Simple Rate-1/3 Convolutional and Tail-Biting Quantum Error-Correcting Codes

G. David Forney, Jr.
Lab. for Inform. and Dec. Syst.
Mass. Inst. Tech.
Cambridge, MA 02139 USA
Email: forneyd@comcast.net

Saikat Guha
Research Lab. Electronics
Mass. Inst. Tech.
Cambridge, MA 02139 USA
Email: saikat@mit.edu

***Abstract*— Simple rate-1/3 single-error-correcting unrestricted and CSS-type quantum convolutional codes are constructed from classical self-orthogonal $\mathbb{F}_4$-linear and $\mathbb{F}_2$-linear convolutional codes, respectively. These quantum convolutional codes have higher rate than comparable quantum block codes or previous quantum convolutional codes, and are simple to decode. A block single-error-correcting $[9, 3, 3]$ tail-biting code is derived from the unrestricted convolutional code, and similarly a $[15, 5, 3]$ CSS-type block code from the CSS-type convolutional code.**

I. Introduction

The field of quantum error-correcting codes (QECCs) has made substantial progress since the first 9-qubit single-error-correcting code was proposed by Shor in 1995 [12]. More efficient 7-qubit and 5-qubit single-error-correcting codes have been discovered [9]. A general theory of stabilizer codes has been elucidated [3], [7], [9]. Within this framework, a theory of $\mathbb{F}_4$-linear stabilizer codes has been developed [4]. Among these codes are Calderbank-Shor-Steane (CSS) codes [2], [13], which are based on binary codes. Using these structures, a large variety of block QECCs have been proposed.

In classical coding, practical systems have mostly used convolutional codes rather than block codes, because convolutional codes are usually superior in terms of their performance-complexity tradeoff. While this tradeoff does not seem to have been much of an issue to date for QECCs, a few attempts have been made to construct quantum convolutional codes (QCCs).

Chau [5], [6] proposed several "quantum convolutional codes," but whether this term is actually appropriate for the Chau codes is debatable. Ollivier and Tillich [10], [11] have given an example of a rate-1/5 single-error-correcting QCC, and have addressed gate-level implementation issues, but unfortunately their example QCC does not improve on the comparable 5-qubit block code in either performance or complexity. Most recently, Almeida and Palazzo [1] have constructed a rate-1/4 single-error-correcting Shor-type concatenated QCC; this code has a higher rate than comparable block codes, but its encoding and decoding appear to be rather complex.

In this paper, we present via simple examples four new classes of quantum codes— namely, $\mathbb{F}_4$-based and CSS-type convolutional and tail-biting codes. We claim to exhibit:

- The first QCCs with clear advantages in both performance and complexity over comparable block codes;
- The first quantum tail-biting codes, with recognition of their complexity advantages as quantum block codes;
- The first CSS-type convolutional codes, with recognition of their complexity advantages over $\mathbb{F}_4$-based codes.

Specifically, we present rate-1/3 single-error-correcting $\mathbb{F}_4$-based and CSS-type QCCs which have higher rate than any of these prior single-error-correcting codes, and which are simple to decode. Moreover, we derive from these codes simple tail-biting block codes, which also have rate 1/3, and which can correct single errors with equally simple decoding algorithms. In future work, we will generalize these examples.

In Section II, using the theory of $\mathbb{F}_4$-linear stabilizer codes developed by Calderbank, Rains, Shor and Sloane [4], we construct a simple rate-1/3 single-error-correcting quantum convolutional code from a classical rate-1/3 self-orthogonal $\mathbb{F}_4$-linear convolutional code. We give a simple decoding algorithm for this code that involves only a 9-entry table lookup. Using tail-biting, we derive a $[9, 3, 3]$ (*i.e.,* 9-qubit, rate-1/3, single-error-correcting) block stabilizer code, which can be decoded by the same simple decoding algorithm.

In Section III, we construct CSS-type codes based on binary codes, which have certain advantages over unrestricted $\mathbb{F}_4$-linear codes; in particular, bit flip and phase flip errors may be corrected independently. For example, the Steane 7-qubit code is a CSS-type code which may be preferred to the 5-qubit single-error-correcting block code, even though it has lower rate. Here we present a rate-1/3 single-error-correcting CSS-type quantum convolutional code which is extremely simple to decode. We derive from this code a $[15, 5, 3]$ tail-biting single-error-correcting block code which has the same rate, and an equally simple decoding algorithm.

II. Codes based on $\mathbb{F}_4$-linear codes

The development of Calderbank, Rains, Shor and Sloane [4] leads to the following proposition:

Proposition A. Given n, k with $0 \leq k \leq n$ and $n - k$ even, and given a classical self-orthogonal $(n, (n - k)/2)$ $\mathbb{F}_4$-linear block code $\mathcal{C}$ over the quaternary field $\mathbb{F}_4$ whose orthogonal $(n, (n + k)/2)$ code $\mathcal{C}^\perp$ under the Hermitian inner product has minimum Hamming distance d, there exists a quantum

$[n, k, d]$ stabilizer code that encodes k qubits into n qubits and can correct any pattern of up to $\lfloor (d-1)/2 \rfloor$ qubit errors.

The codes $\mathcal{C}$ and $\mathcal{C}^\perp$ are the quaternary label codes $L(S)$ and $L(N(S))$ of the stabilizer group S and the normalizer group $N(S)$, respectively, where the quaternary labels of the four Pauli matrices $\{I, X, Y, Z\}$ are respectively the elements $\{0, \omega, 1, \overline{\omega}\}$ of the quaternary field $\mathbb{F}_4$.

As with a classical code, decoding of a stabilizer code involves measuring a set of $(n-k)/2$ $\mathbb{F}_4$-syndromes $S_j = \langle L(\mathbf{E}), \mathbf{g}_j \rangle \in \mathbb{F}_4$, where $\{\mathbf{g}_j\}$ is a set of $(n-k)/2$ generators of $\mathcal{C}$, and $\langle L(\mathbf{E}), \mathbf{g}_j \rangle$ denotes the Hermitian inner product of $\mathbf{g}_j$ with a quaternary error label sequence $L(\mathbf{E})$. The syndromes identify the error label sequence $L(\mathbf{E})$ as belonging to one of $4^{(n-k)/2}$ cosets of the orthogonal code $\mathcal{C}^\perp$.

The decoder then determines the error label sequence of minimum Hamming weight in that coset. If $n - k$ is not too large, then this can be done by a table lookup in a table with $4^{(n-k)/2}$ entries. (The question of decoding complexity for large codes seems hardly to have been addressed previously in the QECC literature, with the notable exception of [8].)

Example A (Five-qubit "quantum Hamming code"). There exists a $(5, 2)$ self-orthogonal linear block code $\mathcal{C}$ over $\mathbb{F}_4$, generated by $\mathbf{g}_1 = (0, \overline{\omega}, \omega, \omega, \overline{\omega})$ and $\mathbf{g}_2 = (\overline{\omega}, 0, \overline{\omega}, \omega, \omega)$, whose orthogonal code $\mathcal{C}^\perp$ is a $(5, 3, 3)$ linear Hamming code over $\mathbb{F}_4$. There therefore exists a quantum $[5, 1, 3]$ code; *i.e.,* a code that encodes 1 qubit into 5 qubits, and corrects any single error. Because the 15 possible single-error label sequences $L(\mathbf{E})$ map one-to-one to the 15 nonzero cosets of $\mathcal{C}^\perp$, this is a "quantum Hamming code." Decoding may be done by a table lookup in a table with 16 entries. □

A. A simple rate-1/3, single-error-correcting QCC

We now construct a rate-1/3 convolutional stabilizer code with minimum Hamming distance $d = 3$ using Proposition A; *i.e.,* we find a classical self-orthogonal rate-1/3 $\mathbb{F}_4$-linear convolutional code $\mathcal{C}$ whose orthogonal code $\mathcal{C}^\perp$ under the Hermitian inner product has minimum distance 3.

Consider the classical rate-1/3 $\mathbb{F}_4$-linear shift-invariant convolutional code $\mathcal{C}$ with generators:

$$\begin{array}{c|ccc|ccc|ccc|c}
 & & & & & \cdots & & & & & \\
\cdots & 1 & 1 & 1 & 1 & \omega & \overline{\omega} & 0 & 0 & 0 & \cdots \\
\cdots & 0 & 0 & 0 & 1 & 1 & 1 & 1 & \omega & \overline{\omega} & \cdots \\
 & & & & & \cdots & & & & &
\end{array}$$

In other words, for every block of $n = 3$ qubits, there is one generator, so the classical rate is $1/3$. The generators are of the "sliding block" type; that is, every generator is a shift by an integral number of blocks of a single basic generator $(\ldots, 000, 111, 1\omega\overline{\omega}, 000, \ldots)$.[1] In D-transform notation, the code generators are $D^k(1 + D, 1 + \omega D, 1 + \overline{\omega} D), k \in \mathbb{Z}$.

[1]The stabilizer group S is actually generated by sequences of Pauli matrices that correspond to multiples by ω and $\overline{\omega}$ of the above generators; *i.e.,* the generators of S are the shifts by an integral number of blocks of the two basic generators $(\ldots, III, XXX, XZY, III, \ldots)$ and $(\ldots, III, ZZZ, ZYX, III, \ldots)$.

In principle, $\mathcal{C}$ has an infinite number of generators covering an infinite number of blocks. Later we will discuss methods for making such a code into a finite block code. However, the code constraints are localized; the code symbols in any block are a function only of the "current" and "previous" generators. Such a convolutional code is said to have a "memory" or "constraint length" of one block ($\nu = 1$).

All generators are orthogonal under the Hermitian inner product, so $\mathcal{C}$ is self-orthogonal. We will take $\mathcal{C}$ as the quaternary label code $L(S)$ of a convolutional stabilizer code.

The rate of this convolutional stabilizer code in quantum terms is also 1/3; *i.e.,* the code encodes one qubit stream into a second stream at a rate of three qubits per original qubit.

The orthogonal code $\mathcal{C}^\perp$ under the Hermitian inner product is a rate-2/3 $\mathbb{F}_4$-linear shift-invariant convolutional code whose generators are as follows (in D-transform notation, multiples of $(\overline{\omega}, \omega, 1)$ and $(1 + D, 1 + \omega D, 1 + \overline{\omega} D)$ by D^k):

$$\begin{array}{c|ccc|ccc|ccc|c}
 & & & & & \cdots & & & & & \\
\cdots & \overline{\omega} & \omega & 1 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\
\cdots & 1 & 1 & 1 & 1 & \omega & \overline{\omega} & 0 & 0 & 0 & \cdots \\
\hline
\cdots & 0 & 0 & 0 & \overline{\omega} & \omega & 1 & 0 & 0 & 0 & \cdots \\
\cdots & 0 & 0 & 0 & 1 & 1 & 1 & 1 & \omega & \overline{\omega} & \cdots \\
\hline
 & & & & & \cdots & & & & &
\end{array}$$

The minimum Hamming distance of $\mathcal{C}^\perp$ is 3, and the only codewords of weight 3 are single-block codewords. This is easily seen because $(\overline{\omega}\omega 1)$ and (111) generate a $(3, 2, 2)$ $\mathbb{F}_4$-linear block code, so every codeword accumulates a Hamming weight of at least 2 in its first block; similarly, every codeword accumulates a Hamming weight of at least 2 in its last block. The only single-block codewords are multiples of $(\overline{\omega}\omega 1)$, which have Hamming weight 3. The convolutional stabilizer code defined by $\mathcal{C}$ thus has minimum Hamming distance 3, so it is a single-error-correcting code.

B. Decoding algorithms

For decoding, we first measure each generator $\mathbf{g}_j$ of $\mathcal{C}$ to obtain a sequence of $\mathbb{F}_4$-syndromes $S_j = \langle L(\mathbf{E}), \mathbf{g}_j \rangle \in \mathbb{F}_4$, where $L(\mathbf{E})$ denotes the quaternary error label sequence $L(\mathbf{E})$, at a rate of one $\mathbb{F}_4$-syndrome for each block. This determines a coset of the orthogonal convolutional code $\mathcal{C}^\perp$. We then need to find the minimum-weight coset leader in that coset.

For any convolutional code, a standard way of finding coset leaders is by a Viterbi algorithm (VA) search. It can easily be seen that $\mathcal{C}^\perp$ has a trellis diagram with 4 states at each block boundary, and with 64 transitions between trellis states during each block. A VA search through such a trellis is not difficult, but requires of the order of 64 computations per block.

If our objective is merely correction of single errors, however, then we can use a much simpler algorithm, as follows. As long as all syndromes are zero, we assume that no errors have occurred. Then, if a nonzero syndrome S_j occurs, we assume that a single error has occurred in one of the three qubits in block j; the error is characterized by a label 3-tuple $\mathbf{e}_j = L(\mathbf{E}_j)$. The nine possible weight-1 error 3-tuples $\mathbf{e}_j$ lead to the following syndromes (S_j, S_{j+1}):

$\mathbf{e}_j$	(S_j, S_{j+1})
100	$(1,1)$
$\omega00$	(ω,ω)
$\overline{\omega}00$	$(\overline{\omega},\overline{\omega})$
010	$(\overline{\omega},1)$
$0\omega0$	$(1,\omega)$
$0\overline{\omega}0$	$(\omega,\overline{\omega})$
001	$(\omega,1)$
00ω	$(\overline{\omega},\omega)$
$00\overline{\omega}$	$(1,\overline{\omega})$

Since these nine syndrome pairs (S_j, S_{j+1}) are distinct, we can map (S_j, S_{j+1}) to the corresponding single-error label 3-tuple $\mathbf{e}_j$ using a simple 9-entry table lookup, and then correct the error as indicated. (If $S_{j+1} = 0$, *i.e.,* if S_j is an isolated nonzero syndrome, then we have detected a weight-2 error.)

We see that this simple algorithm can correct any single-error pattern $\mathbf{E}_j$, provided that there is no second error during blocks j and $j+1$. The decoder synchronizes itself properly whenever a zero syndrome occurs, and subsequently can correct one error in every second block, provided that every errored block is followed by an error-free block.

C. Terminated and tail-biting block codes

A standard method for reducing a convolutional code to a block code without loss of minimum distance is to terminate it; *i.e.,* to take as the block code the set of all convolutional code sequences that are nonzero only during a given interval of N blocks. The resulting code is a linear block code which is a subcode of the convolutional code, and thus has at least the same minimum distance.

For example, if $\mathcal{C}^\perp$ is terminated to an interval of N blocks, then it becomes an $\mathbb{F}_4$-linear block code with parameters $(3N, 2N-1, 3)$, because there are $2N-1$ generators that are nonzero only in the defined N-block interval. For instance, if $N = 3$, then we obtain a classical linear $(9,5,3)$ block code, which yields a quantum $[9,1,3]$ stabilizer code. As $N \to \infty$, the classical rate approaches 2/3, and the corresponding quantum rate approaches $1/3$.

Another, better idea for creating a linear block code from $\mathcal{C}^\perp$ is to use tail-biting, which preserves rate but possibly not minimum distance. For tail-biting, we take the set of all generators that "start" during a given interval of N blocks, and wrap around any blocks that do not fit within the given interval back to the beginning in cyclic "end-around" fashion.

For our orthogonal code $\mathcal{C}^\perp$, it turns out that there is no loss of minimum distance whenever $N \geq 3$. In particular, the following set of tail-biting generators generate a $(9,6,3)$ $\mathbb{F}_4$-linear block code, which is the normalizer label code of a quantum $[9,3,3]$ stabilizer code:

$\overline{\omega}$	ω	1	0	0	0	0	0	0
1	1	1	1	ω	$\overline{\omega}$	0	0	0
0	0	0	$\overline{\omega}$	ω	1	0	0	0
0	0	0	1	1	1	1	ω	$\overline{\omega}$
0	0	0	0	0	0	$\overline{\omega}$	ω	1
1	ω	$\overline{\omega}$	0	0	0	1	1	1

The second, fourth and sixth generators generate the dual $(9,3)$ tail-biting stabilizer label code.

To decode this code, we can use the same decoding algorithm as before, but now on a "circular" time axis. Specifically, if only a single error occurs, then one of the three resulting $\mathbb{F}_4$-syndromes will be zero, and the other two nonzero. The zero syndrome tells which block the error is in; the remaining two nonzero syndromes determine the error pattern according to the 9-entry table given earlier. Thus again we need only a 9-entry table lookup.

D. Error probability

We now briefly consider decoding error probability. We assume that the probability of an error in any qubit is p, independent of errors in other qubits. Our estimates do not depend on the relative probabilities of X, Y or Z errors.

For the 5-qubit block code of Example A, a decoding error may occur if there are 2 errors in any block, so the error probability is of the order of $\binom{5}{2}p^2 = 10p^2$ per block, or per encoded qubit.

For the rate-1/3 convolutional code, for each 3-qubit block, a decoding error may occur if there are 2 errors in that block, or 1 in that block and 1 in the subsequent block. The error probability is therefore of the order of $(3 + 3^2)p^2 = 12p^2$ per 3-qubit block, or per encoded qubit.

Finally, for the $[9,3,3]$ tail-biting block code, a decoding error may occur if there are 2 errors in a block of 9 qubits, so the error probability is of the order of $\binom{9}{2}p^2 = 36p^2$ per block, or $12p^2$ per encoded qubit.

We conclude that the decoding error probability is very nearly the same for any of these codes.

E. Discussion

Our quantum convolutional code has rate 1/3, which is greater than that of any previous simple single-error-correcting quantum code, block or convolutional. Our decoding algorithm involves only a 9-entry table lookup, which is at least as simple as that of any previous quantum code.

Our convolutional code rate and error-correction capability are comparable to those of a $[6,2,3]$ block stabilizer code. However, by the "quantum Hamming bound," there exists no $[6,2,3]$ block stabilizer code.

Our tail-biting code is a $[9,3,3]$ block stabilizer code. A code with the same parameters may be obtained by shortening a $[21,15,3]$ quantum Hamming code. However, such a shortened code would not have such a simple structure as our tail-biting code, nor such a simple decoding algorithm.

III. CSS-TYPE CODES

The binary field $\mathbb{F}_2$ is a subfield of the quaternary field $\mathbb{F}_4$. The $(n-k)/2$ generators of a classical self-orthogonal $(n,(n-k)/2)$ $\mathbb{F}_2$-linear code may therefore be taken as the generators of a self-orthogonal $(n,(n-k)/2)$ $\mathbb{F}_4$-linear code as in Proposition A. The resulting quantum stabilizer code is then of the type proposed by Calderbank and Shor [2] and Steane [13], which we call a *CSS-type code*.

Proposition B. Given n, k with $0 \leq k \leq n$ and $n-k$ even, and given a classical self-orthogonal $(n, (n-k)/2)$ $\mathbb{F}_2$-linear block code $\mathcal{C}$ over the binary field $\mathbb{F}_2$ whose orthogonal $(n, (n+k)/2)$ code $\mathcal{C}^\perp$ has minimum Hamming distance d, there exists an $[n, k, d]$ CSS-type code.

For CSS-type codes, we may think of the four Pauli matrices $\{I, X, Y, Z\}$ as having two-bit labels $\{00, 10, 11, 01\}$, respectively. The first bit is called the bit flip bit, and the second the phase flip bit. Thus an X error is a bit flip error, a Z error is a phase flip error, and a Y error is a combined bit and phase flip error.

CSS-type codes have the advantage that these two types of error bits are protected by two independent binary codes, which may be independently decoded. On the other hand, the parameters $[n, k, d]$ of CSS-type codes are not generally as good as those of unrestricted codes, because the parameters of binary codes are not generally as good as those of quaternary codes.

Decoding of a CSS-type code involves measuring a set of $(n-k)/2$ pairs of $\mathbb{F}_2$-syndromes $(s_{1,j}, s_{2,j}) = (\langle \ell_1(\mathbf{E}), \mathbf{g}_j \rangle, \langle \ell_2(\mathbf{E}), \mathbf{g}_j \rangle) \in (\mathbb{F}_2)^2$, where $\{\mathbf{g}_j\}$ is a set of $(n-k)/2$ generators of the binary code $\mathcal{C}$, and $\langle \ell_1(\mathbf{E}), \mathbf{g}_j \rangle$ and $\langle \ell_2(\mathbf{E}), \mathbf{g}_j \rangle$ denote the binary inner products of $\mathbf{g}_j$ with the two binary label sequences $\ell_1(\mathbf{E}), \ell_2(\mathbf{E})$, which respectively denote sequences of bit flip and phase flip errors. These syndromes identify each of the two error label sequences $\ell_1(\mathbf{E}), \ell_2(\mathbf{E})$ as belonging to one of $2^{(n-k)/2}$ cosets of the orthogonal code $\mathcal{C}^\perp$.

Two identical decoders may operate independently on each of these two syndrome sequences to determine the two error bit sequences of minimum Hamming weight in these respective cosets. If $n-k$ is not too large, then this can be done by two table lookups in a table with $2^{(n-k)/2}$ entries. Thus the decoding complexity is roughly twice the square root of the decoding complexity for a comparable quaternary code.

Example B (Seven-qubit Steane code [13]). There exists a $(7, 3)$ self-orthogonal linear block code $\mathcal{C}$ over $\mathbb{F}_2$, whose orthogonal code $\mathcal{C}^\perp$ is a $(7, 4, 3)$ linear Hamming code over $\mathbb{F}_2$. Thus there exists a $[7, 1, 3]$ CSS-type code. Decoding may be done by two table lookups in an 8-entry table. □

A. A simple rate-1/3, single-error-correcting CSS-type QCC

In this section we will construct a rate-1/3 CSS-type convolutional stabilizer code with minimum Hamming distance $d = 3$ using Proposition B. That is, we will find a binary self-orthogonal rate-1/3 linear convolutional code whose orthogonal code has minimum distance 3.

Consider the binary rate-1/3 convolutional code $\mathcal{C}$ whose generators are as follows:

$$\begin{array}{c|ccc|ccc|ccc|ccc|c} & & & & & & \cdots & & & & & & & \\ \cdots & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & \cdots \\ \cdots & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & \cdots \\ & & & & & & \cdots & & & & & & & \end{array}$$

(or $D^k(1 + D + D^2, 1 + D^2, 1), k \in \mathbb{Z}$, in D-transform notation). In other words, the classical rate is $1/3$, and every generator is a shift by an integral number of blocks of a single basic generator $(\ldots, 000, 111, 100, 110, 000, \ldots)$. Thus $\mathcal{C}$ has a "memory" of two blocks (constraint length $\nu = 2$).[2]

Each generator is orthogonal to all generators under the usual binary inner product, so the code is self-orthogonal. The generators of the orthogonal rate-2/3 binary convolutional code $\mathcal{C}^\perp$ are the shifts of two basic generators (in D-transform notation, multiples of $(1, 1 + D, D)$ and $(1, 1, D)$ by D^k):

$$\begin{array}{c|ccc|ccc|ccc|c} & & & & & \cdots & & & & & \\ \cdots & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & \cdots \\ \cdots & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & \cdots \\ \hline \cdots & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & \cdots \\ \cdots & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & \cdots \\ \hline & & & & & \cdots & & & & & \end{array}$$

It is easily verified that the minimum distance of $\mathcal{C}^\perp$ is $d = 3$.

Following Proposition B, we thus obtain from the binary self-orthogonal convolutional code $\mathcal{C}$ a CSS-type QCC of quantum rate $1/3$ and minimum distance $d = 3$.

B. Decoding algorithms

We will discuss only decoding of bit flip errors; phase flip errors are corrected independently and identically.

For decoding of bit flip errors, we measure each generator $\mathbf{g}_j$ of $\mathcal{C}$ to obtain a sequence of binary syndromes $s_j = \langle \ell_1(\mathbf{E}), \mathbf{g}_j \rangle$, the binary inner products of the generators $\mathbf{g}_j$ with the bit flip error label sequence $\ell_1(\mathbf{E})$, at a rate of one binary syndrome for each block.

Again, rather than VA decoding the 4-state trellis of the rate-2/3 code $\mathcal{C}^\perp$, we use a simple single-error-correction algorithm, as follows. As long as all syndromes are zero, we assume that no errors have occurred. Then, if a nonzero syndrome s_j occurs, we assume that a single error has occurred in one of the three bit flip bits in block j; the error is characterized by a label 3-tuple $\mathbf{e}_j = \ell_1(\mathbf{E}_j)$. The three possible weight-1 error 3-tuples $\mathbf{e}_j$ lead to the following bit flip syndromes:

$\mathbf{e}_j$	(s_j, s_{j+1}, s_{j+2})
100	$(1, 1, 1)$
010	$(1, 0, 1)$
001	$(1, 0, 0)$

Since the three syndrome pairs (s_{j+1}, s_{j+2}) are distinct, we can map (s_{j+1}, s_{j+2}) to the corresponding single-error pattern $\mathbf{e}_j$ using a simple 3-entry table lookup, and then correct it.

We see that this simple algorithm can correct any single-error pattern $\mathbf{e}_j$, provided that there is no second error during blocks j through $j + 2$. The decoder synchronizes itself properly whenever a zero syndrome occurs, and subsequently can correct one error in every third block, provided that every errored block is followed by two error-free blocks.

[2]The stabilizer group S is actually generated by sequences of Pauli matrices that correspond to multiples of the above generators by ω and $\overline{\omega}$. Thus the generators of S are the shifts by an integral number of blocks of two basic generators, $(\ldots, III, XXX, XII, XXI, III, \ldots)$ and $(\ldots, III, ZZZ, ZII, ZZI, III, \ldots)$. Note that these stabilizers affect only bit flip bits and phase flip bits, respectively.

C. Terminated and tail-biting block codes

For our normalizer code $\mathcal{C}^\perp$, it turns out that a tail-biting termination after N blocks results in no loss of minimum distance whenever $N \geq 5$. In particular, the following set of tail-biting generators generate a $(15, 10, 3)$ binary linear block code, which is the normalizer label code of a quantum $[15, 5, 3]$ CSS-type code:

1	1	0	0	1	1	0	0	0	0	0	0	0	0	0
0	0	1	1	1	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	0	0	1	1	0	0	0	0	0	0
0	0	0	0	0	1	1	1	0	0	0	0	0	0	0
0	0	0	0	0	0	1	1	0	0	1	1	0	0	0
0	0	0	0	0	0	0	0	1	1	1	0	0	0	0
0	0	0	0	0	0	0	0	0	1	1	0	0	1	1
0	0	0	0	0	0	0	0	0	0	0	1	1	1	0
0	1	1	0	0	0	0	0	0	0	0	0	1	1	0
1	1	0	0	0	0	0	0	0	0	0	0	0	0	1

To decode this code, we can use the same simple decoding algorithm as for the corresponding convolutional code, but now on a "circular" time axis. If only a single error occurs, then the first syndrome 1 after two zeroes (on a circular time axis) identifies the 3-tuple block of the error, and the next two bits determine its position within the block, according to the 3-entry table above.

D. Error probability

Again, we estimate the decoding error probabilities for these codes when qubit errors are independent and have probability p. We do not take into account that, because of the independence of the two decoders, there are some weight-2 error patterns that can be corrected (*e.g.,* X and Z); this would yield a minor improvement in our estimates.

For the 7-qubit block code of Example B, a decoding error may occur if there are 2 errors in any block, so the error probability is of the order of $\binom{7}{2}p^2 = 21p^2$ per block, or per encoded qubit.

For the rate-1/3 convolutional code, for each 3-qubit block, a decoding error may occur if there are 2 errors in that block, or 1 in that block and 1 in the two subsequent blocks. The error probability is therefore of the order of $(3+3\cdot 6)p^2 = 21p^2$ per 3-qubit block, or per encoded qubit.

Finally, for the $[15, 5, 3]$ tail-biting block code, a decoding error may occur if there are 2 errors in a block of 15 qubits, so the error probability is of the order of $\binom{15}{2}p^2 = 105p^2$ per block, or $21p^2$ per encoded qubit.

Again, we conclude that the decoding error probability is very nearly the same for any of these codes, and is about twice that of the codes of Section II.

E. Discussion

Our CSS-type quantum convolutional code has rate 1/3, which is greater than that of any previous simple CSS-type single-error-correcting quantum code, block or convolutional. Our decoder only requires using a 3-entry table lookup twice. It is arguably simpler than that of Section II.

Our convolutional code rate and error-correction capability are comparable to those of a $[9, 3, 3]$ CSS-type block code. However, no $[9, 3, 3]$ CSS-type block code exists, since there exists no $(9, 6, 3)$ binary linear block code, by the classical Hamming bound.

Our tail-biting code is a $[15, 5, 3]$ CSS-type block code. A code with the same parameters may be obtained by shortening a $[31, 21, 3]$ CSS-type block code. However, such a shortened code would not have such a simple structure as our tail-biting code, nor such a simple decoding algorithm.

IV. Future work

Using the same code construction principles, we have found rate-1/3 $\mathbb{F}_4$-based and CSS-type codes with up to 1024 states and minimum distances up to 8. We expect to present further examples of such codes at the ISIT.

Acknowledgments

We wish to acknowledge helpful comments by Robert Calderbank and David MacKay. Saikat Guha acknowledges the support of Prof. Jeffrey H. Shapiro and the U.S. Army Research Office (DoD MURI Grant No. DAAD-19-00-1-0177).

References

[1] A. C. A. Almeida and R. Palazzo, Jr., "A concatenated [(4, 1, 3)] quantum convolutional code," *Proc. 2004 IEEE Inform. Theory Workshop* (San Antonio, TX), Oct. 2004.

[2] A. R. Calderbank and P. W. Shor, "Good quantum error-correcting codes exist," *Phys. Rev. A*, vol. 54, pp. 1098–1105, Aug. 1996.

[3] A. R. Calderbank, E. M. Rains, P. W. Shor and N. J. A. Sloane, "Quantum error correction and orthogonal geometry," *Phys. Rev. Lett.*, vol. 78, pp. 405–408, 1997. ArXiv: `quant-ph 9605005`.

[4] 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, pp. 1369–1387, July 1998. ArXiv: `quant-ph 9608006`.

[5] H. F. Chau, "Quantum convolutional error-correcting codes," *Phys. Rev. A*, vol. 58(2), pp. 905–909, 1998.

[6] H. F. Chau, "Good quantum convolutional error-correction codes and their decoding algorithm exist," *Phys. Rev. A*, vol. 60(3), pp. 1966–1974, 1999.

[7] D. Gottesman, "A theory of fault-tolerant quantum computation," *Phys. Rev. A*, vol. 57, pp. 127–137, 1998. ArXiv: `quant-ph 9702029`.

[8] D. J. C. MacKay, G. Mitchison and P. L. McFadden, "Sparse-graph codes for quantum error correction," *IEEE Trans. Inform. Theory*, vol. 50, pp. 2315–2330, Oct. 2004.

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

[10] H. Ollivier and J.-P. Tillich, "Description of a quantum convolutional code," *Phys. Rev. Lett.*, vol. 91(17), pp. 1779021–4, 2003. ArXiv: `quant-ph 0304189`.

[11] H. Ollivier and J.-P. Tillich, "Quantum convolutional codes: Fundamentals," submitted to *IEEE Trans. Inform. Theory*, 2004. ArXiv: `quant-ph 0401134`.

[12] P. Shor, "Scheme for reducing decoherence in quantum computer memory," *Phys. Rev. A*, vol. 52, pp. 2493–2496, 1995.

[13] A. M. Steane, "Error-correcting codes in quantum theory," *Phys. Rev. Lett.*, vol. 77, pp. 793–797, July 1996.