

# Degenerate Viterbi Decoding

Emilie Pelchat and David Poulin

Département de Physique, Université de Sherbrooke, Sherbrooke, Québec, Canada, J1K 2R1

**Abstract**—We present a decoding algorithm for quantum convolutional codes that finds the class of degenerate errors with the largest probability conditioned on a given error syndrome. The algorithm runs in time linear with the number of qubits. Previous decoding algorithms for quantum convolutional codes optimized the probability over individual errors instead of classes of degenerate errors. Using Monte Carlo simulations, we show that this modification to the decoding algorithm results in a significantly lower block error rate.

**Index Terms**—Convolutional-codes, Quantum error correction, Viterbi algorithm

## I. INTRODUCTION

The stabilizer formalism [1] for quantum error correction reveals many similarities between classical and quantum codes. In particular, it shows that quantum codes can be obtained from classical self-dual codes over  $\mathbb{F}_4$  [2], and inherit many of their properties. Despite this connection, there exists fundamental distinctions between classical and quantum codes, among which degeneracy stands out [3], [4], [5]. Whereas two distinct bit-flip patterns  $e$  and  $e'$  applied to a given bit string  $s$  always produces two distinct strings  $s + e$  and  $s + e'$ , it is possible in quantum mechanics to find two distinct errors  $E$  and  $E'$  that have exactly the same effect on every code state, i.e.  $E|\bar{\psi}\rangle = E'|\bar{\psi}\rangle$  for every state  $|\bar{\psi}\rangle$  in the code space  $\mathcal{C}$ . We say that errors  $E$  and  $E'$  are degenerate for the code  $\mathcal{C}$ . In fact, every quantum code has degenerate errors, but we say that a code is degenerate when it has degenerate errors that are typical.<sup>1</sup>

Degeneracy is a purely quantum mechanical effect with no classical counterpart, and leads to important distinctions between classical and quantum information theory. In particular, Shannon's bound to the capacity of a memoryless channel  $\mathcal{N}$

$$C(\mathcal{N}) \leq \max_X I(X : \mathcal{N}(X)) \quad (1)$$

is established by counting the number channel outputs that are produced by applying typical errors on all possible input codewords, and demanding that it does not exceed the total number of strings. If we apply the same counting argument to a degenerate quantum code, because some errors have the same effect on all codewords, we overestimate the number of distinct channel outputs and hence underestimate the channel capacity. Indeed, the quantum analog of the mutual information  $I(X : \mathcal{N}(X))$  for an input distribution  $X$  and channel  $\mathcal{N}$  is the coherent information  $I^c(I \otimes \mathcal{N}(\phi_{AB}))$  for a purification  $\phi_{AB}$  of an input distribution  $\phi_B$  and a quantum channel  $\mathcal{N}$ . Based

on this analogy, one would expect that the quantity

$$Q_1(\mathcal{N}) = \max_{\phi_B} I^c(I \otimes \mathcal{N}(\phi_{AB})) \quad (2)$$

upper bounds the quantum channel capacity. But due to the existence of degenerate errors, the true quantum capacity is instead given by regularized quantity [6], [7], [8]

$$Q(\mathcal{N}) = \lim_{n \rightarrow \infty} \frac{1}{n} \max_{\phi_{B^n}} I^c(I \otimes \mathcal{N}^{\otimes n}(\phi_{A^n B^n})). \quad (3)$$

The quantity  $Q_1$  can be achieved by non-degenerate codes—for Pauli noise defined below it is achieved by random stabilizer codes similar to those used in Shannon's original construction in the classical setting. However,  $Q$  requires the use of degenerate codes. Because  $Q$  involves an optimization over an input distribution correlated across many channel uses, it cannot be computed in general. However, degenerate codes of finite rate have been tailored for channels with  $Q_1 = 0$  [3], [4], [5], a striking demonstration that  $Q$  can be greater than  $Q_1$ .

To benefit from the degeneracy of a quantum code, it is necessary to take it into consideration in the decoding process. Maximum a posteriori (MAP) decoding usually consists in identifying the error with largest probability conditioned on a given error syndrome. But since degenerate errors have the same effect on all code states, they can all be corrected the same way. Hence, for a degenerate quantum code, MAP decoding should instead consist of identifying the class of degenerate errors with the largest probability conditioned on a given error syndrome, the probability of a class of errors being the sum of the probabilities of its elements. This can be substantially more complicated than standard MAP decoding. Topological codes [9], [10] are a good example of codes for which a standard MAP decoder exists [9] but no degenerate MAP decoder is known, and where we know that degeneracy can provide an advantage [11]. Thus, the design of degenerate MAP decoding algorithms is an important problem for quantum information theory.

In this article, we study this problem for quantum convolutional codes. MAP decoding of classical convolutional codes can be formally classified as a **MAX-PROD** problem, and is solved using **Viterbi's algorithm** [12]. The MAX is to optimize the conditional probability over all errors, while the PROD reflects the fact that the probability of a given error on a memoryless channel is the product of the probability of each of its components. Viterbi's algorithm makes use of the distributive law  $\max_{x,y} x \cdot y = \max_x x \cdot \max_y y$  to solve the MAP decoding problem [13] in a time that scales linearly with the length of the code. Using the stabilizer formalism, all this

<sup>1</sup>Thus, the notion of degeneracy makes implicit reference to an error model.

machinery can be imported to the quantum realm and yields a MAP decoder for quantum convolutional codes [14], [15].

A degenerate MAP decoder has a different formal structure however; the need to add-up the probabilities of all degenerate errors entails a MAX-SUM-PROD problem. By making use of the distributive law of the product over the sum and of the max over the product, together with the particular factorized structure of the problem, we conceive a generalization of Viterbi's algorithm that exactly achieves degenerate MAP decoding of quantum convolutional code in linear time. Using Monte Carlo simulations, we find that the degenerate decoder suppresses more errors than the standard non-degenerate decoder; the improvement becoming more important at low error rates. Thus, convolutional codes provide an example where degenerate MAP decoding can be performed efficiently and yield a significant performance gain.

The rest of this article is organized as follows. The next section introduces basic concepts. In Sec. III, we define the decoding problem for quantum codes, emphasizing on the role of degenerate errors. Section IV summarizes Viterbi's algorithm for quantum convolutional codes, and explains how it can be modified to take degeneracy into consideration. Lastly, we present numerical result in Sec. V. Our presentation of quantum convolutional codes and their decoding algorithms follows [15], we refer the reader to this article for more technical details on these basic concepts.

## II. DEFINITIONS

### A. Stabilizer codes

A quantum state of  $n$  qubits is specified by a vector  $|\psi\rangle$  in a  $2^n$ -dimensional vector space  $\mathcal{H}$ . A quantum code  $\mathcal{C}$  encoding  $k$  qubits into  $n$  qubits is a  $2^k$ -dimensional subspace of  $\mathcal{H}$ . The stabilizer formalism offers a compact description of such a subspace, making use of the Pauli group. The Pauli matrices are defined as follows

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \quad (4)$$

Together with the  $2 \times 2$  identity matrix  $I$  and the imaginary unit  $i$ , they form a group under multiplication, the Pauli group  $\mathcal{P}_1$ . The  $n$ -qubit Pauli group is the  $n$ -fold tensor product of the single-qubit Pauli group  $\mathcal{P}_n = \mathcal{P}_1^{\otimes n}$ . An important fact about  $\mathcal{P}_n$  is that all of its elements either commute or anti-commute, i.e. for all  $P, Q \in \mathcal{P}_n$ ,  $PQ = \pm QP$ . Using the fact that  $ZX = iY$ , we find that  $\{i, X_a, Z_a\}$  is a generating set of  $\mathcal{P}_n$ , where  $Z_a$  ( $X_a$ ) stands for the Pauli operator  $Z$  ( $X$ ) acting on qubit  $a$ , and is the identity elsewhere, i.e.

$$Z_a = \underbrace{I \otimes I \otimes \dots \otimes I}_{a-1} \otimes Z \otimes \underbrace{I \otimes I \otimes \dots \otimes I}_{n-a-1}. \quad (5)$$

A stabilizer group  $\mathcal{S}$  is an Abelian subgroup of  $\mathcal{P}_n$  that does not contain the element  $-I$ . It can be specified by  $s \leq n$  independent stabilizer generators  $\{S_a\}_{a=1,\dots,s}$ . The quantum code  $\mathcal{C}$  associated to the stabilizer group  $\mathcal{S}$  is defined by a set of eigenvalue equations

$$\mathcal{C} = \{|\bar{\psi}\rangle : S|\bar{\psi}\rangle = |\bar{\psi}\rangle, \forall S \in \mathcal{S}\}. \quad (6)$$

Note that the condition  $S_a|\bar{\psi}\rangle = |\bar{\psi}\rangle$  for all generators  $S_a$  of  $\mathcal{S}$  is enough to ensure Eq. (7). The dimension of the code  $\mathcal{C}$  is  $2^k$  with  $k = n - s$ , so we say that the code encodes  $k$  logical qubits into  $n$  physical qubits, so has rate  $\frac{k}{n}$ . Clearly, the stabilizer generators  $S_a$  play a role analogous to the linearly independent rows of the parity check matrix of a classical linear code.

The centralizer  $C(\mathcal{S})$  of  $\mathcal{S}$ —i.e. elements of  $\mathcal{P}_n$  that commute with all elements of  $\mathcal{S}$ —are called Pauli codewords. Multiplying a code state  $|\bar{\psi}\rangle \in \mathcal{C}$  by a Pauli codeword  $L \in C(\mathcal{S})$  produces another code state. This can be verified directly since  $S(L|\bar{\psi}\rangle) = L(S|\bar{\psi}\rangle) = L|\bar{\psi}\rangle$  which shows that  $L|\bar{\psi}\rangle$  is an eigenstate of eigenvalue  $+1$  of all stabilizers  $S$ , where we have used the commutativity of  $L$  and  $S$ . More generally, all code states can be generated starting from an arbitrary fiducial code state  $|\psi_0\rangle$  and multiplying it by a logical operator  $L \in \text{Alg}(C(\mathcal{S}))$  in the algebra generated by the Pauli codewords.

The encoding circuit is another method to specify a stabilizer code. The main advantages of this circuit-based definition is that all commutation constraints discussed above are automatically satisfied, and it is more suited for the definition of quantum convolutional codes. The Clifford group on  $n$  qubits is the normalizer of the Pauli group in  $U(2^n)$ . In other words, Clifford transformations consist of unitary transformation  $U$  on  $n$  qubits for which  $UPU^\dagger$  is a Pauli operator, for all Pauli operators  $P$ . Any Clifford transformation on  $n$  qubits can be decomposed into a quantum circuit composed of controlled-not gates, Hadamard gates  $H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ , and phase gates  $R = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$ . Given an  $n$ -qubit Clifford transformation  $U$ , we can construct a stabilizer code with stabilizer generators

$$S_a = UZ_aU^\dagger, \quad a = 1, \dots, s \quad (7)$$

and Pauli codewords

$$\bar{X}_a = UX_{s+a}U^\dagger, \quad \bar{Z}_a = UZ_{s+a}U^\dagger, \quad a = 1, \dots, k. \quad (8)$$

For later use, it is also convenient to define operators that we call pure errors as

$$T_a = UX_aU^\dagger, \quad a = 1, \dots, s. \quad (9)$$

Since conjugation by a unitary matrix preserves the commutation relations, the following commutation relations are a straightforward consequence of definitions Eqs. 7-9:

$$[S_a, S_b] = 0 \quad (10)$$

$$[S_a, \bar{Z}_b] = [S_a, \bar{X}_b] = 0 \quad (11)$$

$$S_a T_b = (-1)^{\delta_{ab}} T_b S_a \quad (12)$$

where Eq. (10) expresses the fact that  $\mathcal{S}$  is Abelian, Eq. (11) shows that Pauli codewords are in the center of  $\mathcal{S}$ , and Eq. (12) will be helpful in formulating the decoding problem. Also, because the operators  $\{Z_a, X_a\}$  generate  $\mathcal{P}_n$  and because completeness is not affected by a unitary transformation, it follows that  $\{i, Z_a, \bar{X}_a, S_b, T_b\}_{a=1,\dots,k,b=1,\dots,s}$  generate  $\mathcal{P}_n$ . Lastly,  $U$  is called the encoding circuit because any state obtained from the following circuit



where  $|\psi\rangle$  can be an arbitrary  $k$ -qubit state, is a code state, i.e.  $|\bar{\psi}\rangle = U(|0\rangle^{\otimes s} \otimes |\psi\rangle) \in \mathcal{C}$ . In this circuit, we call the  $s$  top input qubits the **ancillary qubits**, while the bottom  $k$  input qubits are called the **data qubits** as they carry the information prior to encoding into  $\mathcal{C}$ .

### B. Pauli noise

The natural noise models to study using stabilizer codes are Pauli noise models. In these models, the noise alters an encoded state  $|\bar{\psi}\rangle$  by multiplying it by an element of the Pauli group  $E \in \mathcal{P}_n$ , i.e.  $|\bar{\psi}\rangle \rightarrow E|\bar{\psi}\rangle$ . The errors  $E$  are chosen at random according to some probability  $P(E)$  that specifies the noise model. A common assumption is that of an **i.i.d.** noise model where  $P(E)$  is the  $n$ -fold tensor product of a distribution  $(1 - p_x - p_y - p_z, p_x, p_y, p_z)$  over the single qubit Pauli group.<sup>2</sup> The symmetric choice  $p_x = p_y = p_z = p$  is called the depolarization channel of rate  $p$ .

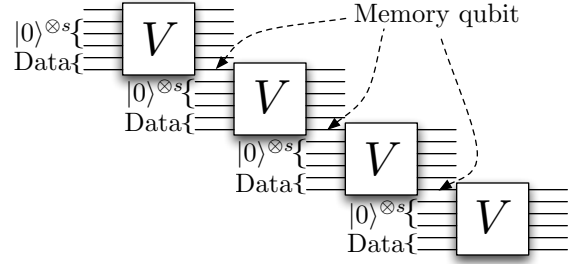
### C. Quantum convolutional codes

Quantum convolutional codes were introduced in [16], [17], but here we follow essentially the definitions of [15], and refer the reader to this article for more technical definitions. An  $(n, k, \eta)$ -quantum convolutional code is a stabilizer code with stabilizer generators of the form

$$\{S_{a,t} = I^{\otimes t \times n} \otimes S_a\}_{a=1,\dots,s; t=1,\dots,T} \quad (13)$$

where  $S_a \in \mathcal{P}_{n+\eta}$  and  $\eta$  is called the constraint length of the code. In other words, there are  $s$  distinct  $(n + \eta)$ -qubit Pauli operators  $S_a$  that are translated by integer multiples of  $n$  to generate the entire set of generators. Note that the total number of qubits used in the code is left unspecified in this definition; the maximum value  $\tau$  of  $t$  determines the length of the convolutional code. Thus, it is implicitly assumed in Eq. (13) that the operators are padded to the right with identity matrices so that they all act on  $N = n\tau + \eta$  qubits. Of course, the  $S_a$  should all commute with one another to ensure  $[S_{a,t}, S_{a',t}] = 0$ , and also with their translations by integer multiples of  $n$  to ensure  $[S_{a,t}, S_{a',t'}] = 0$ . For codes with large constraint length  $\eta$ , this last commutation condition can lead to a large number of constraints on the  $S_a$  that are difficult to fulfill. But as for stabilizer codes discussed above, **there exists an equivalent circuit-based definition of quantum convolutional codes that circumvents this difficulty.**

In this circuit-based definition, a  $(n, k, m)$ -quantum convolutional code is a stabilizer code whose encoding circuit takes the particular form



which illustrates the case  $n = 5$ ,  $k = 2$  and  $m = 1$ . The encoding circuit is composed of a seed transformation  $V$ , element of the Clifford group, that is applied periodically to non-disjoint sets of qubits. We refer to a given period of this circuit as a time frame of the code. The qubits that overlap two consecutive applications of the seed transformation are referred to as memory qubits, and there are in general  $m$  of them in every frame. In each frame (except the very first one), there are a total of  $n$  input qubits,  $k$  of which carry quantum information (marked “Data” in the circuit) and  $s = n - k$  are ancillary qubits in the state  $|0\rangle$ . The rate of the code is therefore  $\frac{k}{n}$ . A quantum convolutional code can contain an arbitrary number  $\tau$  of frames. For a given value of  $\tau$ , we obtain a code block containing a total of  $N = \tau \times n$  physical qubits and encoding  $\tau \times k$  logical qubits.<sup>3</sup>

A generating set for the stabilizer is given by  $S_a = U Z_a U^\dagger$  where  $U$  is the  $N$ -qubit Clifford transformation resulting from the entire sequence of seed transformation  $V$ , and the qubit **index  $a$  varies over all ancillary qubit index**, i.e. locations on the encoding circuit marked with qubits in a  $|0\rangle$  input state.

The notation used in the definition based on Eq. (13) and the definition based on the encoding circuit differ slightly because **there is no direct relation between the constraint length  $\eta$  and the number of memory qubits  $m$** . However, one can easily show that a convolutional code defined by a circuit with  $m$  memory qubits admits a set of generators of constraint length  $\eta \leq 4^m$ .

### III. DEGENERATE DECODING

As in the classical setting, the first step in the decoding process is the error syndrome extraction. In quantum mechanics, this is done by measuring the stabilizer generators. Remember that in quantum mechanics, the measurement of an observable—i.e. an Hermitian matrix—gives an outcome equal to the eigenvalue of the operator corresponding to the state of the system. By definition, all code states have **+1** eigenvalues for all stabilizer generators, i.e.  $S_a|\bar{\psi}\rangle = |\bar{\psi}\rangle$ . If an error  $E$  corrupts the state to  $E|\bar{\psi}\rangle$ , the eigenvalue of  $S_a$  will remain the same when  $E$  and  $S_a$  commute, and will be changed to  $-1$  when  $E$  and  $S_a$  anti commute. To see this, note that

$$S_a(E|\bar{\psi}) = \begin{cases} ES_a|\bar{\psi}\rangle = E|\bar{\psi}\rangle & \text{if } S_a E = E S_a \\ -ES_a|\bar{\psi}\rangle = -E|\bar{\psi}\rangle & \text{if } S_a E = -E S_a. \end{cases}$$

<sup>2</sup>Note that the absolute phase of the noise—e.g. whether error  $X$ , or  $iX$ , or  $-X$  was applied—has no observable effect, so our description of errors is over equivalent classes of Pauli operators defined modulo a phase.

<sup>3</sup>Depending on the way the code is terminated, there can be a constant number of extra physical qubits in the block, which decreases the code’s rate by a quantity that vanishes as  $\frac{1}{\tau}$ , see [18], [15].



The  $\pm 1$  measurement outcome of  $S_a$  gives the  $a$ th syndrome bit  $s_a$ , and we denote the collection of  $s$  syndrome bits  $\mathbf{s} = (s_1, s_2, \dots, s_s)$ .

The goal of the non-degenerate (ND) decoder can be stated quite simply: find the most probable error conditioned on the measured syndrome. If we denote  $\mathbf{s}(E) = (s_1(E), \dots, s_s(E))$  the syndrome that error  $E$  would produce, non-degenerate MAP decoding consists of the optimization problem

$$E_{\text{MAP}}^{\text{ND}}(\mathbf{s}) = \underset{E: \mathbf{s}(E)=\mathbf{s}}{\operatorname{argmax}} P(E) \quad (14)$$

where  $P(E)$  is given by the noise model, and  $\operatorname{argmax}$  denotes the argument that achieves the maximum.

To explain the **degenerate decoding** problem, it is convenient to express Pauli errors in a basis tailored to the stabilizer code. As explained in Sec. II-A, the set  $\{i, \bar{Z}_a, \bar{X}_a, S_b, T_b\}$  forms a basis for  $\mathcal{P}_n$ . Thus, an error  $E$  has a unique decomposition into a product of logical operators, elements of the stabilizer group, pure errors, and an irrelevant phase factor that we will henceforth ignore. In other words, we can uniquely decompose any error as  $E = \bar{L}ST$  where  $\bar{L} \in C(\mathcal{S})$ ,  $S \in \mathcal{S}$ , and  $T \in \langle T_a \rangle$ . We can therefore interpret the noise model  $P(E)$  as a probability distribution over  $L$ ,  $S$ , and  $T$ , simply setting

$$P(\bar{L}, S, T) = P(E = \bar{L}ST). \quad (15)$$

Observe that the syndrome  $\mathbf{s}$  is in **one-to-one correspondence with the pure error component  $T$** , namely, an error with syndrome  $\mathbf{s}$  has the pure error component

$$T(\mathbf{s}) = \prod_a T_a^{(1-s_a)/2} \quad (16)$$

where  $T_a^0 = I$  and  $T_a^1 = T_a$ . This can be seen from the commutation relation Eq. (12), which shows that  $S_a$  anti-commutes with  $E$  if and only if  $E$  contains  $T_a$  when decomposed in the basis  $\{\bar{Z}_a, \bar{X}_a, S_b, T_b\}$ . In addition, Eqs. 10-11 show that the  $S$  and  $L$  components of an error  $E$  have no effect on its syndrome. Thus, knowledge of the error syndrome is equivalent to knowledge of  $T$ , so the error probability conditioned on the error syndrome is obtained by Bayes' rule

$$P(E|\mathbf{s}) = P(\bar{L}, S|T(\mathbf{s})) = \frac{P(\bar{L}, S, T(\mathbf{s}))}{P(T(\mathbf{s}))} \quad (17)$$

where the marginal probability is defined as usual  $P(T(\mathbf{s})) = \sum_{\bar{L}, S} P(\bar{L}, S, T(\mathbf{s}))$ .

**Degenerate errors are those that differ only by their  $S$  component. Indeed,** the errors  $E$  and  $E' = ES$  for  $S \in \mathcal{S}$  have exactly the same effect on all code states by definition. Thus, only the logical component  $\bar{L}$  needs to be identified to correct the error; the  $T$  component is known given the syndrome and the  $S$  component has no effect on the encoded information. The degenerate MAP decoding therefore consists of the optimization problem

$$L_{\text{MAP}}^{\text{D}}(\mathbf{s}) = \underset{\bar{L}}{\operatorname{argmax}} P(\bar{L}|T(\mathbf{s})), \quad (18)$$

where  $P(\bar{L}|T(\mathbf{s})) = \sum_S P(\bar{L}, S|T(\mathbf{s}))$  is the marginal conditional distribution obtained from Eq. (17).

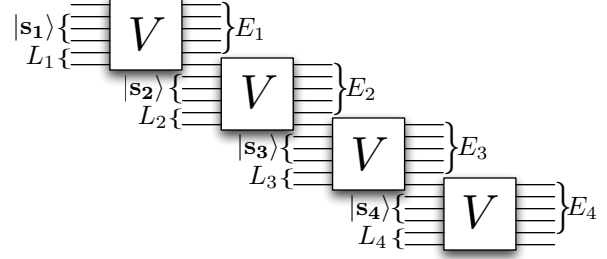


Fig. 1. Decomposition of an error  $E$  into a tensor product of errors  $E_t$  in each  $n$ -qubit frame. The error produces a syndrome  $\mathbf{s}$  and an un-encoded logical component  $L$ , that are also broken-up into components on each frame  $\mathbf{s}_t$  and  $L_t$ .

An equivalent way to explain the degenerate decoding problem is to imagine un-encoding the corrupted encoded state  $E|\bar{\psi}\rangle$ . This produces the state  $U^\dagger E|\bar{\psi}\rangle = U^\dagger E U(|\psi\rangle \otimes |0\rangle^{\otimes s}) = L|\psi\rangle \otimes Q|0\rangle^{\otimes s}$  where  $L \otimes I^{\otimes s} = U^\dagger \bar{L} U$  is the un-encoded version of the logical component  $\bar{L}$  of  $E$  and  $I^{\otimes k} \otimes Q = U^\dagger T S U$  is the un-encoded version of the stabilizer  $S$  and pure error  $T$  components of  $E$ . Note that the un-encoded version of  $S$  contains only  $Z$  operators on the ancillary qubits by definition, c.f. Eq. (7). Since  $Z|0\rangle = |0\rangle$ , we conclude that the  $S$  component has no effect on the un-encoded corrupted state, a direct manifestation of degeneracy. The  $T$  component on the other hand map to  $X$  operators under  $U^\dagger$ , c.f. Eq. (9). We conclude that the final state is  $U^\dagger E|\bar{\psi}\rangle = L|\psi\rangle \otimes |\mathbf{s}\rangle$  where  $|\mathbf{s}\rangle$  is a shorthand for a  $s$ -qubit state with qubit  $a$  in a state  $|\frac{1-s_a}{2}\rangle$ . The error syndrome can be directly obtained by measuring the ancillary qubits, and the degenerate decoding problem consists in identifying the most likely  $L$  given  $\mathbf{s}$ .

#### IV. DEGENERATE VITERBI ALGORITHM

With convolutional code, it is natural to use a notation that reflects the periodic structure of the encoding circuit. We use a two-index notation to identify qubits: qubit  $(t, a)$  corresponds to the  $a$ th qubit of the  $t$ th time frame, in other words the  $[(t-1)n + a]$ th qubit altogether since there are  $n$  qubit per time frame. Similarly, the error syndrome  $\mathbf{s}$  naturally breaks into  $\tau$  distinct  $s$ -bit syndromes  $\mathbf{s} = (\mathbf{s}_1, \dots, \mathbf{s}_\tau)$  with  $\mathbf{s}_t = (s_{t,1}, \dots, s_{t,s})$ . Following the last paragraph of the previous section, syndrome bit  $s_{t,a}$  is obtained by measuring the ancillary qubit  $(t, a)$  after having un-encoded the corrupted state, see Figure 1. Similarly, it is natural to also break  $L$  and  $E$  into a product of operators on each frame  $L_t$  and  $E_t$ , see Figure 1. Since we assume that the quantum channel is memoryless, the error probability factors as  $P(E) = \prod_t P(E_t)$ .

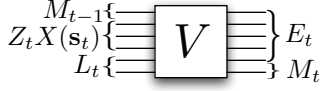
**The standard, non-degenerate, decoding of quantum convolutional code uses a trellis.** Trellis-based decoding of quantum convolutional codes was introduced in [14], but here we follow the presentation of [15], and refer the reader to this article for more details of the construction. The trellis for an  $(n, k, m)$ -quantum convolutional code of length  $\tau$  is a directed multigraph whose vertices can be grouped into  $\tau$  sets  $\Lambda_t$ . Each element of  $\Lambda_t$  is labeled by a distinct element  $M_t \in \mathcal{P}_m$ ,



so  $|\Lambda_t| = 4^m$ . Given an error syndrome  $\mathbf{s}$ , there is an edge between  $M_{t-1} \in \Lambda_{t-1}$  and  $M_t \in \Lambda_t$  if and only if there exists an  $L_t \in \mathcal{P}_k$ ,  $Z_t \in \langle I, Z \rangle^s$ , and  $E_t \in \mathcal{P}_{n-m}$  such that

$$V(M_{t-1} \otimes Z_t X(\mathbf{s}_t) \otimes L_t) V^\dagger = E_t \otimes M_t, \quad (19)$$

where  $X(\mathbf{s}_t) = \prod_{a=1}^s X_a^{(1-s_{t,a})/2}$ . There can in general be different choices of  $E_t$  that fulfill this condition, and we label the multiple edges by the associate  $E_t$ . This condition is illustrated by the circuit



which can be understood, reading from right to left, as one segment of the un-encoding circuit which proceeds recursively starting at  $t = \tau$  and making its way to  $t = 1$ . The interpretation is a situation where errors in the previous un-encoding steps have resulted in a memory state  $M_t$  which, combined with the error segment  $E_t$ , produces the right syndrome bits  $\mathbf{s}_t$  for the time frame  $t$ , modifies the logical state by the application of  $L_t$  in this frame, and sets the memory in a new state  $M_{t-1}$ .

With this interpretation we see that paths in the trellis correspond to all the distinct errors  $E$  with syndrome  $\mathbf{s}$ . To find the most likely error (or path) among them, we associate a probability (or weight) to each edge. The probability associated to the edge linking  $M_{t-1}$  to  $M_t$  labeled  $E_t$  is  $P(E_t)$ , the probability of the corresponding error, specified by the error model. The non-degenerate MAP decoding then consists in finding the path in the trellis with the largest probability, where the probability of a path is equal to the product of the probability of each of its segments. It is convenient to speak instead of the length of a path, which is equal to the sum of the weight of its segments, the weight of segment  $w(M_{t-1}, M_t; E_t)$  labeled  $E_t$  being  $-\ln P(E_t)$ , and we set  $w(M_{t-1}, M_t; E_t) = \infty$  if there is no edge labeled  $E_t$  between  $M_{t-1}$  and  $M_t$ . Then, the decoding problem becomes that of finding the shortest path in the trellis.

Viterbi's algorithm solves this problem recursively, starting at  $t = \tau$  and decreasing the value of  $t$  by 1 at each iteration, by associating a cumulative distance to the vertices of the graph. The cumulative distance of vertex  $M_t$ ,  $d(M_t)$ , is equal to the length of the shortest path starting at  $t = \tau$  and leading to  $M_t$ . It obeys the recursive equation

$$d(M_{t-1}) = \min_{M_t, E_t} [d(M_t) + w(M_{t-1}, M_t; E_t)], \quad (20)$$

so the algorithm keeps only the edges that realize the maximum for each  $M_{t-1}$ , the other edges are erased. The initialization of the algorithm is given by  $d(M_\tau) = -\ln P(M_\tau)$  where  $P(M_\tau)$  is the error model on the last  $m$  bits of the circuit. To explain the termination of the algorithm, we must specify that the first  $m$  qubits of the quantum circuit (the memory qubits of frame 0) are used as additional ancillary qubits. Thus, they are initialized in state  $|0\rangle^{\otimes m}$  prior to encoding and measured

in the  $Z$  basis after the un-encoding, see [15] for more details. The output of this measurement reveals the  $X$  component of  $M_0$ , and only the values of  $M_0$  with the correct component are kept. The shortest path is the one connecting to the  $M_0$  with the shortest cumulative distance  $d(M_0)$ .

The reason why the previous algorithm does not take degeneracy into account is that each path is associated to a distinct physical error. For degenerate decoding, errors need to be associated to equivalent classes of states instead, where errors that differ by an element of  $\mathcal{S}$  are joined in the same class. One way to keep track of such classes is to label them by their logical un-encoded component  $L = L_1 \otimes \dots \otimes L_\tau$ . The degenerate Viterbi algorithm will thus use the same trellis structure as above, but instead of labeling an edge that fulfills Eq. (19) by the corresponding  $E_t$ , it will be labeled by the corresponding un-encoded logical operator segments  $L_t \in \mathcal{P}_k$ . As before, the probability associated to an edge will be equal to  $P(E_t)$  for the  $E_t$  fulfilling the condition Eq. (19). Thus, at this point, the construction of the trellis follows exactly the original prescription of [14], but its edges are labeled differently.

For a given syndrome segment  $\mathbf{s}_t$ , un-encoded logical segment  $L_t$ , and memory states  $M_{t-1}$  and  $M_t$ , there can be multiple distinct  $E_t$  that fulfill the condition Eq. (19). Call this set of solutions  $\Omega(M_{t-1}, M_t, L_t, \mathbf{s}_t)$ . Thus, the trellis contains multiple edges between some memory states  $M_{t-1}$  and  $M_t$  with the same label  $L_t$ . These multiple edges  $\Omega(M_{t-1}, M_t, L_t, \mathbf{s}_t)$  correspond to degenerate paths, i.e. paths that differ only by an element of the stabilizer group. Therefore, they should be viewed as equivalent paths so we merge them into a single super edge as illustrated in Fig. 2. The probability of a super edge is equal to the sum of the probabilities of the merged edges, or in terms of weight:

$$w(M_{t-1}, M_t; L_t) = -\ln \left( \sum_{E_t \in \Omega(M_{t-1}, M_t, L_t, \mathbf{s}_t)} P(E_t) \right). \quad (21)$$

In terms of these weights, the degenerate decoding problem becomes the problem of finding the shortest path in the trellis, for which Viterbi's algorithm can be used.

In summary, our modification to Viterbi's decoding algorithm consists in a pre-processing phase that modifies the trellis by merging edges that correspond to degenerate errors and computing the associated weights. Given this modified trellis, the degenerate decoding problem becomes identical to the non-degenerate decoding problem, and can be solved with Viterbi's original algorithm.

## V. RESULTS

We have benchmarked our degenerate Viterbi decoder using Monte Carlo simulations on quantum convolution codes of finite block length  $\tau = 600$  with various code parameters  $(n, k, m)$ , results are presented in Fig. 3. The codes were generated by choosing the seed transformation  $V$  at random in the Clifford group. For each set of parameters  $(n, k, m)$ , we have simulated a few dozens of such randomly generated codes

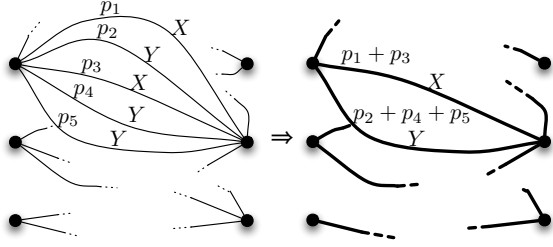


Fig. 2. Merging procedure. Multiple edges with the same label linking  $M_{t-1}$  to  $M_t$  are merged into a unique super edge with the corresponding label. The probability of a super edge is equal to the sum of the probabilities of the merged edges.

and kept the most interesting results, although our conclusions extend qualitatively to all codes we have simulated. The codes are terminated by padding, as explained in [15].

The Monte Carlo simulations were realized as follows. An  $N$  qubit error  $E$  is generated randomly according to the depolarizing error probability of rate  $p$  (c.f. Sec. II-B) and the corresponding error syndrome  $s$  is calculated. The syndrome is fed as input to two distinct algorithms: a degenerate and a non-degenerate decoding algorithm. The degenerate decoder outputs the most likely logical error component  $L_{\text{MAP}}^D(s)$  given by Eq. (18). It is declared successful if the output  $L_{\text{MAP}}^D(s)$  belongs to the equivalence class of the randomly generated error  $E$ , and failed otherwise. The procedure is repeated  $N_{\text{sample}}$  times to accumulate statistics, and the block error rate equals the frequency of its failures. The number of samples  $N_{\text{sample}}$  is adjusted such that at least 30 significant events are observed, resulting in a relative error at most  $1/\sqrt{30}$  on the reported data. Figure 3 reports the block error rate of the degenerate decoder as a function of the channel error rate for different convolutional codes.

A similar procedure is performed in parallel with the non-degenerate Viterbi decoder during our Monte Carlo simulations. Given the input syndrome  $s$ , the non-degenerate decoder outputs the most likely error  $E_{\text{MAP}}^{\text{ND}}(s)$  given by Eq. (14). It is declared successful if the output  $E_{\text{MAP}}^{\text{ND}}(s)$  belongs to the equivalence class of the randomly generated error  $E$ , and failed otherwise. Note that the non-degenerate decoder therefore benefit from the existence of degeneracy because its output  $E_{\text{MAP}}^{\text{ND}}(s)$  is not required to exactly match the randomly generated error  $E$  to be accepted as a successful decoding, as long as they belong to the same equivalence class. The distinction is that degeneracy is not explicitly taken into account to estimate the error.

To evaluate the impact of degeneracy on the decoder's performance, we compare its block error rate with the one obtained with the non-degenerate Viterbi decoder. In the inset of Fig. 3, we show the ratio between the non-degenerate decoder's block error rate to the one of the degenerate decoder. The fact that all data points are above 1 indicates that the non-degenerate decoder always yields a lower block error rate than the non-degenerate decoder. We also observe in the inset of Fig. 3 that the benefit becomes more prominent at low

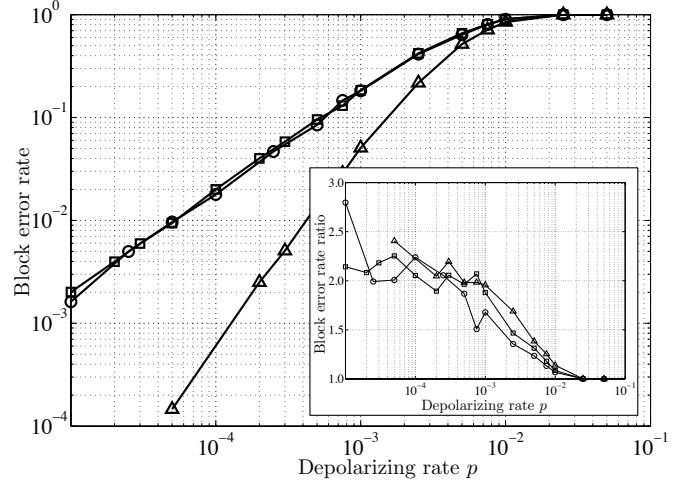


Fig. 3. Block error rate as a function of the channel depolarizing rate  $p$  for different  $(n, k, m)$ -quantum convolutional codes: circles  $(4, 1, 1)$ ; squares  $(4, 1, 2)$ ; triangles  $(5, 1, 3)$ . Since all codes have  $k = 1$  and the length of the code is  $\tau = 600$ , all codes encode 600 logical qubits. The triangles represent a code of lower rate, so as expected it shows overall better performances. Inset: the ratio of block error rates obtained with the non-degenerate Viterbi decoder and the degenerate Viterbi decoder. The degenerate Viterbi decoder always yields a lower block error rate, and this advantage increases as the channel error rate decreases.

depolarizing rates. This effect appears to be independent of the code parameters: the block error rate ratios of all the codes we have simulated show a clear monotonic increase as  $p$  decreases. We have observed gains as large as 4.4 dB.

## VI. CONCLUSION

Degeneracy is a key feature of quantum codes that is at the origin of important distinctions between classical and quantum information theory. To gain from the existence of degeneracy, it must be taken into account during the decoding process of a code. Here, we have developed a decoding algorithm for quantum convolutional codes that exactly solves the maximum a posteriori decoding problem over equivalent classes of degenerate errors. The main modification to the standard Viterbi algorithm is a pre-processing phase that modifies the trellis. Our Monte Carlo simulations show that degenerate decoding improves the error suppression of the code, and that this effect becomes more prominent at low error rates.

Because convolutional codes are the main ingredient of quantum turbo codes, a natural next step would be to use our degenerate decoder in a concatenated scheme, where its benefit could be further amplified. The development of a fault-tolerant computing scheme based on these codes is also desirable. The importance of our decoder may be particularly important in this setting since fault-tolerant quantum computers operate at low error rates, where the effect of degenerate decoding are most prominent.

*Acknowledgements*—This research was partially funded by Mprime, NSERC, FQRNT and the IARPA QCS program. Numerical resources were provided by Calcul Québec and Compute Canada.

*Disclaimer*— The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI/NBC, or the U.S. Government.

## REFERENCES

- [1] D. Gottesman, “Stabilizer codes and quantum error correction,” Ph.D. dissertation, California Institute of Technology, Pasadena, CA, 1997.
- [2] A. R. Calderbank, E. M. Rains, P. W. Shor, and N. J. A. Sloane, “Quantum error correction via codes over  $GF(4)$ ,” *IEEE Trans. Info. Theor.*, vol. 44, p. 1369, 1998.
- [3] P. W. Shor and J. A. Smolin, “Quantum error-correcting codes need not completely reveal the error syndrome,” AT&T Research, Murray Hill, NJ 07974, 1996, preprint.
- [4] D. P. DiVincenzo, P. W. Shor, and J. A. Smolin, “Quantum-channel capacity of very noisy channels,” *Phys. Rev. A*, vol. 57, pp. 830–839, 1998.
- [5] G. Smith and J. A. Smolin, “Degenerate coding for Pauli channels,” *Phys. Rev. Lett.*, vol. 98, p. 030501, 2007.
- [6] S. Lloyd, “Capacity of the noisy quantum channel,” *Phys. Rev. A*, vol. 55, p. 1613, 1997.
- [7] P. Shor, “The quantum channel capacity and coherent information,” 2002, mSRI Workshop on quantum computation.
- [8] I. Devetak, “The private classical capacity and quantum capacity of a quantum channel,” *IEEE Trans. Info. Theor.*, vol. 51, p. 44, 2005.
- [9] E. Dennis, A. Kitaev, A. Landahl, and J. Preskill, “Topological quantum memory,” *J. Math. Phys.*, vol. 43, p. 4452, 2002.
- [10] A. Y. Kitaev, “Fault-tolerant quantum computation by anyons,” *Ann. Phys.*, vol. 303, p. 2, 2003.
- [11] T. Stace and S. Barrett, “Error correction and degeneracy in surface codes suffering loss,” 2009.
- [12] A. J. Viterbi, “Error bounds for convolutional codes and an asymptotically optimum decoding algorithm,” *IEEE Trans. Info. Theor.*, vol. 13, no. 2, pp. 260–269, 1967.
- [13] S. Aji and R. McEliece, “The generalized distributive law,” *IEEE Trans. Info. Theor.*, vol. 46, no. 2, p. 325, 2000.
- [14] H. Ollivier and J.-P. Tillich, “Trellises for stabilizer codes: definition and uses,” *Phys. Rev. A*, vol. 74, p. 032304, 2006.
- [15] D. Poulin, J.-P. Tillich, and H. Ollivier, “Quantum serial turbo-codes,” *IEEE Trans. Info. Theor.*, vol. 55, no. 6, p. 2776, 2009.
- [16] H. Ollivier and J.-P. Tillich, “Description of a quantum convolutional code,” *Phys. Rev. Lett.*, vol. 91, no. 17, p. 177902, 2003.
- [17] —, “Quantum convolutional codes: fundamentals,” 2004.
- [18] J. G. D. Forney, M. Grassl, and S. Guha, “Convolutional and tail-biting quantum error-correcting codes,” *IEEE Trans. Info. Theor.*, vol. 53, p. 865, 2007.