Description of a Quantum Convolutional Code

Harold Ollivier and Jean-Pierre Tillich
INRIA, Projet CODES, BP 105, F-78153 Le Chesnay, France
(Received 29 April 2003; published 23 October 2003)

We describe a quantum error correction scheme aimed at protecting a flow of quantum information over long distance communication. It is largely inspired by the theory of classical convolutional codes which are used in similar circumstances in classical communication. The particular example shown here uses the stabilizer formalism. We provide an explicit encoding circuit and its associated error estimation algorithm. The latter gives the most likely error over *any* memoryless quantum channel, with a complexity growing only linearly with the number of encoded qubits.

DOI: 10.1103/PhysRevLett.91.177902 PACS numbers: 03.67.Pp, 03.67.Hk, 03.67.Lx

In recent years, the discovery and development of quantum computation and communication has shed new light on quantum physics. The potential applications of these new fields encompass a wide variety of subjects, ranging from unconditionally secure secret key generation protocols [1] to efficient integer factoring algorithms [2] or enhancement of communication complexity [3]. However, the practical realization of such protocols and algorithms remains a very involved task mainly because of the inherent instability of quantum superpositions [4] as well as intrinsic imprecisions of the physical devices that process quantum information. These errors wipe out the quantum superpositions together with entanglement, which are usually seen as key resources of the power of quantum algorithms and protocols [5]. Hence, protecting the quantum nature of information became one of the most important challenges to prove the feasibility of quantum computers. The discovery of quantum error correction schemes [6,7] notably opened the future of large scale quantum information processing: a certain, but unfortunately very small, degree of imprecision can be tolerated at each step of a quantum transformation and still allow a speedup over classical information processing [8,9]. However, building a fault-tolerant quantum computer remains largely out of reach of the present day practical realizations, principally because of the large number of physical qubits required to account for the error correction.

On the other hand, quantum cryptography and more generally the field of quantum communication seems more promising in the near future. Some quantum key distribution protocols have been implemented and the associated devices seem to be close to commercialization [10]. Within this context, we construct a new family of codes—quantum convolutional codes—aimed at protecting a stream of quantum information in a long distance communication. They are the correct generalization to the quantum domain of their classical analogs, and hence inherit their most important properties. First, they have a *maximum likelihood* error estimation algorithm for *all* memoryless channels with a complexity growing linearly with the number of encoded qubits. This is an important issue since finding the most likely error—a strategy which allows one to determine the most likely sent codeword—is in general a hard task: for a generic family of block codes with constant rate, the maximum likelihood error estimation algorithm has a complexity growing exponentially with the number of encoded qubits. Hence, generic block codes rapidly require one to employ suboptimal error estimation procedures which, as a consequence, do not exploit the whole error correcting capabilities of the code. Moreover, our algorithm can easily handle variations in the properties of the communication channel (i.e., a change in the single qubit error probabilities). The second advantage of quantum convolutional codes is their ability to perform the encoding of the qubits on-line (i.e., as they arrive in the encoder). Thus, it is not necessary to wait for all the qubits to be ready to start sending the encoded state through the communication channel: it reduces the overall processing time of the qubits which is an additional source of decoherence. Note that an attempt at defining quantum convolutional codes was made some time ago [11], but missed some crucial points concerning the error estimation algorithm as well as error propagation properties.

In this Letter, we deal with a specific example drawn from our general theory. We construct a rate $1/5$ quantum convolutional code: we explain how to encode and decode a stream of qubits efficiently, and we expose the maximum likelihood error estimation algorithm. This will give the necessary intuition to understand how to generalize the present results to a wider framework [12].

Description of the code.—The particular code we wish to present is best described by using the stabilizer formalism [13]. This provides a simple way to understand the encoding and decoding operations. Moreover, the error syndromes can be easily identified, which considerably simplifies the description of the error estimation algorithm. We use the following standard notations for the Pauli operators acting on a single qubit:

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \qquad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \qquad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \tag{1}$$

so that $XY = iZ$, while I denotes the identity matrix. Since convolutional codes are designed to deal with infinite streams of information qubits, the number of generators of the stabilizer group is infinite as well. In practice, transmission starts and ends at a given time, thus we consider only generators made of a finite number of Pauli operators.

The code subspace is described by the generators of its stabilizer group, S. These generators are given by

$$\begin{aligned} M_0 &= XZIIIIII\ldots, & M_1 &= ZXXZIIII\ldots, \\ M_2 &= IZXXZIII\ldots, & M_3 &= IIZXXZII\ldots, \\ M_4 &= IIIZXXZI\ldots, \\ M_{4i+j} &= I^{\otimes 5i} \otimes M_j, & 0 < i, &\quad 1 \le j \le 4, \\ M_\infty &= \ldots IIIIZX. \end{aligned} \tag{2}$$

It is easy to check that all the generators commute and are independent. Thus, the code subspace (i.e., the largest common eigenspace of the generators with eigenvalue $+1$) is nontrivial.

An important point to address when considering stabilizer codes is the ability to manipulate encoded information. Namely, we want to find the encoded Pauli operators $\overline{X}_i$, $\overline{Z}_i$ corresponding to logical qubit i. These operators must satisfy the following relations:

$$\overline{X}_i, \overline{Z}_i \in N(S) - S, \tag{3}$$

$$\forall i \neq j, \quad [\overline{X}_i, \overline{X}_j] = [\overline{Z}_i, \overline{Z}_j] = [\overline{X}_i, \overline{Z}_j] = 0, \tag{4}$$

where $N(S)$ denotes the normalizer of S. Equation (3) states that encoded Pauli operators leave the code subspace globally invariant, but have a nontrivial action on its elements, while Eq. (4) ensures that manipulating qubit i does not affect other qubits. There exists a great choice of different sets of such operators; however, they are not equivalent in the perspective of effectively manipulating encoded quantum information in an easy way: in practice only those with a small number of terms different from the identity are useful. For our particular example, such set exists and has a structure invariant by a shift of five qubits:

$$\begin{aligned} \overline{X}_1 &= IZIXIZII\ldots, & \overline{Z}_1 &= IZZZZZII\ldots, \\ \overline{X}_n &= I^{\otimes 5n} \otimes \overline{X}_1, & n &> 1, \\ \overline{Z}_n &= I^{\otimes 5n} \otimes \overline{Z}_1, & n &> 1. \end{aligned} \tag{5}$$

Hence, a unitary transformation on a single encoded qubit will in general be implemented by a unitary transformation on five physical qubits.

At this point, one can wonder what in this code differs from a generic block code. The answer to this question comes from the particular structure of the stabilizer generators: beside M_0 and M_∞, the generators of the stabilizer group can be cast into sets of constant size (e.g., four), each set acting on a fixed number (e.g., seven) of consecutive qubits. In addition, each set has a fixed overlap (e.g., of two qubits) with the set immediately before and immediately after. This very peculiar structure defines quantum convolutional codes and we can prove [12] that this implies the possibility of on-line encoding and the existence of an efficient error estimation algorithm.

Encoding circuit.—As explained by Gottesman [13], there are various ways to realize the encoding into the code subspace. However, for convolutional codes, they are not equivalent: standard encoding circuits usually require one to wait until the last "to-be-protected" qubit has been obtained before sending the encoded state. In this section, we explain how to take advantage of the structure of the stabilizer generators to overcome this limitation and encode the qubits on-line. We first exhibit a map from the computational basis of the to-be-protected qubits to a basis of the code subspace. As a second step, we derive the quantum circuit implementing this map in a unitary way.

More precisely, consider the following set of states:

$$\{|\psi(c_1, c_2, c_3, \ldots)\rangle\}_{c_i \in \{0,1\}} = \{P|0, 0, 0, 0, 0, c_1, 0, 0, 0, 0, c_2, 0, 0, 0, 0, c_3, \ldots\rangle\}_{c_i \in \{0,1\}}, \tag{6}$$

where $P = \prod_i (I + M_i)/\sqrt{2}$ is the projection operator onto the code subspace. Since $\overline{Z}_i$ commutes with all the generators of the stabilizer group, the following equation holds for any element of the set:

$$\overline{Z}_i P|0, 0, 0, 0, 0, c_1, 0, 0, 0, 0, c_2, 0, 0, 0, 0, c_3, \ldots\rangle = (-1)^{c_i} P|0, 0, 0, 0, 0, c_1, 0, 0, 0, 0, c_2, 0, 0, 0, 0, c_3, \ldots\rangle. \tag{7}$$

This implies that $\{|\psi(c_1, c_2, c_3, \ldots)\rangle\}_{c_i \in \{0,1\}}$ is an orthonormal basis of the code subspace. Hence, the natural encoding consists in mapping the computational basis of the to-be-protected qubits, $\{|c_1, c_2, c_3, \ldots\rangle\}_{c_i \in \{0,1\}}$ into the basis $\{|\psi(c_1, c_2, c_3, \ldots)\rangle\}_{c_i \in \{0,1\}}$.

In practice, to encode a stream of qubits q_i, we first add to it ancillary qubits in the $|0\rangle$ state such that the to-be-protected qubit i is now at the position $5i + 1$. Then, we need to implement P for these specific input states as a unitary transformation onto the whole Hilbert space. This can be done in full generality as explained in [13], and gives the encoding circuit of Fig. 1. From this simple example, it is easy to understand that the possibility of on-line encoding for quantum convolutional codes is a consequence of the finite extension of the support of the generators of the stabilizer group and of the encoded Pauli operators. Also note that alternative encoding methods can be found and can be relevant when considering some specific applications, but these issues are beyond the scope of this Letter.

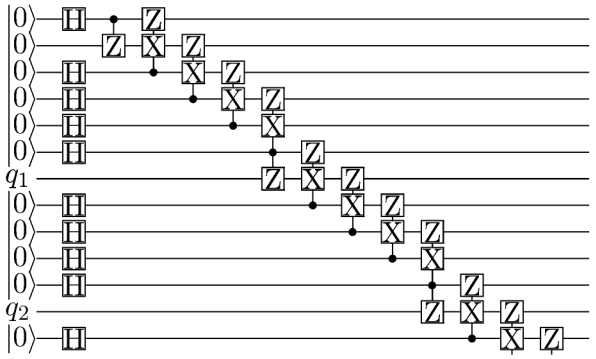

FIG. 1. Beginning of the encoding circuit. H is the Hadamard transform, and the circles represent the control qubit for a given gate. The circuit is run from left to right. Ancillary qubits are in the $|0\rangle$ state while the to-be-protected quantum information is in qubits $q_1, q_2, \ldots$.

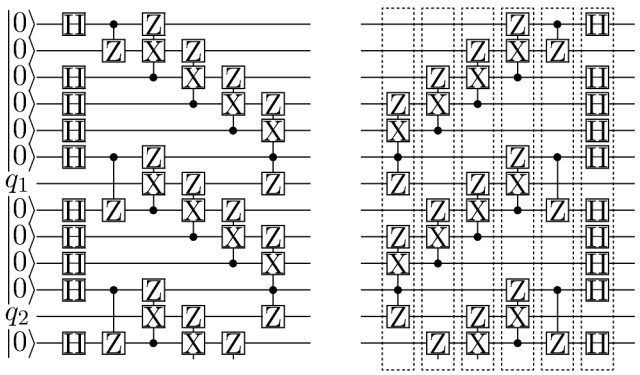

FIG. 2. Encoding (left) and decoding (right) circuits satisfying the pearl-necklace structure. The dashed boxes in the decoding circuit define the different layers. The gates inside a layer commute with each other, granting noncatastrophicity and forward decoding.

Error propagation and on-line decoding.—Because of their very specific nature, convolutional codes propagate information contained in a given qubit to its successors (see again Fig. 1). During the decoding process (i.e., the inverse of encoding) this can actually become a problem: an error affecting a *finite* number of qubits before decoding can propagate through the decoding circuit and finally affect an *infinite* number of qubits. Such error is called *catastrophic*. It is worth mentioning that this issue is not specific to the quantum domain: classical convolutional encoders might also be catastrophic [14,15]. Fortunately, in both cases, noncatastrophic encoders exist. More precisely, given an encoder one can determine whether it has catastrophic errors. For classical codes this is a well-known result established by Massey and Sain [16]. For the quantum setting, the following condition is both necessary and sufficient for noncatastrophicity: the gates of the decoding circuit can be arranged in a finite number of layers, such that they commute with each other inside a layer. Figure 2 illustrates this "pearl-necklace" structure for our example, and thus proves that our quantum convolutional code is noncatastrophic.

In this paragraph, we briefly mention how this result can be derived. The sufficiency of the above condition is easily obtained by realizing that errors can only propagate through noncommuting gates of the decoding circuit. Hence, for a finite size error the pearl-necklace structure imposes that after each layer only a finite number of qubits are potentially erroneous. Thus, no such error can propagate infinitely through the whole decoding circuit. On the other hand, necessity is obtained by explicitly exhibiting the pearl-necklace structure for a noncatastrophic decoding circuit. Without loss of generality, we can assume this circuit is obtained by running the encoding circuit in reverse order. To group the gates into the pearl-necklace structure, we start by changing the order of two noncommuting gates in this circuit. This can be regarded as an error which, by hypothesis, can be corrected by a finite size unitary operation. In turn, this finiteness allows one to repeat the above procedure at regular intervals in the circuit and allows the appropriate grouping (compare Figs. 1 and 2). The formal proof is straightforward but cumbersome and we leave it for a separate paper [12]. Note also that this procedure is an explicit algorithm to check for catastrophicity of a code.

Moreover, this condition implies the existence of a *forward* decoding scheme: there is no need to wait for the last qubit to start decoding (see Fig. 2). For noncatastrophic codes, both encoding and decoding can be done on-line [12].

Maximum likelihood error estimation.—An error correcting code aims at protecting information sent over a noisy communication channel by letting the receiver infer which error possibly affected the information. This is the role of the error estimation algorithm. On average, the correct information is most often retrieved when the estimated error coincides with the most likely error. Thus, it is both of theoretical and practical relevance to have an efficient maximum likelihood error estimation algorithm for our quantum convolutional codes. In this section, we exhibit such algorithm. It is indeed the quantum analog of the well-known Viterbi algorithm for classical convolutional codes [14,15]. The Viterbi algorithm realizes a maximum likelihood error estimation on all memoryless channels with a complexity linear in the number of encoded bits. This explains why classical convolutional codes are so widely used for reducing the noise on communication channels.

Our algorithm for quantum convolutional codes processes the information obtained through the syndrome in order to infer the most likely error. The circuit for obtaining the syndromes follows the usual phase estimation scheme: an ancillary qubit is prepared in the $|0\rangle$ state; undergoes a Hadamard transform; controls the application of one of the generator M_i of the stabilizer group; again undergoes a Hadamard transform; and is finally measured in the $\{|0\rangle, |1\rangle\}$ basis. Then, the algorithm updates a list of maximum likelihood error candidates by looking at a small number of syndromes at a time, and by

taking local decisions. It is preceded and followed by appropriate initialization and termination steps.

The initialization step lists all error candidates, $\{E_j^0\}_j$, for the first two qubits which are compatible with the syndrome M_0. There are exactly $8 = 4^2/2$ of them (there are 4^2 different operators with support on the first two qubits, but the constraint associated with M_0 divides this set into two equal parts). This list constitutes the input of the main loop of the algorithm. At step i, the algorithm constructs a list of some most likely error candidates, $\{E_j^i\}_j$, compatible with the syndromes M_0 to M_{4i}. Each candidate E_j^i is thus specified only on qubits 1 to $5i + 2$. The crucial point of the algorithm is to maintain a fixed size of this list, and hence to avoid the exponential blow up that would arise when listing all error candidates compatible with these syndromes. More precisely, E_j^i is a most likely candidate whose restriction on qubit $5i + 1$ and $5i + 2$ is prescribed by the index j running over the set of 16 possible errors affecting those two qubits. The computation of any error candidate E_k^{i+1} is easily achieved provided $\{E_j^i\}_j$: consider the set of all possible extensions of the error candidates E_j^i to qubit $5i + 3$ to $5i + 7$ that are compatible with syndromes M_{4i+1} to $M_{4(i+1)}$, and which have the prescribed error k at position $5i + 6$ and $5i + 7$. It is easy to check that any such element is now compatible with syndromes M_0 to $M_{4(i+1)}$. The specific candidate E_k^{i+1} is chosen to be the most likely operator among the elements of the latter set (in the case of a tie, one is chosen at random). This procedure is continued until reaching M_∞, which again selects half of the candidates. The termination of the algorithm outputs the most likely candidate among the remaining ones. This constitutes the most likely error given the value of all the syndromes for the received stream of qubits [12].

The main property used to prove this fact is related to the structure of the generators of the stabilizer group: the value of the syndromes associated to M_{4i+1} to M_{4i+4} depends on the syndromes M_0 to M_{4i} only through the error operators at position $5i + 1$ and $5i + 2$. Thus, taking a sequence of local decisions allows one to construct a list of error candidates among which one will coincide with the most likely error until qubit $5i + 2$, while on the other hand maintaining a linear complexity of the algorithm as the number of encoded qubits increases. Note that the error maximizing the likelihood is known when the last syndrome is measured. Hence, it is in principle necessary to wait until the end of the transmission to actually correct the estimated error. However, as for the classical Viterbi algorithm, numerical simulations show that the different candidates at a given step coincide with the most likely error except on their last few positions. Thus, in practice it is possible to estimate the error online. In addition, we want to stress that without increasing its complexity, this algorithm can take into account all memoryless quantum channels even if the single qubit error probabilities are not constant in time. For example, one could imagine that the qubits are photons sent through an optical fiber, and that the probabilities are evaluated by sending probe photons containing no useful information. Finally, as the codes described here are the exact translation to the quantum setting of the classical convolutional codes, one can also derive suboptimal error estimation algorithms (for their classical analogs see [14,15]). Most importantly, quantum convolutional codes can be decoded iteratively and should allow quantum turbo decoding [12].

Conclusion.—In this Letter, we presented the theory of quantum convolutional codes by an example. We gave explicitly the associated encoding and decoding circuits, as well as a low complexity maximum likelihood error estimation algorithm. We believe that such codes could be used to reduce errors for long distance quantum communications provided that we are able to perform a small and fixed number of quantum gates with good fidelity. Moreover, the tools developed for quantum convolutional codes can be used to translate other families of classical codes to the quantum domain, like, for instance, low density parity check codes.

Part of this work was done when H. O. was visiting the Perimeter Institute and the Institute for Quantum Computing in Waterloo. Useful discussions with J. Kempe, R. Laflamme, and D. Poulin are gratefully acknowledged.

[1] C. H. Bennett and G. Brassard, in *Proceedings of the IEEE International Conference on Computers, Systems and Signal Processing, Bangalore, India, 1984* (IEEE, New York, 1984), p. 175.
[2] P. W. Shor, SIAM J. Comput. **26**, 1484 (1997).
[3] H. Buhrman, R. Cleve, and A. Wigderson, in *Proceedings of the 13th Annual ACM Symposium on Theory of Computing (STOC'98), Dallas, Texas, 1998* (ACM Press, New York, 1998), pp. 63–68.
[4] W. H. Zurek, Phys. Today **44**, No. 10, 36 (1991).
[5] R. Jozsa and N. Linden, arXiv:quant-ph/0201143.
[6] P. W. Shor, Phys. Rev. A **52**, 2493 (1995).
[7] R. Laflamme, C. Miquel, J.-P. Paz, and W. H. Zurek, Phys. Rev. Lett. **77**, 198 (1996).
[8] D. Aharonov and M. Ben-Or, arXiv quant-ph/9906129.
[9] C. Zalka, arXiv:quant-ph/9612028.
[10] D. Stucki, N. Gisin, O. Guinnard, G. Ribordy, and H. Zbinden, New J. Phys. **4**, 41 (2002).
[11] H. Chau, Phys. Rev. A **60**, 1966 (1999); H. Chau, Phys. Rev. A **58**, 905 (1998).
[12] H. Ollivier and J.-P. Tillich (to be published).
[13] D. Gottesman, Ph.D. thesis, California Institute of Technology, 1997; arXiv:quant-ph/9705052.
[14] R. Johannesson and K. Zigangirov, *Fundamentals of Convolutional Coding* (IEEE, New York, 1999).
[15] L. H. C. Lee, *Convolutional Coding: Fundamentals and Applications* (Artech House, Boston, 1997).
[16] J. L. Massey and M. K. Sain, IEEE Trans. Comput. **C17**, 330 (1968).