and correspond to $c-s$ ebits per frame, and the last set of $n-k-c$ rows corresponds to $n-k-c$ ancilla qubits per frame. The above matrix has $n+c$ columns on both the Z and X side so it stabilizes k information qubits per frame. Bob possesses the first c qubits and Alice possesses the next n qubits. Alice performs the encoding operations in Eqs. (15)–(20) to get the following stabilizer:

$$\begin{bmatrix} I & 0 & I & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & I & 0 & G(D) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & I & 0 & I & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & I & 0 & \Gamma_2(D) & 0 & L(D) & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & I & 0 & 0 \end{bmatrix} \quad (22)$$

where $G(D) \equiv \Gamma_2^{-1}(D^{-1})$.

We perform several row operations to get the quantum check matrix in Eq. (13). Premultiply the middle set of rows by $\Gamma_1(D)$ (one actually does not have to do this row operation—the result is a "subcode" of the original code [3]). Premultiply the last set of rows by $E'_{2,2a}(D)$ and add the result to the set of rows above the last set. Premultiply the last set of rows by $\Gamma(D)$ (one also does not have to do this row operation and the result is again a subcode). Finally, premultiply the first two sets of rows by

$$E''_1(D) \equiv E'_1(D)[I \oplus \Gamma_2(D^{-1})],$$

and add the result to the last three sets of rows. The Z side of the quantum check matrix becomes

$$\begin{bmatrix} I & 0 & I & 0 & 0 & 0 & 0 \\ 0 & I & 0 & \Gamma_2^{-1}(D^{-1}) & 0 & 0 & 0 \\ & & & & 0 & 0 & 0 \\ E''_1(D) & & E'_1(D) & & 0 & 0 & 0 \\ & & & & 0 & 0 & 0 \end{bmatrix}, \quad (23)$$

and the X side becomes

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \Gamma_1(D) & 0 & \Gamma_1(D) & 0 & 0 & 0 & 0 \\ 0 & \Gamma_2(D) & 0 & \Gamma_2(D) & E'_{2,2a}(D) & L(D) & 0 \\ 0 & 0 & 0 & 0 & \Gamma(D) & 0 & 0 \end{bmatrix}. \quad (24)$$

The last three rows of Alice's part of the above quantum check matrix are equivalent to the quantum check matrix in

Eq. (13). The first two sets of rows represent extra shared entanglement that Alice and Bob use to resolve the anticommutativity present in the original set of generators. Alice performs all finite-depth encoding operations (column operations) in Eqs. (1)–(13) in reverse order to obtain the desired quantum check matrix in the statement of the theorem.

Decoding first consists of performing all the operations in Eqs. (1)–(13). We now illustrate a way to decode the stabilizer in Eq. (19) and information-qubit matrix in Eq. (20) so that the information qubits appear at the output of the decoding circuit. Bob performs CNOT gates from the first set of qubits to the third set of qubits corresponding to the entries in $L(D)$. The stabilizer becomes

$$\begin{bmatrix} I & \Gamma_2^{-1}(D^{-1}) & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I & \Gamma_2(D) & 0 \end{bmatrix}, \quad (25)$$

and the information-qubit matrix becomes

$$\begin{bmatrix} L^T(D^{-1}) & L^T(D^{-1})\Gamma_2^{-1}(D^{-1}) & I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & I \end{bmatrix}. \quad (26)$$

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

$$\begin{bmatrix} 0 & 0 & I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & I \end{bmatrix}, \quad (27)$$

by multiplying the first $c-s$ rows of the stabilizer by $L^T(D^{-1})$ and adding to the first $c-s$ rows of the information-qubit matrix. The information qubits are available at the receiving end of the channel because the above information-qubit matrix is equivalent to the original one in Eq. (16).

The above code is an $[[n,k;c]]$ entanglement-assisted code because the code uses a noisy quantum communication channel n times per frame to send k information qubits per frame. The parameter $c = \text{rank}[X(D)]$ because the matrices $\Gamma_1(D)$ and $\Gamma_2(D)$ determine the rank of the matrix $X(D)$ by the Smith algorithm [10]. ∎

III. EXAMPLE

We present an example of a classical quaternary code over $GF(4)$ that we import for use as an entanglement-assisted quantum convolutional code [15];

$$(\cdots|0000|1\bar{\omega}10|1101|0000|\cdots). \quad (28)$$

The above code is a convolutional version of the classical quaternary block code from Ref. [11]. We multiply the above generator by $\bar{\omega}$ and ω as prescribed in Refs. [7,16] and use the following map:

$$0 \to I, \quad \omega \to X, \quad 1 \to Y, \quad \bar{\omega} \to Z, \quad (29)$$

to obtain the following two Pauli generators:

$$(\cdots|IIII|ZXZI|ZZIZ|IIII|\cdots),$$

$$(\cdots|IIII|XYXI|XXIX|IIII|\cdots). \quad (30)$$

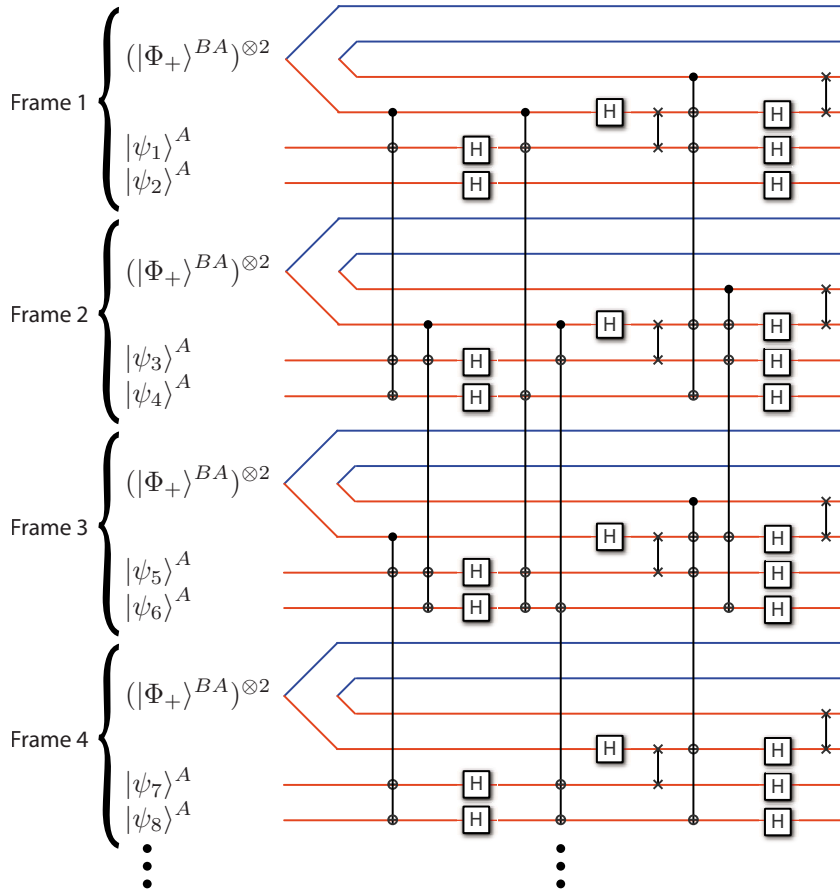We write the above two generators as a quantum check matrix;

FIG. 1. (Color online) An online encoding circuit for an entanglement-assisted quantum convolutional code. The receiver Bob possesses the first two qubits in the two ebits and the sender Alice possesses the second two qubits in the two ebits. The sender encodes two information qubits per frame with the help of her half of the two ebits.

$$\begin{bmatrix} 1+D & D & 1 & D & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1+D & 1+D & 1 & D \end{bmatrix}. \quad (31)$$

We now show how entanglement helps in implementing the above code as an entanglement-assisted quantum convolutional code. We encode two qubits per frame with the help of two ebits. The stabilizer matrix for the unencoded qubit stream is as follows:

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}. \quad (32)$$

Rows 1 and 3 correspond to one ebit and rows 2 and 4 correspond to the other. Multiply row 1 by D and add the

result to row 3, multiply row 1 by $g(D)=1+D^{-1}+D^2$ and add the result to row 4, and multiply row 2 by $f(D)=1+D^{-2}$ and add the result to row 4. These row operations give the following equivalent stabilizer:

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & D & D & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ f(D) & g(D) & g(D) & f(D) & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}. \quad (33)$$

Figure 1 illustrates the operations that transform the unencoded stabilizer to the encoded one in an online encoding circuit (see Refs. [3,12] for details of translating gate-level operations to polynomial operations on the stabilizer matrix). The final stabilizer is as follows:

$$\begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & D & l(D) & D & 1 & D & 0 & 1 & 0 & 1 & 0 & 0 \\ f(D) & h(D) & 0 & 1 & 0 & 0 & 1 & l(D) & l(D) & l(D) & 1 & D \end{bmatrix},$$

where $h(D) \equiv D^{-1} + 1 + D$ and $l(D) \equiv 1 + D$. Compare Alice's generators in the last two rows of the above matrix to the quantum check matrix in Eq. (31). We have constructed a code with the same error-correcting properties because these two matrices are equivalent. The above code is a [[4,2;2]] entanglement-assisted convolutional code because it encodes two information qubits and consumes two ebits for every four uses of the noisy quantum channel.

It is interesting to compare the above example to the similar example in Sec. VII of Ref. [13]. In that example, we expanded the generators by a factor of 2 and were able to encode six information qubits and consume two ebits for every eight uses of the channel. The code in Ref. [13] makes a more efficient use of entanglement at the price of a doubling of the frame size (and thus a doubling of the memory requirements). In addition, the code in Ref. [13] requires infinite-depth operations, but the code in the above example uses finite-depth operations only.

IV. CLASSICAL ENHANCEMENT

It is possible to "piggyback" classical information along with the extra entanglement in a fashion similar to the superdense coding effect [17]. One can then decide to send classical information only or combine the transmitted classical information with extra entanglement to teleport more qubits [18]. We mention that Refs. [19,20] offer a different way of piggybacking classical information along with an entanglement-assisted code.

Consider the stabilizer in Eqs. (23) and (24). Recall that the first two sets of rows correspond to "extra entanglement" generators and the second two sets of rows are their corresponding generators. It is only clear to us that this piggybacking effect can work for rows that are not affected by infinite-depth operations—the first and third sets of rows in Eqs. (23) and (24).

Consider the first and third sets of rows in Eqs. (23) and (24),

$$\left[\begin{array}{ccc|ccc} I & 0 & I & 0 & 0 & 0 \\ E''_{1,a}(D) & 0 & E'_{1,a}(D) & \Gamma_1(D) & 0 & \Gamma_1(D) \end{array} \right], \quad (34)$$

where we omit the last four sets of columns of zeros and label the respective submatrices of $E''_1(D)$ and $E'_1(D)$ as $E''_{1,a}(D)$ and $E'_{1,a}(D)$. Consider the following set of operators:

$$\left[\begin{array}{ccc|ccc} 0 & 0 & E'^T_{1,a}(D^{-1})\Gamma_1(D) & 0 & 0 & I \end{array} \right]. \quad (35)$$

The rows in Eq. (35) anticommute with the first set of rows in Eq. (34) but commute with the second set of rows in Eq. (34) according to the shift symplectic product [15,12]. Furthermore, the rows in Eq. (35) anticommute with the first set of rows in Eq. (34) in such a way that the result of the shifted symplectic product is the identity matrix because

$$I \times I + 0 \times \Gamma_1^T(D^{-1})E'_{1,a}(D) = I.$$

The operators in Eq. (35) therefore give Alice a way to send s extra classical bits in analogy to superdense coding (we define s at the beginning of the proof of the main theorem). Alice sets any of the s bits to "1" by applying the corre-

sponding "encoded" version of the operator in Eq. (35) (one can determine the encoded version of the operator by applying the rest of the column operations of the encoding circuit to the operator). The fact that the generators in Eq. (35) commute with the second set of rows in Eq. (34) guarantees that encoding this extra classical information does not "throw off" the operation of the code for error correction. Bob simply has to measure the encoded version of the operators corresponding to the first set of rows in Eq. (34) to determine the values of the s classical bits.

One can use this classical enhancement just to send classical information and the result is an $[[n,k:s;c]]$ classically-enhanced entanglement-assisted convolutional code according to the notation of Ref. [19]. Alternatively, one can use the classical communication and extra entanglement to teleport more quantum information [18]. Teleporting then gives an $[[n,k+s/2;c+s/2]]$ entanglement-assisted convolutional code.

We mention that there is a trade-off between sending extra classical information and extra error-correcting capability. Exploiting the technique reduces number of errors that the code corrects. The encoded versions of the operators in Eq. (35) are actually a basis for the extra errors that the code corrects. Bob measures the encoded version of the operators corresponding to the first set of rows in Eq. (34) to retrieve the syndrome bits for these errors. It may be useful to correct these additional errors, but typically, one begins with a given set of generators that have desirable error-correcting properties. If one chooses to construct the entanglement-assisted quantum code with the techniques developed in the previous section, the resulting quantum code possesses the original desired error-correcting properties. One can then choose whether to exploit the extra entanglement for extra error-correcting capability or for classical enhancement.

Classical enhancement of the example

We show how to enhance the example so that it also sends classical information. Consider the following two operators:

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

The first row anticommutes with the first row in Eq. (33) and commutes with all other rows in Eq. (33). The second row anticommutes with the second row in Eq. (33) and commutes with all other rows in Eq. (33). These commutation relations imply that the above operators are useful for encoding classical information in a superdense-coding-like fashion. These operators encode two classical bits into the code and make use of the first two rows in Eq. (33) instead of just "wasting" them. Measuring the first two rows in Eq. (33) reveals the values of the two classical bits. We can determine the encoded versions of these "classical-information-encoding" operators by tracing how the operators change in the Heisenberg picture through the rest of the encoding circuit. The result is a [[4,2:2;2]] classically-enhanced entanglement-assisted convolutional code.

Alternatively, we can use these two classical bits and consume one ebit to teleport an additional information qubit.

This technique produces a [[4,3]] entanglement-assisted convolutional code.

V. CONCLUSION

We have constructed a theory of entanglement-assisted quantum convolutional coding for codes that do not have the CSS structure. The method of this paper uses entanglement less efficiently than the protocol in Ref. [13], but it does not require expanding a set of generators and therefore does not require a heuristic convergence argument as do the codes in Ref. [13]. The "extra entanglement" method results in encoding and decoding circuits that act on smaller numbers of qubits than the circuits in Ref. [13] and therefore requires less memory and circuit complexity. It should be of interest to find solutions in between the entanglement-efficient codes [13] and the entanglement-inefficient codes discussed in this paper.

It would be ideal to know the exact tradeoff between entanglement for the codes in this paper and the entanglement for the codes in Ref. [13], but we cannot establish this relationship right now because it is still an open question to determine the exact amount of entanglement that the codes in Ref. [13] require. Also, it would be ideal to know the trade-off between the frame size for codes in this paper and the frame size for codes in Ref. [13]. Note that it also remains an open question to determine the frame size for codes in Ref. [13] because we have not yet shown the exact step at which the algorithm from Ref. [13] converges.

It may be possible to avoid either using extra entanglement as outlined in this paper or expanding the generator set as outlined in Ref. [13]; we have no proof that these constructions are optimal. The fact that CSS codes do not need these methods provides some evidence that it may be possible [12]. However, after extensive exploration we have not found such a better technique, making us believe that it is unlikely.

[1] H. Ollivier and J.-P. Tillich, Phys. Rev. Lett. **91**, 177902 (2003).

[2] H. Ollivier and J.-P. Tillich, e-print arXiv:quant-ph/0401134.

[3] M. Grassl and M. Rötteler, in Proceedings of the IEEE International Symposium on Information Theory (IEEE, Piscataway, NJ, 2006), pp. 1109–1113; e-print arXiv:quant-ph/0602129.

[4] M. Grassl and M. Rötteler, Proceedings of the 44th Annual Allerton Conference on Communication, Control, and Computing (IEEE, Piscataway, NJ, 2006), pp. 510–519.

[5] M. Grassl and M. Rötteler, in Proceedings of the IEEE International Symposium on Information Theory (IEEE, Piscataway, NJ, 2007), pp. 816–820; e-print arXiv:quant-ph/0703182.

[6] G. D. Forney and S. Guha, in Proceedings of the IEEE International Symposium on Information Theory (IEEE, Piscataway, NJ, 2005), pp. 1028–1032; e-print arXiv:quant-ph/0501099.

[7] G. D. Forney, M. Grassl, and S. Guha, IEEE Trans. Inf. Theory **53**, 865 (2007).

[8] S. A. Aly, M. Grassl, A. Klappenecker, M. Rötteler, and P. K. Sarvepalli, in Proceedings of the 10th Canadian Workshop on Information Theory (IEEE, Piscataway, NJ, 2007),

pp. 180–183; e-print arXiv:quant-ph/0703113.

[9] S. A. Aly, A. Klappenecker, and P. K. Sarvepalli, e-print arXiv:quant-ph/0701037.

[10] R. Johannesson and K. S. Zigangirov, *Fundamentals of Convolutional Coding* (Wiley-IEEE, New York, NY, 1999).

[11] T. A. Brun, I. Devetak, and M.-H. Hsieh, Science **314**, 436 (2006).

[12] M. M. Wilde and T. A. Brun, e-print arXiv:0712.2223.

[13] M. M. Wilde and T. A. Brun, e-print arXiv:0807.3803.

[14] A. J. Viterbi, IEEE Trans. Inf. Theory **13**, 260 (1967).

[15] M. M. Wilde, H. Krovi, and T. A. Brun, e-print arXiv:0708.3699.

[16] A. Calderbank, E. Rains, P. Shor, and N. Sloane, IEEE Trans. Inf. Theory **44**, 1369 (1998).

[17] C. H. Bennett and S. J. Wiesner, Phys. Rev. Lett. **69**, 2881 (1992).

[18] C. H. Bennett, G. Brassard, C. Crépeau, R. Jozsa, A. Peres, and W. K. Wootters, Phys. Rev. Lett. **70**, 1895 (1993).

[19] I. Kremsky, M.-H. Hsieh, and T. A. Brun, Phys. Rev. A **78**, 012341 (2008).

[20] M. M. Wilde and T. A. Brun, in Proceedings of the IEEE International Symposium on Information Theory (IEEE, Piscataway, NJ, 2008), pp. 359–363; e-print arXiv:0801.0821.