

Report of Introduction to Quantum Computing Quantum Error Correction

Student: Diego Arcelli

Student ID: 647979

Professors: Gianna Maria Del Corso,
Anna Bernasconi, Roberto Bruni

Academic Year 2021-2022



UNIVERSITÀ DI PISA

Contents

1	Quantum Error Correction	2
2	General Framework	2
3	Correcting single qubit errors	2
3.1	Correcting bit flip errors	2
3.2	Correcting phase flip errors	3
3.3	Nine-qubits correcting code	4
3.4	Correcting generic errors	5
4	Stabilizer codes	5
4.1	Stabilizer formalism	5
4.2	Thee and nine qubits encoding revisited	6
4.3	Errors with stabilizer codes	6
4.4	Five qubit correction code	7

1 Quantum Error Correction

The hardware devices used to implement quantum circuits are sensitive to noise (much more than classical computers). This might cause the information carried by the circuit to be modified in an unpredictable way. For this reason being able to detect and correct these errors it's a topic of major importance.

Sadly handling errors in the quantum case it's more troublesome than classical computing: first, we cannot copy arbitrary qubits (because of the no cloning theorem), then we cannot measure a qubit to check its state because this would destroy the superposition and, moreover, errors are continuous (while in classical computing we only have bits flip errors).

2 General Framework

The general framework applied for quantum error correction is similar to the one used in classical error correction. The first step is defining an error model, which is a description of the type of error we want to defend against. Once we have the error model, in order to defend a quantum state $|\psi\rangle$ from the error that might occur, we'll define an encoding operator U_{enc} which, by adding some extra ancillary qubits to the state, will produce an encoded state $|\psi_{enc}\rangle$:

$$U_{enc} |\psi\rangle |00\dots 0\rangle = |\psi_{enc}\rangle$$

The space of all the possible encoded space will form a sub-space of the Hilbert space that we'll call the Quantum Error Correcting Code (QECC). If $|\psi\rangle$ is composed by k qubit, and with the encoding we add n ancillary qubits, then the code \mathcal{C} will be a sub-space of the 2^{k+n} Hilbert space. The idea of adding ancillary qubits, is to use them to store the information about the error that occurred, which we call the **error syndrome**. Then we define a recovery operator \mathcal{R} that, if an error occurs on the encoded state $|\psi_{enc}\rangle$, will undo the effect of the error, based on the information gained with the error syndrome. Finally we'll decode $|\psi_{enc}\rangle$ back to $|\psi\rangle$ that, with high probability, will not contain effects of the error.

In general given a quantum code $\mathcal{C} \subset \mathcal{H}$, we say that a noise operator \mathcal{E} is recoverable if there exists a recovery operator \mathcal{R} such that for every density matrix ρ associated with a state $|\psi\rangle \in \mathcal{C}$:

$$\mathcal{R}(\mathcal{E}(\rho)) = \rho$$

In the following sections we'll analyze some of the most relevant type of errors and we'll show circuits which can detect and correct the errors, using the framework we just explained.

3 Correcting single qubit errors

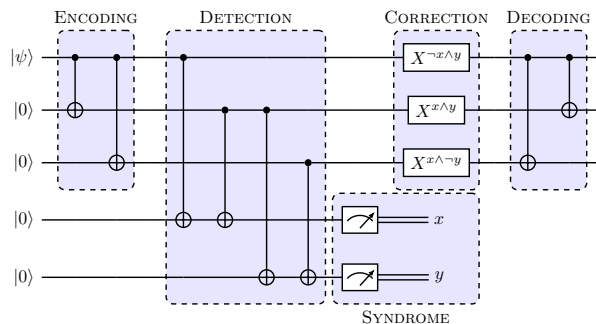
3.1 Correcting bit flip errors

Let's start with the simplest error that we might have, the bit flip error, where a generic qubit in state $\alpha|0\rangle + \beta|1\rangle$ it's changed into $\alpha|1\rangle + \beta|0\rangle$ with probability p . This can be seen as an application of the X Pauli's gate to the qubit. To correct this kind of errors we use the following three bit encoding:

$$\alpha|0\rangle + \beta|1\rangle \rightarrow \alpha|000\rangle + \beta|111\rangle$$

So we encoded physical qubits to a sequence of logical qubits. Note that this is not a violation of the no cloning theorem since we're not mapping $\alpha|0\rangle + \beta|1\rangle$ to $(\alpha|0\rangle + \beta|1\rangle)^{\otimes 3}$.

The idea we want to apply is to check if the value of the three qubit is the same, if not, we impose the value of the majority bit. As we said earlier we cannot do that by measuring the state, otherwise the superposition will be destroyed. What we can do instead is using some extra qubits, to store in them the information of which qubit of the state has been flipped. All this can be done with the following circuit:



The first two CNOT gates are used to apply the three bits encoding. Then with the second two CNOT gates we check if the first two qubits are equal. If they are equal and they're both $|0\rangle$ then the first ancilla qubit is not flipped. If they are equal and they're both $|1\rangle$ the first ancilla qubit it's flipped twice (so it goes back to $|0\rangle$). If they're different the ancilla qubit is flipped and so it is set to $|1\rangle$. With the last two CNOT gates we repeat the same procedure, but this time we compare the second and the third qubits and we act on the second ancillary qubit. Finally we measure the ancillary qubits and we get the values x and y . The value of x tells us if the first and the second qubit were the same, while the value of y tells us if the second and the third qubit were the same. Note that if two or all the three qubits are flipped, this circuit will produce a wrong state. In general this strategy can be extended by using an encoding of length n .

3.2 Correcting phase flip errors

A more tricky type of error which we might have in quantum computing is the phase flip error where a generic qubit $\alpha|0\rangle + \beta|1\rangle$ is changed into $\alpha|0\rangle - \beta|1\rangle$. This can be seen as an application of the Z Pauli gate. It turns out that solving phase flip errors is very similar to solve bit flip errors, since phase flip errors in the canonical basis become bit flip errors in the Hadamard basis since $X = HZH$ and because of that, $Z|+\rangle = |-\rangle$ and $Z|-\rangle = |+\rangle$. We can exploit this fact by using the following encoding:

$$\alpha|0\rangle + \beta|1\rangle \rightarrow \alpha|+++\rangle + \beta|---\rangle$$

which can be obtained from the bit flip encoding, by applying to each one of the three qubit the H gate. Suppose that with this encoding a phase flip error occurs in the second qubit. Then, since in the Hadamard basis, a phase flip acts like a bit flip, the encoded state becomes:

$$\alpha|+-+\rangle + \beta|-+-\rangle$$

if we now apply an H gate to each qubit of the state, it becomes:

$$\alpha|010\rangle + \beta|101\rangle$$

which is like if a bit flip error occurred in the second bit of the encoding for the bit flip error. Therefore we can simply run the circuit, for the bit flip correction, and we'll correctly recover the initial phase flip error.

3.3 Nine-qubits correcting code

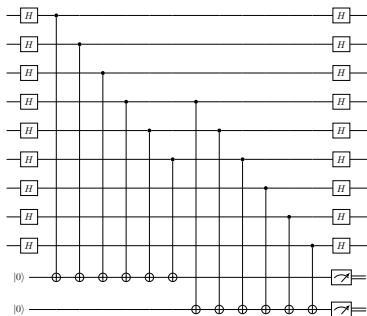
We have seen a circuit which can correct bit flip errors and a circuit which can correct phase flip errors, but what if a qubit is affected by both the errors? We can also easily solve those errors, using the 9-qubit encoding defined by Shor [1], which can be derived from the previous encoding as follows: first we start from the encoding for the phase flip errors:

$$\alpha|0\rangle + \beta|1\rangle \rightarrow \alpha|+++\rangle + \beta|---\rangle$$

then we apply to each qubit of the encoded sequence the circuit of the bit flip encoding and we get:

$$\alpha|0\rangle + \beta|1\rangle \rightarrow \frac{\alpha}{\sqrt{8}}(|000\rangle + |111\rangle)^{\otimes 3} + \frac{\beta}{\sqrt{8}}(|000\rangle - |111\rangle)^{\otimes 3}$$

So we have three blocks of three qubits each. Within each block, all the qubits are supposed to be the same, so we can run the bit flip error correction circuit in every one of the blocks to protect each block by bit flip errors. Now, if we consider the three blocks, the sign of every block should be the same, but if there's been a phase flip error in one of the blocks, the sign on that block will be flipped. The idea is the same but this time we cannot apply the circuit we saw before, but what we can do is use this circuit:



it rather messy, but what the circuit does is comparing the sign of the first block with the sign of the second block and if they're different set the first ancillary qubit to 1, and then it does the same thing with the second and third block, so that we'll measure a syndrome for each possible error.

To show why the circuit works, after the application of $H^{\otimes 9}$ the blocks will be transformed like that:

$$\begin{aligned}(|000\rangle + |111\rangle)^{\otimes 3} &\rightarrow 2(|000\rangle + |011\rangle + |101\rangle + |110\rangle)^{\otimes 3} = |\psi^+\rangle^{\oplus 3} \\(|000\rangle - |111\rangle)^{\otimes 3} &\rightarrow 2(|001\rangle + |010\rangle + |100\rangle + |111\rangle)^{\otimes 3} = |\psi^-\rangle^{\oplus 3} \\ \alpha|0\rangle + \beta|1\rangle &\rightarrow \alpha|\psi^+\rangle|\psi^+\rangle|\psi^+\rangle + \beta|\psi^-\rangle|\psi^-\rangle|\psi^-\rangle\end{aligned}$$

showing all the steps would be too messy and not even interesting, but the idea is that now each qubit of each block is composed a series of binary string where for the encoding of $|0\rangle$ each string is composed by an even number of 1s, while for the encoding of $|1\rangle$ by an odd number of 1s. The circuit exploit this property, since if there's been no phase error, the first three CNOT gates will flip the first ancillary qubit an even number of times for the encoding of $|0\rangle$, and an odd number of times for the encoding of $|1\rangle$. The the second three CNOTs will flip the first qubits again an even number of times for the encoding of $|0\rangle$, and an odd number of times for the encoding of $|1\rangle$. Since an even number plus an even number is an even number, and an odd number plus an odd number is an odd number, for both the encoding of $|0\rangle$ and $|1\rangle$ it gets flipped an even number of times and so it remains $|0\rangle$.

Suppose that there's been a phase flip error on the first block. Then state after the application of the Hadamard gates will be:

$$\alpha|\psi^-\rangle|\psi^+\rangle|\psi^+\rangle + \beta|\psi^+\rangle|\psi^-\rangle|\psi^-\rangle$$

so this time when we apply the first three CNOTs for the encoding of $|0\rangle$ the first ancillary qubit will be flipped an odd number of times, and an even number of times for the encoding of $|1\rangle$. Then with the second three CNOTs the ancillary qubits will be flipped an even number of times for the encoding of $|0\rangle$ and an odd number of times for the encoding of $|1\rangle$. Since an even number plus an odd number is an odd number, the first ancillary is flipped in total an odd number of times for both the encoding of $|0\rangle$ and $|1\rangle$. The same reasoning can be applied with the second and third triples of CNOTs and with the third and the fourth.

3.4 Correcting generic errors

Since now we only saw bit flip and phase flip errors, but how do we deal with other type of errors, like a generic rotation of the qubit in the Bloch sphere. Consider a generic error on a single qubit defined by a matrix $E \in \mathbb{C}^{2 \times 2}$, and consider the four Pauli matrices I , X , Y and Z . As we already saw X and Z represent respectively bit flip and phase flip errors. Since $Y = iXZ$, it can be seen as bit flip error followed by a phase flip error (the global phase factor i does not matter), while the gate I can be simply seen as the no-error gate. These four matrices span the space of the 2×2 complex matrices, so the before error matrix E can be written as:

$$E = \alpha_0 I + \alpha_1 X + \alpha_2 Y + \alpha_3 Z, \quad \alpha_0, \alpha_1, \alpha_2, \alpha_3 \in \mathbb{C}$$

Basically we've rewritten E as a linear combination of bit flip and phase flip errors, which we already know how to solve. This is important because there's a theorem which state that if a code \mathcal{C} corrects a set of errors $\mathcal{E} = \{E_i\}$, then \mathcal{C} corrects also the span of \mathcal{E} . So if we're able to correct bit flip and phase flip errors, then we're able to correct any kind of error.

4 Stabilizer codes

It turns out that the codes that we saw so far and many others can be described using a particular formalism, introduced by Gottesman in [2], which makes it easier to define quantum error correcting codes. In the following section we'll introduce the stabilizer formalism and we'll see how we can apply it to quantum error correction.

4.1 Stabilizer formalism

We say that a unitary operator U stabilizes the state $|\psi\rangle$ if $U|\psi\rangle = |\psi\rangle$, which means that $|\psi\rangle$ is an eigenstate of U with eigenvalue 1. If V and U stabilize $|\psi\rangle$ the also their product and their inverses stabilize $|\psi\rangle$. Since matrix multiplication is an associative operation, this means that the set of unitary matrices that stabilizes $|\psi\rangle$ form a group. Consider the set of Pauli operators $\Pi = \{I, X, Y, Z\}$, we then define Π^n as:

$$\Pi^n = \{A_1 \otimes \cdots \otimes A_n : A_j \in \Pi \forall j\}$$

which is the set $2^n \times 2^n$ matrices which can be obtained by the tensor product of n Pauli matrices. A stabilizer \mathcal{S} is an abelian subgroup of Π^n .

Given a stabilizer \mathcal{S} we can define the code-space generated by \mathcal{S} as $\mathcal{T}(\mathcal{S}) = \{|\psi\rangle \in \mathcal{H} : M|\psi\rangle = |\psi\rangle \ \forall M \in \mathcal{S}\}$ which is the set of state stabilized by all the elements of \mathcal{S} . We can interpret $\mathcal{T}(\mathcal{S})$ as a QECC where the states in $\mathcal{T}(\mathcal{S})$ are the encoded states.

4.2 Thee and nine qubits encoding revisited

Let's now revisit the three qubit code for bit flip correction. What we were trying to do was check if the first and the second qubits where the same, and the same thing for the second and the third qubits. We can equivalently say that we where verifying if the encodes state has eigenvalue 1 with respect to the operators $Z_1 = Z \otimes Z \otimes I$ and $Z_2 = I \otimes Z \otimes Z$, since:

$$(Z \otimes Z \otimes I)(\alpha|000\rangle + \beta|111\rangle) = \alpha|000\rangle + \beta|111\rangle$$

if instead there's a bit flip error in the first qubit we get that:

$$(Z \otimes Z \otimes I)(\alpha|010\rangle + \beta|101\rangle) = -(\alpha|010\rangle + \beta|101\rangle)$$

so the eigenvalue of the state wit respect to that operator becomes -1 . So what we were doing was checking if the encoded state has eigenvalue 1 with respect both of Z_1 and Z_2 , while if for Z_1 has eigenvalue -1 it means that the first and the second qubits are different, while if it has eigenvalue 1 with Z_2 , it means that the second and the second qubits are different. It's easy to check that we could have done the same bit flip correcting procedure, by comparing first the first and the second qubits, and then the first and the third, the syndrome would have been different but it could be used to correct the error. In therm of stabilizers, this corresponds to using the matrices Z_1 (the same we used before) and $Z_3 = Z \otimes I \otimes Z$, since also for Z_3 we have $Z_3|\psi_{enc}\rangle = |\psi_{enc}\rangle$. In fact the set of stabilizers of $|\psi_{enc}\rangle$ is $\mathcal{S} = \{Z_1, Z_2, Z_3\}$, but we can only use Z_1 and Z_2 because they are generators of Z_3 , since $Z_1 Z_2 = Z_2 Z_1 = Z_3$.

Another way to put this is by saying that the correcting code must have eigenvalue 1 with respect of the operators:

$$\begin{array}{c|ccc} g_1 & Z & Z & I \\ g_2 & I & Z & Z \end{array}$$

The same reasoning applies to the phase flip correction but with the matrices $X_1 = X \otimes X \otimes I$ and $X_2 = I \otimes X \otimes X$, and we won't show it since it's straightforward to verify that is true. Instead it is more interesting to see. In instead we think about the nine qubit encoding the corresponding stabilizer representation is the following:

$$\begin{array}{c|ccccccccc} g_1 & Z & Z & I & I & I & I & I & I & I \\ g_2 & I & Z & Z & I & I & I & I & I & I \\ g_3 & I & I & I & Z & Z & I & I & I & I \\ g_4 & I & I & I & I & Z & Z & I & I & I \\ g_5 & I & I & I & I & I & I & Z & Z & I \\ g_6 & I & I & I & I & I & I & I & Z & Z \\ g_7 & X & X & X & X & X & X & I & I & I \\ g_8 & I & I & I & X & X & X & X & X & X \end{array}$$

the first size basically correspond to the bit flip stabilizer applied to each group of three qubits.

4.3 Errors with stabilizer codes

Consider an error matrix $E \in \Pi^n$. E can either commute or anti-commute with the operators in \mathcal{S} . If $\forall M \in \mathcal{S} ME = -EM$ then $ME|\psi\rangle = -EM|\psi\rangle = -E|\psi\rangle$, thus the error defined by E can be detected. If instead $\forall M \in \mathcal{S} ME = EM$ then $ME|\psi\rangle = EM|\psi\rangle = E|\psi\rangle$, which means that this error cannot be detected. So we define $\mathcal{N}(\mathcal{S}) = \{E \in \Pi^n : EM = ME, \forall M \in \mathcal{S}\}$ which can be seen as the set of errors which cannot be detected by the code $\mathcal{T}(\mathcal{S})$. We can also define the distance of a code, which is the smallest weight of an operator in the set $\mathcal{N}(\mathcal{S}) \setminus \mathcal{S}$, where the weight is the number of elements in the tensor product which compose the operator that are not the identity.

4.4 Five qubit correction code

The fine qubit correcting code is the smallest error correcting code which is able to detect both bit flip and phase flip error on one qubit. For this code we'll do the opposite procedure we did for the previous codes. We

first define the stabilizer, and then we try to infer the circuit from the stabilizer.

$$\begin{array}{l|cccccc} M_1 & X & Z & Z & X & I \\ M_2 & I & X & Z & Z & X \\ M_3 & X & I & X & Z & Z \\ M_4 & Z & X & I & X & Z \end{array}$$

References

- [1] Peter W. Shor. Scheme for reducing decoherence in quantum computer memory. *Phys. Rev. A*, 52:R2493–R2496, Oct 1995.
- [2] Daniel Gottesman. Stabilizer codes and quantum error correction. *arXiv: Quantum Physics*, 1997.