Quantum Error Correction

Louis Golowich Wenjie Gong Ari Hatzimemos
Dylan Li Dylan Zhou

Physics 160
Harvard University

Final Project Presentation, 13 May 2020

Table of Contents

Introduction

"To be an Error and to be Cast out is part of God's Design."

William Blake

- Noise as a longstanding problem in information processing systems
  - *e.g.*, classical computers, modems, CD players, etc.
  - Noise is still a problem in quantum information
- Key idea: to protect a message against noise, *encode* the message by adding redundant information; even if some information is corrupted, redundancy allows us to *decode* and recover the original message

Project Framework

- Goals:
  - to implement various quantum error-correcting codes
    - we chose the 3-qubit, 9-qubit, 7-qubit codes
  - to analyze and compare their performances
    - *when are they effective?*
    - *when should we use error-correcting codes?*
- Tools:
  - Python's Qiskit package
  - IBM's quantum machines

3-Qubit Codes: Classical Inspiration

Classical Error Correction

- Encoding by *repetition codes*:

$$0 \to 000$$
$$1 \to 111.$$

- Decoding by *majority voting*:

$$\textit{Ex.:}\quad 001 \to 0.$$

- Analysis: Let p be the probability that a bit is flipped. This method fails when 2 or more bits are flipped, which occurs with probability $3p^2(1-p)+p^3$, so the probability of error is $p_e = 3p^2 - 2p^3$. Then this method is preferred when $p_e < p$, or $p < 1/2$.

Noisy Channels: The Bit Flip Channel

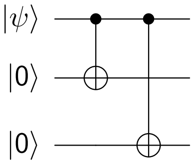
- One model for noise is the *bit flip channel* (analogous to classical channel).
- The bit flip channel flips qubits with probability p and leaves them untouched with probability $1 - p$.
- Equivalent to applying X gate with probability p.
- We protect qubits from this channel with the *bit flip code*.

3-Qubit Bit Flip Code: Encoding Logical Bits

- The goal is to correct bit flip errors.
- Encoding:

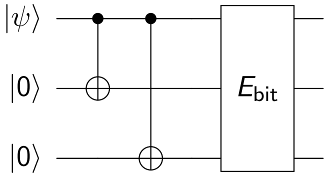
$$
\begin{aligned}
|0\rangle &\rightarrow |0_L\rangle \equiv |000\rangle \\
|1\rangle &\rightarrow |1_L\rangle \equiv |111\rangle \,.
\end{aligned}
$$

- Encoding circuit for 3-qubit bit flip code:



3-Qubit Bit Flip Code: Detecting Errors

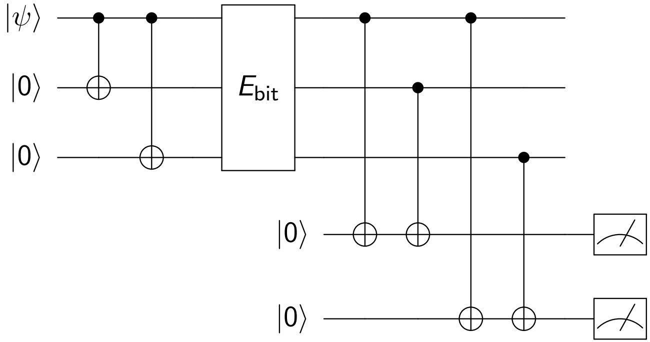
- Suppose there is a bit flip error after encoding:



- Error Detection (or *syndrome diagnosis*):
  - we would like to determine which, if any, of the qubits have been corrupted
  - four error syndromes: no error, bit flip on qubit one, bit flip on qubit two, bit flip on qubit three

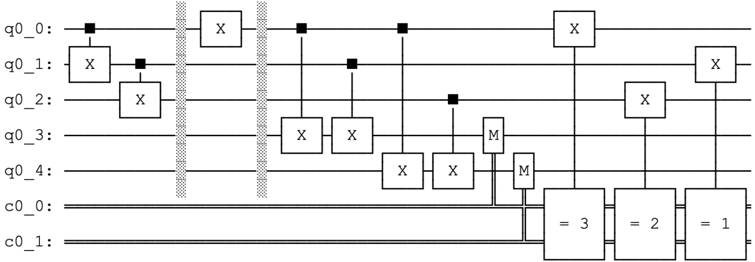
3-Qubit Bit Flip Code: Detecting Errors

- We can diagnose the syndrome using two ancillary qubits:



- Based on measurement results, we know where the error occured.

- Complete circuit for error correction (or *recovery*):



Analyzing the Bit Flip Code: Simulation

- Let's look at the performance of the 3-qubit bit flip code against bit flip channels of varying error probabilities p.

Analyzing the Bit Flip Code: Simulation

- Let's look at the performance of the 3-qubit bit flip code against bit flip channels of varying error probabilities p.
- Setup:
  1. encode a single qubit in state $|0\rangle$ into a logical state $|0_L\rangle = |000\rangle$
  2. create a bit flip channel which adds X gates with probability p
  3. run error correcting code
  4. measure final state

Analyzing the Bit Flip Code: Simulation

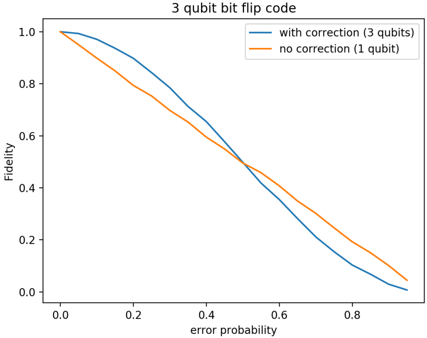
- Let's look at the performance of the 3-qubit bit flip code against bit flip channels of varying error probabilities p.
- Setup:
  1. encode a single qubit in state $|0\rangle$ into a logical state $|0_L\rangle = |000\rangle$
  2. create a bit flip channel which adds X gates with probability p
  3. run error correcting code
  4. measure final state
- We can calculate the accuracy of the error correcting code for a given p by repeating many times and taking the number of times we measure a correct final state $|000\rangle$ and dividing by the total number of trials.

Analyzing the Bit Flip Code: Simulation

- Let's look at the performance of the 3-qubit bit flip code against bit flip channels of varying error probabilities p.
- Setup:
  1. encode a single qubit in state $|0\rangle$ into a logical state $|0_L\rangle = |000\rangle$
  2. create a bit flip channel which adds X gates with probability p
  3. run error correcting code
  4. measure final state
- We can calculate the accuracy of the error correcting code for a given p by repeating many times and taking the number of times we measure a correct final state $|000\rangle$ and dividing by the total number of trials.
- We can compare this to the accuracy of a single qubit (without encoding or error correction) that goes through a bit flip channel with the same p to see when error correction is effective.

Analyzing the Bit Flip Code: Simulation

- Ran tests on Qiskit's simulator
- Probability p ranging from 0 to 1; 10000 trials for each p



- Observe crossover point at $p = 0.5$.
- For $p < 0.5$, error correcting code performs better than a single qubit with no correction.

Noisy Channels: Phase Flip Channel

- Another quantum channel is the *phase flip* error model.
- With probability p the relative phase of states $|0\rangle$ and $|1\rangle$ is flipped, with probability $1 - p$ it is left alone.
- Equivalent to applying Z operator with probability p.
- We fight this channel with the *phase flip code*.

3-Qubit Phase Flip Code

- No classical analog, but it is easy to turn the phase flip channel into a bit flip channel.
- Use x-basis for encoding:

$$
\begin{aligned}
|0\rangle &\to |0_L\rangle \equiv |+++\rangle \\
|1\rangle &\to |1_L\rangle \equiv |---\rangle\,.
\end{aligned}
$$

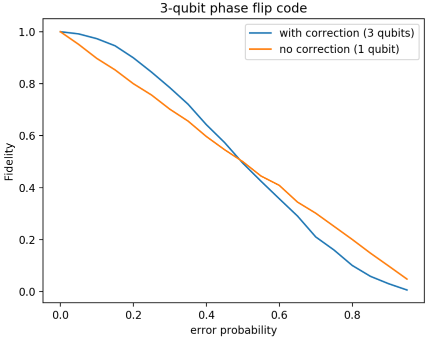
- Phase flip Z acts as bit flip for this encoding!

3-Qubit Phase Flip Code

- TODO

Analyzing the Phase Flip Code: Simulation

- Ran tests on Qiskit's simulator
- Probability p ranging from 0 to 1; 10000 trials for each p



- Observe crossover point at $p = 0.5$.
- For $p < 0.5$, error correcting code performs better than a single qubit with no correction.
- Nearly identical result to the bit flip code.

3-Qubit Codes on IBM's Machines

TODO

The Shor Code

- Can we protect against *arbitrary* errors?

The Shor Code

- Can we protect against *arbitrary* errors?
- Yes! $\longrightarrow$ The *Shor code*

The Shor Code: Encoding

- By combining the 3-qubit phase flip and bit flip codes, the Shor code protects against arbitrary errors.

The Shor Code: Encoding

- By combining the 3-qubit phase flip and bit flip codes, the Shor code protects against arbitrary errors.
- First encode the qubit using the phase flip code:

$$|0\rangle \rightarrow |+++\rangle\,, \quad |1\rangle \rightarrow |---\rangle\,.$$

The Shor Code: Encoding

- By combining the 3-qubit phase flip and bit flip codes, the Shor code protects against arbitrary errors.
- First encode the qubit using the phase flip code:

$$|0\rangle \to |+++\rangle\,, \quad |1\rangle \to |---\rangle\,.$$

- Then encode each of those qubits with the bit flip code:

$$|+\rangle \to (|000\rangle + |111\rangle)/\sqrt{2}, \quad |-\rangle \to (|000\rangle - |111\rangle)/\sqrt{2}.$$

The Shor Code: Encoding

- By combining the 3-qubit phase flip and bit flip codes, the Shor code protects against arbitrary errors.
- First encode the qubit using the phase flip code:

$$|0\rangle \to |+++\rangle\,, \quad |1\rangle \to |---\rangle\,.$$

- Then encode each of those qubits with the bit flip code:

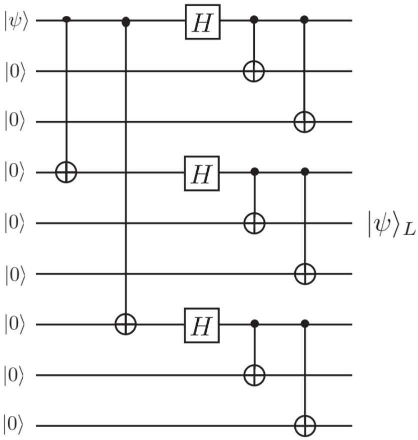
$$|+\rangle \to (|000\rangle + |111\rangle)/\sqrt{2}, \quad |-\rangle \to (|000\rangle - |111\rangle)/\sqrt{2}.$$

- The result is a 9-qubit code with codewords

$$|0\rangle \to |0_L\rangle \equiv \frac{(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)}{2\sqrt{2}}$$

$$|1\rangle \to |1_L\rangle \equiv \frac{(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)}{2\sqrt{2}}.$$

Encoding circuit for 9-qubit code:



The Shor 9-Qubit Code: Correcting Errors

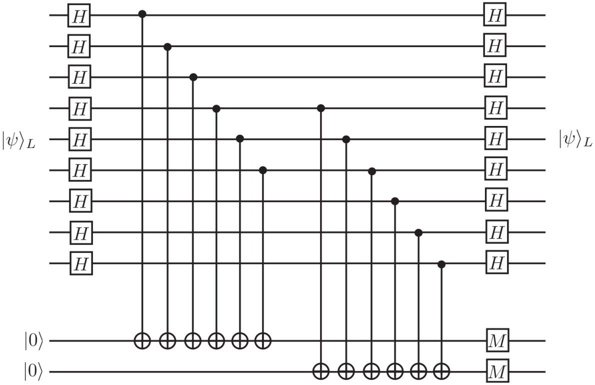
Bit Flip Error Correction

- On each block of three (i.e. qubits 0-2, 3-5, and 6-8), the 3-qubit circuit is utilized to correct for bit flips.

Phase Flip Error Correction

- The phase of the first two blocks of three (qubits 0-2 and 3-5) and the second two blocks of three (qubits 3-5 and 6-8) are compared to correct for phase flips.
- The phase correction necessitates two ancillary qubits. Thus, we need 8 ancilla: 6 for bit flip correction, and 2 for phase flip correction.

- The phase correction circuit, shown below, converts the qubits from the x-basis to the z-basis and checks parity of each block of two.



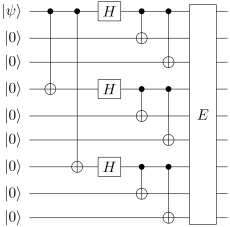
The Shor 9-Qubit Code: Correcting Phase Errors

- The following corrections are performed depending on the measured ancilla for phase flip correction:

$$10 \rightarrow \sigma_z \text{ on block 1}$$
$$01 \rightarrow \sigma_z \text{ on block 2}$$
$$11 \rightarrow \sigma_z \text{ on blocks 1 and 2.}$$

The Shor 9-Qubit Code: Error Correction Methodology

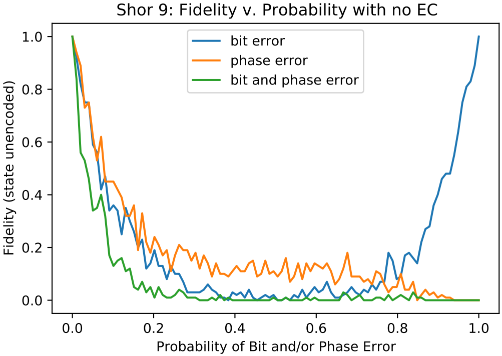
- We only consider error that occurs between the encoding step and the correcting step, thus simulating a memory error.



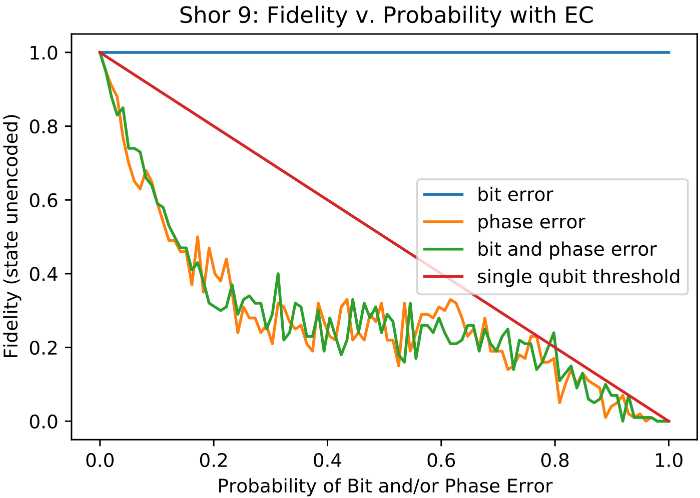
- Specifically, we consider a complete phase flip and/or bit flip (i.e. X or Z) that occurs independently on each of the 9 physical qubits with probability p.
- After the error, we measure the ancilla and apply the appropriate error correcting steps. Finally, we run the encoding circuit in reverse and measure the output to determine fidelity.

The Shor 9-Qubit Code: Simulation Performance with No Error Correction

- Initial state:
  $|0\rangle \to |0_L\rangle \equiv \frac{1}{\sqrt{8}}(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)$.
- Fidelity of un-encoded state measured against $|000000000\rangle$.



The Shor 9-Qubit Code: Simulation Performance with Error Correction



7-Qubit Code

Encodes 1 logical qubit using 7 physical qubits:

$$|\bar{0}\rangle = \frac{|0000000\rangle + |1010101\rangle + |0110011\rangle + |1100110\rangle + |0001111\rangle + |1011010\rangle + |0111100\rangle + |1101001\rangle}{\sqrt{8}}$$
$$|\bar{1}\rangle = \frac{|1111111\rangle + |0101010\rangle + |1001100\rangle + |0011001\rangle + |1110000\rangle + |0100101\rangle + |1000011\rangle + |0010110\rangle}{\sqrt{8}}$$

$$H^{\otimes 7}|\bar{0}\rangle = \frac{|\bar{0}\rangle + |\bar{1}\rangle}{\sqrt{2}}$$
$$H^{\otimes 7}|\bar{1}\rangle = \frac{|\bar{0}\rangle - |\bar{1}\rangle}{\sqrt{2}}$$

7-Qubit Code

$$|\bar{0}\rangle = \frac{|0000000\rangle + |1010101\rangle + |0110011\rangle + |1100110\rangle + |0001111\rangle + |1011010\rangle + |0111100\rangle + |1101001\rangle}{\sqrt{8}}$$

$$|\bar{1}\rangle = \frac{|1111111\rangle + |0101010\rangle + |1001100\rangle + |0011001\rangle + |1110000\rangle + |0100101\rangle + |1000011\rangle + |0010110\rangle}{\sqrt{8}}$$

$$H^{\otimes 7}|\bar{0}\rangle = \frac{|\bar{0}\rangle + |\bar{1}\rangle}{\sqrt{2}}$$

$$H^{\otimes 7}|\bar{1}\rangle = \frac{|\bar{0}\rangle - |\bar{1}\rangle}{\sqrt{2}}$$

- Of the 16 bit strings above, any two differ by ≥ 3 bits
- Intuition: therefore a single bit flip can be recovered
  - X error flips bit in $|\bar{0}\rangle, |\bar{1}\rangle$
  - Z error flips bit in $H^{\otimes 7}|\bar{0}\rangle, H^{\otimes 7}|\bar{1}\rangle$

Example recovery for X error in qubit 3

$$|\bar{0}\rangle = \frac{|0000000\rangle + |1010101\rangle + |0110011\rangle + |1100110\rangle + |0001111\rangle + |1011010\rangle + |0111100\rangle + |1101001\rangle}{\sqrt{8}}$$

$$X^{(3)}\,|\bar{0}\rangle = \frac{|0010000\rangle + |1000101\rangle + |0100011\rangle + |1110110\rangle + |0011111\rangle + |1001010\rangle + |0101100\rangle + |1111001\rangle}{\sqrt{8}}$$

$$\begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} \\ \\ \\ \\ \\ \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} = |3\rangle \text{ (in binary)}$$

Example recovery for X error in qubit 3

$$|\bar{0}\rangle = \frac{|0000000\rangle + |1010101\rangle + |0110011\rangle + |1100110\rangle + |0001111\rangle + |1011010\rangle + |0111100\rangle + |1101001\rangle}{\sqrt{8}}$$

$$X^{(3)}\,|\bar{0}\rangle = \frac{|0010000\rangle + |1000101\rangle + |0100011\rangle + |1110110\rangle + |0011111\rangle + |1001010\rangle + |0101100\rangle + |1111001\rangle}{\sqrt{8}}$$

$$\begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} \\ \\ \\ \\ \\ \\ \\ \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} = |3\rangle \text{ (in binary)}$$

Example recovery for X error in qubit 3

$$|\bar{0}\rangle = \frac{|0000000\rangle + |1010101\rangle + |0110011\rangle + |1100110\rangle + |0001111\rangle + |1011010\rangle + |0111100\rangle + |1101001\rangle}{\sqrt{8}}$$

$$X^{(3)}|\bar{0}\rangle = \frac{|0010000\rangle + |1000101\rangle + |0100011\rangle + |1110110\rangle + |0011111\rangle + |1001010\rangle + |0101100\rangle + |1111001\rangle}{\sqrt{8}}$$

$$\begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} \\ \\ \\ \\ \\ \\ \\ \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} = |3\rangle \text{ (in binary)}$$

Example recovery for X error

- In fact:

$$\begin{pmatrix}1 & 0 & 1 & 0 & 1 & 0 & 1\\ 0 & 1 & 1 & 0 & 0 & 1 & 1\\ 0 & 0 & 0 & 1 & 1 & 1 & 1\end{pmatrix} X^{(i)} \left|\overline{0}\right\rangle = |i\rangle \text{ (in binary) for all } i = 1, \ldots, 7$$

- Let H be matrix above. To recover from single X error, apply map

$$|v\rangle \otimes |0\rangle_A \mapsto |v\rangle \otimes |Hv\rangle_A$$

  and measure subsystem A. Result will be index i of bit flip, in binary!
- Also works for logical state 1, and for phase flips.

7-qubit code: Why does it work?

- The kernel of the matrix

$$H = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \in \mathbb{F}_2^{3\times 7}$$

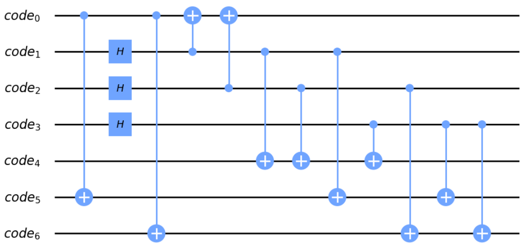
  consists of the 16 bit strings defining $\left|\overline{0}\right\rangle, \left|\overline{1}\right\rangle$
- A bit flip at position i of a vector v adds the ith row of H to Hv (basic linear algebra)
- The ith row of H is i in binary
- Same reasoning for phase flips = bit flips in rotated basis

7-qubit code: Initialization

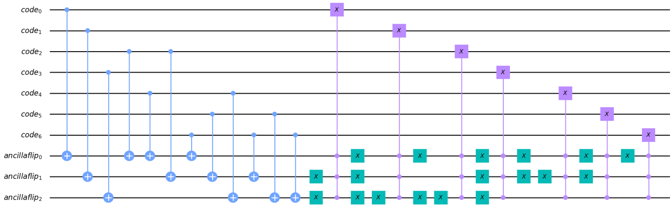
$$|\bar{0}\rangle = \frac{|0000000\rangle + |1010101\rangle + |0110011\rangle + |1100110\rangle + |0001111\rangle + |1011010\rangle + |0111100\rangle + |1101001\rangle}{\sqrt{8}}$$

$$|\bar{1}\rangle = \frac{|1111111\rangle + |0101010\rangle + |1001100\rangle + |0011001\rangle + |1110000\rangle + |0100101\rangle + |1000011\rangle + |0010110\rangle}{\sqrt{8}}$$

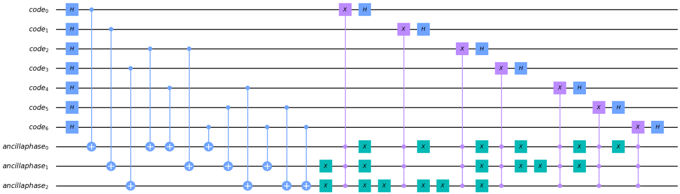
$$|\psi\rangle \rightarrow$$
$$|0\rangle \rightarrow$$
$$|0\rangle \rightarrow$$
$$|0\rangle \rightarrow$$
$$|0\rangle \rightarrow$$
$$|0\rangle \rightarrow$$
$$|0\rangle \rightarrow$$



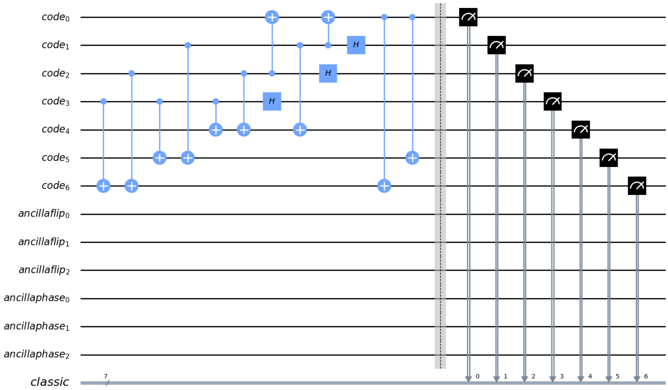
7-qubit code: Flip correction



7-qubit code: Phase correction

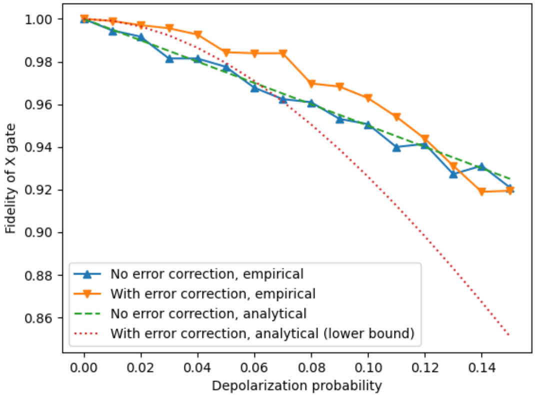


7-qubit code: Measurement



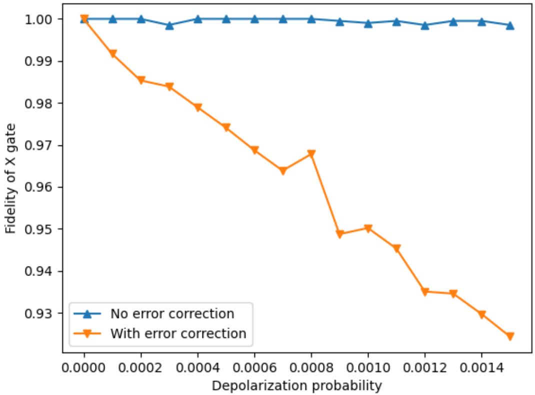
7-qubit code: Fidelity of X Gate under Depolarization

Encode $\|\bar{0}\rangle$	X	Correct flip	Correct phase	Decode, measure



7-qubit code: Fidelity of X Gate under Depolarization

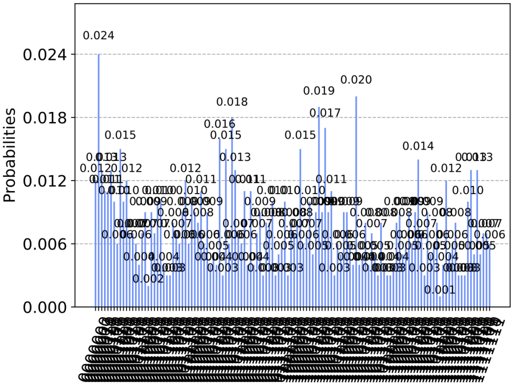
Encode $\lvert\bar{0}\rangle$	X	Correct flip	Correct phase	Decode, measure



7-qubit code: Simulation vs running on quantum computers?

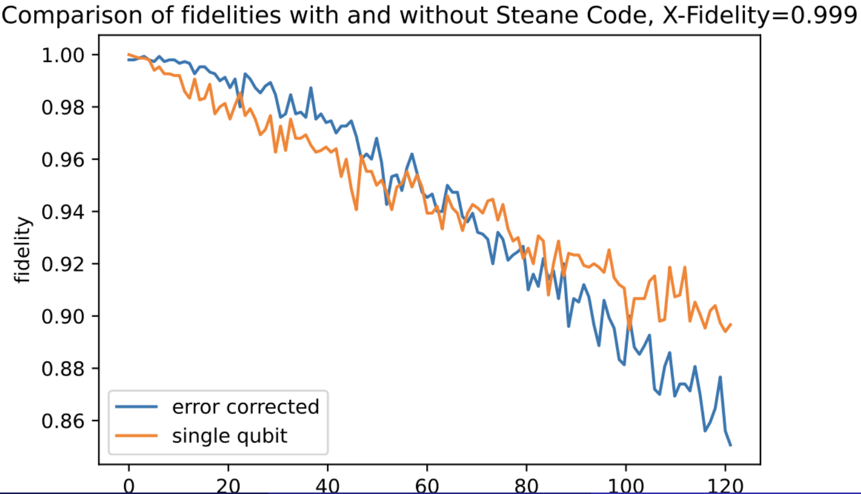
- The states should be clearly defined, but noise dominates the system

```
[ ] # using optimization_level=3
    counts = job.result().get_counts(projection_circuit)
    plot_histogram(counts)
```



7-qubit code: Useful with lower error probability

Encode $\lvert\bar{0}\rangle$	X	Correct flip	Correct phase	Decode, measure



7-qubit code: Adding Error Correction at different Timesteps

Encode $\|\bar{0}\rangle$	X	Correct flip	Correct phase	Decode, measure