A Markov chain description of error correction

Marcus Silva (U. Waterloo), Martin Rötteler (NEC)

FTQC 2005, Aug 30

University of Waterloo

IQC Institute *for* Quantum Computing

Overview

(again)

i. The need to calculate error rates

ii. Erasure error models

iii. The Markov chain description of error correction

iv. Symmetries in the code and redundancy in the Markov chain

Building a QC

We want to know whether we can do useful computation even in the presence of noise

- we know how to encode information
- we know how to perform encoded operations

But we need to know how much effort needed to make the jump to useful computation:

- a threshold and more

Erasures vs. general errors

Classically an erasure corresponds to complete loss of information at a known location

$$\mathbb{E}(\rho) = \frac{1}{2}\mathbf{1} = \frac{1}{4}\left(\rho + \mathbf{X}\rho\mathbf{X} + \mathbf{Y}\rho\mathbf{Y} + \mathbf{Z}\rho\mathbf{Z}\right)$$

In QC, I will take it to correspond to *some* type of state corruption at a known location, e.g. it could also be

$$\mathbb{Z}(\rho) = \frac{1}{2}\left(\rho + \mathbf{Z}\rho\mathbf{Z}\right)$$

Erasure threshold theorem (roughly)

After one level of encoding, the error rate can be higher or lower. For a $[[n, k, d \geq 2]]$ code,

$$\epsilon^{(1)} \approx c_d \epsilon^d$$

is the encoded error rate. The break even point is called *the error threshold* for the given model

$$c_d \epsilon^d \approx \epsilon \to \epsilon \approx \left(\frac{1}{c_d}\right)^{\frac{1}{d-1}}$$

If we are below the threshold, concatenated codes give efficient use of resources.

We want more

It is necessary to know

- how the whole error model change under concatenated coding

  - (for reliable threshold estimation)

- what the trade off between different strategies under different error rates is

  - (for optimization of strategies)

Error rate estimation

- Exact calculation of the error recursion relation is very complex if done naively
  - requires tracking propagation of all the errors
- Most common method of estimating the threshold is via Monte Carlo simulation
  - which is slow for error models with multiple parameters, since they require simulation of multiple encoding levels [Svore et. al]

Correcting Erasures

Correction is greatly simplified by knowledge of error locations

- we only need to measure stabilizers that act non-trivially on the error

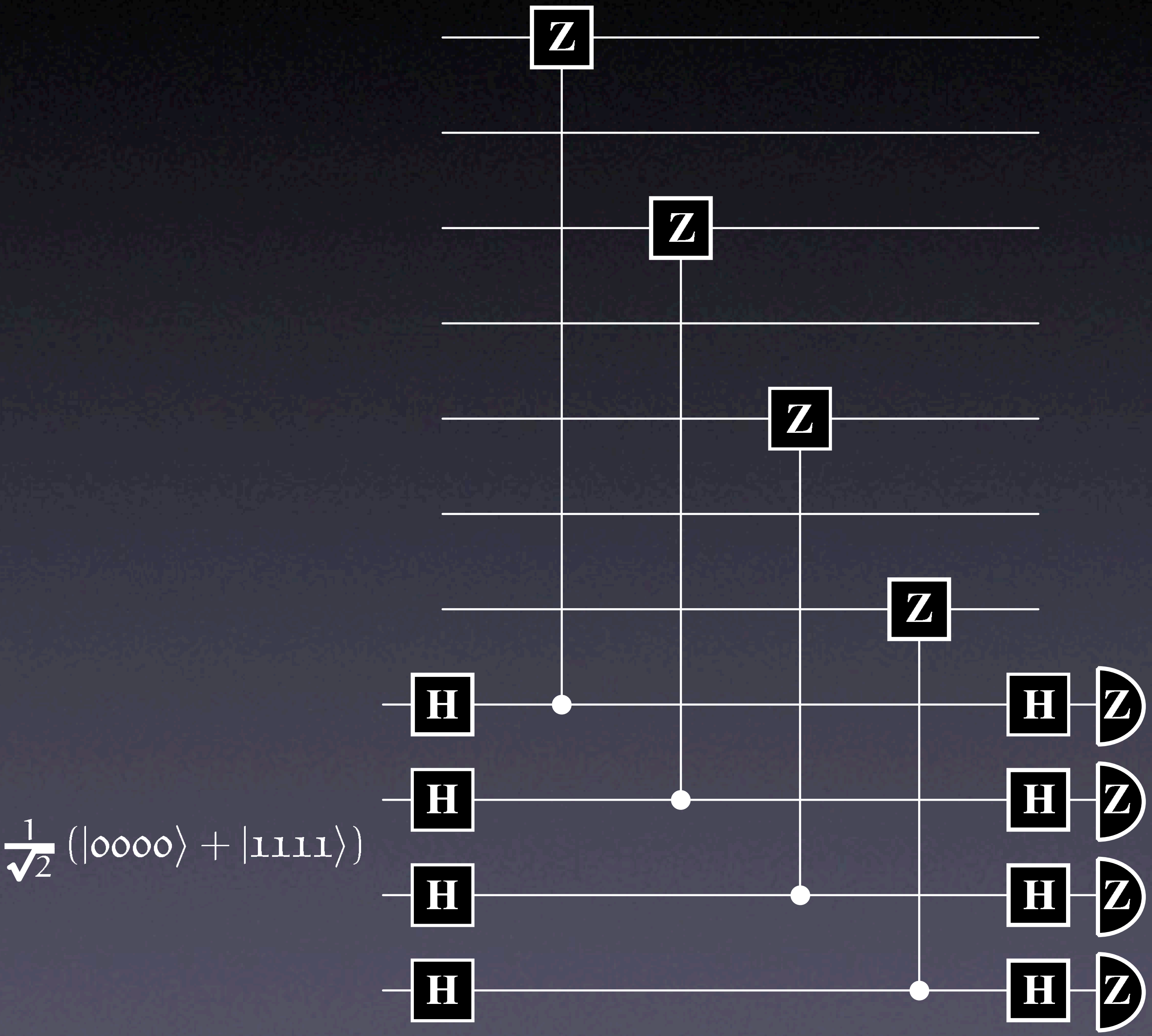
We can also continue correcting until we are *certain* the data is error free, or that it is uncorrectable.

Although not general enough, it is not an unreasonable error model

- see Linear Optics QC proposal [KLM]

The correction circuitry

e.g.



Tracking Errors

Say we have a universal set of Q operations.

The error model tells us how an error affects each of these fundamental operations.

$\Pr_{|0\rangle}(\mathbb{E})$ $\Pr_{\mathbf{H}}(\mathbb{E}|\mathbf{1})$ $\Pr_{\mathrm{CNOT}}(\mathbb{E}\otimes\mathbf{1}|\mathbb{Z}\otimes\mathbf{1})$

Notice that

we only track the error, not the state of the data.

we assume the error model is independent of the qubit position.

Markov Chain

The error model, in this form, can describe transition probabilities between erasures

$$\Pr_{\mathbf{H}^{\otimes 7}}\left(\mathbb{E}\otimes \mathbf{1}^{\otimes 6}|\mathbf{1}^{\otimes 7}\right)$$

as well as the initial distribution of erasures from state preparation

$$\Pr_{|0\rangle^{\otimes 7}}\left(\mathbb{E}\otimes \mathbb{Z}\otimes \mathbf{1}^{\otimes 5}\right)$$

So we can compute all error probabilities

$$\Pr(\mathbb{W}) = \sum_{\mathbb{V}} \Pr(\mathbb{W}|\mathbb{V})\Pr(\mathbb{V})$$

The encoded error rate

Define the transition matrix for error correction $[\mathcal{P}]_{ij} = \Pr(\mathbb{W}_i|\mathbb{W}_j)$

Given the initial error distribution $\mathcal{I}_0$, the distribution after N rounds of error correction is given by $\mathcal{P}^N \mathcal{I}_0$

The probability of encoded error, e.g. $\Pr(\mathbb{E}^{(1)})$ is then given by the sum of erasure patterns that correspond to $\mathbb{E}^{(1)}$

We can calculate error probabilities at *any* level of encoding!

The cost, vaguely

It is easy to see that the Markov chain is exponentially large in the number of qubits:

2187 elements for $\mathbb{E}, \mathbb{Z}$ and 7 qubits.

But there is a lot of symmetry that we are ignoring that:

the error model is "symmetric".

in the [[7,1,3]] CSS code, e.g., all single qubit erasures have very similar transition probabilities

Symmetries in error correction

Some permutations of the qubits leave the code space unchanged

- similar to elementary row operations in matrices

The group generated by these permutations is called the *autopermutation group of the code*. Formally,

$$\mathrm{AutPerm}(\mathcal{C}) = \{\pi | \pi \in S_n, \langle \pi G_i \pi^\dagger \rangle = \langle G_i \rangle\}$$

Equivalence of errors

We say that two Pauli operators $\mathbf{E}, \mathbf{F}$ are equivalent $\mathbf{E} \equiv \mathbf{F}$ if

$$\mathbf{E} = \pi \mathbf{F} \pi^{\dagger}, \pi \in \mathrm{AutPerm}(\mathcal{C})$$

Equivalence can be similarly defined for erasure patterns:

two erasure patterns $\mathbb{W}, \mathbb{V}$ are equivalent if there is a $\pi \in \mathrm{AutPerm}(\mathcal{C})$ which is a bijection between the Pauli decompositions of $\mathbb{W}, \mathbb{V}$

Operational Equivalence

If we require that the error correction circuitry have the same symmetries, that is

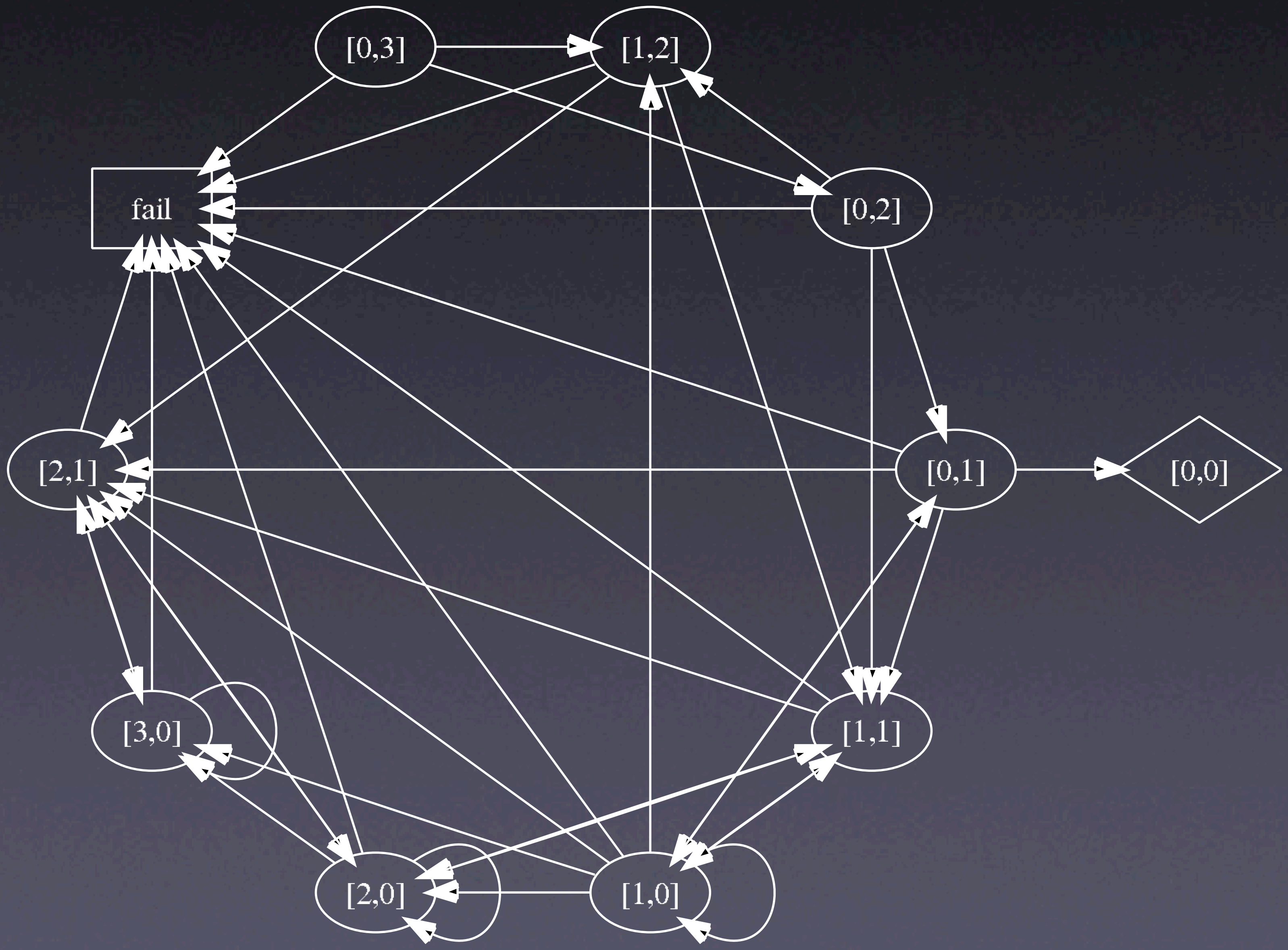
> measure stabilizer M to correct erasure $\mathbb{W}$ and measure stabilizer $\pi\mathrm{M}\pi^\dagger$ to correct erasure $\pi\mathbb{W}\pi^\dagger$

then the transition probabilities from equivalent erasures are identical

> and the Markov chain description of the error model becomes significantly smaller!

The reduced cost

With $\mathbb{E}.\mathbb{Z}$ and the [[7,1,3]] CSS code, we go from 2187 elements to 11



The gains

The run time of the method I described is independent of the error probabilities

- does not suffer from the long simulation times that can be a problem in Monte Carlo simulations.

Even in error models with multiple parameters, one can find true threshold.

Numerical approximations can be easily made, given the simplicity of Markov chains.

Open Questions

Can a similar approach be taken for general error models?

- syndrome extraction à la Steane has enough symmetry, but errors in syndrome extraction cause problems.
- Even if not for general computation, could it be useful for post-selected state preparation?

Questions?