QCIPU Lectures on Quantum Error Correction

Hersh Singh
Fermilab

Introduction

Any physical system used to store or process information is subject to noise. In classical computing, bit-flip errors caused by thermal fluctuations or electromagnetic interference are suppressed by engineering — transistors are engineered to have very high noise thresholds, and redundancy is added at the hardware level almost invisibly. Quantum computers face a far more hostile environment: qubits are exquisitely sensitive to their surroundings, and even the act of measuring them to check for errors risks destroying the delicate superpositions that give quantum computation its power.

Quantum error correction (QEC) is the mathematical framework that makes fault-tolerant quantum computation possible despite these obstacles. Its development in the mid-1990s — beginning with Shor's 9-qubit code (1995) and rapidly followed by the stabiliser formalism of Gottesman and the threshold theorem — fundamentally changed the outlook for practical quantum computing. The central message is both surprising and profound: it *is* possible to protect quantum information against noise, even though qubits are continuous, measurement causes collapse, and quantum states cannot be cloned.

These notes develop QEC from first principles, starting with its classical analogue. Lecture 1 introduces the *repetition code* and distils the two ideas — **redundancy** and **thresholds** — that underpin all error-correcting codes. Lecture 2 confronts the three apparent obstructions to quantum error correction, resolves each in turn through the bit-flip and phase-flip codes, and culminates in the **Shor code**: the first example of a quantum error-correcting code, and one that already illustrates why arbitrary continuous errors on qubits can be corrected by a code that only needs to handle a discrete set of Pauli errors.

Classical Error Correction

Repetition Code

We encode a single logical bit (a 1-bit message) into n identical physical bits, called *codewords*:

$$0 \longrightarrow \underbrace{00\cdots0}_{n}, \qquad 1 \longrightarrow \underbrace{11\cdots1}_{n}.$$

- We can **detect** an error whenever the bits are not all identical.
- We can **correct** an error by **majority vote**: for $n = 2t + 1$, the code corrects up to t bit-flip errors.

Example: $n = 3$**.**

	Codeword	+ Error	Detect	Correct
1-bit error	000	100	✓	✓
2-bit error	000	110	✓	×
3-bit error	000	111	takes one codeword to another	

A 3-bit error maps one codeword to another and *cannot be detected*.

Code distance. The *code distance* d is the minimum number of bit flips needed to take one codeword to another.

Does the 3-Bit Code Actually Help?

Adding more bits also introduces more opportunities for error. We need to check that the code actually suppresses errors on balance.

Let p be the probability of a single-bit error. For the 3-bit code:

$$\begin{aligned} p_0 &= (1-p)^3, && \text{(no error)} \\ p_1 &= 3p(1-p)^2, && \text{(1-bit error)} \\ p_2 &= 3p^2(1-p), && \text{(2-bit error)} \\ p_3 &= p^3. && \text{(3-bit error)} \end{aligned}$$

The probability of an **uncorrectable** error is $P_e = p_2 + p_3$. For the scheme to be beneficial, we require $P_e < p$:

$$\begin{aligned} 3p^2(1-p) + p^3 &< p \\ 2p^2 - 3p + 1 &> 0 \\ (2p-1)(p-1) &> 0, \end{aligned}$$

which holds for $p < \frac{1}{2}$ (the case $p > 1$ is unphysical).

There is therefore a **threshold** $p_c = \frac{1}{2}$: if $p < \frac{1}{2}$ the code further suppresses errors; if $p > \frac{1}{2}$ the code makes things worse.

Two Key Ideas

The repetition code illustrates two principles generic to *all* error-correcting codes:

1. **Redundancy** — encode information into a larger space.
2. **Thresholds** — error correction only works if the physical error rate is below a threshold.

How do we apply these ideas to build quantum error-correcting codes?

Quantum Error Correction

Three Reasons to Be Sceptical

1. **Redundancy?** The *no-cloning theorem* prohibits copying an unknown quantum state:
$$|\psi\rangle |0\rangle \longrightarrow |\psi\rangle |\psi\rangle \quad \text{is } \textit{impossible}.$$
So how do we create redundancy for a quantum state?
2. **Measurement causes collapse.** The classical code requires reading all bits. But measuring a qubit collapses its state — we destroy the very thing we are trying to protect. How can we detect errors without measuring the qubits directly?
3. **Qubits are continuous.** A general qubit state
$$|\psi\rangle = \cos\tfrac{\theta}{2}\,|0\rangle + e^{i\phi}\sin\tfrac{\theta}{2}\,|1\rangle$$
carries continuous parameters (θ, ϕ). Classical error correction handles discrete bits — it seems impossible to correct a *continuous* variable perfectly.

We resolve (1) and (2) by constructing the quantum analogue of the 3-bit repetition code. Issue (3) is addressed at the end via the Shor code.

Bit-Flip Error Correction Code

Consider a qubit $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ subject to *bit-flip* errors:
$$|\psi\rangle \xrightarrow{\text{error}} X|\psi\rangle = \alpha|1\rangle + \beta|0\rangle .$$

Step 1: Encoding. Introduce two ancilla qubits in state $|00\rangle$ and apply CNOT gates with the data qubit as control:
$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \xrightarrow{\text{CNOT}^{\otimes 2}} |\psi\rangle_L = \alpha|000\rangle + \beta|111\rangle .$$
This is **not** cloning — we are *entangling* qubits in the computational basis, not copying the quantum state. The **code space** $\mathcal{C} = \text{span}\{|000\rangle, |111\rangle\}$; any 3-qubit state outside $\mathcal{C}$ signals an error.

Step 2: Measuring the Error Syndrome. A single bit-flip on qubit i produces one of three orthogonal error states:

$$\begin{aligned}|\psi_1\rangle &= X_1 |\psi\rangle_L = \alpha |100\rangle + \beta |011\rangle ,\\ |\psi_2\rangle &= X_2 |\psi\rangle_L = \alpha |010\rangle + \beta |101\rangle ,\\ |\psi_3\rangle &= X_3 |\psi\rangle_L = \alpha |001\rangle + \beta |110\rangle .\end{aligned}$$

We identify i by measuring the **stabilizer operators** $E_1 = Z_1Z_2$ and $E_2 = Z_1Z_3$. Since all four states are eigenstates of these operators, the measurement *does not disturb* the amplitudes (α, β):

	$\|\psi\rangle_L$	$\|\psi_1\rangle$	$\|\psi_2\rangle$	$\|\psi_3\rangle$
Z_1Z_2	$+1$	-1	-1	$+1$
Z_1Z_3	$+1$	-1	$+1$	-1

The syndrome (Z_1Z_2, Z_1Z_3) uniquely identifies which of the four subspaces the state occupies.

Step 3: Correction. Given syndrome outcome i, apply X_i to return to the code space: $X_i |\psi_i\rangle = |\psi\rangle_L$.

This code already resolves the first two objections. Redundancy is achieved through *entanglement* rather than cloning. And measurement does not collapse the logical state, because the stabilizer measurements reveal only which error occurred — nothing about the amplitudes (α, β) themselves.

Phase-Flip Error

A *phase-flip* error acts as:

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle \xrightarrow{Z} \alpha |0\rangle - \beta |1\rangle .$$

This has no classical analogue. More generally, a **coherent error** is an arbitrary unitary:

$$|\psi\rangle \longrightarrow e^{i(\theta_1 X + \theta_2 Y + \theta_3 Z)} |\psi\rangle ,$$

making qubits subject to genuinely *continuous* errors.

Correcting phase flips Change from the Z-basis $\{|0\rangle , |1\rangle\}$ to the X-basis

$$|\pm\rangle = \frac{1}{\sqrt{2}}\big(|0\rangle \pm |1\rangle\big).$$

In this basis a phase flip becomes a *bit flip*. The basis change is effected by the **Hadamard gate** H:

$$|+\rangle = \tfrac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \xrightarrow{Z} |-\rangle = \tfrac{1}{\sqrt{2}}(|0\rangle - |1\rangle).$$

We can therefore reuse the bit-flip code after conjugating by Hadamards. The syndrome operators transform as:

$$H(Z_1Z_2)H = X_1X_2, \qquad H(Z_1Z_3)H = X_1X_3.$$

These detect phase-flip errors on $|\pm\rangle$ states.

Combining the bit-flip and phase-flip codes gives a $3 \times 3 = 9$-qubit code that corrects both error types simultaneously.

The 9-Qubit Shor Code

The combined 9-qubit code — the **Shor code**, the first quantum error-correcting code — does something remarkable: it corrects *arbitrary* single-qubit errors, not merely the discrete bit-flip and phase-flip errors it was designed for.

Why arbitrary errors can be corrected — resolution of Issue (3) Consider a coherent error: we intend to apply $U(\vec{\alpha}) = e^{i(\alpha_1 X + \alpha_2 Y + \alpha_3 Z)}$ but instead apply $U(\vec{\alpha} + \vec{\varepsilon})$ for small $|\vec{\varepsilon}|$. The error operator is

$$\delta U = U(\vec{\alpha} + \vec{\varepsilon})\, U^\dagger(\vec{\alpha}) \approx e^{i(\varepsilon_1 X + \varepsilon_2 Y + \varepsilon_3 Z)}.$$

Expanding to first order in ε and using $Y = -iZX$:

$$\delta U \approx \underbrace{\mathbf{1}}_{\text{no error}} + \underbrace{i\varepsilon_1 X}_{\text{bit flip}} + \underbrace{i\varepsilon_3 Z}_{\text{phase flip}} + \underbrace{\varepsilon_2\,(ZX)}_{\substack{\text{bit flip +} \\ \text{phase flip}}} + \mathcal{O}(\varepsilon^2).$$

Every coherent error thus decomposes into a linear combination of $\{\mathbf{1}, X, Z, ZX\}$ — precisely the errors the Shor code corrects. Measuring the syndrome *projects* the continuous error onto one of these discrete outcomes, effectively digitising it.

> *Qubits appear "analogue", but arbitrary errors on qubits have an underlying* ***digital structure****: they decompose into bit-flip and phase-flip components under measurement. A code correcting those discrete errors therefore corrects* any *coherent single-qubit error.*