

Quantum Computing (CST Part II)

Lecture 13: Quantum Error Correction

*We have learned that it is possible to fight
entanglement with entanglement.*

John Preskill

Resources for this lecture

Nielsen and Chuang p425-453 (sections 10.1-10.4) covers the material of this lecture.

The next section (10.5) of Nielsen and Chuang introduces the *stabiliser formalism* which, whilst being beyond the scope of this course, is very commonly used to describe error correction, so may be of interest to students considering further study / work in quantum computing.

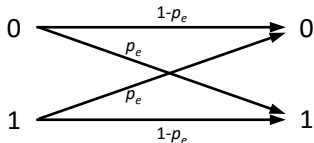
Why do we need quantum error correction?

Modern (classical) digital computers are, to all intents and purposes, error-free. The same cannot, however, be said of quantum computers, therefore error correction is required, in particular:

- Even though near-term hybrid quantum-classical algorithms have been conceptualised to achieve classically-intractable simulations, even in the presence of noise, it is becoming increasingly apparent that some amount of error correction is crucial to achieve satisfactory performance.
- More fundamentally, in order to assert that quantum algorithms will indeed achieve super-classical performance in practise, it is necessary to understand the “asymptotic significance” of quantum errors, and the possibility and efficiency of correction.

Classical errors: the binary symmetric channel

One of the simplest models for single-bit (classical) errors is the *binary symmetric channel*, in which each possible state of the bit, 0 and 1 “flips” to the other with some probability p_e :



Note that, without loss of generality we can assume $p_e \leq 0.5$, because if $p_e > 0.5$ then it is more likely than not that a bit-flip has occurred, so we can interpret a received 0 as a 1 and vice-versa. In the case where $p_e = 0.5$ we cannot recover any information from the channel.

Classical error correction: the three-bit repetition code

If we wish to send a single bit over a binary symmetric channel, then we can **encode** the bit, by simply repeating it three times. That is, if we wish to transmit a **0**, we send three bits (sequentially) in the state **0**, and likewise for **1**. This can be denoted as:

$$0 \rightarrow 000$$

$$1 \rightarrow 111$$

Once the three bits have been received, they are **decoded** by a “majority vote”. So in order for an error to occur, it is necessary that either two of the three bits have been flipped (which can occur in three different ways), or all three have been, that is:

$$p'_e = 3p_e^2(1 - p_e) + p_e^3$$

Which is less than p_e if $p_e < 0.5$. Typically, p_e is small, and we can describe this as **suppressing the error to** $\mathcal{O}(p_e^2)$.

Complicating factors with quantum error correction

Ostensibly, it appears that we cannot directly transfer classical error correction techniques to the problem of quantum error correction for three reasons:

1. The no-cloning principle forbids the copying of quantum states.
2. Measurement destroys quantum information.
3. Quantum states are continuous: $\alpha|0\rangle + \beta|1\rangle$.

Nevertheless, we shall see that with some ingenuity **we can correct quantum errors**.

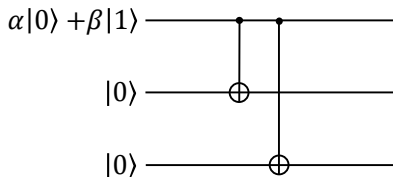
The three-qubit bit-flip code

The three-bit repetition code guarantees to return the correct bit value, so long as at most one of the bits in the code is flipped. We now use this as inspiration for the *three-qubit bit-flip code*, in which **entanglement rather than cloning** plays the role of the repetition. That is, we encode the computational basis states:

$$|0\rangle \rightarrow |000\rangle$$

$$|1\rangle \rightarrow |111\rangle$$

Which is achieved using the following circuit:

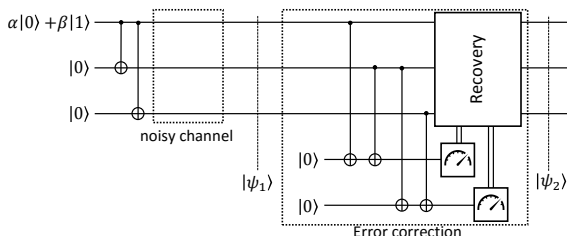


This has the following action on an arbitrary qubit state:

$$(\alpha|0\rangle + \beta|1\rangle)|0\rangle^{\otimes 2} \rightarrow \alpha|000\rangle + \beta|111\rangle$$

The three-qubit bit-flip code: error detection and recovery

To detect and recover errors, we supplement the circuit with two ancillas that we use for error detection:



We can thus detect and recover single-qubit bit-flips:

Bit-flip	$ \psi_1\rangle$	M_1	M_2	Recovery	$ \psi_2\rangle$
-	$\alpha 000\rangle + \beta 111\rangle$	0	0	$I \otimes I \otimes I$	$\alpha 000\rangle + \beta 111\rangle$
1	$\alpha 100\rangle + \beta 011\rangle$	1	0	$X \otimes I \otimes I$	$\alpha 000\rangle + \beta 111\rangle$
2	$\alpha 010\rangle + \beta 101\rangle$	1	1	$I \otimes X \otimes I$	$\alpha 000\rangle + \beta 111\rangle$
3	$\alpha 001\rangle + \beta 110\rangle$	0	1	$I \otimes I \otimes X$	$\alpha 000\rangle + \beta 111\rangle$

That is, we have made comparative *parity-check* measurements that tell us only about the error and not about the quantum state itself, **and so these measurements have not destroyed the quantum state.**

The three-qubit phase-flip code

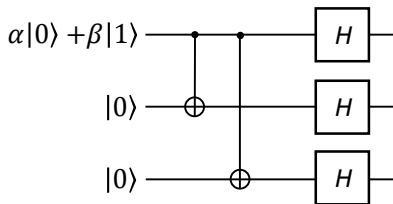
The three-qubit bit-flip code demonstrates how we can overcome two of the possible problems with quantum error correction that we previously identified:

- We can use entanglement to enable repetition.
- We can detect errors using parity-check measurements that do not destroy the quantum information.

However, we still have not addressed the fact that quantum states are continuous. To begin to do this, we'll look at an error correction code for a different type of error. The three-qubit phase-flip code has the following action on an arbitrary single-qubit state:

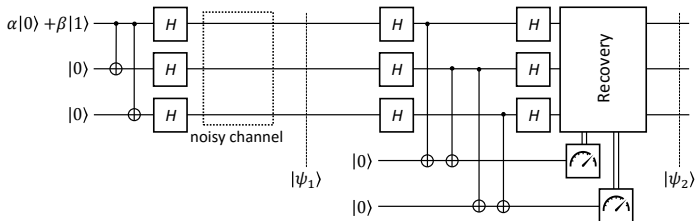
$$(\alpha|0\rangle + \beta|1\rangle)|0\rangle^{\otimes 2} \rightarrow \alpha|+++ \rangle + \beta|--- \rangle$$

Which is achieved by the following circuit:



Three-qubit phase-flip code: error detection and recovery

Once again, to detect and recover errors, we supplement the circuit with two ancillas that we use for error detection:



By definition, a phase flip sends:

$$|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \rightarrow \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = |-\rangle$$

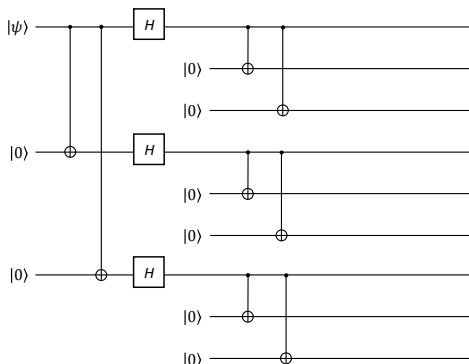
$$|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \rightarrow \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = |+\rangle$$

Thus we have:

Ph-flip	$ \psi_1\rangle$	M_1	M_2	Recovery	$ \psi_2\rangle$
-	$\alpha +++ \rangle + \beta --- \rangle$	0	0	$I \otimes I \otimes I$	$\alpha +++ \rangle + \beta --- \rangle$
1	$\alpha + + - \rangle + \beta + - - \rangle$	1	0	$Z \otimes I \otimes I$	$\alpha +++ \rangle + \beta --- \rangle$
2	$\alpha + - + \rangle + \beta - + - \rangle$	1	1	$I \otimes Z \otimes I$	$\alpha +++ \rangle + \beta --- \rangle$
3	$\alpha + + - \rangle + \beta - - + \rangle$	0	1	$I \otimes I \otimes Z$	$\alpha +++ \rangle + \beta --- \rangle$

The Shor code

The Shor code is a 9-qubit code which is constructed by *concatenating* the three-qubit bit-flip and three-qubit phase-flip codes:



This encodes the computational basis states as follows:

$$|0\rangle \rightarrow |0_L\rangle = \frac{1}{2\sqrt{2}} (|000\rangle + |111\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)$$

$$|1\rangle \rightarrow |1_L\rangle = \frac{1}{2\sqrt{2}} (|000\rangle - |111\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)$$

Correcting bit-flips with the Shor code

The Shor code can detect and correct a bit-flip on any single qubit. For example, suppose we have an arbitrary quantum state $\alpha|0\rangle + \beta|1\rangle$ which we encode with the Shor code as:

$$\frac{1}{2\sqrt{2}} \left(\alpha(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle) \right. \\ \left. + \beta(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle) \right)$$

If a bit-flip occurs on the first qubit, the state becomes:

$$\frac{1}{2\sqrt{2}} \left(\alpha(|100\rangle + |011\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle) \right. \\ \left. + \beta(|100\rangle - |011\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle) \right)$$

Which can be detected (and thus recovered from) by parity-check measurements between the first three qubits as in the three-qubit bit-flip code. By symmetry we can see that the same principle applies to all of the nine qubits.

Correcting phase-flips with the Shor code

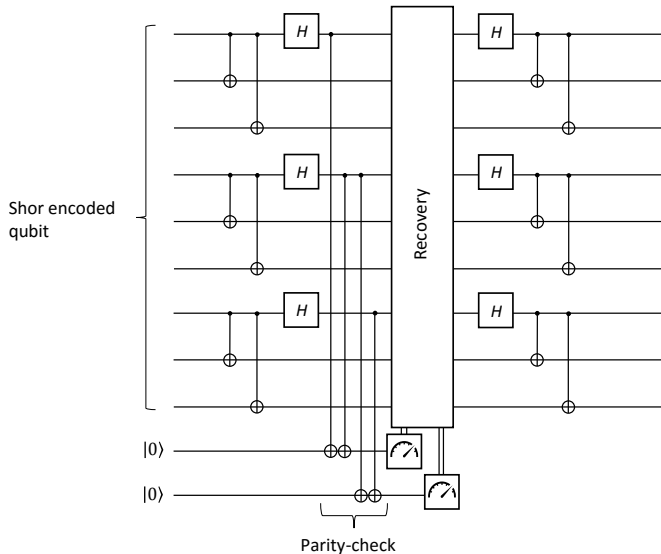
The Shor code can also detect and correct a phase-flip on any single qubit. If a phase-flip occurs on the first qubit, the state becomes:

$$\frac{1}{2\sqrt{2}} \left(\alpha(|000\rangle - |111\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle) \right. \\ \left. + \beta(|000\rangle + |111\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle) \right)$$

The key idea here is to detect which of the three blocks of three qubits has experienced a change of sign. This is achieved using the circuit shown on the following slide.

We can also correct combinations of bit- and phase-flips in this way.

Circuit for correcting phase-flips with the Shor code



The depolarising channel

When studying the (classical) three-bit repetition code, we saw that in practise it is more useful to think of it as a code that suppresses the error in the binary symmetric channel from p_e to $\mathcal{O}(p_e^2)$.

In the quantum case, we can see something similar: Consider the **depolarising channel**, in which a qubit is left unchanged with probability $1 - p_e$; experiences a bit-flip with probability $\frac{p_e}{3}$; experiences a phase-flip with probability $\frac{p_e}{3}$; or experiences both a bit- and phase-flip with probability $\frac{p_e}{3}$.

An analogous argument to that made for the binary symmetric channel can be made to show that the **Shor code suppresses the error from p_e to $\mathcal{O}(p_e^2)$ in the depolarising channel.**

Correcting any single qubit error with the Shor code (1)

Suppose the first qubit encounters an error which sends $|0\rangle \rightarrow a|0\rangle + b|1\rangle$ and $|1\rangle \rightarrow c|0\rangle + d|1\rangle$. We thus have the state:

$$\frac{1}{2\sqrt{2}} \left(\alpha(a|000\rangle + b|100\rangle + c|011\rangle + d|111\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle) \right. \\ \left. + \beta(a|000\rangle + b|100\rangle - c|011\rangle - d|111\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle) \right)$$

Letting $k + m = a$, $k - m = d$, $l + n = b$ and $l - n = c$, we get

$$\frac{1}{2\sqrt{2}} \left(k \left(\alpha(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle) \right. \right. \\ \left. \left. + \beta(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle) \right) \right. \\ \left. + l \left(\alpha(|100\rangle + |011\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle) \right. \right. \\ \left. \left. + \beta(|100\rangle - |011\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle) \right) \right. \\ \left. + m \left(\alpha(|000\rangle - |111\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle) \right. \right. \\ \left. \left. + \beta(|000\rangle + |111\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle) \right) \right. \\ \left. + n \left(\alpha(|100\rangle - |011\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle) \right. \right. \\ \left. \left. + \beta(|100\rangle + |011\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle) \right) \right)$$

Correcting any single qubit error with the Shor code (2)

As before, we first perform parity-check measurements to detect a bit-flip. The parity check for a bit-flip in the first block of three qubits requires two ancillas (the first comparing the first and second qubits, the second comparing the second and third qubits), whose state (after the parity-check CNOTs) we can append to the Shor code state:

$$\begin{aligned} & \frac{1}{2\sqrt{2}} \left(k \left(\alpha(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle) \right. \right. \\ & \quad \left. \left. + \beta(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle) \right) |00\rangle \right. \\ & + l \left(\alpha(|100\rangle + |011\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle) \right. \\ & \quad \left. + \beta(|100\rangle - |011\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle) \right) |10\rangle \\ & + m \left(\alpha(|000\rangle - |111\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle) \right. \\ & \quad \left. + \beta(|000\rangle + |111\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle) \right) |00\rangle \\ & + n \left(\alpha(|100\rangle - |011\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle) \right. \\ & \quad \left. + \beta(|100\rangle + |011\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle) \right) |10\rangle \left. \right) \end{aligned}$$

Correcting any single qubit error with the Shor code (3)

If the parity-check measurement outcome is 00, the state collapses to (un-normalised):

$$\begin{aligned} & k \left(\alpha(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle) \right. \\ & \quad \left. + \beta(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle) \right) \\ & + m \left(\alpha(|000\rangle - |111\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle) \right. \\ & \quad \left. + \beta(|000\rangle + |111\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle) \right) \end{aligned}$$

In which case there is no bit-flip. Or if the measurement outcome is 10:

$$\begin{aligned} & l \left(\alpha(|100\rangle + |011\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle) \right. \\ & \quad \left. + \beta(|100\rangle - |011\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle) \right) \\ & + n \left(\alpha(|100\rangle - |011\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle) \right. \\ & \quad \left. + \beta(|100\rangle + |011\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle) \right) \end{aligned}$$

i.e., a bit-flip has occurred which we can then correct.

Correcting any error by correcting only bit- and phase-flips

Following the bit-flip parity-check measurement (and correction if necessary) we perform a parity-check measurement to check for a phase flip. Using the same argument as for the bit-flip detection, if we measure **0** the state collapses to:

$$\begin{aligned} & \alpha(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle) \\ & + \beta(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle) \end{aligned}$$

Or if we measure a **1** we get:

$$\begin{aligned} & \alpha(|000\rangle - |111\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle) \\ & + \beta(|000\rangle + |111\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle) \end{aligned}$$

i.e., a phase-flip has occurred which we can then correct. Therefore we have recovered the original state.

Therefore performing bit- and phase-flip parity-check measurements collapses a general state into the case where either the bit / phase flip has occurred or not as per the measurement outcome. This remarkable property allows us to correct a continuum of errors by performing only bit- and phase-flip checks.

Quantum error correction

We have seen that the Shor code can correct an arbitrary error on any one qubit. In practise, however, **we are interested in the case where each qubit experiences some amount of noise** – we assume this is independent and identically distributed (iid).

- We saw that the Shor code suppresses errors in the (iid) depolarising channel to $\mathcal{O}(p_e^2)$.
- The general quantum case is slightly more complicated, but essentially the same principle applies: **in the *low noise* setting it often suffices to use codes which guarantee to correct any single error**, because multiple errors are much less likely.

This principle only works because of the way that quantum error correction works: the fact that the parity-checking measurements collapse the state such that either a bit-flip or a phase-flip or both or neither has occurred means that quantum error correction is analogous to digital rather than analogue classical error correction in many important ways.

More sophisticated classical error correction

Repetition codes are useful for demonstrating the principle of error correction, but are rather too inefficient to use in practise. One particularly elegant code is the $(7,4)$ Hamming code, a linear code that encodes a 4-bit *data-word*, d , as a 7-bit *code-word*, c , according to $c = Gd \pmod 2$, where G is the generator matrix:

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

Any errors are detected by applying the parity-check matrix, H , to a given code-word.

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

Letting $p = Hc \pmod 2$, the three bits of p are all zero if c is a valid code-word, and otherwise the seven possible three-bit strings with at least one 1 encode the position of a single error. Thus the $(7,4)$ Hamming code can detect and correct any single bit error.

Quantum codes from classical codes

- Classical *linear* codes are efficient, in the sense that code-words are generated by multiplying the data-word by a matrix, which can be compactly described.
- There is a technique for using classical linear codes to find quantum error correction codes. These codes are known as CSS (Calderbank-Shor-Steane) codes.
- To study CSS codes it is helpful to know the *Stabiliser formalism*, which is beyond the scope of this course.
- However, it is worth being aware of one particular CSS code, the *Steane code*, which is constructed from the (7, 4) Hamming code. The Steane code encodes one data-qubit in seven physical qubits, such that any single-qubit error can be detected and corrected. The Steane code encodes the logical states 0 and 1 as follows:

$$\begin{aligned} |0_L\rangle &= \frac{1}{\sqrt{8}} (|0000000\rangle + |1010101\rangle + |0110011\rangle + |1100110\rangle \\ &\quad |0001111\rangle + |1011010\rangle + |0111100\rangle + |1101001\rangle) \\ |1_L\rangle &= \frac{1}{\sqrt{8}} (|1111111\rangle + |0101010\rangle + |1001100\rangle + |0011001\rangle \\ &\quad |1110000\rangle + |0100101\rangle + |1000011\rangle + |0010110\rangle) \end{aligned}$$

What to remember

We have seen that there are three obstacles to applying the techniques and principles of classical error correction directly to quantum error correction, each of which can be worked around:

- The no-cloning principle means that we cannot simply copy quantum states in repetition codes – instead we can use entangling to “copy” the information.
- Measurements destroy quantum information: so instead we design the error correcting codes so that the measurements only tell us whether an error has occurred, and nothing about the quantum state itself.
- Quantum errors are continuous: but we have seen that the process of error correction effectively digitises the errors.

Additionally, we have seen that, in practise, classical error correction codes are typically more sophisticated and efficient than simple repetition codes, and that these can be used to design quantum error correction codes, of which the Steane code is an important example.