CPEN 400Q Lecture 20
Conditions for quantum error correction; intro to stabilizers

Monday 24 March 2025

Announcements

- Quiz 9 today
- Last tutorial assignment tomorrow
- Sign up for project presentations, and final oral interview (Canvas calendar)
- Project rubric will be available in next few days
- A3 due tomorrow 23:59

Last time

We discussed the motivation for quantum error correction, and developed a mathematical (/graphical) model for noise.

Last time

We saw our first quantum error correcting code: the quantum repetition code, or bit flip code.

Last time

We designed logical operations for the bit flip code.

Learning outcomes

Today:

1. Correct bit *and* phase flip errors with the 9-qubit code
2. Outline the conditions under which errors can be corrected
3. Define the stabilizers of a quantum error correcting code

Phase flip errors

With our encoding

and appropriate circuitry, we can correct *bit flip errors* but not phase flip errors.

Phase flip code: encoding circuit

Main idea: make phase flip errors “look like” bit flip errors.

Shor code

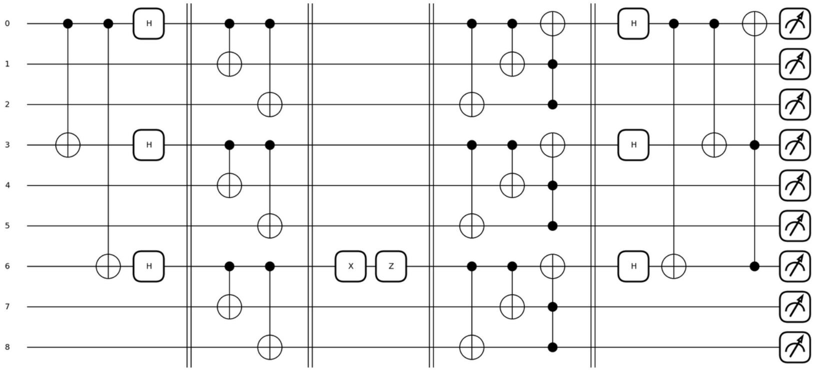
To correct a combination of one bit flip and/or phase flip error, we can *concatenate* codes.

Shor code

The Shor code can correct one *arbitrary* error *on a single qubit*.

Imagine a bit + phase flip on qubit 6:

Shor code



Correcting arbitrary errors

The 9-qubit code can correct *arbitrary* errors on a *single qubit*.

Represent possible errors as $\{E_i\}$, full error channel as

We can correct $\{E_i\} = \{I, X_j, Z_j, X_j Z_j\}_j$. If we can write an arbitrary error $\mathcal{E}_k$ on qubit j as

then we can correct it too.

Conditions for quantum error correction

How do we know if a particular error is correctable by a QECC?

Formal definition of a quantum error correcting code is a *subspace*, C, called the **codespace**.

Example: bit flip code.

- Codewords:
- Codespace: contents have the form

Define a **projector** onto the codespace,

Conditions for quantum error correction

The errors this code corrects are all mapped to different **orthogonal** subspaces of the full Hilbert space.

Conditions for quantum error correction

Theorem 10.1: (**Quantum error-correction conditions**) Let C be a quantum code, and let P be the projector onto C. Suppose $\mathcal{E}$ is a quantum operation with operation elements $\{E_i\}$. A necessary and sufficient condition for the existence of an error-correction operation $\mathcal{R}$ correcting $\mathcal{E}$ on C is that

(10.16)

for some Hermitian matrix α of complex numbers.

If such an $\mathcal{R}$ exists, $\{E_i\}$ is called a *correctable set of errors*.

Correctable errors: bit flip code

Consider $\{I, E_1 = XII, E_2 = IXI, E_3 = IIX\}$

Try including a 2-qubit error:

Example: amplitude damping

Not immediately clear how to express this in terms of no errors, bit, phase or bit/phase flip errors. Can we still correct it?

Straightforward to re-express K_1:

For K_0,

Bit flip code: recovery revisited

Last class, we considered two recovery circuits for the bit flip code.

Stabilizers

We can consider a more general invariant than parity: eigenvalues w.r.t. special subsets of the Pauli group.

The n-qubit Pauli group is

Example: Which two-qubit Paulis is this Bell state a $+1$ eigenstate of?

Bit flip code: stabilizers

Consider our logical states:

Which three-qubit Paulis are these states $+1$ eigenstates of?

Stabilizers

Let S be a subgroup of $\mathcal{P}_n$.

Let V_S be a set of states that are $+1$ eigenstates for all $P \in S$.

Then, S is the **stabilizer** of V_S.

Facts about S:

- $-I$ is never in S
- all items of S commute
- choosing S uniquely defines the fixed subspace, V_S

Bit flip code: stabilizers generators

Let's determine a *minimal* representation of the group in terms of its **stabilizer generators**.

Bit flip code: stabilizer measurement

We can use this formalism to construct new circuits for error detection and recovery: simply *measure the stabilizer generators*.

Next time

Next class:

- More stabilizer codes: phase flip, $[[5, 1, 3]]$ code, Steane code

Action items:

1. A3 (due 25 March 23:59)
2. Work on project

Recommended reading:

- From this class: Codebook EC; 10.3, 10.5
- For next class: Codebook EC; 10.1-10.5