9-25-2013

Recursive quantum convolutional encoders are catastrophic: A simple proof

Monireh Houshmand
Imamreza University of Mashhad

Mark M. Wilde
Université McGill

Recommended Citation

Houshmand, M., & Wilde, M. (2013). Recursive quantum convolutional encoders are catastrophic: A simple proof. *IEEE Transactions on Information Theory, 59* (10), 6724-6731. https://doi.org/10.1109/TIT.2013.2272932

Recursive quantum convolutional encoders are catastrophic: A simple proof

Monireh Houshmand
Electrical Engineering Department, Imam Reza International University, Mashhad, Iran

Mark M. Wilde
School of Computer Science, McGill University, Montreal, Quebec H3A 2A7, Canada

October 24, 2018

Abstract

Poulin, Tillich, and Ollivier discovered an important separation between the classical and quantum theories of convolutional coding, by proving that a quantum convolutional encoder cannot be both non-catastrophic and recursive. Non-catastrophicity is desirable so that an iterative decoding algorithm converges when decoding a quantum turbo code whose constituents are quantum convolutional codes, and recursiveness is as well so that a quantum turbo code has a minimum distance growing nearly linearly with the length of the code, respectively. Their proof of the aforementioned theorem was admittedly "rather involved," and as such, it has been desirable since their result to find a simpler proof. In this paper, we furnish a proof that is arguably simpler. Our approach is group-theoretic—we show that the subgroup of memory states that are part of a zero physical-weight cycle of a quantum convolutional encoder is equivalent to the centralizer of its "finite-memory" subgroup (the subgroup of memory states which eventually reach the identity memory state by identity operator inputs for the information qubits and identity or Pauli-Z operator inputs for the ancilla qubits). After proving that this symmetry holds for any quantum convolutional encoder, it easily follows that an encoder is non-recursive if it is non-catastrophic. Our proof also illuminates why this no-go theorem does not apply to entanglement-assisted quantum convolutional encoders—the introduction of shared entanglement as a resource allows the above symmetry to be broken.

1 Introduction

Quantum convolutional coding is an approach to quantum error correction that allows a sender to encode a stream of quantum data in an online fashion, by the repeated application of some encoding unitary [7]. As soon as the qubits have been encoded, the sender can then transmit them over a noisy quantum channel to a receiver, whereupon the receiver can begin to decode the qubits as he receives them. An advantage of this approach is that the encoding complexity is linear in the number of qubits that are encoded, and the most clear application of these codes is in the context of quantum communication. One can also realize a quantum turbo code by serially concatenating two quantum convolutional encoders [8]. Quantum turbo codes have linear encoding and decoding complexity, and they are known (from numerical simulations) to exhibit good performance when the noisy channel is a depolarizing channel [8, 11].

The notions of a state diagram, catastrophicity, and recursiveness are prominent in the classical theory of convolutional coding [9, 4, 6], and Poulin, Tillich, and Ollivier (PTO) were the first to provide satisfying definitions of these notions for a quantum convolutional encoder [8]. In fact, a casual glance at the development in Ref. [8] might lead one to believe that there is little difference between the classical and quantum theories of convolutional coding, given that the classical definitions of state diagram, catastrophicity, and recursiveness were essentially imported and then adapted to the quantum setting.

In spite of the apparent similarities, PTO found a striking separation between the classical and quantum theories of convolutional coding: in the quantum case, encoders cannot be both non-catastrophic and recursive [8]. Recall that in the classical case, convolutional encoders can certainly possess both properties, and in fact, one looks for encoders satisfying both when constructing classical turbo codes [2]. Non-catastrophicity ensures that an iterative decoding algorithm for a turbo code converges, while recursiveness of one of the convolutional encoders ensures that the turbo code has a minimum distance growing nearly linearly with the number of qubits being encoded (this is true for both the classical and quantum cases [5, 8, 1]). The important theorem of PTO was then later extended to apply to quantum convolutional encoders for subsystem codes [10], but Ref. [10] also pointed out that this theorem does not apply if a sender and receiver share entangled Bell states before communication begins (in fact, the theory of entanglement-assisted quantum convolutional coding bears more similarities with the classical theory than does the "unassisted" theory of quantum convolutional coding).

The purpose of the present paper is to simplify the proof of the PTO theorem, stated as "recursive quantum convolutional encoders are catastrophic." One might deem it unnecessary to do so given that PTO have already provided a proof, but it is often the case that if a theorem is sufficiently important, then alternate proofs can provide more insight into the theorem itself, perhaps spark new developments, or be useful for pedagogical purposes. Since this theorem gives such a strong separation between the classical and quantum theories, it clearly stands as one of the most important results in the theory of quantum convolutional coding. In the case of the PTO theorem, a quick glance over its original proof indicates that it is complicated. Indeed, Ref. [8] states that "the proof of Theorem 1 is rather involved," and as such, they leave its many details to an appendix.

We now provide a brief summary of our proof of the PTO theorem for the expert in quantum convolutional coding (this summary should become more understandable for the non-expert after reviewing the background material in the next section). First, it is apparently simpler to prove the logically-equivalent contrapositive of the PTO theorem. That is, we prove the statement "If a quantum convolutional encoder is non-catastrophic, then it is non-recursive." Our approach relies heavily on the use of group theory. We define a finite-memory subgroup $\mathcal{F}_0$, which consists of all the memory states of a quantum convolutional encoder that eventually reach the identity memory state via a path in the state diagram resulting from applying combinations of identity and Pauli-Z operators on the ancilla qubits and identity operators on the information qubits (we call such paths "finite standard paths").[1] We define the infinite-memory set to consist of all memory states that do not have this property. This set on its own is not a subgroup, but if we quotient the full Pauli group acting on the memory qubits by the finite-memory subgroup, then we obtain a subgroup of infinite-memory states that obey the group property of staying within the infinite-memory set

[1] This formulation of the finite-memory subgroup is different from that in Ref. [8], and it is a further step that is helpful in simplifying the proof of the PTO theorem.

when they are multiplied by each other. We let $\mathcal{I}_0$ denote this subgroup, and for simplicity, we refer to it as the "infinite-memory subgroup." We then show that all memory states in the finite-memory subgroup $\mathcal{F}_0$ commute with the memory states that are part of a zero physical-weight cycle (these latter states form a subgroup that we denote by $\mathcal{P}_0$). After doing so, we prove the converse statement, namely, that any memory state which commutes with all elements of the finite-memory subgroup is part of some zero-physical weight cycle. This establishes that the zero physical-weight cycle subgroup $\mathcal{P}_0$ is equivalent to the centralizer $C(\mathcal{F}_0)$ of the finite-memory subgroup $\mathcal{F}_0$. This last observation easily leads to the statement of the theorem. For a non-catastrophic encoder, any weight-one logical input drives it to a memory state that commutes with all the elements of $C(\mathcal{F}_0)$, implying that this memory state belongs to $\mathcal{F}_0$ (here we are invoking a double centralizer theorem that applies for the Pauli group). Since we can conclude that this memory state belongs to $\mathcal{F}_0$, it follows that there is a finite standard path from it to the identity memory state, and as such, a non-catastrophic encoder is non-recursive.

We structure this paper as follows. In the next section, we review the definition of a quantum convolutional code, a quantum convolutional encoder, a state diagram, catastrophicity, and recursiveness. Ref. [8] established these definitions, but we repeat them here for convenience and invite the expert to skip the next section. Section 3 presents our proof of the PTO theorem along the lines outlined in the previous paragraph. Section 4 provides a brief discussion to conclude this paper.

2 Background

2.1 Quantum convolutional codes

We begin by recalling some standard facts, and then we review the definition of a quantum convolutional code. A Pauli sequence is a countably-infinite tensor product of Pauli matrices:

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

where each operator A_i in the sequence is an element of the Pauli group $\Pi \equiv \{I, X, Y, Z\}$.[2] Let $\Pi^{\mathbb{Z}^+}$ denote the set of all Pauli sequences. A Pauli sequence is finite-weight if only finitely many operators A_i in the sequence are equal to X, Y, or Z, and it is an infinite-weight sequence otherwise.

Definition 1 (Quantum Convolutional Code) *A rate-k/n quantum convolutional code admits a representation with a basic set $\mathcal{G}_0$ of $n-k$ generators and all of their n-qubit shifts:*

$$\mathcal{G}_0 \equiv \left\{ \mathbf{G}_i \in \Pi^{\mathbb{Z}^+} : 1 \leq i \leq n-k \right\}.$$

In order to form a quantum convolutional code, these generators should commute with themselves and all of the n-qubit shifts of themselves and the other generators.

[2] In our work, to be precise, we are dealing with the quotient of the Pauli group by its center, so that global phases are irrelevant. For simplicity, in this paper we refer to the quotient of the Pauli group by its center as "the Pauli group."

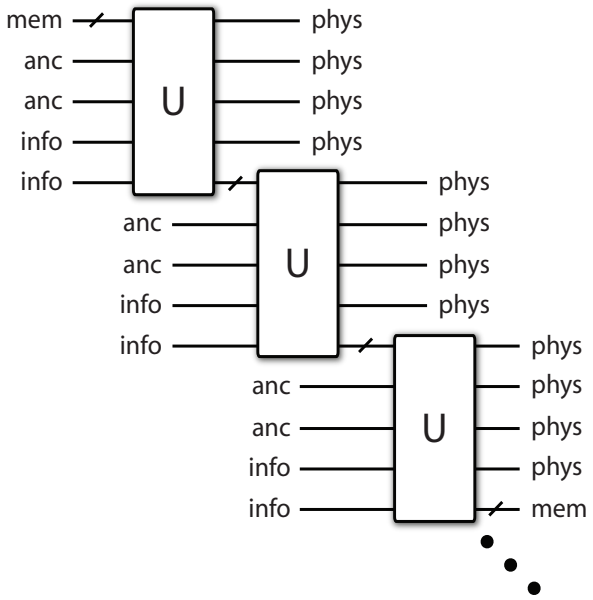

Figure 1: The encoder U for a quantum convolutional code that has four physical qubits for every two information qubits. The encoder U acts on m memory qubits, two ancilla qubits, and two information qubits to produce four output physical qubits to be sent over the channel and m output memory qubits to be fed into the next round of encoding.

Equivalently, a rate-k/n quantum convolutional code is specified by $n-k$ generators h_1, h_2, ..., h_{n-k}, where

$$\begin{array}{ccccccccc} h_1 & = & h_{1,1} & | & h_{1,2} & | & \cdots & | & h_{1,d_1} \\ h_2 & = & h_{2,1} & | & h_{2,2} & | & \cdots & | & h_{2,d_2} \\ \vdots & & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots \\ h_{n-k} & = & h_{n-k,1} & | & h_{n-k,2} & | & \cdots & | & h_{n-k,d_{n-k}} \end{array} . \tag{1}$$

Each entry $h_{i,j}$ is an n-qubit Pauli operator and d_i is the degree of generator h_i (in general, the degrees d_i can be different from each other). We obtain the other generators of the code by shifting the above generators to the right by multiples of n qubits. (In the above, note that the entries h_{1,d_1}, h_{2,d_2}, ..., $h_{n-k,d_{n-k}}$ are not required to be in the same column, but we have written them in the above way for convenience.) Note also that the generators could have an infinite degree d_i, but in this case, they should have some periodicity. We can obtain infinite-weight generators from a finite-weight representation, for example, by multiplying the first generator by the second one and all of its shifts, and this gives a different representation of the same code.

2.2 Quantum convolutional encoders

Figure 1 depicts an example of an encoder for a quantum convolutional code. The encoder depicted there has four physical qubits for every two information qubits. The unencoded qubit stream might have the following form:

$$|0\rangle|0\rangle|\phi_1\rangle|\phi_2\rangle|0\rangle|0\rangle|\phi_3\rangle|\phi_4\rangle\cdots , \tag{2}$$

so that an ancilla qubit appears as every first and second qubit and an information qubit appears as every third and fourth qubit (generally, these information qubits can be entangled with each other and even with an inaccessible reference system, but we write them as product states for simplicity).

More generally, a convolutional encoder acts on some number m of memory qubits, $n-k$ ancilla qubits, and k information qubits, and it produces n output physical qubits and m output memory qubits to be fed into the next round of encoding. It should transform an unencoded Pauli Z operator acting on the i^{th} ancilla qubit to the i^{th} stabilizer generator h_i in (1). That is, it should be some Clifford transformation.[3] The first application of the encoder U results in some Pauli operator $g_{i,1}$ acting on the m output memory qubits. The second application of the encoder U results in some other Pauli operator $g_{i,2}$ acting on the m output memory qubits and so on. The shift invariance of the overall encoding guarantees that shifts of the unencoded Z Pauli operators transform to appropriate shifts of the generators. A convolutional encoder for the code performs the following transformation:

$$
\begin{array}{c|c|c}
\text{Mem.} & \text{Anc.} & \text{Info.} \\
\hline\hline
I^{\otimes m} & Z(1) & I^{\otimes k} \\
g_{1,1} & I^{\otimes n-k} & I^{\otimes k} \\
\vdots & \vdots & \vdots \\
g_{1,d_1-2} & I^{\otimes n-k} & I^{\otimes k} \\
g_{1,d_1-1} & I^{\otimes n-k} & I^{\otimes k} \\
\hline
I^{\otimes m} & Z(2) & I^{\otimes k} \\
g_{2,1} & I^{\otimes n-k} & I^{\otimes k} \\
\vdots & \vdots & \vdots \\
g_{2,d_2-2} & I^{\otimes n-k} & I^{\otimes k} \\
g_{2,d_2-1} & I^{\otimes n-k} & I^{\otimes k} \\
\hline
\vdots & \vdots & \vdots \\
\hline
I^{\otimes m} & Z(s) & I^{\otimes k} \\
g_{s,1} & I^{\otimes n-k} & I^{\otimes k} \\
\vdots & \vdots & \vdots \\
g_{s,d_s-2} & I^{\otimes n-k} & I^{\otimes k} \\
g_{s,d_s-1} & I^{\otimes n-k} & I^{\otimes k}
\end{array}
\rightarrow
\begin{array}{c|c}
\text{Phys.} & \text{Mem.} \\
\hline\hline
h_{1,1} & g_{1,1} \\
h_{1,2} & g_{1,2} \\
\vdots & \vdots \\
h_{1,d_1-1} & g_{1,d_1-1} \\
h_{1,d_1} & I^{\otimes m} \\
\hline
h_{2,1} & g_{2,1} \\
h_{2,2} & g_{2,2} \\
\vdots & \vdots \\
h_{2,d_2-1} & g_{2,d_2-1} \\
h_{2,d_2} & I^{\otimes m} \\
\hline
\vdots & \vdots \\
\hline
h_{s,1} & g_{s,1} \\
h_{s,2} & g_{s,2} \\
\vdots & \vdots \\
h_{s,d_s-1} & g_{s,d_s-1} \\
h_{s,d_s} & I^{\otimes m}
\end{array}
\tag{3}
$$

where, as a visual aid, we have separated the Pauli operators acting on the memory qubits, ancilla qubits, and information qubits at the input with a vertical bar and we have done the same for those acting on the physical qubits and memory qubits at the output. The parameter m is the number of memory qubits, k is the number of information qubits, $n-k$ is the number of ancilla qubits, and we make the abbreviation $s \equiv n-k$. A horizontal bar separates the entries of the encoder needed to encode the first generator from the entries needed to encode the second generator. Each $g_{i,j}$ is a Pauli operator acting on some number m of memory qubits—these operators should be consistent with the input-output commutation relations of the encoder [3]. We stress that the above input-output relations only partially specify the encoder such that it produces a code with the stabilizer generators in (1), and there is still a fair amount of freedom remaining in the encoding. Note that it could be the case that the encoder transforms a Pauli-Z operator acting on an ancilla qubit to

[3] A Clifford transformation is a unitary operator that preserves the Pauli group under unitary conjugation.

a Pauli operator with infinite weight (though, the resulting operator would have some periodicity because the encoder acts on a finite number of qubits).

2.3 State diagram

The state diagram for a quantum convolutional encoder is the most important tool for determining whether it is catastrophic or recursive [8]. The state diagram is similar to one for a classical encoder [9, 4, 6], with an important exception for the quantum case that incorporates the fact that the logical operators of a quantum code are unique up to multiplication by the stabilizer generators. The state diagram allows us to analyze the "flow" of the logical operators through the quantum convolutional encoder.

Definition 2 (State Diagram) *The state diagram for a quantum convolutional encoder is a directed multigraph with 4^m vertices that we can think of as "memory states," where m is the number of memory qubits in the encoder. Each memory state corresponds to an m-qubit Pauli operator M that acts on the memory qubits. We connect two vertices M and M' with a directed edge from M to M' and label this edge as (L, P) if the encoder takes the m-qubit Pauli operator M, an $(n-k)$-qubit Pauli operator $S \in \{I, Z\}^{n-k}$ acting on the $n-k$ ancilla qubits, and a k-qubit Pauli operator L acting on the information qubits, to an n-qubit Pauli operator P acting on the n physical qubits and an m-qubit Pauli operator M' acting on the m memory qubits:*

$$\begin{array}{c|c|c} \text{Mem.} & \text{Anc.} & \text{Info.} \\ \hline M & S & L \end{array} \quad \xrightarrow{\textit{encoder}} \quad \begin{array}{c|c} \text{Phys.} & \text{Mem.} \\ \hline P & M' \end{array} \quad .$$

The labels L and P are the respective logical and physical labels of the edge.

2.4 Catastrophicity

We now review the definition of catastrophicity from Ref. [8], which is based on the classical notion of catastrophicity from Refs. [9, 6]. The essential idea behind catastrophic error propagation is that an error with finite weight, after being fed through the inverse of the encoder, could propagate infinitely throughout the decoded information qubit stream without triggering syndromes corresponding to these errors. The only way that this catastrophic error propagation can occur is if there is some cycle in the state diagram where all of the edges along the cycle have physical labels equal to the identity operator, while at least one of the edges has a logical label that is not equal to the identity. If such a cycle exists, it implies that the finite-weight channel error produces an infinite-weight information qubit error without triggering syndrome bits corresponding to this error (if it did trigger syndrome bits, this cycle would not be in the state diagram), and an iterative decoding algorithm such as that presented in Ref. [8] is not able to detect these errors. So, we can now state the definition of a catastrophic encoder.

Definition 3 (Catastrophic Encoder) *A quantum convolutional encoder acting on memory qubits, information qubits, and ancilla qubits is catastrophic if there exists a cycle in its state diagram where all edges in the cycle have zero physical weight, but there is at least one edge in the cycle with non-zero logical weight.*

2.5 Recursiveness

Recursiveness or lack thereof is a fundamental property of a quantum convolutional encoder as discussed in Refs. [8, 1].

Definition 4 (Path) *A path in the state diagram is a sequence of memory states M_1, M_2, ... (with corresponding edges) such that $M_i \to M_{i+1}$ belongs to the state diagram.*

Definition 5 (Recursive encoder) *An admissable path is a path in the state diagram for which its first edge is not part of a zero physical-weight cycle. Consider any vertex belonging to a zero physical-weight loop and any admissable path beginning at this vertex that also has logical weight one. The encoder is recursive if all such paths do not contain a zero physical-weight loop.*

We can gain some intuition behind the above definition by recalling the definition of a recursive classical convolutional encoder. In the classical case, an encoder is recursive if it has an infinite impulse response—that is, if it outputs an infinite-weight, periodic sequence in response to an input consisting of a single "one" followed by an infinite number of "zeros." Definition 5 above for the quantum case ensures that the response to a single non-identity Pauli operator (one of $\{X, Y, Z\}$) at a single logical input along with the identity operator at all other logical inputs leads to a periodic output sequence of Pauli operators with infinite weight. Though, the definition above ensures that this is not only the case for the above sequence but also for one in which the ancilla qubit inputs can be chosen arbitrarily from $\{I, Z\}$. Thus, it is a much more stringent condition for a quantum convolutional encoder to be recursive.

3 Main result: A simple proof that recursive quantum convolutional encoders are catastrophic

We now provide a proof of the PTO theorem. Our strategy is to show that the subgroup consisting of memory states in zero physical-weight cycles is equivalent to the centralizer of the finite-memory subgroup (which we define below). After doing so, it follows rather easily that a non-catastrophic encoder must be non-recursive.

A zero physical-weight cycle has the following structure:

$$\begin{array}{c|c|c} \text{Mem.} & \text{Anc.} & \text{Info.} \\ \hline\hline M_1 & S_1 & L_1 \\ M_2 & S_2 & L_2 \\ \vdots & \vdots & \vdots \\ M_p & S_p & L_p \end{array} \quad \to \quad \begin{array}{c|c} \text{Phys.} & \text{Mem.} \\ \hline\hline I^{\otimes n} & M_2 \\ I^{\otimes n} & M_3 \\ \vdots & \vdots \\ I^{\otimes n} & M_1 \end{array} \quad , \tag{4}$$

where $M_1, \ldots, M_p$ are arbitrary m-qubit Pauli operators acting on the m memory qubits, the operators $S_i \in \{I, Z\}^{\otimes(n-k)}$ act on the $n-k$ ancilla qubits, and the operators L_i are arbitrary k-qubit Pauli operators acting on the k information qubits. A zero physical-weight cycle with non-zero logical weight is a zero physical-weight cycle such that at least one of the operators L_i is not equal to the identity operator.

Let $\mathcal{P}_0$ denote the subgroup of the Pauli group on m qubits consisting of memory states that are part of a zero physical-weight cycle. That $\mathcal{P}_0$ is a subgroup follows from the observation that

if M_1 is part of a zero physical-weight cycle and if M_2 is part of a zero physical-weight cycle, then $M_1 \cdot M_2$ must be as well because we can combine the cycles of which M_1 and M_2 are a part simply by multiplying each transition of the individual cycles to get the new transitions for the zero physical-weight cycle of which $M_1 \cdot M_2$ is involved.

Definition 6 (Standard path) *Consider a given memory state M_0. A standard path originating from M_0 is a path with the following structure:*

$$\begin{array}{c|c|c} \textit{Mem.} & \textit{Anc.} & \textit{Info.} \\ \hline\hline M_0 & S_0 & I^{\otimes k} \\ M_1 & S_1 & I^{\otimes k} \\ M_2 & S_2 & I^{\otimes k} \\ \vdots & \vdots & \vdots \end{array} \rightarrow \begin{array}{c|c} \textit{Phys.} & \textit{Mem.} \\ \hline\hline P_1 & M_1 \\ P_2 & M_2 \\ P_3 & M_3 \\ \vdots & \vdots \end{array} , \tag{5}$$

where $S_i \in \{I, Z\}^{\otimes(n-k)}$ and $P_i \in \{I, X, Y, Z\}^{\otimes n}$. From the above, we can see that a standard path originating from M_0 is such that the output memory state of the i^{th} row is the same as the input memory state of the $(i+1)^{th}$ row, with the ancilla operators chosen from $\{I, Z\}^{\otimes(n-k)}$ and the input logical operators equal to the identity for every transition.

We now define two types of standard paths: finite standard paths and infinite standard paths.

Definition 7 (Finite standard path) *A finite standard path is a standard path which has "finite duration," i.e., it ends in the identity memory state after traversing a finite number of memory states for some $S_i \in \{I, Z\}^{\otimes(n-k)}$:*

$$\begin{array}{c|c|c} \textit{Mem.} & \textit{Anc.} & \textit{Info.} \\ \hline\hline M_0 & S_0 & I^{\otimes k} \\ M_1 & S_1 & I^{\otimes k} \\ M_2 & S_2 & I^{\otimes k} \\ \vdots & \vdots & \vdots \\ M_{t-2} & S_{t-2} & I^{\otimes k} \\ M_{t-1} & S_{t-1} & I^{\otimes k} \end{array} \rightarrow \begin{array}{c|c} \textit{Phys.} & \textit{Mem.} \\ \hline\hline P_1 & M_1 \\ P_2 & M_2 \\ P_3 & M_3 \\ \vdots & \vdots \\ P_{t-1} & M_{t-1} \\ P_t & I^{\otimes m} \end{array} , \tag{6}$$

where M_0, M_1, ..., M_{t-1} are not equal to the identity. A finite-memory state is a memory state from which a finite standard path originates. (In (6), M_0, M_1, ..., M_{t-1} are finite-memory states.)

Fact 8 (Finite-memory subgroup) *The finite-memory states form a subgroup of the Pauli group on m qubits, with the group property being that each element is the starting memory state for some finite standard path. That is, if M_1 and M_2 are finite-memory states, then the finite standard path originating from $M_1 \cdot M_2$ is constructed by multiplying the finite standard paths originating from M_1 and M_2 which individually lead to the identity memory state. Thus, this path eventually ends in the identity memory state and as such is a finite standard path. Let $\mathcal{F}_0$ denote this subgroup (we will refer to it as the finite-memory subgroup).*

Definition 9 (Infinite standard path) *An infinite standard path is a standard path which has infinite duration, i.e., it does not ever end in the identity memory state when the input on the ancilla*

qubits is chosen from the set $\{I,Z\}^{\otimes(n-k)}$ *and the input for the logical qubits is the identity* $I^{\otimes k}$*. Since memory state operators act on a finite number of qubits, the input memory state of the* i^{th} *row is the same as the output memory state of the* j^{th} *row, for some* $j \geq i$ *and* $S_0, S_1, \ldots, S_i, \ldots, S_j \in \{I,Z\}^{\otimes(n-k)}$*:*

$$\begin{array}{c|c|c} \textit{Mem.} & \textit{Anc.} & \textit{Info.} \\ \hline\hline M_0 & S_0 & I^{\otimes k} \\ M_1 & S_1 & I^{\otimes k} \\ M_2 & S_2 & I^{\otimes k} \\ \vdots & \vdots & \vdots \\ M_i & S_i & I^{\otimes k} \\ \vdots & \vdots & \vdots \\ M_j & S_j & I^{\otimes k} \end{array} \rightarrow \begin{array}{c|c} \textit{Phys.} & \textit{Mem.} \\ \hline\hline P_1 & M_1 \\ P_2 & M_2 \\ P_3 & M_3 \\ \vdots & \vdots \\ P_{i+1} & M_{i+1} \\ \vdots & \vdots \\ P_{j+1} & M_i \end{array} . \tag{7}$$

In the above, the memory states M_0*,* M_1*, . . . ,* M_j *are not equal to the identity operator. Observe that different sequences* S_0*, . . . ,* S_j *can lead to different cycles starting at some memory state* M_i*, but that for any fixed sequence* S_0*, . . . ,* S_j *we always have a cycle of this form. An infinite-memory state is a memory state which is never the starting memory state for a finite standard path.*

Fact 10 (Infinite-memory set) *The set of all infinite-memory states does not form a subgroup. Indeed, let* M_1 *be an infinite-memory state and let* M_2 *be a finite-memory state. The state* $M_3 \equiv M_1 \cdot M_2$ *is then an infinite-memory state that eventually enters the same loop as* M_1*. However, it is clear that the state* $M_3 \cdot M_1 = M_2$ *is a finite-memory state obtained by multiplying two infinite-memory states, so that all states in the infinite-memory set need not obey the group property.*

Fact 11 (Infinite-memory subgroup) *If we take the quotient of the full Pauli group acting on* m *qubits by the finite-memory subgroup, we obtain a set of infinite-memory states that do obey the group property. That is, if* M_1 *and* M_2 *are infinite-memory states in this subgroup, then every standard path originating from* $M_1 \cdot M_2$ *is constructed by multiplying every standard path originating from* M_1 *and* M_2*. The resulting paths will never end in the identity memory state no matter which operators in* $\{I,Z\}^{\otimes(n-k)}$ *are input for the ancilla qubits. Let* $\mathcal{I}_0$ *denote this subgroup (for simplicity, we will refer to it as the infinite-memory subgroup, and our convention is that the identity memory state is part of this subgroup). Thus, all memory states in this subgroup are infinite-memory and we obtain only infinite-memory states when multiplying them together.*

One can easily construct a generating set for the infinite-memory subgroup explicitly if a generating set $\{T_1, \ldots, T_l\}$ for the finite-memory subgroup is available. Indeed, we would just need to find $2m - l$ independent generators $\{U_{l+1}, \ldots, U_{2m}\}$ such that $\{T_1, \ldots, T_l, U_{l+1}, \ldots, U_{2m}\}$ constitutes a generating set for the Pauli group on the m memory qubits. The generating set $\{U_{l+1}, \ldots, U_{2m}\}$ then generates the infinite-memory subgroup mentioned above.

Observe that the finite-memory subgroup $\mathcal{F}_0$ and the infinite-memory subgroup $\mathcal{I}_0$ are independent subgroups, and together they constitute the full Pauli group acting on m qubits.

Lemma 12 *The memory states in a zero physical weight cycle commute with each element of the finite-memory subgroup* $\mathcal{F}_0$*.*

Proof. The proof of this lemma is similar to the proof of Theorem 7 in Ref. [3]. For completeness, we repeat it here. Consider that the RHS of the last row of (6) commutes with the RHS of any of the rows of (4). This then implies that the LHS of the last row of (6) commutes with the LHS of any of the rows of (4). Since the S_i operators already commute, this implies that $M_1, \ldots, M_p$ in (4) commute with M_{t-1} in (6). This in turn implies that the RHS of the second-to-last row of (6) commutes with the RHS of any row of (4), and furthermore, that the LHS of the second-to-last row of (6) commutes with the LHS of any row of (4). So we have that $M_1, \ldots, M_p$ in (4) commutes with M_{t-2} in (6). Continuing this argument inductively, we can conclude the statement of the lemma. ■

Let $C(\mathcal{F}_0)$ denote the subgroup consisting of memory states that commute with each element of the finite-memory subgroup $\mathcal{F}_0$ (so that $C(\mathcal{F}_0)$ is the *centralizer* of $\mathcal{F}_0$). The above lemma states that the zero physical-weight cycle subgroup is a subset of the centralizer of the finite-memory subgroup: $\mathcal{P}_0 \subseteq C(\mathcal{F}_0)$. The rest of our proof is dedicated to proving the other containment: $C(\mathcal{F}_0) \subseteq \mathcal{P}_0$, implying that the zero physical-weight cycle subgroup $\mathcal{P}_0$ is equivalent to the centralizer of the finite-memory subgroup: $\mathcal{P}_0 = C(\mathcal{F}_0)$. Having this will then allow us to easily prove that all non-catastrophic encoders are non-recursive.

Now consider the following transformation corresponding to the response of the encoder to a Pauli-Z operator on one of the ancilla qubits:

$$\begin{array}{c|c|c} \text{Mem.} & \text{Anc.} & \text{Info.} \\ \hline\hline I^{\otimes m} & Z(i) & I^{\otimes k} \end{array} \quad \rightarrow \quad \begin{array}{c|c} \text{Phys.} & \text{Mem.} \\ \hline\hline h_{i,1} & g_{i,1} \end{array} \quad , \tag{8}$$

where $i \in \{1, \ldots, n-k\}$. The encoder acts on a finite number of qubits, so it must either end in the identity memory state or cycle after following a path from $g_{i,1}$ in which the identity operator is always input for both the ancilla and information qubits. Thus, we either have

$$\begin{array}{c|c|c} \text{Mem.} & \text{Anc.} & \text{Info.} \\ \hline\hline I^{\otimes m} & Z(i) & I^{\otimes k} \\ g_{i,1} & I^{\otimes (n-k)} & I^{\otimes k} \\ g_{i,2} & I^{\otimes (n-k)} & I^{\otimes k} \\ \vdots & \vdots & \vdots \\ g_{i,d_i-1} & I^{\otimes (n-k)} & I^{\otimes k} \end{array} \quad \rightarrow \quad \begin{array}{c|c} \text{Phys.} & \text{Mem.} \\ \hline\hline h_{i,1} & g_{i,1} \\ h_{i,2} & g_{i,2} \\ h_{i,3} & g_{i,3} \\ \vdots & \vdots \\ h_{i,d_i} & I^{\otimes m} \end{array} \quad ,$$

in which case the input $I^{\otimes m} \,|\, Z(i) \,|\, I^{\otimes k}$ leads to a finite standard path, or we have

$$\begin{array}{c|c|c} \text{Mem.} & \text{Anc.} & \text{Info.} \\ \hline\hline I^{\otimes m} & Z(i) & I^{\otimes k} \\ g_{i,1} & I^{\otimes n-k} & I^{\otimes k} \\ \vdots & \vdots & \vdots \\ g_{i,j-1} & I^{\otimes n-k} & I^{\otimes k} \\ \vdots & \vdots & \vdots \\ g_{i,t} & I^{\otimes n-k} & I^{\otimes k} \\ g_{i,j} & I^{\otimes n-k} & I^{\otimes k} \\ \vdots & \vdots & \vdots \\ g_{i,t} & I^{\otimes n-k} & I^{\otimes k} \\ \vdots & \vdots & \vdots \end{array} \quad \rightarrow \quad \begin{array}{c|c} \text{Phys.} & \text{Mem.} \\ \hline\hline h_{i,1} & g_{i,1} \\ h_{i,2} & g_{i,2} \\ \vdots & \vdots \\ h_{i,j} & g_{i,j} \\ \vdots & \vdots \\ h_{i,t+1} & g_{i,j} \\ h_{i,j+1} & g_{i,j+1} \\ \vdots & \vdots \\ h_{i,t+1} & g_{i,j} \\ \vdots & \vdots \end{array} \quad ,$$

so that the encoder enters some cycle. In this latter case, the following input sequence drives the encoder back to the identity memory state:

Mem.	Anc.	Info.		Phys.	Mem.
$I^{\otimes m}$	$Z(i)$	$I^{\otimes k}$		$h_{i,1}$	$g_{i,1}$
$g_{i,1}$	$I^{\otimes n-k}$	$I^{\otimes k}$		$h_{i,2}$	$g_{i,2}$
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$
$g_{i,j-1}$	$I^{\otimes n-k}$	$I^{\otimes k}$		$h_{i,j}$	$g_{i,j}$
$g_{i,j}$	$I^{\otimes n-k}$	$I^{\otimes k}$	$\rightarrow$	$h_{i,j+1}$	$g_{i,j+1}$
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$
$g_{i,t-j+1}$	$Z(i)$	$I^{\otimes k}$		$h_{i,t-j+2}$	$g_{i,t-j+2} \cdot g_{i,1}$
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$
$g_{i,t-1} \cdot g_{i,j-2}$	$I^{\otimes n-k}$	$I^{\otimes k}$		$h_{i,t}$	$g_{i,t} \cdot g_{i,j-1}$
$g_{i,t} \cdot g_{i,j-1}$	$I^{\otimes n-k}$	$I^{\otimes k}$		$h_{i,t+1}$	$g_{i,j} \cdot g_{i,j} = I^{\otimes m}$

So the memory states resulting from the input $I^{\otimes m} \mid Z(i) \mid I^{\otimes k}$ are always part of the finite-memory subgroup $\mathcal{F}_0$, and based on Lemma 12, these memory states commute with the memory states in any zero-physical-weight cycle.

Lemma 13 *Suppose that a memory state $M' \in C(\mathcal{F}_0)$ (that is, it commutes with each element of $\mathcal{F}_0$). Then there is a unique memory state $M \in C(\mathcal{F}_0)$, an $S \in \{I, Z\}^{\otimes(n-k)}$, and an $L \in \{I, X, Y, Z\}^{\otimes k}$ such that the encoder transforms $M \mid S \mid L$ as follows:*

$$M \mid S \mid L \rightarrow I^{\otimes n} \mid M'. \tag{9}$$

Proof. First, the transition in (9) is part of the encoder because we can always act with the inverse of the encoder on $I^{\otimes n} \mid M'$, producing some unique $M \mid S \mid L$ as the input operators. Then, by assumption, $M' \in C(\mathcal{F}_0)$, implying that the RHS of (9) commutes with the RHS of (8) for all $i \in \{1, \ldots, n-k\}$ because $g_{i,1} \in \mathcal{F}_0$. Thus, the LHS of (9) should commute with the LHS of (8) for all $i \in \{1, \ldots, n-k\}$. This implies that $S \in \{I, Z\}^{\otimes(n-k)}$. Also, M commutes with all states in $\mathcal{F}_0$ ($M \in C(\mathcal{F}_0)$) for the same reason that the zero physical weight cycles from Lemma 12 commute with the finite-memory subgroup $\mathcal{F}_0$. This is due to the particular form in (9), the need for input-output commutativity consistency between (9) and (6), and the fact that $M' \in C(\mathcal{F}_0)$. ■

Lemma 14 *Suppose that a memory state $M \in C(\mathcal{F}_0)$ (that is, it commutes with each element of $\mathcal{F}_0$). Then there is a unique memory state $M' \in C(\mathcal{F}_0)$, an $S \in \{I, Z\}^{\otimes(n-k)}$, and an $L \in \{I, X, Y, Z\}^{\otimes k}$ such that the encoder transforms $M \mid S \mid L$ as follows:*

$$M \mid S \mid L \rightarrow I^{\otimes n} \mid M'. \tag{10}$$

Proof. Let $\{T_1, \ldots, T_l\}$ be a generating set for the finite-memory subgroup $\mathcal{F}_0$ and let $\{U_{l+1}, \ldots, U_{2m}\}$ be a generating set for the infinite-memory subgroup $\mathcal{I}_0$. For every $i \in \{l+1, \ldots, 2m\}$, there exists a P_i and T_i such that

$$U_i \mid I^{\otimes(n-k)} \mid I^{\otimes k} \rightarrow P_i \mid T_i. \tag{11}$$

Furthermore, the T_i operators form a generating set $\{T_{l+1}, \ldots, T_{2m}\}$ for $\mathcal{I}_0$. If it were not so (that these T_i operators were not independent), we would be able to take certain multiplicative

combinations of the U_i operators at the input and construct a path to the identity memory state, contradicting the fact that the U_i operators form a generating set for elements of the infinite-memory subgroup $\mathcal{I}_0$. Thus, $\{T_{l+1}, \ldots, T_{2m}\}$ is a generating set for $\mathcal{I}_0$ as well.

Now, we fix M' to be the unique Pauli operator on m qubits that commutes with each element of $\{T_1, \ldots, T_l\}$ and such that its commutation relation with each T_i in $\{T_{l+1}, \ldots, T_{2m}\}$ is the same as the commutation relation of M with the corresponding U_i in $\{U_{l+1}, \ldots, U_{2m}\}$ (the correspondence between U_i and T_i being set by (11)). The uniqueness of M' follows from the fact that specifying the commutation relations of an m-qubit Pauli operator with each element of a generating set for the Pauli group on m qubits completely specifies that operator up to an irrelevant global phase. Additionally, the above choice for M' implies that $M' \in C(\mathcal{F}_0)$.

To obtain the conclusion of the lemma, we consider applying the inverse of the encoder to $I^{\otimes n} \mid M'$. By Lemma 13, there exists an $M'' \in C(\mathcal{F}_0)$, an $S \in \{I, Z\}^{\otimes(n-k)}$, and an $L \in \{I, X, Y, Z\}^{\otimes k}$ such that

$$M'' \mid S \mid L \rightarrow I^{\otimes n} \mid M'. \tag{12}$$

But this M'' must be equal to M by construction—the encoder preserves the commutation relations between the RHSs of (11-12) and the LHSs of (11-12) and we already know that $M'', M' \in C(\mathcal{F}_0)$. As we stated above, these $2m$ commutation relations completely specify M'' and we can thus conclude that $M'' = M$. This concludes the proof of the lemma since we have constructed a unique $M' \in C(\mathcal{F}_0)$, $S \in \{I, Z\}^{\otimes(n-k)}$, and $L \in \{I, X, Y, Z\}^{\otimes k}$ satisfying (10) for all $M \in C(\mathcal{F}_0)$. ■

Remark 15 *The above proof works just as well by replacing all the transitions in (11) with the following ones:*

$$U_i \mid S' \mid I^{\otimes k} \rightarrow P_i' \mid T_i',$$

for some fixed $S' \in \{I, Z\}^{\otimes(n-k)}$.

Lemma 16 *Suppose that a memory state* $M \in C(\mathcal{F}_0)$. *Then* M *is a part of a zero physical-weight cycle, such that all memory states of this cycle are in* $C(\mathcal{F}_0)$.

Proof. From Lemma 14, we know that there is a unique memory state $M_1 \in C(\mathcal{F}_0)$, an $S_1 \in \{I, Z\}^{\otimes(n-k)}$, and an $L_1 \in \{I, X, Y, Z\}^{\otimes k}$ such that

$$M \mid S_1 \mid L_1 \rightarrow I^{\otimes n} \mid M_1,$$

This is also the case for M_1. That is, there exists a unique memory state $M_2 \in C(\mathcal{F}_0)$, an $S_2 \in \{I, Z\}^{\otimes(n-k)}$, and an $L_2 \in \{I, X, Y, Z\}^{\otimes k}$ such that

$$M_1 \mid S_2 \mid L_2 \rightarrow I^{\otimes n} \mid M_2.$$

This process cannot go on indefinitely (that of finding a new $M_{i+1} \in C(\mathcal{F}_0)$ that has not yet already appeared, for a given $M_i \in C(\mathcal{F}_0)$). The centralizer $C(\mathcal{F}_0)$ of the finite-memory subgroup $\mathcal{F}_0$ has a finite number of elements, and at some point, the transition is guaranteed to return back to M. Also, note that these zero physical-weight transitions cannot go back to any of M_1 through M_j before going back to M—this would contradict Lemma 13. Thus, M is part of a zero

physical-weight cycle of the following form:

$$\begin{array}{c|c|c} \text{Mem.} & \text{Anc.} & \text{Info.} \\ \hline\hline M & S_1 & L_1 \\ M_1 & S_2 & L_2 \\ \vdots & \vdots & \vdots \\ M_{j-1} & S_{j-1} & L_{j-1} \\ M_j & S_j & L_j \end{array} \quad \rightarrow \quad \begin{array}{c|c} \text{Phys.} & \text{Mem.} \\ \hline\hline I^{\otimes n} & M_1 \\ I^{\otimes n} & M_2 \\ \vdots & \vdots \\ I^{\otimes n} & M_j \\ I^{\otimes n} & M \end{array} \quad . \tag{13}$$

■

Corollary 17 *From Lemma 16, we conclude that all memory states of $C(\mathcal{F}_0)$ are part of some zero physical-weight cycle, and when combined with Lemma 12, we have that $\mathcal{P}_0 = C(\mathcal{F}_0)$. So if all memory states of all zero-physical weight cycles of a transformation commute with a given memory state, that memory state belongs to $\mathcal{F}_0$.*

Theorem 18 (Poulin, Tillich, Ollivier [8]) *A non-catastrophic encoder is non-recursive.*

Proof. Consider the following transition, resulting from a weight-one logical input:

$$\begin{array}{c|c|c} \text{Mem.} & \text{Anc.} & \text{Info.} \\ \hline\hline I^{\otimes m} & I^{\otimes n-k} & X(i) \end{array} \quad \rightarrow \quad \begin{array}{c|c} \text{Phys.} & \text{Mem.} \\ \hline\hline h & g \end{array} \quad . \tag{14}$$

For a non-catastrophic encoder, all of the edges in a zero physical-weight cycle have zero logical weight, so that the LHS of (14) commutes with the LHS of the transitions in these zero physical-weight cycles. Thus, the RHSs commute as well. Since zero physical-weight cycles have the identity operator acting on physical qubits, the above memory state g commutes with all memory states of zero-physical weight cycles, so that $g \in C(\mathcal{P}_0)$. According to Corollary 17, the memory state $g \in \mathcal{F}_0$, and according to Fact 8, there is a standard path from g to the identity memory state, implying that the encoder is non-recursive. ■

Remark 19 *Our proof of the above theorem illustrates a particular property of any non-catastrophic quantum convolutional encoder. Transitions of the following form*

$$\begin{array}{c|c|c} \textit{Mem.} & \textit{Anc.} & \textit{Info.} \\ \hline\hline I^{\otimes m} & S & L \end{array} \quad \rightarrow \quad \begin{array}{c|c} \textit{Phys.} & \textit{Mem.} \\ \hline\hline h_{S,L} & g_{S,L} \end{array} \quad , \tag{15}$$

for all $S \in \{I,Z\}^{\otimes n-k}$ and $L \in \{I,X,Y,Z\}^{\otimes k} \backslash \{I^{\otimes k}\}$ lead to a memory state $g_{S,L} \in \mathcal{F}_0$, from which there is a standard path to the identity memory state.

4 Discussion

We have provided a proof of the statement "recursive quantum convolutional encoders are catastrophic" that is arguably simpler than the one originally furnished by Poulin, Tillich, and Olliver in Ref. [8]. Our approach was to show that the subgroup of memory states that are part of zero physical-weight cycles is equivalent to the centralizer of the finite-memory subgroup. From there, it was straightforward to prove the PTO theorem.

The proof given here also provides insight into why entanglement-assisted quantum convolutional encoders can circumvent this no-go theorem. In particular, an examination of the example from Figure 4 of Ref. [10] reveals that its zero physical-weight cycle subgroup consists of only the identity memory state, so that its centralizer contains all memory states. Yet, the finite-memory subgroup for this example is empty. Thus, the symmetry between these two subgroups is broken by the introduction of entanglement, and the no-go theorem no longer applies.

Acknowledgements We acknowledge Jean-Pierre Tillich for suggesting to us at QIP 2011 in Singapore that it would be worthwhile to pursue a simpler proof of Theorem 18. We also acknowledge useful discussions with Min-Hsiu Hsieh on this topic, and we are grateful for the suggestions of the anonymous referee. MMW acknowledges support from the Centre de Recherches Mathématiques at the University of Montreal.

References

[1] Mamdouh Abbara and Jean-Pierre Tillich. The minimum distance of classical and quantum turbo-codes. September 2011. arXiv:1109.0215.

[2] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara. Serial concatenation of interleaved codes: performance analysis, design, and iterative decoding. *IEEE Transactions on Information Theory*, 44(3):909–926, May 1998.

[3] Monireh Houshmand, Saied Hosseini-Khayat, and Mark M. Wilde. Minimal-memory, noncatastrophic, polynomial-depth quantum convolutional encoders. *Accepted into the IEEE Transactions on Information Theory*, 2012. arXiv:1105.0649.

[4] Rolf Johannesson and Kamil Sh. Zigangirov. *Fundamentals of Convolutional Coding*. Wiley-IEEE Press, 1999.

[5] Nabil Kahale and Rüdiger Urbanke. On the minimum distance of parallel and serially concatenated codes. In *Proceedings of the International Symposium on Information Theory*, page 31, Cambridge, Massachussetts, USA, August 1998.

[6] Robert J. McEliece. *The Theory of Information and Coding*. Cambridge University Press, 2002.

[7] Harold Ollivier and Jean-Pierre Tillich. Description of a quantum convolutional code. *Physical Review Letters*, 91(17):177902, October 2003.

[8] David Poulin, Jean-Pierre Tillich, and Harold Ollivier. Quantum serial turbo-codes. *IEEE Transactions on Information Theory*, 55(6):2776–2798, June 2009.

[9] Andrew J. Viterbi. Convolutional codes and their performance in communication systems. *IEEE Transactions on Communication Technology*, 19(5):751–772, October 1971.

[10] Mark M. Wilde and Min-Hsiu Hsieh. Entanglement boosts quantum turbo codes. *Proceedings of the 2011 IEEE International Symposium on Information Theory*, pages 445–449, August 2011. Saint-Petersburg, Russia.

[11] Mark M. Wilde, Min-Hsiu Hsieh, and Zunaira Babar. Entanglement-assisted quantum turbo codes. May 2013. arXiv:1010.1256v3.