Convolutional and Block Quantum Error-Correcting Codes

Markus Grassl
Institut für Algorithmen und Kognitive Systeme
Fakultät für Informatik, Universität Karlsruhe (TH)
Am Fasanengarten 5, 76128 Karlsruhe, Germany
Email: grassl@ira.uka.de

Abstract — **In this paper, we briefly recall the connection of both convolutional and block quantum error-correcting codes to their classical counterparts. Based on this correspondence, algorithms for computing encoding circuits for both types of codes are derived.**

I. Stabilizer Codes

The development of quantum error-correcting codes has been an important step towards the realization of a quantum computer. While the theory of quantum block codes [1, 4] is fairly advanced by now, the investigation of the quantum version of convolutional codes has only started more recently [2, 3, 7, 10]. Most of the quantum codes are based on the theory of so-called stabilizer codes. We briefly recall the required definitions for both quantum block codes and quantum convolutional codes. This will then allow to present the algorithms for encoding block codes (see also [6]) and convolutional codes (see also [7]). For more details on quantum computing in general see e. g. [9].

Definition 1 (Finite Pauli Group) *Let*

$$X=\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix},\quad Z=\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix},\quad Y=XZ=\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$$

be the (real version of the) 2×2 *Pauli matrices. The finite Pauli group* $\mathcal{G}_n$ *is the group generated by* n*-fold tensor products of the Pauli matrices* X*,* Y*,* Z *and the identity matrix* I_2*.*

Every element of the finite Pauli group $\mathcal{G}_n$ can be expressed in the form

$$(-1)^c X^{a_1}Z^{b_1}\otimes\ldots\otimes X^{a_n}Z^{b_n} := (-1)^c X^{\boldsymbol{a}}Z^{\boldsymbol{b}}$$

where the exponents have been combined to vectors $\boldsymbol{a},\boldsymbol{b}\in\mathbb{F}_2^n$. They can also be expressed by a single vector $\boldsymbol{a}+\omega\boldsymbol{b}\in\mathbb{F}_4^n$, where ω denotes a primitive element of $\mathbb{F}_4$. Two elements of $\mathcal{G}_n$ commute if and only if the symplectic inner product

$$(\boldsymbol{a},\boldsymbol{b})\cdot(\boldsymbol{a}',\boldsymbol{b}') := \boldsymbol{a}\cdot\boldsymbol{b}'-\boldsymbol{b}\cdot\boldsymbol{a}' \qquad (1)$$

of the corresponding exponent vectors vanishes. Quantum stabilizer block codes are defined as the common eigenspace of an abelian subgroup of $\mathcal{G}_n$, the stabilizer group. In terms of the representation as vectors over $\mathbb{F}_2$ or $\mathbb{F}_4$, they correspond to additive codes which are self-orthogonal with respect to the inner product (1). The generators of the stabilizer group are given by a generator matrix of this code, which is referred to as *stabilizer matrix*. It is usually written in the form

$$(X|Z)\in\mathbb{F}_2^{r\times 2n}.$$

The condition that the code is self-orthogonal can be compactly expressed as

$$XZ^t-ZX^t=0.$$

Quantum convolutional codes can be seen as infinite versions of quantum stabilizer block codes. First we extend the definition of the Pauli group:

Definition 2 (Infinite Pauli Group) *Consider an infinite set of qubits labeled by the nonnegative integers* $\mathbb{N}$*. Let* $M\in\{X,Y,Z\}$ *be a Pauli matrix. We denote by* M_i *the semi-infinite tensor product* $I_2\otimes\ldots\otimes I_2\otimes M\otimes I_2\otimes\ldots$*, where* M *operates on qubit* i *and* I_2 *denotes the identity matrix of size* 2×2*. The group generated by all* X_i *and* Z_i *for* $i\in\mathbb{N}$ *is called the infinite Pauli group* $\mathcal{G}_\infty$*. For an element* $A=A_1\otimes A_2\otimes\ldots\in\mathcal{G}_\infty$ *the positions in which* A_i *is not equal to* $\pm I_2$ *is called the support of* A*.*

Similar as the elements of the finite Pauli group $\mathcal{G}_n$ can be represented by two binary vectors, we can label the elements of the infinite Pauli group $\mathcal{G}_\infty$ by a tuple of binary sequences, each of which is represented by a formal power series. Hence we get the correspondence

$$\begin{aligned}(-1)^c X^{\boldsymbol{\alpha}}Z^{\boldsymbol{\beta}} &:= (-1)^c\bigotimes_{\ell\geq 0}X^{\alpha_\ell}Z^{\beta_\ell}\\ &\hat{=}\left(\sum_{\ell\geq 0}\alpha_\ell D^\ell,\sum_{\ell\geq 0}\beta_\ell D^\ell\right)\end{aligned}$$

where $c\in\mathbb{F}_2$ and $\boldsymbol{\alpha}=\sum_{\ell\geq 0}\alpha_\ell D^\ell$ and $\boldsymbol{\beta}=\sum_{\ell\geq 0}\beta_\ell D^\ell$ are formal power series with coefficients in $\mathbb{F}_2$. In this representation, multiplication of elements of $\mathcal{G}_\infty$ corresponds to addition of power series. Furthermore, shifting an element $A\in\mathcal{G}_\infty$ one qubit to the right corresponds to the multiplication of the power series by D. As we also allow to shift the operators by a bounded number of qubits to the left, we use Laurent series instead of power series to

1-4244-0067-8/06/$20.00 ©2006 IEEE.

represent the elements of $\mathcal{G}_\infty$. An element $A \in \mathcal{G}_\infty$ with finite support corresponds to a tuple of Laurent polynomials. Recall that the field of Laurent series in the variable D with coefficients in $\mathbb{F}_2$ is denoted by $\mathbb{F}_2((D))$ and recall further that it contains the ring $\mathbb{F}_2[D, D^{-1}]$ of Laurent polynomials.

We are interested in subgroups of $\mathcal{G}_\infty$ which can be generated by a finite number of elements with finite support and their (right) shifted versions. The following definition introduces a shorthand notation for describing such subgroups.

Definition 3 (Stabilizer Matrix) *Let $\mathcal{S}$ be an abelian subgroup of $\mathcal{G}_\infty$ which has trivial intersection with the center of $\mathcal{G}_\infty$. Furthermore, let $\{g_1, g_2, \ldots, g_r\}$ where $g_i = (-1)^{c_i} X^{\boldsymbol{\alpha}_i} Z^{\boldsymbol{\beta}_i}$ with $c_i \in \{0, 1\}$ and $(\boldsymbol{\alpha}_i, \boldsymbol{\beta}_i) \in \mathbb{F}_2((D))^n \times \mathbb{F}_2((D))^n$ be a minimal set of generators for $\mathcal{S}$ when including their shifted versions, too. Then a stabilizer matrix of the corresponding quantum convolutional (stabilizer) code $\mathcal{C}$ is a generator matrix of the (classical) additive convolutional code $C \subseteq \mathbb{F}_2((D))^n \times \mathbb{F}_2((D))^n$ generated by $(\boldsymbol{\alpha}_i, \boldsymbol{\beta}_i)$. We will write this matrix in the form*

$$S(D) = (X(D)|Z(D)) =\in \mathbb{F}_2((D))^{r \times 2n}. \tag{2}$$

Additionally, we require that every generator has finite support, so the entries of the stabilizer matrix $S(D)$ are polynomials.

Writing the stabilizer in the form as in (2), it was shown in [10] that the condition of symplectic orthogonality can be expressed compactly in the form

$$X(D)Z(1/D)^t - Z(D)X(1/D)^t = 0. \tag{3}$$

On the other hand, we can start with an arbitrary self-orthogonal additive convolutional code over $\mathbb{F}_2((D))^n \times \mathbb{F}_2((D))^n$ to define a convolutional quantum code. In general, the generator matrix for such a code may contain rational functions, but there is always an equivalent description in terms of a matrix with polynomial entries [8]. Therefore we assume that the stabilizer matrix contains only polynomials or Laurent polynomials.

II. Encoding Block Codes

The simplest way of encoding k qubits into n qubits is to add $n-k$ qubits in the state $|0\rangle$ to the k-qubit input state $|\phi\rangle$. While this code is useless for the correction of errors, we take it as the starting point for our algorithm. The stabilizer of such a code consists of error operators that equal identity on the k positions corresponding to $|\phi\rangle$. At the remaining $n-k$ positions, we have either σ_z or identity. If the qubits in the state $|0\rangle$ are at the first $n-k$ positions, the stabilizer matrix has the form

$$\left(\begin{array}{ccc|ccc} 0 & \cdots\cdots & 0 & 1 & & 0 \;\cdots\; 0 \\ \vdots & & \vdots & & \ddots & \vdots \quad\quad \vdots \\ 0 & \cdots\cdots & 0 & & & 1\;0 \;\cdots\; 0 \end{array}\right) = (0\,0 \mid I\,0). \tag{4}$$

Tab. 1: Single-qubit operations permuting the Pauli matrices.

Hadamard matrix H		
$HXH = Z$	$HYH = -Y$	$HZH = X$
$(1,0) \mapsto (0,1)$	$(1,1) \mapsto (1,1)$	$(0,1) \mapsto (1,0)$
matrix P		
$PXP^\dagger = Y$	$PYP^\dagger = -X$	$PZP^\dagger = Z$
$(1,0) \mapsto (1,1)$	$(1,1) \mapsto (1,0)$	$(0,1) \mapsto (0,1)$

It can be shown that by a change of basis, any stabilizer can be transformed into one of type (4). The operations needed for this are exactly those of the so-called *Clifford group* which plays an important role in the context of fault-tolerant quantum computation, too (see e. g. [5]).

The first of those operations is the Hadamard transform H. Conjugating Pauli matrices with the Hadamard transform interchanges X and Z, the matrix Y gets a minus sign which is irrelevant. The second operation is given by

$$P := \begin{pmatrix} 1 & 0 \\ 0 & e^{\pi i/2} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \in \mathbb{C}^{2\times 2}.$$

As P is a diagonal matrix, it commutes with Z. The conjugation of X with P yields the matrix Y. The action of H and P on the Pauli matrices is summarized in Table 1. Additionally, the action on the binary tuples (a, b) corresponding to a Pauli matrix $X^a Z^b$ is given. It can be seen that this action can be described by binary 2×2 matrices

$$\overline{H} \mathrel{\hat{=}} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad \text{and} \quad \overline{P} \mathrel{\hat{=}} \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}. \tag{5}$$

When the matrices H and P act on the jth qubit of a stabilizer, we have to apply the matrices $\overline{H}$ and $\overline{P}$ to the jth column in the X-part and the jth column of the Z-part of the stabilizer matrix.

Finally, we need the CNOT gate that acts on pairs of qubits. The action of CNOT on the Pauli matrices is shown in Fig. 1. On the stabilizer matrix, a CNOT gate with source qubit i and target j acts as follows: in the X-part, we add column i (the source) to column j (the target). In the Z part, we add column j (the target) to column i (the source). Note that in the arithmetic mod 2, addition and subtraction are the same operation.

III. Example: Five Qubit Code

We illustrate the transformation of a stabilizer to the form (4) for the five qubit code $[\![5, 1, 3]\!]$ with stabilizer matrix

$$\begin{bmatrix} XXZIZ \\ ZXXZI \\ IZXXZ \\ ZIZXX \end{bmatrix} \mathrel{\hat{=}} \left(\begin{array}{ccccc|ccccc} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \end{array}\right). \tag{6}$$

For each row of the stabilizer matrix, we first transform the corresponding stabilizer into an X-only generator.

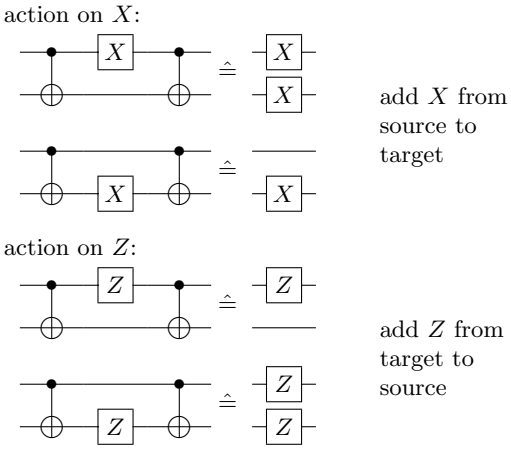

Fig. 1: Modifying stabilizers using the CNOT gate.

Then, using CNOT gates we obtain an X-generator of weight one. Finally, using row operations we further simplify the generators.

Using the matrices H and P, we can change the Pauli matrices Z and Y into X. In the first row of the stabilizer matrix (6) there are Z operators at position three and five, so we apply the operation $T_1 = I \otimes I \otimes H \otimes I \otimes H$ and obtain the stabilizer matrix

$$\left(\begin{array}{ccccc|ccccc} 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \end{array}\right).$$

Using the gate $\mathrm{CNOT}^{(i,j)}$ with source qubit i and target j, we can transform a pair of X-operators into a single X-operator. Hence by a sequence of CNOT gates, we can transform the first operator into an operator where the only tensor factor different from identity equals X. For the gate $\mathrm{CNOT}^{(1,2)}$, the transformation on the stabilizer matrix is given by

$$\left(\begin{array}{ccccc|ccccc} 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{array}\right) \longmapsto \left(\begin{array}{ccccc|ccccc} 1 & \mathbf{0} & 1 & 0 & 1 & \mathbf{0} & 0 & 0 & 0 & 0 \\ 0 & \mathbf{1} & 0 & 0 & 0 & \mathbf{1} & 0 & 1 & 1 & 0 \\ 0 & \mathbf{0} & 0 & 1 & 1 & \mathbf{1} & 1 & 1 & 0 & 0 \\ 0 & \mathbf{0} & 1 & 1 & 0 & \mathbf{1} & 0 & 0 & 0 & 1 \end{array}\right).$$

In the X-part, we add the first column to the second, in the Z part, we add the second column to the first. Proceeding analogously, we obtain the stabilizer matrix

$$\left(\begin{array}{c|cccc|c|cccc} 1 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right). \quad (7)$$

After this step, in the first row we have zero at all but the first position. Note that automatically all other entries in the first column of the Z-part are zero, too, because all generators commute with the operator $X^{(1)}$ acting on the first position. The other non-zero entries in the first column of the X-part can be cancelled by subtracting the first row. This corresponds to multiplying one generator of the stabilizer group by another which does not change the group.

We continue in the same way with the submatrices of the stabilizer matrix as indicated in (7) and obtain a stabilizer matrix of the form $(I0|00)$. Finally, we apply Hadamard gates to the first $n-k=4$ positions and obtain a stabilizer matrix as in (4).

Combining all transformations, we get a quantum circuit mapping the encoded state to an unencoded state plus the ancilla states (see Fig. 2). The inverse of the circuit gives an algorithm for encoding a quantum state.

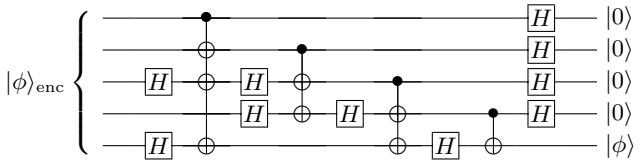

Fig. 2: Quantum circuit mapping a state $|\phi\rangle_{\mathrm{enc}}$ of the code $[\![5,1,3]\!]$ to an unencoded one-qubit state $|\phi\rangle$.

IV. Encoding Quantum Convolutional Codes

Similar as for quantum block codes, the stabilizer matrix of a quantum convolutional code can be transformed into a simple form (4) using Clifford operations. Note that each row of the stabilizer matrix corresponds to an infinite number of generators which are obtained by repeatedly shifting the stabilizer by n qubits respectively multiplying the generator by powers of D. Hence when transforming the stabilizer matrix we have to apply the corresponding Clifford operations in all its shifted versions, too. So for convolutional codes, for example $\overline{H}$ applied to first position corresponds to the semi-infinite tensor product $(H \otimes I_2 \otimes \ldots \otimes I_2) \otimes (H \otimes I_2 \otimes \ldots \otimes I_2) \otimes \ldots$

The gate CNOT from the Clifford group requires special attention. If the source qubit is say the first qubit in one block and the target qubit is the first qubit in the next block, we get a semi-infinite cascade of CNOT gates that cannot be parallelized. The resulting quantum circuit would have infinite depth and would be hard to invert. What is even worse, such an infinite cascade might cause an infinite spread of a local error, similar to the situation for catastrophic classical convolutional codes. But it will turn out that we do not need such a CNOT gate. Instead, we will use the gate P_ℓ which is a controlled-sign gate CSIGN acting on the same qubit in different blocks. These gates are diagonal and hence can be re-ordered to have finite depth.

The action of all semi-infinite Clifford operations which we are using is summarized in Table. 2. If a controlled operation acts on qubits in different blocks of n qubits, there are entries D^ℓ and $D^{-\ell}$ in the corresponding matrix.

Tab. 2: Action of various Clifford operations. Conjugation of the stabilizer group $\mathcal{S}$ by the unitary gate U corresponds to the action of the matrices $\overline{U}$ on the columns of the stabilizer matrix $S(D) = (X(D)|Z(D))$.

unitary gate U	matrix $\overline{U}$
$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \in \mathbb{C}^{2\times 2}$	$\overline{H} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \in \mathbb{F}_2^{2\times 2}$
$P = \begin{pmatrix} 1 & 0 \\ 0 & \exp(i\pi/2) \end{pmatrix} \in \mathbb{C}^{2\times 2}$	$\overline{P} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \in \mathbb{F}_2^{2\times 2}$
$\mathrm{CNOT}^{(i,j+\ell n)}, i \not\equiv j \pmod n$	$\overline{\mathrm{CNOT}} = \left(\begin{array}{cc\|cc} 1 & D^\ell & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 \\ 0 & 0 & D^{-\ell} & 1 \end{array}\right)$
$P_\ell := \mathrm{CSIGN}^{(i,i+\ell n)}, \ell \neq 0$	$\overline{P_\ell} = \begin{pmatrix} 1 & D^{-\ell} + D^\ell \\ 0 & 1 \end{pmatrix}$

For quantum block codes, every generator can be transformed into an X-only generator using the gates H and P. This is no longer true for quantum convolutional codes as the entries in one row of the stabilizer matrix may have different degrees in D. If the stabilizer operates non-trivially on only one position of the block of n qubits (or if $n = 1$), we can use the following lemma:

Lemma 4 *Let $S(D) = (f(D)|g(D)) \in \mathbb{F}_2[D, D^{-1}]^{1\times 2}$ be a stabilizer matrix. Then using $\overline{H}$, $\overline{P}$, and $\overline{P}_\ell$ we can transform $S(D)$ into the form $(h(D)|0)$, where $h(D) \in \mathbb{F}_2[D, D^{-1}]$.*

Proof: As the stabilizer given by $(f(D)|g(D))$ commutes with all its shifted versions, the first and the last non-trivial element must be equal. Without loss of generality we can assume that

$$\begin{aligned} f(D) &= \qquad\qquad D^{\nu_0} + \ldots + \quad D^{\nu_1} \\ g(D) &= D^{\mu_0} + \ldots + g_{\nu_0} D^{\nu_0} + \ldots + g_{\nu_1} D^{\nu_1} + \ldots + D^{\mu_1} \end{aligned}$$

where $\mu_0 < \nu_0 \leq \nu_1 < \mu_1$. In this case, the first and the last non-trivial element of the stabilizer is Z. If both are X or Y, the gates $\overline{H}$ and $\overline{P}$, respectively, yield the desired form. Using $\overline{P}_\ell$ with $\ell = \min(\nu_0 - \mu_0, \mu_1 - \nu_1)$, we obtain $(f(D)|g'(D))$ with $g'(D) = (D^{-\ell} + D^\ell) f(D) + g(D)$. Either $g'(D)$ is zero or the degree of $g'(D)$ (as Laurent polynomial) is strictly smaller than the degree of $g(D)$. The result follows by induction from repeating this procedure. ∎

This allows us to formulate an algorithm to transform the stabilizer matrix of a convolutional quantum code into a simple Z-only form:

Algorithm 5 *Apply the following steps to each row $i = 1, 2, \ldots$ of the stabilizer matrix:*

1. *Use* CNOT *gates to achieve that all entries in the columns j, $j > i$ of the X-part are zero.*
2. *Use Lemma 4 to cancel the ith entry of the Z-part.*
3. *Use Hadamard gates to swap the non-trivial elements of the Z-part into the X-part.*
4. *Use again* CNOT *gates to achieve that all entries in the columns j, $j > i$ of the X-part are zero.*

Finally, use Hadamard gates to obtain a Z-only stabilizer matrix.

It should be noted that one could combine the final Hadamard gates and the Hadamard gates of Step 3 to conjugate the CNOT gates into CSIGN gates, slightly reducing the total number of gates.

Instead of giving a detailed proof of the correctness of the algorithm which is very similar to the algorithm presented in [7], we will illustrate the algorithm in the next section.

V. Example: Rate-1/3 Convolutional Code

Consider the $\mathbb{F}_4$-linear rate-1/3 convolutional code from [3, Table VI]) with generator matrix

$$G(D) = \begin{pmatrix} 1 + D & 1 + \omega D & 1 + \overline{\omega} D \end{pmatrix}.$$

The corresponding stabilizer matrix is

$$S(D) = \left(\begin{array}{ccc|ccc} 1+D & 1 & 1+D & 0 & D & D \\ 0 & D & D & 1+D & 1+D & 1 \end{array}\right).$$

Using $\mathrm{CNOT}^{(1,3)}$, we add the first column of the X-part to the last column, clearing the third entry. In the Z-part, we have to add the last column to the first. The new stabilizer matrix is

$$\left(\begin{array}{ccc|ccc} 1+D & 1 & 0 & D & D & D \\ 0 & D & D & D & 1+D & 1 \end{array}\right).$$

Then we add D times the second column of the X-part to the first column. In the Z-part we have to add $1/D$ times the first column to the second. We get

$$\left(\begin{array}{ccc|ccc} 1 & 1 & 0 & D & 1+D & D \\ D^2 & D & D & D & D & 1 \end{array}\right).$$

Using $\mathrm{CNOT}^{(1,2)}$ we achieve that the X-part of the first generator has weight one, completing the first step:

$$\left(\begin{array}{ccc|ccc} 1 & 0 & 0 & 1 & 1+D & D \\ D^2 & D+D^2 & D & 0 & D & 1 \end{array}\right).$$

Applying the gate P to the first qubit, we obtain

$$\left(\begin{array}{ccc|ccc} 1 & 0 & 0 & 0 & 1+D & D \\ D^2 & D+D^2 & D & D^2 & D & 1 \end{array}\right).$$

The first stabilizer has now an X-only element on the first position and Z-only elements on the others. Using Hadamard gates, this generator can be transformed into an X-only generator:

$$\left(\begin{array}{ccc|ccc} 1 & 1+D & D & 0 & 0 & 0 \\ D^2 & D & 1 & D^2 & D+D^2 & D \end{array}\right).$$

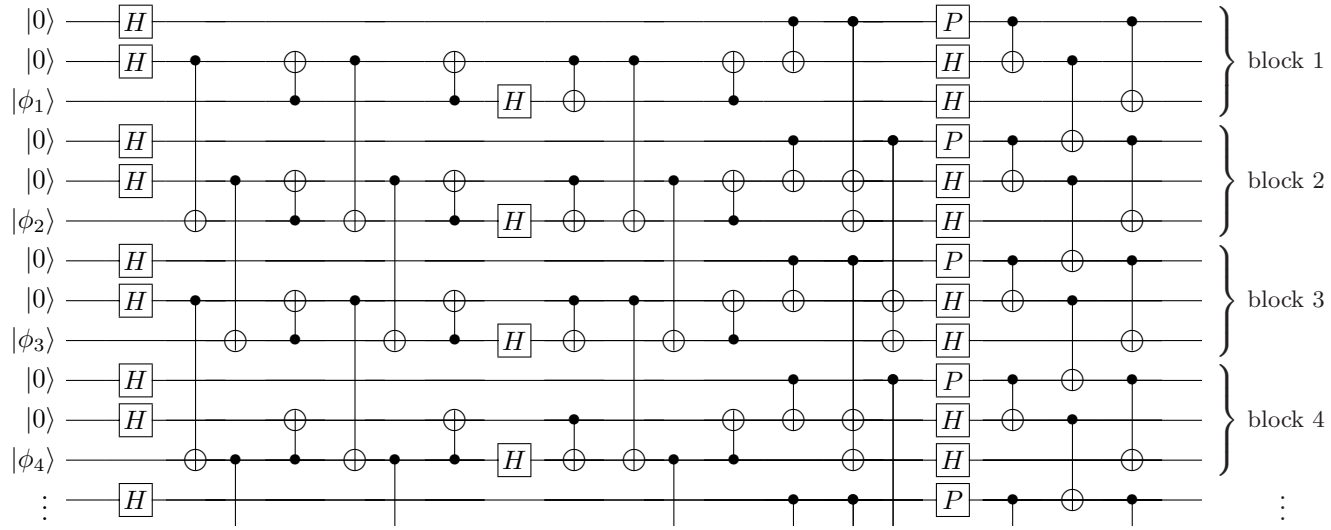

Fig. 3: Encoding circuit for a rate 1/3 convolutional quantum code. Every gate has to be repeatedly applied shifted by one block.

Next we transform the first generator into an X-only generator of weight one using CNOT gates:

$$\left(\begin{array}{ccc|ccc} 1 & 0 & 0 & 0 & 0 & 0 \\ D^2 & D+D^2+D^3 & 1+D^3 & 0 & D+D^2 & D \end{array}\right). \quad (8)$$

Similar as in the case of block codes, all entries in the first column of the Z part are zero because the corresponding generators commute with the first generator. We continue with the 1×2 submatrix as indicated in (8). Using CNOT gates we cancel all but one entry in the X-part of the submatrix:

$$\left(\begin{array}{ccc|ccc} 1 & 0 & 0 & 0 & 0 & 0 \\ D^2 & 1+D+D^2 & 1+D^3 & 0 & D+D^2 & D^2 \end{array}\right)$$
$$\left(\begin{array}{ccc|ccc} 1 & 0 & 0 & 0 & 0 & 0 \\ D^2 & 1+D+D^2 & 1+D+D^2 & 0 & D^2 & D^2 \end{array}\right)$$
$$\left(\begin{array}{ccc|ccc} 1 & 0 & 0 & 0 & 0 & 0 \\ D^2 & 1+D+D^2 & 0 & 0 & 0 & D^2 \end{array}\right)$$

A Hadamard gate applied to the third qubit yields

$$\left(\begin{array}{ccc|ccc} 1 & 0 & 0 & 0 & 0 & 0 \\ D^2 & 1+D+D^2 & D^2 & 0 & 0 & 0 \end{array}\right).$$

Using a sequence of CNOT gates, the stabilizer matrix can be transformed as follows:

$$\left(\begin{array}{ccc|ccc} 1 & 0 & 0 & 0 & 0 & 0 \\ D^2 & 1+D & D^2 & 0 & 0 & 0 \end{array}\right)$$
$$\left(\begin{array}{ccc|ccc} 1 & 0 & 0 & 0 & 0 & 0 \\ D^2 & 1+D & D & 0 & 0 & 0 \end{array}\right)$$
$$\left(\begin{array}{ccc|ccc} 1 & 0 & 0 & 0 & 0 & 0 \\ D^2 & 1 & D & 0 & 0 & 0 \end{array}\right)$$
$$\left(\begin{array}{ccc|ccc} 1 & 0 & 0 & 0 & 0 & 0 \\ D^2 & 1 & 0 & 0 & 0 & 0 \end{array}\right)$$

Now we have an X-only stabilizer which can be transformed into a Z-only stabilizer using Hadamard gates. The corresponding simple quantum convolutional code consists of blocks with three qubits in each block. The first two qubits in each block are zero, and the information is encoded in the third qubit. Taking the inverse of these operations we obtain the encoding circuit shown in Fig. 3.

Acknowledgments

This paper is based on joint work with Martin Rötteler. We also acknowledge fruitful discussions with David Forney, Saikat Guha, Harold Ollivier, David Poulin, and Jean-Pierre Tillich.

References

[1] A. R. Calderbank, E. M. Rains, P. W. Shor, and N. J. A. Sloane, "Quantum Error Correction Via Codes Over GF(4),", *IEEE Trans. Inform. Theory*, vol. 44, no. 4, pp. 1369–1387, 1998.

[2] H. F. Chau, "Quantum convolutional error-correcting codes," *Phys. Rev. A*, vol. 58, no. 2, pp. 905–909, 1998.

[3] G. D. Forney Jr., M. Grassl, and S. Guha, "Convolutional and tail-biting quantum error-correcting codes," 2005, preprint quant-ph/0511016, submitted to IEEE Trans. Inform. Theory.

[4] D. Gottesman, "A Class of Quantum Error-Correcting Codes Saturating the Quantum Hamming Bound," *Phys. Rev. A*, vol. 54, no. 3", pp. 1862–1868, 1996.

[5] D. Gottesman, "Fault-Tolerant Quantum Computation with Higher-Dimensional Systems,", in *Quantum Computing & Quantum Communications (QCQC'98)*, C. P. Williams (ed.), LNCS 1509, Springer, Heidelberg, 1998, pp. 302–313.

[6] M. Grassl, Th. Beth, and M. Rötteler, "Efficient quantum circuits for non-qubit quantum error-correcting codes," *Int. J. Found. Comput. Sci.*, vol. 14, no. 5, pp. 757–775, 2003.

[7] M. Grassl and M. Rötteler, "Non-catastrophic Encoders and Encoder Inverses for Quantum Convolutional Codes," in *Proc. 2006 IEEE Int. Symposium Information Theory (Seattle, USA)*, 2006, pp. 1109-1114.

[8] R. Johannesson and K. S. Zigangirov, *Fundamentals of Convolutional Coding*. New York: IEEE Press, 1999.

[9] M. Nielsen and I. Chuang, *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

[10] H. Ollivier and J.-P. Tillich, "Quantum convolutional codes: fundamentals," 2004, preprint quant-ph/0401134.