6-19-2008

Optimal entanglement formulas for entanglement-assisted quantum coding

Mark M. Wilde
University of Southern California

Todd A. Brun
University of Southern California

Recommended Citation
Wilde, M., & Brun, T. (2008). Optimal entanglement formulas for entanglement-assisted quantum coding. *Physical Review A - Atomic, Molecular, and Optical Physics, 77* (6) https://doi.org/10.1103/PhysRevA.77.064302

Optimal Entanglement Formulas for Entanglement-Assisted Quantum Coding

Mark M. Wilde* and Todd A. Brun
Center for Quantum Information Science and Technology,
Communication Sciences Institute, Department of Electrical Engineering,
University of Southern California, Los Angeles, California 90089, USA
(Dated: October 29, 2018; Received October 29, 2018; Revised; Accepted; Published)

We provide several formulas that determine the optimal number of entangled bits (ebits) that a general entanglement-assisted quantum code requires. Our first theorem gives a formula that applies to an arbitrary entanglement-assisted block code. Corollaries of this theorem give formulas that apply to a code imported from two classical binary block codes, to a code imported from a classical quaternary block code, and to a continuous-variable entanglement-assisted quantum block code. Finally, we conjecture two formulas that apply to entanglement-assisted quantum convolutional codes.

PACS numbers: 03.67.Hk, 03.67.Mn, 03.67.Pp
Keywords: entanglement-assisted quantum error-correcting codes, entanglement-assisted quantum convolutional codes

The entanglement-assisted stabilizer formalism provides a useful framework for constructing quantum codes [1]. It has a number of advantages over the standard stabilizer formalism [2], including the ability to produce a quantum code from any classical linear code.

Entanglement is a valuable resource, and we would like to optimize the amount that a sender and receiver need to consume for quantum coding. Hsieh, Devetak, and Brun previously addressed this issue by determining a useful formula that gives the optimal number of ebits required by a Calderbank-Shor-Steane (CSS) entanglement-assisted code [3]. In particular, the number c of ebits that a CSS entanglement-assisted code requires is

$$c = \operatorname{rank}\left(HH^T\right), \tag{1}$$

where H corresponds to the parity check matrix of a classical binary block code that we import to correct quantum bit and phase flips. The same authors also determined an upper bound for the number of ebits that an entanglement-assisted quantum LDPC code requires [4].

In this Brief Report, we present several generalizations of the above formula. Our first theorem gives a formula for the optimal number of ebits that an arbitrary (non-CSS) entanglement-assisted quantum block code requires. This formula is also a bipartite entanglement measure for a stabilizer state, because it is equivalent to the measure in Ref. [5]. We find special cases of this formula that apply to an entanglement-assisted quantum block code produced from two arbitrary classical binary block codes, and to codes from a classical block code over $GF(4)$. Our last formula applies to *continuous-variable* entanglement-assisted codes [6]. We finally conjecture two formulas for the optimal number of ebits that a general (non-CSS) entanglement-assisted quantum convolutional code or one imported from a classical quaternary convolutional code [7] need to consume per frame of operation. We show that this conjecture holds for a particular example.

Theorem 1 *Suppose we want to build an entanglement-assisted quantum code from generators corresponding to the rows in a quantum check matrix*

$$H = \left[\begin{array}{c|c} H_Z & H_X \end{array}\right], \tag{2}$$

where H is an $(n-k)\times 2n$-dimensional binary matrix representing the quantum code [2], and both H_Z and H_X are $(n-k)\times n$-dimensional binary matrices. Then the resulting code is an $[[n, k+c; c]]$ entanglement-assisted code and requires c ebits, where

$$c = \operatorname{rank}\left(H_X H_Z^T + H_Z H_X^T\right)/2, \tag{3}$$

and addition is binary.

Proof. Consider the matrix $\Omega_H = H_X H_Z^T + H_Z H_X^T$. Its entries are the symplectic products between all rows of H so that

$$[\Omega_H]_{ij} = h_i \odot h_j, \tag{4}$$

where h_i is the i^{th} row of H and $\odot$ denotes the symplectic product [8]. Both h_i and h_j are each $2n$-dimensional binary vectors and we write them as follows:

$$h_i = \left(h_i^Z \mid h_i^X\right), \tag{5}$$
$$h_j = \left(h_j^Z \mid h_j^X\right), \tag{6}$$

so that the first n components are the "Z" part and the last n components are the "X" part. Then the symplectic product $h_i \odot h_j$ is equal to

$$h_i \odot h_j = h_i^Z \cdot h_j^X + h_i^X \cdot h_j^Z, \tag{7}$$

where $\cdot$ is the standard inner product and addition is binary. The matrix Ω_H is a $(n-k)\times(n-k)$-dimensional binary matrix. We call Ω_H the "symplectic product matrix" for the purposes of this Brief Report.

Refs. [1, 8] outline a symplectic Gram-Schmidt orthogonalization procedure (SGSOP) that uniquely determines the optimal (i.e., minimal) number of ebits that the code requires, and Ref. [9] proves that the SGSOP gives the optimal number of ebits. The code construction in Refs. [1, 8] shows that the resulting entanglement-assisted quantum code requires at most c ebits. The essence of the argument in Ref. [9] is that the resulting entanglement-assisted quantum code requires at least c ebits because any fewer ebits would not be able to resolve the anticommutativity of the generators on Alice's side of the code.

The SGSOP performs row operations that do not change the error-correcting properties of the quantum code (because the code is additive), but these row operations do change the symplectic product relations. These row operations are either a row swap $S(i,j)$, where $S(i,j)$ is a full-rank $(n-k)\times(n-k)$ matrix that swaps row i with j, or a row addition $A(i,j)$, where $A(i,j)$ is a full-rank $(n-k)\times(n-k)$ matrix that adds row i to row j. These row operations multiply the matrix H from the left. The SGSOP then is equivalent to a full-rank $(n-k)\times(n-k)$ matrix G that contains all of the row operations and produces a new quantum check matrix $H' = GH$ with corresponding symplectic product matrix $\Omega_{H'} = G\left(H_X H_Z^T + H_Z H_X^T\right) G^T$. In particular, the resulting symplectic product matrix $\Omega_{H'}$ is in a standard form so that

$$\Omega_{H'} = \bigoplus_{i=1}^{c} J \oplus \bigoplus_{j=1}^{n-k-2c} [0], \tag{8}$$

where the small and large $\oplus$ correspond to the direct sum operation, J is the matrix

$$J = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \tag{9}$$

and $[0]$ is the one-element zero matrix. Each matrix J in the direct sum corresponds to a *symplectic pair* and has rank two. Each symplectic pair corresponds to exactly one ebit that the code requires [1, 8]. Each matrix $[0]$ has rank zero and corresponds to an ancilla qubit. The optimal number of ebits required for the code is $\operatorname{rank}(\Omega_{H'})/2$:

$$\operatorname{rank}(\Omega_{H'}) = \operatorname{rank}\left(\bigoplus_{i=1}^{c} J \oplus \bigoplus_{j=1}^{n-k-2c} [0]\right) \tag{10}$$

$$= \sum_{i=1}^{c} \operatorname{rank}(J) + \sum_{j=1}^{n-k-2c} \operatorname{rank}([0]) \tag{11}$$

$$= 2c. \tag{12}$$

The second line follows because the rank of a direct sum is the sum of the individual matrix ranks, and the third line follows from the individual matrix ranks given above. The number c of ebits is also equal to $\operatorname{rank}(\Omega_H)/2$ because the matrix G is full rank. The code is an $[[n, k+c; c]]$ entanglement-assisted quantum block code by the construction in Refs. [1, 8]. ■

Our formula (3) is equivalent to the formula at the top of page 14 in Ref. [8], but it provides the quantum code designer with a quick method to determine how many ebits an entanglement-assisted code requires, by simply "plugging in" the generators of the code.

The formula (3), like the CSS formula in (1), is a measure of how far a set of generators is from being a commuting set, or equivalently, how far it is from giving a standard stabilizer code.

Corollary 1 below gives a formula for the optimal number of ebits required by a CSS entanglement-assisted quantum code. It is generally a bit less difficult to compute than the above formula in (3). This reduction in complexity occurs because of the special form of a CSS quantum code and because the size of the matrices involved are generally smaller for a CSS code than for a general code with the same number of generators and physical qubits.

Corollary 1 *Suppose we import two classical $[n, k_1, d_1]$ and $[n, k_2, d_2]$ binary codes with respective parity check matrices H_1 and H_2 to build an entanglement-assisted quantum code. The resulting code is an $[[n, k_1 + k_2 - n + c, \min(d_1, d_2); c]]$ entanglement-assisted code, and requires c ebits where*

$$c = \operatorname{rank}\left(H_1 H_2^T\right). \tag{13}$$

Proof. The quantum check matrix has the following form:

$$H = \left[\begin{array}{c|c} H_1 & 0 \\ 0 & H_2 \end{array}\right]. \tag{14}$$

The symplectic product matrix Ω_H is then

$$\Omega_H = \begin{bmatrix} H_1 \\ 0 \end{bmatrix} \begin{bmatrix} 0 & H_2^T \end{bmatrix} + \begin{bmatrix} 0 \\ H_2 \end{bmatrix} \begin{bmatrix} H_1^T & 0 \end{bmatrix} \tag{15}$$

$$= \begin{bmatrix} 0 & H_1 H_2^T \\ H_2 H_1^T & 0 \end{bmatrix}. \tag{16}$$

The above matrix is equivalent by a full rank permutation matrix to the matrix $H_1 H_2^T \oplus H_2 H_1^T$, so the rank of Ω_H is

$$\operatorname{rank}(\Omega_H) = \operatorname{rank}\left(H_1 H_2^T \oplus H_2 H_1^T\right) \tag{17}$$

$$= \operatorname{rank}\left(H_1 H_2^T\right) + \operatorname{rank}\left(H_2 H_1^T\right) \tag{18}$$

$$= 2 \operatorname{rank}\left(H_1 H_2^T\right) \tag{19}$$

The second line follows because the rank of a direct sum is equivalent to the sum of the individual ranks, and the third line follows because the rank is invariant under matrix transposition. The number of ebits

required for the resulting entanglement-assisted quantum code is $\operatorname{rank}\left(H_1 H_2^T\right)$, using the result of the previous theorem. The construction in Refs. [1, 8] produces an $[[n, k_1 + k_2 - n + c, \min(d_1, d_2); c]]$ entanglement-assisted quantum block code. ■

Corollary 2 *Suppose we import an $[n, k, d]_4$ classical code over $GF(4)$ with parity check matrix H for use as an entanglement-assisted quantum code according to the construction in Ref. [1, 8]. Then the resulting quantum code is an $[[n, 2k - n + c; c]]$ entanglement-assisted quantum code where $c = \operatorname{rank}\left(HH^\dagger\right)$ and $\dagger$ denotes the conjugate transpose operation over matrices in $GF(4)$.*

Proof. Ref. [1] shows how to produce an entanglement-assisted quantum code from the parity check matrix H of a classical code over $GF(4)$. The resulting quantum parity check matrix H_Q in symplectic binary form is

$$H_Q = \gamma\left(\begin{bmatrix}\omega H \\ \bar{\omega} H\end{bmatrix}\right), \tag{20}$$

where γ denotes the isomorphism between elements of $GF(4)$ and symplectic binary vectors that represent Pauli matrices. Specifically, $\gamma^{-1}(h) = \omega h_x + \bar{\omega} h_z$ where h is a symplectic binary vector and h_x and h_z denote its "X" and "Z" parts respectively. The symplectic product between binary vectors is equivalent to the trace product of their $GF(4)$ representations (see, e.g., Ref. [1]):

$$h_i \odot h_j = \operatorname{tr}\left\{\gamma^{-1}(h_i) \cdot \overline{\gamma^{-1}(h_j)}\right\}, \tag{21}$$

where h_i and h_j are any two rows of H_Q, $\cdot$ denotes the inner product, the overbar denotes the conjugate operation, and $\operatorname{tr}\{x\} = x + \bar{x}$ denotes the trace operation over elements of $GF(4)$. We exploit these correspondences to write the symplectic product matrix Ω_{H_Q} for the quantum check matrix H_Q as follows:

$$\begin{aligned}
\Omega_{H_Q} &= \operatorname{tr}\left\{\begin{bmatrix}\omega H \\ \bar{\omega} H\end{bmatrix}\begin{bmatrix}\omega H \\ \bar{\omega} H\end{bmatrix}^\dagger\right\} && (22)\\
&= \operatorname{tr}\left\{\begin{bmatrix}\omega H \\ \bar{\omega} H\end{bmatrix}\begin{bmatrix}\bar{\omega} H^\dagger & \omega H^\dagger\end{bmatrix}\right\} && (23)\\
&= \operatorname{tr}\left\{\begin{bmatrix}HH^\dagger & \bar{\omega} HH^\dagger \\ \omega HH^\dagger & HH^\dagger\end{bmatrix}\right\} && (24)\\
&= \operatorname{tr}\left\{\begin{bmatrix}1 & \bar{\omega} \\ \omega & 1\end{bmatrix} \otimes HH^\dagger\right\} && (25)\\
&= \begin{bmatrix}1 & \bar{\omega} \\ \omega & 1\end{bmatrix} \otimes HH^\dagger + \begin{bmatrix}1 & \omega \\ \bar{\omega} & 1\end{bmatrix} \otimes \bar{H} H^T && (26)
\end{aligned}$$

where the "tr" operation above is an element-wise trace operation over $GF(4)$ (it is not the matrix trace operation.) The matrix Ω_{H_Q} is over the field $GF(2)$, but we can consider it as being over the field $GF(4)$ without changing its rank. Therefore, we can multiply it by matrices over the field $GF(4)$. Consider the following full-rank $GF(4)$ matrices:

$$A_1 = \begin{bmatrix}1 & \bar{\omega} \\ 0 & 1\end{bmatrix} \otimes I, \quad A_2 = \begin{bmatrix}1 & 0 \\ 1 & 1\end{bmatrix} \otimes I. \tag{27}$$

We premultiply and postmultiply the matrix Ω_{H_Q} as follows and obtain a matrix with the same rank as Ω_{H_Q}:

$$\begin{aligned}
A_2 A_1 \Omega_{H_Q} A_1^\dagger A_2^\dagger &= \begin{bmatrix}0 & 0 \\ 0 & 1\end{bmatrix} \otimes HH^\dagger + \begin{bmatrix}1 & 0 \\ 0 & 0\end{bmatrix} \otimes \bar{H} H^T && (28)\\
&= \begin{bmatrix}\bar{H} H^T & 0 \\ 0 & HH^\dagger\end{bmatrix} && (29)\\
&= \bar{H} H^T \oplus HH^\dagger && (30)
\end{aligned}$$

Therefore, the rank of Ω_{H_Q} is

$$\begin{aligned}
\operatorname{rank}\left(\Omega_{H_Q}\right) &= \operatorname{rank}\left(\bar{H} H^T \oplus HH^\dagger\right) && (31)\\
&= \operatorname{rank}\left(\bar{H} H^T\right) + \operatorname{rank}\left(HH^\dagger\right) && (32)\\
&= 2 \operatorname{rank}\left(HH^\dagger\right). && (33)
\end{aligned}$$

The second line holds because the rank of a direct sum is the sum of the individual ranks and the third holds because the rank is invariant under the matrix transpose operation. Therefore, the resulting entanglement-assisted quantum code requires $c = \operatorname{rank}\left(HH^\dagger\right)$ ebits by applying the result of the original theorem. The construction in Refs. [1, 8] produces an $[[n, 2k - n + c; c]]$ entanglement-assisted quantum code. ■

Corollary 3 *We can construct a continuous-variable entanglement-assisted quantum code from generators corresponding to the rows in quantum check matrix $H = \left[\begin{array}{c|c} H_Z & H_X \end{array}\right]$ where H is $(n-k) \times 2n$-dimensional, H is a real matrix representing the quantum code [6], and both H_Z and H_X are $(n-k) \times n$-dimensional. The resulting code is an $[[n, k + c; c]]$ continuous-variable entanglement-assisted code and requires c entangled modes where*

$$c = \operatorname{rank}\left(H_X H_Z^T - H_Z H_X^T\right)/2. \tag{34}$$

Proof. The proof is similar to the proof of the first theorem but requires manipulations of real vectors instead of binary vectors. See Ref. [6] for details of the symplectic geometry required for continuous-variable entanglement-assisted codes. ■

Remark 1 *A similar formula holds for entanglement-assisted qudit codes by replacing the subtraction operation above with subtraction modulo d. Specifically, we can construct a qudit entanglement-assisted quantum code from generators corresponding to the rows in check matrix $H = \left[\begin{array}{c|c} H_Z & H_X \end{array}\right]$ whose matrix entries are elements of the finite field $\mathbb{Z}_d$. The code requires c edits (a d-dimensional state $\left(\sum_{i=0}^{d-1} |i\rangle |i\rangle\right)/\sqrt{d}$) where*

$$c = \operatorname{rank}\left(H_X H_Z^T \ominus_d H_Z H_X^T\right)/2$$

and $\ominus_d$ is subtraction modulo d. We use subtraction modulo d because the symplectic form over d-dimensional variables includes subtraction modulo d.

We can also consider the case when the binary, d-variable, or real matrix H specifies one party's generators of a respective bipartite qubit, qudit, or continuous-variable stabilizer state. In that case, our formula is equivalent to the entanglement measure found in Ref. [5].

We finally conjecture two formulas for the number of ebits required in a general (non-CSS) entanglement-assisted quantum convolutional code.

Conjecture 1 *The optimal number c of ebits necessary per frame for an entanglement-assisted quantum convolutional code is*

$$c = \operatorname{rank}\left(H_X(D)H_Z^T(D^{-1}) + H_Z(D)H_X^T(D^{-1})\right)/2 \tag{35}$$

where $H(D) = \left[\begin{array}{c|c} H_Z(D) & H_X(D) \end{array}\right]$ represents the parity check matrix for a set of quantum convolutional generators that do not necessarily form a commuting set.

Conjecture 2 *The optimal number c of ebits required per frame for an entanglement-assisted quantum convolutional code imported from a classical quaternary convolutional code with parity check matrix $H(D)$ is*

$$c = \operatorname{rank}\left(H(D)H^\dagger(D^{-1})\right). \tag{36}$$

We know the number of ebits required for a CSS entanglement-assisted quantum convolutional code is $\operatorname{rank}\left(H_1(D)H_2^T(D^{-1})\right)$ [7] where $H_1(D)$ and $H_2(D)$ correspond to the classical binary convolutional codes that we import to correct respective bit and phase flips. Comparing the CSS convolutional formula, the CSS block formula in Corollary 1, and the general block formula in Theorem 1, the above conjecture seems natural.

We finally provide an example of the above conjecture. It is a slight modification of the code presented in Ref. [10].

Example 1 *Consider the quantum convolutional code with quantum check matrix as follows:*

$$\left[\begin{array}{ccccc|ccccc} 0 & 0 & 0 & 0 & 0 & h(D) & 0 & D & 1 & h(D) \\ h(D) & D & 0 & 1 & h(D) & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & D & D & D & 0 & 1 & 0 & 1 & 1 \\ 0 & \frac{1}{D} & 1 & \frac{1}{D} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{D} & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{array}\right],$$

where $h(D) = 1 + D$. This code requires two ebits and one ancilla qubit for quantum redundancy and encodes two information qubits. The shifted symplectic product matrix $H_X(D)H_Z^T(D^{-1}) + H_Z(D)H_X^T(D^{-1})$ [7, 11] for this code is as follows:

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{37}$$

The rank of the above matrix is four and the code requires two ebits. Therefore, the conjecture holds for this example.

MMW and TAB acknowledge support from NSF Grants CCF-0545845 and CCF-0448658.

* Electronic address: mark.wilde@usc.edu

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

[2] D. Gottesman, Ph.D. thesis, California Institue of Technology (1997).

[3] M.-H. Hsieh, I. Devetak, and T. Brun, Phys. Rev. A **76**, 062313 (2007).

[4] M.-H. Hsieh, T. A. Brun, and I. Devetak, in *Asian Conference on Quantum Information Science* (2007), pp. 101–102.

[5] D. Fattal, T. S. Cubitt, Y. Yamamoto, S. Bravyi, and I. L. Chuang, arXiv:quant-ph/0406168 (2004).

[6] M. M. Wilde, H. Krovi, and T. A. Brun, Phys. Rev. A **76**, 052308 (2007).

[7] M. M. Wilde and T. A. Brun, arXiv:0712.2223 (2007).

[8] T. A. Brun, I. Devetak, and M.-H. Hsieh, arXiv:quant-ph/0608027 (2006).

[9] D. Gottesman, URL `http://www.perimeterinstitute.ca/personal/`

[10] M. M. Wilde and T. A. Brun, arXiv:0801.0821 (2008).

[11] M. M. Wilde, H. Krovi, and T. A. Brun, arXiv:0708.3699 (2007).