

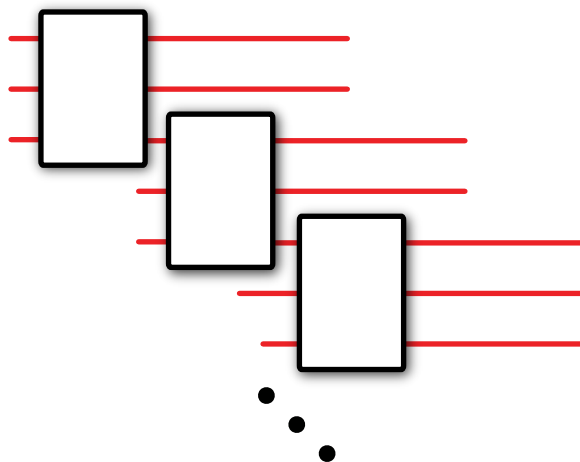
Minimal-Memory, Non-Catastrophic Quantum Convolutional Encoders

Monireh Houshmand and Saied Hosseini-Khayat

*Department of Electrical Engineering, Ferdowsi University of Mashhad,
Mashhad, Iran*

Mark M. Wilde

*School of Computer Science, McGill University,
Montreal, Quebec, Canada*



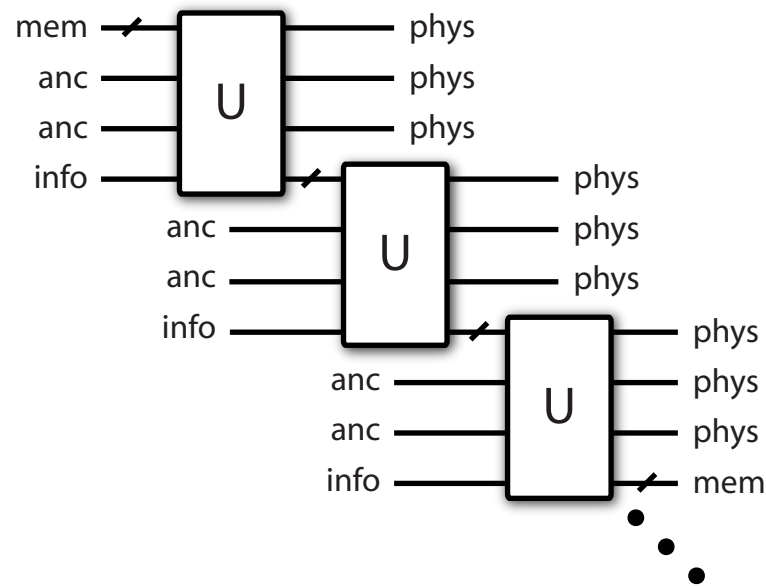
ISIT 2011, St. Petersburg, Russia

Quantum Convolutional Codes

A **quantum convolutional code** has a mathematical description with a set of stabilizer operators. An example code is below:

$$\begin{array}{ccc|ccc|ccc|} X & X & X & X & Z & Y & I & I & I & \dots \\ Z & Z & Z & Z & Y & X & I & I & I & \end{array}$$

A **quantum convolutional code** also requires an encoder—it would be ideal for this encoder to be **minimal-memory**



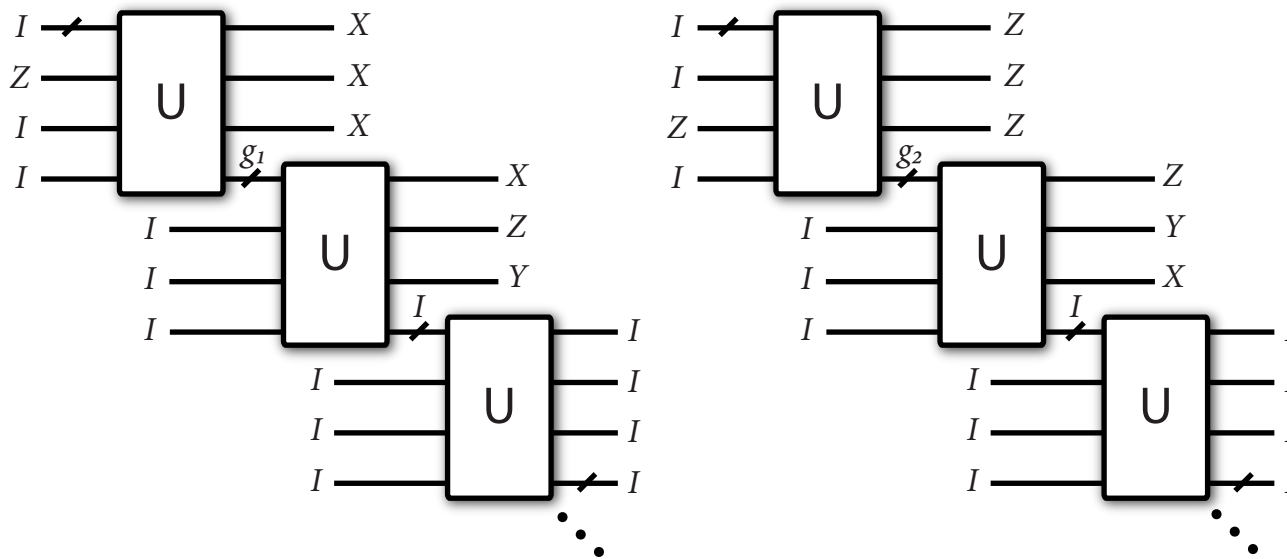
Ollivier and Tillich. *arXiv:quant-ph/0304189*
Forney, Grassl, Guha. *arXiv:quant-ph/0511016*
Grassl and Roetteler. *arXiv:quant-ph/0602129*

Finding a Minimal-Memory Encoder

The encoder transforms **unencoded operators** to **encoded operators**:

$$\begin{array}{c} I^{\otimes m} \\ I^{\otimes m} \end{array} \left| \begin{array}{ccc} Z & I & I \\ I & Z & I \end{array} \right| \begin{array}{ccc} I & I & I \\ I & I & I \end{array} \left| \dots \rightarrow \begin{array}{ccc} X & X & X \\ Z & Z & Z \end{array} \right| \begin{array}{ccc} X & Z & Y \\ Z & X & Y \end{array} \left| \begin{array}{ccc} I & I & I \\ I & I & I \end{array} \right| \dots$$

We can depict this transformation graphically:



So the encoder U should act as follows:

$$\begin{array}{cccc} I^{\otimes m} & Z & I & I & X & X & X & g_1 \\ I^{\otimes m} & I & Z & I & Z & Z & Z & g_2 \\ g_1 & I & I & I & X & Z & Y & I^{\otimes m} \\ g_2 & I & I & I & Z & Y & X & I^{\otimes m} \end{array} \rightarrow$$

Finding a Min-Memory Encoder (Ctd.)

The memory Pauli operators g_1 and g_2 should respect the input-output commutation relations:

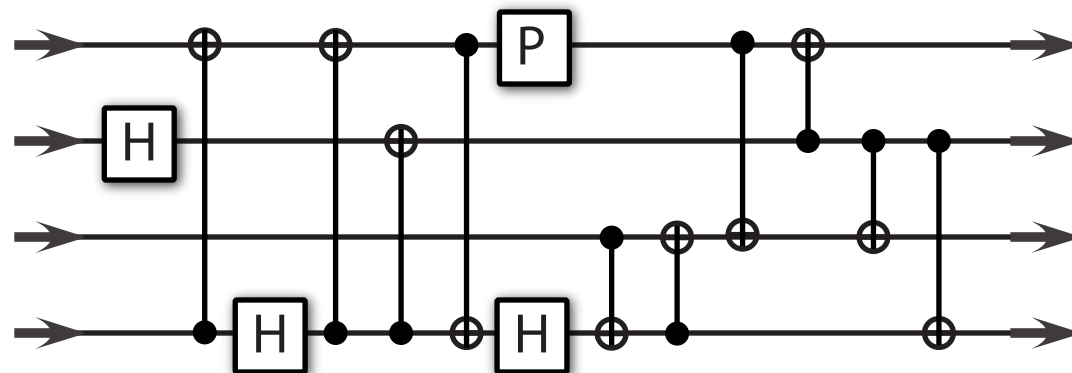
$$\{g_1, g_2\} = 0$$

A good choice for these operators is then $g_1 = X$ and $g_2 = Z$.

The encoder acts as follows and requires **only one memory qubit!**

$$\begin{array}{cccc} I & Z & I & I \\ I & I & Z & I \\ X & I & I & I \\ Z & I & I & I \end{array} \rightarrow \begin{array}{cccc} X & X & X & X \\ Z & Z & Z & Z \\ X & Z & Y & I \\ Z & Y & X & I \end{array}$$

One realization of the encoder is below:



Non-Catastrophicity

An encoder is **non-catastrophic** if *all zero physical-weight cycles have zero logical weight*.

This property is desirable because errors do not propagate infinitely for such encoders.

The minimal-memory encoder from the previous slide is **non-catastrophic**. Why?

Suppose (for a contradiction) that the encoder is **catastrophic**.

That is, there is some cycle of the following form:

$$\begin{array}{cccccccc} h_1 & s_{1,1} & s_{1,2} & l_1 & & I & I & I & h_2 \\ \vdots & \vdots & \vdots & \vdots & \rightarrow & \vdots & \vdots & \vdots & \vdots \\ h_p & s_{p,1} & s_{p,2} & l_p & & I & I & I & h_1 \end{array}$$

Then, the one-qubit memory states of this cycle should commute with both X and Z .

But this can only happen if the memory states are equal to the identity!

Thus, the encoder is non-catastrophic...

Finding an Online Decoder

Would like to find an **online decoder** for the encoder found from before

The encoder transforms the **unencoded logical operators** as follows:

$$\begin{array}{l}
 I^{\otimes m} \\
 I^{\otimes m}
 \end{array}
 \left| \begin{array}{ccc|ccc}
 I & I & X & I & I & I \\
 I & I & Z & I & I & I
 \end{array} \right| \cdots \rightarrow
 \begin{array}{ccc|ccc}
 Z & Y & I & X & Z & Y \\
 Y & I & Z & X & Z & Y
 \end{array}
 \left| \begin{array}{ccc}
 I & I & I \\
 I & I & I
 \end{array} \right| \cdots$$

Need to decode these in addition to decoding the encoded stabilizer operators

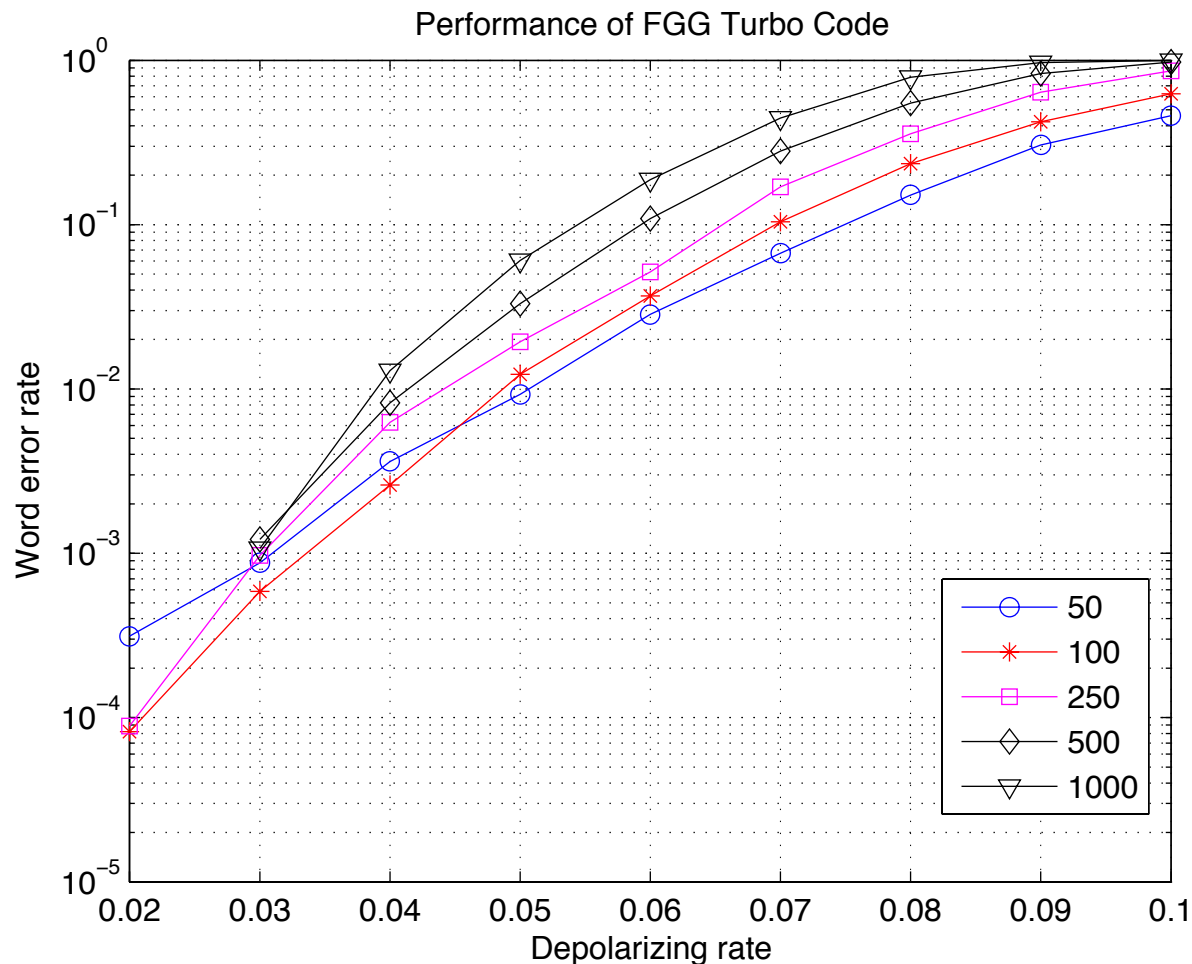
Can use the same technique!

$$\begin{array}{l}
 I^{\otimes m} \\
 I^{\otimes m} \\
 I^{\otimes m} \\
 I^{\otimes m} \\
 g'_1 \\
 g'_2 \\
 g'_3 \\
 g'_4
 \end{array}
 \begin{array}{ccc}
 Z & Y & I \\
 Y & I & Z \\
 X & X & X \\
 Z & Z & Z \\
 X & Z & Y \\
 X & Z & Y \\
 X & Z & Y \\
 Z & Y & X
 \end{array}
 \rightarrow
 \begin{array}{ccc}
 I & I & I \\
 I & I & I \\
 I & I & I \\
 I & I & I \\
 I & I & X \\
 I & I & Z \\
 Z & I & I \\
 I & Z & I
 \end{array}
 \begin{array}{l}
 g'_1 \\
 g'_2 \\
 g'_3 \\
 g'_4 \\
 I^{\otimes m} \\
 I^{\otimes m} \\
 I^{\otimes m} \\
 I^{\otimes m}
 \end{array}$$

Online decoder requires two memory qubits and is non-catastrophic

Simulation Results

The encoder for the FGG code requires only one memory qubit, and so we can simulate its performance in a quantum turbo code.



The performance curve for this quantum turbo code does not feature a pseudoshreshold, as is the case for other quantum turbo codes.

Poulin, Tillich, and Ollivier. arXiv:0712.2888

Wilde, Houshmand, and Hosseini-Khayat. arXiv:1011.5535

More General Example

How to handle **more general** examples?

$$\begin{array}{l}
 h_1 = X \ X \ X \ X \mid X \ X \ I \ X \mid I \ X \ I \ I \mid I \ I \ X \ X \\
 h_2 = Z \ Z \ Z \ Z \mid Z \ Z \ I \ Z \mid I \ Z \ I \ I \mid I \ I \ Z \ Z
 \end{array}$$

Similar idea for encoder:

Mem.	Anc.	Info.	Phys.	Mem.
$I^{\otimes m}$	$Z \ I$	$I \ I$	$X \ X \ X \ X$	$g_{1,1}$
$g_{1,1}$	$I \ I$	$I \ I$	$X \ X \ I \ X$	$g_{1,2}$
$g_{1,2}$	$I \ I$	$I \ I$	$I \ X \ I \ I$	$g_{1,3}$
$g_{1,3}$	$I \ I$	$I \ I$	$I \ I \ X \ X$	$I^{\otimes m}$
$I^{\otimes m}$	$I \ Z$	$I \ I$	$Z \ Z \ Z \ Z$	$g_{2,1}$
$g_{2,1}$	$I \ I$	$I \ I$	$Z \ Z \ I \ Z$	$g_{2,2}$
$g_{2,2}$	$I \ I$	$I \ I$	$I \ Z \ I \ I$	$g_{2,3}$
$g_{2,3}$	$I \ I$	$I \ I$	$I \ I \ Z \ Z$	$I^{\otimes m}$

→

Need to consider commutation relations again...

More General Example (Ctd.)

Enumerate **all of the commutation relations** for memory operators:

$$\begin{aligned} [g_{1,1}, g_{1,2}] &= [g_{1,1}, g_{1,3}] = [g_{1,1}, g_{2,1}] = \{g_{1,1}, g_{2,2}\} = \{g_{1,1}, g_{2,3}\} = 0, \\ [g_{1,2}, g_{1,3}] &= \{g_{1,2}, g_{2,1}\} = \{g_{1,2}, g_{2,2}\} = [g_{1,2}, g_{2,3}] = 0, \\ \{g_{1,3}, g_{2,1}\} &= [g_{1,3}, g_{2,2}] = [g_{1,3}, g_{2,3}] = 0, \\ [g_{2,1}, g_{2,2}] &= [g_{2,1}, g_{2,3}] = 0, \\ [g_{2,2}, g_{2,3}] &= 0, \end{aligned}$$

Can place in a **Memory Commutativity Matrix**

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

More General Example (Ctd.)

Can reduce this matrix by full-rank binary matrices (row operations) to the following **normal form**:

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \oplus \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \oplus \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Corresponds to the Pauli operators:

$X_1, Z_1, X_2, Z_2, X_3,$ and Z_3

Do the row operations *in reverse* to get

$$g_{1,1} = XIX, \quad g_{2,1} = ZXZ,$$

$$g_{1,2} = IIX, \quad g_{2,2} = IIZ,$$

$$g_{1,3} = IZI, \quad g_{2,3} = ZII$$

More General Example (Ctd.)

Encoder finally requires just **three memory qubits**:

Mem.			Anc.		Info.			Phys.				Mem.		
I	I	I	Z	I	I	I		X	X	X	X	X	I	X
X	I	X	I	I	I	I		X	X	I	X	I	I	X
I	I	X	I	I	I	I		I	X	I	I	I	Z	I
I	Z	I	I	I	I	I	\rightarrow	I	I	X	X	I	I	I
I	I	I	I	Z	I	I		Z	Z	Z	Z	Z	X	Z
Z	X	Z	I	I	I	I		Z	Z	I	Z	I	I	Z
I	I	Z	I	I	I	I		I	Z	I	I	Z	I	I
Z	I	I	I	I	I	I		I	I	Z	Z	I	I	I

The encoder is also non-catastrophic!

Follows from the previous argument:

All memory operators in a catastrophic cycle should commute with the above operators and are thus forced to be the identity operator.

(This same argument holds for full-rank memory commutativity matrices)

Summary

We have found a **simpler way to encode quantum convolutional codes**, and we have a general technique for finding a minimal-memory encoder for a given stabilizer representation of a quantum convolutional code.

In some cases, we can show that a minimal-memory encoder **always is non-catastrophic**.

Current Work

Would like to have a minimal memory encoder over all possible representations of the stabilizer (Houshmand has progress here)

Would like to have a general way for making a minimal-memory encoder be non-catastrophic as well (Some progress in arXiv:1105.0649)

Would like to have the same results for an online decoder corresponding to an online encoder