Iterative Decoder for Quantum Turbo Codes: Performance Analysis and Enhancement

PROYECTO

presentado para optar
al Título de Grado en Ingeniería en Sistemas de Telecomunicación por

Jon Ander Iñiguez de Gordoa Pouso

bajo la supervisión de

Pedro Crespo Bofill

Donostia-San Sebastián, junio 2019

tecnun
Universidad de Navarra

tecnun Universidad de Navarra

Proyecto Fin de Grado

INGENIERIA EN SISTEMAS DE TELECOMUNICACIÓN

Iterative Decoder for Quantum Turbo Codes: Performance Analysis and Enhancement

Jon Ander Iñiguez de Gordoa Pouso

San Sebastián, junio de 2019

Pº Manuel Lardizabal, 13. 20018 Donostia-San Sebastián, Gipuzkoa
Tel. 943 219 877 · Fax 943 311 442 · www.tecnun.es

Contents

List of Figures

List of Tables

Abstract

Quantum error correction is necessary in quantum communication. In that sense, quantum turbo codes present a remarkably low probability of error compared to other quantum error correcting codes. The document of this Final Degree Project analyses the performance of iterative SISO decoders in quantum turbo codes, especially when the decoder is under the influence of a channel mismatch. The obtained results suggest that the closer the estimated depolarizing probability is to the actual depolarizing probability of the channel, the lower the WER of the SISO decoder will be.

In this document, chapter 1 contemplates the relevance of quantum error correcting codes in current and future quantum technologies. Chapter 2 provides the basic background on linear algebra and quantum mechanics that are crucial in order to understand quantum error correction. Chapter 3 presents a few notions in quantum error correction and the stabilizer codes, which are the cornerstone in order to export classical error correcting codes into the quantum world. Chapter 4 presents the actual quantum turbo codes, their construction, their decoding algorithm and their performance when the decoder suffers from channel mismatch. Chapter 5 summarizes the main conclusions of this Final Degree Project, and chapter 6 provides the budget of the whole project.

1 State Of The Art

Quantum supremacy is a term popularized by John Preskill which refers to the potential ability of quantum computers to solve problems with a superpolynomial speedup in comparison to classical computers [1]. Even though quantum computing is still in its rudimentary stage, the tech giants are already on a race to dominate quantum computing. In January 2019, IBM revealed the world's first integrated quantum computing system for commercial use called IBM Q System One [2]; however, many experts received the announcement with skepticism. In February 2019, Intel claimed to have created the world's first quantum computing testing tool [3]. In March 2019, Microsoft launched *Microsoft Quantum Network*, a global community of individuals and organizations working with Microsoft to research and launch quantum computing applications [4]. And Google has been working with NASA on the *Quantum Artificial Intelligence Lab* since 2013, and they have been designing quantum processors since then [5].

Similarly, quantum communication has also been on the rise in the past few years. Persuaded by the unconditional security that it offers, many countries have decided to invest in quantum communication, mostly oriented to potential military applications. The main example of this is China, who, among other things, achieved to establish a quantum-encrypted videoconference between Vienna and Beijing in 2017 with the help of a quantum communication satellite [6].

In this context of emerging quantum technologies, *quantum error correction* plays a key role in the development of such technologies. Quantum error correction is a branch of a larger theory named *Quantum Information Theory*. Quantum Information Theory is the field of study of the quantum information stored in the state of quantum systems and it analyses how quantum information behaves in the quantum world. It is the quantum counterpart of Classical Information Theory, which was introduced by Claude Shannon in his article *A Mathematical Theory of Communication* [7] in 1948.

Quantum error correction is achievable thanks to quantum error correcting codes. The primary purpose of quantum error correcting codes is to protect quantum information from quantum noise. These codes are needed in two different situations: quantum computing and quantum communication.

In quantum computing, the *qubits* that contain the quantum information can be affected by the noise introduced by imperfect logical gates, the interaction with the external environment or even when they are just being stored. Quantum error correcting codes must assure a *fault-tolerant* quantum computing, which means that the whole system works perfectly even when some of its components are imperfect. In order to achieve fault tolerance, qubits in a quantum computer must be kept error-free long enough until the overall computation is finished. In computation, reliability is key, and that is why codes with a high distance (such as concatenated codes) are used.

In quantum communication, where Alice (the sender) and Bob (the receiver) are physically separated, Alice uses a channel to send qubits to Bob. This channel is assumed to be a noisy channel, and it might corrupt the quantum information that Alice sent to Bob. In this scenario, when Bob decodes the received qubits, quantum error correcting codes must be able to recover the original quantum information sent by Alice, even when such information was corrupted by the noisy channel. When quantum communication requires sending a large amount of qubits,

codes with a high transmission rate are preferred: codes that often encode qubits into larger blocks, with the aim of getting close to the actual capacity of the channel [8].

Among the existing quantum error correcting codes, Quantum Turbo Codes (QTCs) present an especially interesting performance and low probability of error. These codes are based on the classical turbo codes and they can be imported to quantum error correction with the help of *stabilizer codes* and *entanglement assistance*.

2 Background for Quantum Error Correction

This chapter summarizes the basic concepts of Linear Algebra and Quantum Mechanics that are essential in order to understand quantum communication and the following error correction that will be expained in the next chapters.

2.1 Linear Algebra

A few concepts of Linear Algebra are needed in order to understand the mathematical formulation of Quantum Mechanics. Section 2.1 introduces the concepts of *Bra-Ket notation, Pauli matrices, tensor product* and the *commutator* and *anti-commutator* operators.

2.1.1 Dirac's Bra-Ket notation

The Bra-Ket notation, also known as Dirac's notation, was introduced in 1939 by Paul Dirac. It is the standard notation for Linear Algebra that is used to describe quantum states. In this notation, a *ket* is a column vector and a *bra* is the Hermitian conjugate of a ket. The inner product between a bra and a ket is also called a *bracket*. The Bra-Ket notation is shown in Table 2.1.

Notation	Description
$\|\psi\rangle$	Vector. Also known as *Ket*.
$\langle\psi\|$	The Hermitian conjugate of the ket with the same label, $\langle\psi\|$ =$\|\psi\rangle^\dagger$. Also known as *Bra*.
$\langle\varphi\|\psi\rangle$	Inner product between $\|\varphi\rangle$ and $\|\psi\rangle$.
A^T	Transpose of matrix *A*.
A^*	Complex conjugate of *A*.
$A^\dagger$	Hermitian conjugate of *A*, $A^\dagger = (A^T)^*$.

Table 2.1: Dirac's Bra-Ket notation

2.1.2 Pauli Matices

Pauli matrices are a set of four 2 x 2 complex Hermitian and unitary matrices. These matrices play an important role in Quantum Computation and Quantum Information. The Pauli matrices are defined as:

$$\sigma_0 \equiv I \equiv \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}; \qquad \sigma_1 \equiv \sigma_x \equiv X \equiv \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix};$$

$$\sigma_2 \equiv \sigma_y \equiv Y \equiv \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}; \qquad \sigma_3 \equiv \sigma_z \equiv Z \equiv \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

It can be easily proved that the Pauli matrices meet the following properties:
- $X^2 = Y^2 = Z^2 = -iXYZ = I$
- $XY = iZ, \qquad YZ = iX, \qquad ZX = iY$

• $YX = -iZ, \qquad ZY = -iX, \qquad XZ = -iY$

2.1.3 Tensor Product

The *tensor product* is a mathematical operation that puts vector spaces together in order to form another vector space. Let V and W be two vector spaces of dimension m and n, respectively. The tensor product between V and W, denoted as $V \otimes W$, is another vector space of dimension mn where the elements of this new space are linear combinations of tensor products $|\varphi\rangle \otimes |\psi\rangle$, with $|\varphi\rangle \in V$ and $|\psi\rangle \in W$. If the vectors $|i\rangle$ and $|j\rangle$ are orthonormal bases for V and W respectively, then $|i\rangle \otimes |j\rangle$ is a basis for $V \otimes W$ [9].

When it comes to matrix vector spaces, the tensor product is performed by the *Kronecker product*. Let $A \in \mathbb{C}^{m\times n}$ and $B \in \mathbb{C}^{p\times q}$ be two arbitrary complex matrices. The Kronecker product between A and B, $A \otimes B \in \mathbb{C}^{mp\times nq}$, is defined in equation (2.1).

$$A \otimes B \equiv \begin{pmatrix} a_{11}B & \dots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \dots & a_{mn}B \end{pmatrix} \equiv \begin{pmatrix} a_{11}b_{11} & \dots & a_{11}b_{1q} & \dots & a_{1n}b_{11} & \dots & a_{1n}b_{1q} \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{11}b_{p1} & \dots & a_{11}b_{pq} & \dots & a_{1n}b_{p1} & \dots & a_{1n}b_{pq} \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{m1}b_{11} & \dots & a_{m1}b_{1q} & \dots & a_{mn}b_{11} & \dots & a_{mn}b_{1q} \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{mn}b_{p1} & \dots & a_{mn}b_{pq} & \dots & a_{mn}b_{p1} & \dots & a_{mn}b_{pq} \end{pmatrix} \tag{2.1}$$

In the following document, the Kronecker product will be the considered tensor product.

In the Bra-Ket notation, the tensor product between two arbitrary vectors, $|\psi\rangle \otimes |\varphi\rangle$, is sometimes denoted as $|\psi\rangle\,|\varphi\rangle$, or $|\psi\varphi\rangle$, just for the sake of simplicity. Besides, the notation $A^{\otimes k}$ is often used, which means that $A^{\otimes k} = A \otimes A \otimes A \otimes ... \otimes A$, i.e., the matrix A tensored by itself k times.

2.1.4 Commutator and anti-commutator

The *commutator* between two operators A and B is defined in equation (2.2).

$$[A, B] \equiv AB - BA \tag{2.2}$$

Similarly, the *anti-commutator* is defined in equation (2.3).

$$\{A, B\} \equiv AB + BA \tag{2.3}$$

If $[A, B] = 0$, that is, if $AB = BA$, we say that A *commutes* with B. While if $\{A, B\} = 0$, or $AB = -BA$, we say that A *anti-commutes* with B.

The commutation and anti-commutation relations between Pauli matrices can be easily proved. The commutator relations between Pauli matrices are shown in equation (2.4).

$$[\sigma_j, \sigma_k] = 2i \sum_{l=1}^{3} \epsilon_{jkl}\sigma_l, \tag{2.4}$$

where σ_l represents the Pauli matrices as shown in figure **??**, and $\epsilon_{jkl} = 0$, except for $\epsilon_{123} = \epsilon_{231} = \epsilon_{312} = 1$ and $\epsilon_{321} = \epsilon_{132} = \epsilon_{213} = -1$. As expected, every Pauli matrix commutes with itself.

Likewise, the Pauli matrices obey the anti-commutator relations shown in equation (2.5).

$$\{\sigma_j, \sigma_k\} = 2\delta_{jk} I, \tag{2.5}$$

where δ_{jk} is the Kronecker delta, and I is the 2×2 identity matrix. Note that Pauli matrices always anti-commute between distinct elements.

2.2 Quantum Mechanics

Quantum mechanics describes the behavior of subatomic particles and provides a mathematical and conceptual framework for the development of physical theories.

This section presents the postulates of quantum mechanics, as well as some singular concepts such as the *no-cloning theorem*, *entanglement* and *quantum teleportation*, due to their interesting role in quantum communication and quantum error correction.

2.2.1 The Postulates of Quantum Mechanics

The interconnection between the physical world and the mathematical formalism of quantum mechanics is provided by *the postulates of quantum mechanics* [9]. These postulates were introduced in the early 1930s by physicists John von Neumann and Paul Dirac.

Postulate 1: State Space. Any isolated physical system is associated with a complex Hilbert space known as the *state space* of the system. The system is completely described by its *state vector*, which is a unit vector in the state space of the system.

The simplest quantum mechanical system, and the one we will be working with, is the *qubit*. A qubit has a two dimentional state space. In a qubit, an arbitrary state vector can be written as in equation (2.6)

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \tag{2.6}$$

where $\alpha, \beta \in \mathbb{C}$ are *probability amplitudes*, and $|0\rangle$ and $|1\rangle$ are the standard orthonormal basis for that Hilbert space. This orthonormal basis can be written as in equation (2.7).

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}; \qquad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \tag{2.7}$$

The qubit state described in equation (2.6) is a *superposition* of the basis states, which means that a single qubit can be described as a linear combination of $|0\rangle$ and $|1\rangle$.

Since postulate 1 states that the state vector $|\psi\rangle$ is a unit vector, $\langle\psi|\psi\rangle = 1$ must hold, which is equivalent to say that $|\alpha|^2 + |\beta|^2 = 1$.

Postulate 2: Evolution. The evolution of a closed quantum system is described by a *unitary transformation*. The relationship between the state $|\psi_1\rangle$ of a system at time t_1 and the state $|\psi_2\rangle$ of the same system at time t_2 is described by a unitary operator U [9]. This unitary operator depends on t_1 and t_2 only, and it fulfills equation (2.8).

$$|\psi_1\rangle = U\,|\psi_2\rangle \tag{2.8}$$

When it comes to single qubits, any unitary operator can be performed in real systems. Some interesting unitary qubit operators are the *quantum gates*, which are crucial in quantum computation. Among these quantum gates, the *Pauli gates* (whose unitary operators are described by the Pauli matrices), the *Hadamard gate* and the *phase shift gate* stand out. Their schematic representation is shown in figure 2.1 and figure 2.2.

The X Pauli gate, also known as the *bit flip* gate, flips the probability amplitudes of the standard basis of an arbitrary qubit $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$.

$$X|\psi\rangle = X(\alpha|0\rangle + \beta|1\rangle) = \alpha|1\rangle + \beta|0\rangle$$

The Z Pauli gate, also known as the *phase flip* gate, changes $|1\rangle$ to $-|1\rangle$ of any qubit.

$$Z|\psi\rangle = Z(\alpha|0\rangle + \beta|1\rangle) = \alpha|0\rangle - \beta|1\rangle$$

The Y Pauli gate executes both a bit flip and a phase flip into the qubit, and multiplies the standard bases by $\pm i$.

$$Y|\psi\rangle = Y(\alpha|0\rangle + \beta|1\rangle) = i\alpha|1\rangle - i\beta|0\rangle$$

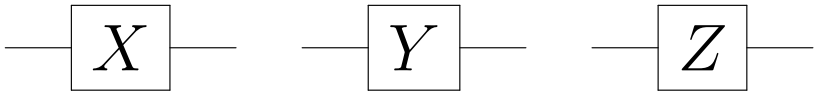


Figure 2.1: Schematic representation of Pauli gates

The Hadamard gate is given by equation (2.9).

$$H \equiv \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}. \tag{2.9}$$

The phase shift gate is defined in equation (2.10).

$$R_\Phi \equiv \begin{pmatrix}1 & 0\\ 0 & e^{i\Phi}\end{pmatrix}. \tag{2.10}$$

Note that the R_π gate is equal to the Z Pauli gate. Another special case of the phase shift gate is $P \equiv R_{\frac{\pi}{2}}$, which is used in quantum error correction.

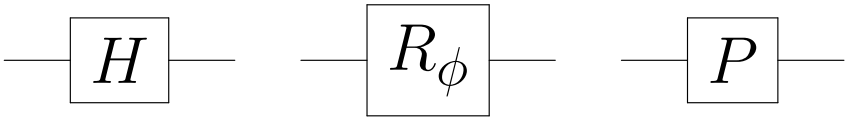


Figure 2.2: Schematic representation of Hadamard, phase shift and P gates

Besides the previously mentioned one-qubit quantum gates, there are also some *two-qubit quantum gates*, which are very important in the construction of error correction code unitaries. The most relevant ones are the *controlled-NOT* or *CNOT gate*, the *controlled-Unitary* or *controlled-U* gate and the *SWAP* gate. Their schematic representation is shown in figure 2.3.

The CNOT gate performs a NOT operation (or flip) on the second qubit (or *target qubit*) if the first qubit (or *control qubit*) is $|1\rangle$. The unitary operator of the CNOT gate is presented in equation (2.11).

$$CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{2.11}$$

The controlled-U gate ($C(U)$) is a more general two-qubit operation. This gate performs an arbitrary U operator to the target qubit if the control qubit is $|1\rangle$. The unitary operator of a controlled-U gate can be represented as in equation (2.12).

$$C(U) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & u_{11} & u_{12} \\ 0 & 0 & u_{21} & u_{22} \end{pmatrix} \tag{2.12}$$

Finally, the SWAP gate just swaps the two qubits of order in a circuit, and its unitary operator is represented in equation (2.13).

$$SWAP = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{2.13}$$

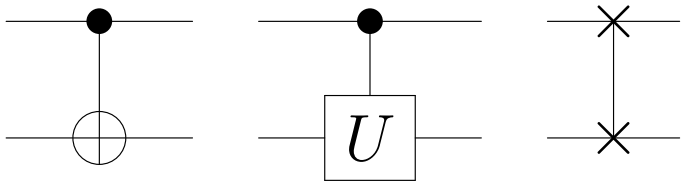


Figure 2.3: Schematic representation of CNOT, controlled-U and SWAP gates

Postulate 3: Quantum Measurement. Measurements in quantum mechanics are described by a set of *measurement operators* $\{M_m\}$. These measurement operators act on the state space of the system that is measured, and index m denotes the results that may be obtained in the measurement. Let the state of a quantum system be $|\psi\rangle$ right before the measurement. Then, the probability that result m is obtained is given by equation (2.14) [9].

$$p(m) = \langle\psi|M_m^\dagger M_m|\psi\rangle \tag{2.14}$$

Besides, the state of the system after the measurement, denoted as $|\psi'_m\rangle$ (being m the obtained result in the measurement), is described by equation (2.15).

$$|\psi'_m\rangle = \frac{M_m\,|\psi\rangle}{\sqrt{\langle\psi|M_m^\dagger M_m|\psi\rangle}} \tag{2.15}$$

Note that the set of measurement operators $\{M_m\}$ must fulfill the following condition so that the probabilities of measurement outcomes sum to one:

$$\sum_m M_m^\dagger M_m = I, \qquad \text{so that} \qquad \sum_m p(m) = \sum_m \langle\psi|M_m^\dagger M_m|\psi\rangle = \langle\psi|I|\psi\rangle = \langle\psi|\psi\rangle = 1.$$

From postulate 3, it can be deduced that when a quantum state is measured, the superposition state held by $|\psi\rangle$ is destroyed, and the post-measurement state changes to a specific state consistent with the measurement results. This conclusion suggests that the received states cannot be measured in quantum error correction, because those states would be destroyed by such operation [10].

Postulate 4: Composite systems. The state space of a composite system is the tensor product of the state spaces associated with the component systems. If there are n numbered systems, with $|\psi_i\rangle$ being the state associated with system number i, then the joint state of the composite system is $|\psi_1\rangle \otimes |\psi_2\rangle \otimes ... \otimes |\psi_n\rangle$.

Postulate 4 allows us to introduce another remarkable concept called *entanglement*.

2.2.2 Entanglement

A composite quantum system is an *entangled system* if it cannot be written as the tensor product of its component systems:

$$\nexists\, |\psi\rangle \in \mathcal{H}_\mathcal{A}, |\varphi\rangle \in \mathcal{H}_\mathcal{B} : |\psi\rangle_{AB} = |\psi\rangle \otimes |\varphi\rangle\,, |\psi\rangle_{AB} \in \mathcal{H}_\mathcal{A} \otimes \mathcal{H}_\mathcal{B}.$$

The *Bell states*, also known as *EPR pairs* (named after Albert Einstein, Boris Podolsky and Nathan Rosen), are specific states that represent the simplest examples of entanglement. These states are defined as

$$|\Phi^+\rangle = \frac{|0\rangle \otimes |0\rangle + |1\rangle \otimes |1\rangle}{\sqrt{2}} = \frac{|00\rangle + |11\rangle}{\sqrt{2}},$$
$$|\Phi^-\rangle = \frac{|0\rangle \otimes |0\rangle - |1\rangle \otimes |1\rangle}{\sqrt{2}} = \frac{|00\rangle - |11\rangle}{\sqrt{2}},$$
$$|\Psi^+\rangle = \frac{|0\rangle \otimes |1\rangle + |1\rangle \otimes |0\rangle}{\sqrt{2}} = \frac{|01\rangle + |10\rangle}{\sqrt{2}},$$
$$|\Psi^-\rangle = \frac{|0\rangle \otimes |1\rangle - |1\rangle \otimes |0\rangle}{\sqrt{2}} = \frac{|01\rangle - |10\rangle}{\sqrt{2}}.$$

An interesting property of these states is that the measurement of either qubit determines the result of the other qubit:

Consider the Bell state $|\Psi^-\rangle$ and the measurement operators $M_1 = |00\rangle\langle 00|$, $M_2 = |01\rangle\langle 01|$, $M_3 = |10\rangle\langle 10|$ and $M_4 = |11\rangle\langle 11|$. As postulate 3 states, the probability for each result ($m = 1, 2, 3, 4$) to be obtained in a measurement is

$$p(1) = \langle\Psi^-|M_1^\dagger M_1|\Psi^-\rangle = 0;$$
$$p(2) = \langle\Psi^-|M_2^\dagger M_2|\Psi^-\rangle = \tfrac{1}{2};$$
$$p(3) = \langle\Psi^-|M_3^\dagger M_3|\Psi^-\rangle = \tfrac{1}{2};$$
$$p(4) = \langle\Psi^-|M_4^\dagger M_4|\Psi^-\rangle = 0.$$

This means that the only possible outcomes are $|01\rangle$ and $|10\rangle$. If we measure one of the qubits, the result obtained will be either $|0\rangle$ or $|1\rangle$ (with a 50% of probability for each outcome), but the outcome of the other qubit will necessarily be the opposite of the first measured qubit.

2.2.3 No-Cloning Theorem

Theorem: No-cloning theorem. It is impossible to create an identical copy of an arbitrary quantum state. In other words, there is no unitary operator U in $\mathcal{H} \otimes \mathcal{H}$ that fulfills $U(|\varphi\rangle\,|\psi\rangle) = |\varphi\rangle\,|\varphi\rangle$, for any qubits $|\varphi\rangle$ and $|\psi\rangle$.

This theorem means a limitation in quantum communication since a qubit cannot be replicated in order to obtain repetition codes and protect the information from quantum noise, as it is done in classical communication.

2.2.4 Quantum Teleportation

Quantum teleportation is a process that allows the transmission of quantum states without the need of a quantum communication channel. In quantum teleportation, some classical communication and quantum entanglement is needed. Assume that Alice wants to deliver an arbitrary qubit $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$ to Bob. The protocol of quantum teleportation [11], which is shown in figure 2.4, is as follows:

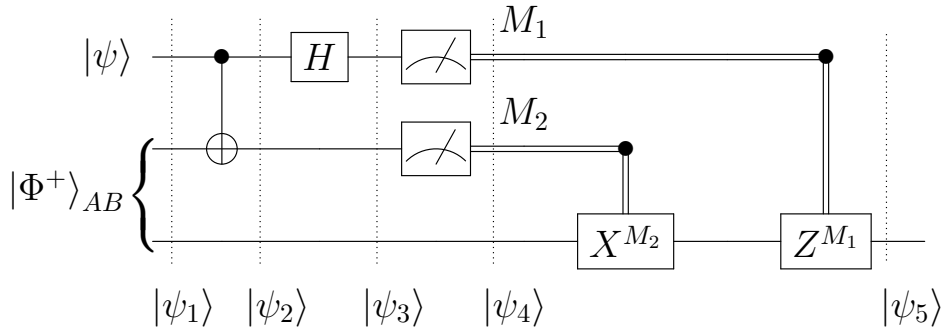


Figure 2.4: Schematic representation of quantum teleportation

1. An ERP pair $|\Phi^+\rangle_{AB}$ is generated. One qubit is given to Alice and the other one is given to Bob. The initial composite state is
$$|\psi_1\rangle = |\psi\rangle\,|\Phi^+\rangle_{AB} = \tfrac{1}{\sqrt{2}}[\alpha\,|0\rangle\,(|00\rangle + |11\rangle) + \beta\,|1\rangle\,(|00\rangle + |11\rangle)$$
2. A CNOT gate is applied to the composite state, with the arbitrary qubit $|\psi\rangle$ as the control qubit and Alice's EPR pair qubit as the target qubit. The composite state now is
$$|\psi_2\rangle = \tfrac{1}{\sqrt{2}}[\alpha\,|0\rangle\,(|00\rangle + |11\rangle) + \beta\,|1\rangle\,(|10\rangle + |01\rangle)$$
3. A Hadamard gate is applied to $|\psi\rangle$. The state becomes

$$|\psi_3\rangle = \tfrac{1}{2}[\alpha\,|0\rangle\,(|00\rangle + |11\rangle) + \beta(|0\rangle - |1\rangle)(|10\rangle + |01\rangle) =$$
$$= \tfrac{1}{2}[|00\rangle\,(\alpha\,|0\rangle + \beta\,|1\rangle) + |01\rangle\,(\alpha\,|1\rangle + \beta\,|0\rangle) + |10\rangle\,(\alpha\,|0\rangle - \beta\,|1\rangle) + |11\rangle\,(\alpha\,|1\rangle - \beta\,|0\rangle)$$

The previous expression is a sum of four terms, where the first two qubits correspond to Alice's qubits and the third one correspond to Bob's. For example, the first term has Alice's qubits at state $|00\rangle$, while Bob's qubit is at state $\alpha\,|0\rangle + \beta\,|1\rangle$, which is the original state $|\psi\rangle$ [9].

4. Alice measures the two qubits that she has, and sends the obtained bits to Bob using a classical communication channel (either 00, 01, 10 or 11).

5. Bob performs a correction to his EPR pair qubit depending on the bits he receives from Alice. Table 2.2 shows what Pauli gate(s) Bob needs to apply to his qubit so that he recovers the original state ($|\psi_5\rangle = |\psi\rangle$).

Alice's measurements	Post-measurement state	Correction needed
00	$\|\psi_4\rangle = \alpha\,\|0\rangle + \beta\,\|1\rangle$	None (I)
01	$\|\psi_4\rangle = \alpha\,\|1\rangle + \beta\,\|0\rangle$	X
10	$\|\psi_4\rangle = \alpha\,\|0\rangle - \beta\,\|1\rangle$	Z
11	$\|\psi_4\rangle = \alpha\,\|1\rangle - \beta\,\|0\rangle$	X, and then Z

Table 2.2: Correction in quantum teleportation depending on measurement results

3 Quantum Error Correction

Chapter 3 deals with Quantum Error Correction (QEC). This chapter introduces a conceptual framework where the error-correcting codes developed in classical communication can be interpolated into quantum communication, obtaining quantum error-correcting codes with similar properties to the original classical codes. As a consequence, Quantum Turbo Codes, which are based on classical turbo codes, can be constructed, and these Quantum Turbo Codes will be analysed in chapter 4.

A quantum error correcting code encodes k *logical qubits*, that is, the useful information in the communication, into n *physical qubits* that are sent through a noisy quantum transmission channel. The purpose of this encoding is being able to recover the original information (the original k logical qubits) in case any qubit is corrupted by the noisy channel.

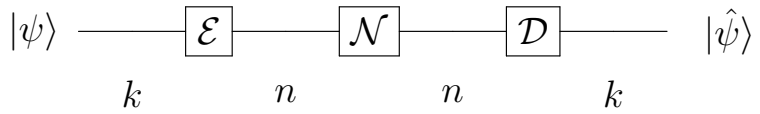


Figure 3.1: Simplified block diagram of quantum error correction

Figure 3.1 shows a simplifed block diagram of quantum error correction for a [[n, k]] quantum error correcting code. The $\mathcal{E}$ gate represents the encoding of k logical qubits into n physical qubits. The $\mathcal{N}$ gate is a noisy quantum channel; as it is explained in section 3.2, a depolarizing channel is considered in this document. $\mathcal{D}$ represents the decoding operator of the received n qubits into the k qubits that allow us to get an estimation $|\hat{\psi}\rangle$ of the original quantum state $|\psi\rangle$.

In the first section of this chapter, the main differences between quantum error correction and classical error correction are presented. Then, *depolarizing channels*, the *Pauli group*, and the *stabilizer codes* are introduced. The last section of this chapter shows the relationship between quantum error-correcting codes and classical error-correcting codes, and how quantum codes can be constructed from classical codes.

3.1 Restrictions in Quantum Error Correction

In the previous chapter we saw some interesting properties of quantum mechanics that might suppose a problem in Quantum Error Correction. There are fundamental differences between the quantum and the classical information processing because a quantum system can exist in the form of superposition state [13]. Next are detailed the main restrictions that arise in the quantum world, as well as the strategies that are used in quantum theory in order to overcome each issue.

Restriction 1: No-cloning theorem. The no-cloning theorem presented in section 2.2.3 implies that qubit repetition cannot be used in quantum error correcting codes since it is impossible to replicate qubits.

However, *qubit redundancy* does not violate such theorem. Note that the left part of equation (3.1) shows a redundancy of an arbitrary qubit $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ and such quantum state can be obtained in reality.

$$\alpha|00\rangle + \beta|11\rangle \neq (\alpha|0\rangle + \beta|1\rangle) \otimes (\alpha|0\rangle + \beta|1\rangle) \tag{3.1}$$

Restriction 2: Measurement destroys superposition. As it is explained in the 3^{rd} postulate of section 2.2.1, the measurement of a quantum state collapses its superposition an the quantum information is lost. This means that error correction must be done without measuring the received states.

In order to do that, instead of measuring the actual qubits, the *error syndromes* are measured. The syndrome tells us what error, if any, happened to the quantum state [9].

An example of how error syndromes work is presented next. Let $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ be a qubit that needs to be transmitted through a noisy channel. The encoder of the qubit works as follows:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \rightarrow |\psi_{enc}\rangle = \alpha|000\rangle + \beta|111\rangle .$$

Let the noisy channel perform a bit flip on the second qubit, ie, the error occurred is $E = I \otimes X \otimes I$. Then, the received state would be:

$$|\psi_N\rangle = E|\psi_{enc}\rangle = (I \otimes X \otimes I)(\alpha|000\rangle + \beta|111\rangle) = \alpha|010\rangle + \beta|101\rangle .$$

For this case, the syndrome measured could be achieved by applying the quantum circuit presented in figure 3.2. As it can be seen, this circuit leaves the received state $|\psi_N\rangle$ unchanged, while computing two classical bits s_1 and s_2 that correspond to the error syndrome. For $|\psi_{enc}\rangle = \alpha|000\rangle + \beta|111\rangle$, it can be proved that:

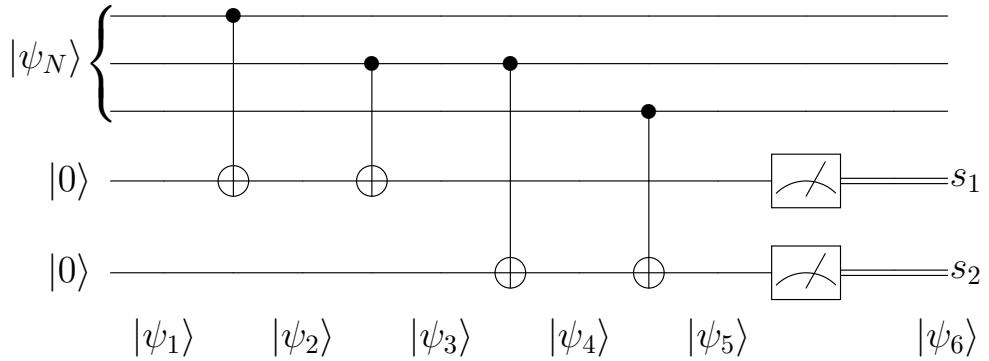


Figure 3.2: Example of a quantum circuit for syndrome measurement

1. $|\psi_1\rangle = |\psi_N\rangle|0\rangle|0\rangle = \alpha|010\rangle|0\rangle|0\rangle + \beta 101|0\rangle|0\rangle$
2. $|\psi_2\rangle = \alpha|010\rangle|0\rangle|0\rangle + \beta|101\rangle|1\rangle|0\rangle$
3. $|\psi_3\rangle = \alpha|010\rangle|1\rangle|0\rangle + \beta|101\rangle|1\rangle|0\rangle$
4. $|\psi_4\rangle = \alpha|010\rangle|1\rangle|1\rangle + \beta|101\rangle|1\rangle|0\rangle$
5. $|\psi_5\rangle = \alpha|010\rangle|1\rangle|1\rangle + \beta|101\rangle|1\rangle|1\rangle = |\psi_N\rangle|1\rangle|1\rangle$
6. $|\psi_6\rangle = |\psi_N\rangle$ and $s_1 = 1$, $s_1 = 1$.

As it can be seen, syndrome measurement does not destroy the superposition of the quantum state $|\psi_N\rangle$. The information given by the syndrome bits can be used in order to correct the received state $|\psi_N\rangle$: the first bit s_1 tells us if the first two qubits of the state are the same or not, and the second bit tells us if the second and third qubit are the same or not. Therefore, an estimation of the occurred error can be obtained, and the quantum state $|\psi_N\rangle$ can be corrected using quantum gates. There might be errors that cannot be corrected with this method (such as phase-flip errors [10]) or errors that share the same syndrome, but this is just an example of how an error can be corrected without destroying the actual superposition of quantum states.

Restriction 3: Errors are continuous. Errors might not be just phase flips or bit flips, but partial changes in the quantum states such as the ones introduced by the quantum gate R_ϕ. A continuum of possible errors may occur on a single qubit, and determining which error occurred in order to correct it would appear to require infinite precision, and therefore infinite resources [13].

However, due to the phenomenon of *error discretization*, it can be proved that if an arbitrary continuous error is a linear combination of regular errors that have associated error syndromes, then such continuous error can be corrected after the syndrome measurement and just applying the correction associated to each syndrome [14]. In other words, if an error correction code C corrects a set of errors $\mathcal{E}$, then C is also able to correct all the linear combinations of the elements in $\mathcal{E}$.

3.2 Depolarizing channel

A depolarizing channel is a noisy quantum channel affecting a quantum state. This channel has particularly nice symmetry properties. The probability of a qubit affected by a depolarizing channel to remain intact is $(1-p)$, while an error occurs with probability p. The channel performs an error X (bit flip) on the qubit with probability $p_x = \frac{p}{3}$, an error Z (phase flip) with probability $p_z = \frac{p}{3}$, and an Y error with probability $p_y = \frac{p}{3}$ [12].

The depolarizing channel represents the errors that are considered in this document.

3.3 Pauli group

The n-fold *Pauli group* G_n is defined to be the set of the n-th tensorial products of Pauli operators [15], $\{I, X, Y, Z\}^{\otimes n}$, together with the possible overall factors ± 1 and $\pm i$.

It can be proved that every element of G_n squares to $\pm I_n = \pm I^{\otimes n}$, and any two elements of G_n either commute or anti-commute. Moreover, every element of G_n is either Hermitian or anti-Hermitian [16].

Note that the Pauli matrices define a basis for the $\mathbb{C}^{2\times 2}$ space, which means that any operator can be expressed as a linear combination of Pauli matrices. Furthermore, as explained in section 2.1.3, if $|i\rangle$ and $|j\rangle$ are orthonormal bases for V and W respectively, then $|i\rangle \otimes |j\rangle$ is a basis for the composite vector space $V \otimes W$. Then, we can deduce that since the Pauli matrices $\{I, X, Y, Z\}$ form a basis for $\mathbb{C}^{2\times 2}$, then the n-fold Pauli group G_n is a basis for the vector space $(\mathbb{C}^{2\times 2})^{\otimes n} = \mathbb{C}^{2^n \times 2^n}$. This means that all the matrices in $\mathbb{C}^{2^n \times 2^n}$ are a linear combination of the elements of G_n. As a consequence, by the discretization of errors presented in section 3.1, we conclude that it is enough to consider the errors that are elements of the Pauli group in order to

design quantum error codes that correct any error in $\mathbb{C}^{2^n \times 2^n}$.

Sometimes, it is useful to represent elements of the Pauli group with the *symplectic representation*. The *symplectic representation* allows us to represent each element of the Pauli group with a pair of bit strings, a and b, that represent if an X or a Z operation is being used in each position of the tensor product, repectively, as

$$\sigma = e^{i\Phi} X^a Z^b, \sigma \in G_n$$

where $X^a = X^{a_1} X^{a_2} ... X^{a_n}$ and $Z^b = Z^{b_1} Z^{b_2} ... Z^{b_n}$. The string of bits a will have a 1 in the i^{th} position if an X is being applied in such position, b will have a 1 in the i^{th} position if Z is being applied in such position, and both a and b will have a 1 if an Y operator is being applied in the i^{th} position. This way, any element in G_n can be represented as $(a|b)$.

3.4 Stabilizer codes

Stabilizer codes, also known as *additive* quantum codes, are an important class of quantum codes whose construction is analogous to classical linear codes. Thanks to the stabilizer codes, the problem of finding Quantum Error Correcting Codes is reduced to that of constructing classical dual-containing quaternary codes [16].

3.4.1 Stabilizer formalism

Let S be an abelian subgroup of the Pauli group G_n that does not contain $-I_n$. A *stabilizer code* $C(S)$ associated with its *stabilizer* S is defined as a subspace fixed by S that fullfills equation (3.2).

$$C(S) = \{|\psi\rangle : M |\psi\rangle = |\psi\rangle, \forall M \in S\} \tag{3.2}$$

In other words, the code space is the simultaneous +1-eigenspace of all elements of the stabilizer S. Since S is an abelian group, all the elements $M \in S$ must commute.

Let $[[n, k]]$ be a stabilizer code that encodes k logical qubits into n physical qubits. Then the dimension of the code space $C(S)$ is 2^k, and the stabilizer S has 2^{n-k} elements [16].

A stabilizer S can be defined by a set of independent generators $\{M_i\}$. These independent generators cannot be expressed as products of each other, and each element of S can be written as a product of elements of the set. If an abelian subgroup S of G_n has 2^{n-k} distinct elements, then S is specified by a set of $n-k$ independent generators. Therefore, if a specific vector $|\Psi\rangle$ is stabilized by the $n-k$ generators of a group S, then $|\Psi\rangle$ is stabilized by S [16].

In stabilizer codes, it is very common to use a notation where all the tensor product symbols $\otimes$ are omitted. For example, an example of a generator M_i of a stabilizer S can be written as:

$$M_i = I \otimes X \otimes Z \otimes Z = I \quad X \quad Z \quad Z.$$

The **Gottesman-Knill theorem** [17] implies that the quantum computation related with stabilizer codes can be efficiently simulated by means of a classical computer, which is very helpful since quantum computers are not an available resource yet.

3.4.2 Encoding stabilizer codes

The encoding process of a [[n,k]] stabilizer code consists of adding $n-k$ *ancilla qubits* to the k logical qubits, and then applying an encoding unitary operator U to the whole quantum system in order to obtain the quantum state that will be sent through a quantum channel.

An *ancilla qubit* is a qubit whose value is known *a priori*, and it is usually chosen to be in state $|0\rangle$. In stabilizer encoding, when $n-k$ ancilla qubits are added to the k logical qubits, the resulting state (before applying the U encoder) is

$$|\varphi\rangle = |0\rangle^{n-k} \otimes |\psi\rangle \,,$$

where $|\psi\rangle$ is the tensor product of the k logical qubits that are wanted to be sent through the noisy channel. Now let $\mathcal{B}$ be a group generated by the following set:

$$\begin{array}{rcccccccc} M_1 & = & Z & I & I & \dots & I & I & \dots & I \\ M_2 & = & I & Z & I & \dots & I & I & \dots & I \\ M_3 & = & I & I & Z & \dots & I & I & \dots & I \\ \vdots & & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ M_{n-k} & = & I & I & I & \dots & Z & I & \dots & I \end{array}$$

It is easy to prove that $\mathcal{B}$ is a stabilizer for the quantum state $|\varphi\rangle$. The groups $\mathcal{B}$ and the stabilizer of the $|\psi\rangle$ state S are said to be *isomorphic* because their elements have the same commutation relationships [10], and it is denoted as $\mathcal{B} \cong S$. Isomorphism leads to the following lemma, which plays a huge role in quantum error correction in stabilizer codes.

Lemma: If $\mathcal{B}$ and S are both subgroups of G_n, and $\mathcal{B} \cong S$, then there exists a unitary operator U such that for all $B \in \mathcal{B}$ there exists a $\mathcal{S} \in S$ such that $B = U\mathcal{S}U^{-1}$, up to an overall phase.

This lemma is very helpful in error correction because the unitary U constructed in the previous lemma can be considered as the *encoding* operator U_{enc} for the code $C(S)$ [8]. In other words, it sufficient to find a unitary operator U that fulfills the previous lemma in order to find an operator that encodes the k logical qubits and ancilla qubits into n physical qubits.

Moreover, it can be proved that encoding operators U_{enc} can be formed as a combination of Hadamard gates, phase gates $R_{\pi/2}$ and CNOT gates [9]. Therefore, the encoding problem is reduced to finding the combination of quantum gates that construct a quantum circuit that performs an operator U_{enc} needed for going from the standard form stabilized by $\mathcal{B}$ to an actual codeword stabilized by S [10].

When it comes to the decoding process, since U_{enc} is an unitary matrix, it is sufficient to apply the unitary operator $U_{dec} = U_{enc}^{\dagger}$. The quantum circuit that will be used in order to perform this operator will be the same as in the encoder, but applied in the opposite direction and with the complex conjugates gates [10].

3.4.3 Error syndrome in stabilizer codes

Let $C(S)$ be a stabilizer code, and $\mathcal{E} = \{E_a\} \subset G_n$ be a set of errors. The error syndrome associated to an error E_a is the vector $\overline{s}_a$ with coefficients $s_{i,a}$. These coefficients express whether if the error E_a commutes or anticommutes with the generators M_i, and they fulfill equation (3.3).

$$M_i E_a = (-1)^{s_{i,a}} E_a M_i \tag{3.3}$$

Coefficients $s_{i,a}$ in $\overline{s}_a$ can take either the value of 0 or 1. If $s_{i,a} = 0$, it means that the error E_a commutes with the generator M_i, and if $s_{i,a} = 1$, it means that E_a anticommutes with M_i.

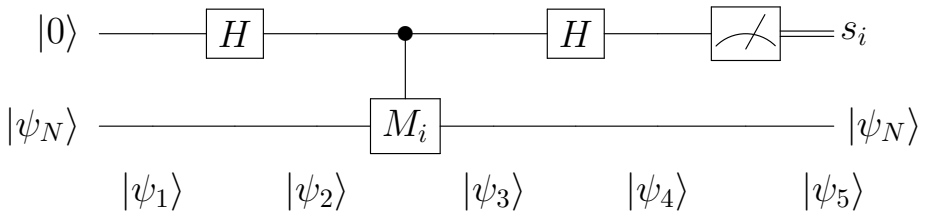


Figure 3.3: Quantum circuit for syndrome measurement in stabilizer codes

Figure 3.3 shows an example of a quantum circuit that measures the s_i syndrome coefficient associated to the generator M_i from a noisy state $|\psi_N\rangle = E\,|\psi_{enc}\rangle$, with $E \in \mathcal{E}$. This syndrome measurement process is as follows:

1. $|\psi_1\rangle = |0\rangle\,|\psi_N\rangle$
2. $|\psi_2\rangle = \left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right)|\psi_N\rangle = \frac{1}{\sqrt{2}}|0\rangle\,|\psi_N\rangle + \frac{1}{\sqrt{2}}|1\rangle\,|\psi_N\rangle$
3. $|\psi_3\rangle = \frac{1}{\sqrt{2}}|0\rangle\,|\psi_N\rangle + \frac{1}{\sqrt{2}}|1\rangle\,M_i\,|\psi_N\rangle = \frac{1}{\sqrt{2}}|0\rangle\,|\psi_N\rangle + \frac{1}{\sqrt{2}}|1\rangle\,\lambda\,|\psi_N\rangle = \left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}\lambda\,|1\rangle\right)|\psi_N\rangle$, where $\lambda = +1$ if $[E, M_i] = 0$, and $\lambda = -1$ if $\{E, M_i\} = 0$.
4. $|\psi_4\rangle = \left(\frac{1}{\sqrt{2}}H\,|0\rangle + \frac{1}{\sqrt{2}}\lambda H\,|1\rangle\right)|\psi_N\rangle = \left(\frac{1}{\sqrt{2}}\left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right) + \frac{1}{\sqrt{2}}\left(\frac{\lambda}{\sqrt{2}}|0\rangle - \frac{\lambda}{\sqrt{2}}|1\rangle\right)\right)|\psi_N\rangle = \left(\frac{1+\lambda}{2}|0\rangle + \frac{1-\lambda}{2}|1\rangle\right)|\psi_N\rangle$. Note that $|\psi_4\rangle = |0\rangle\,|\psi_N\rangle$ for $\lambda = +1$, and $|\psi_4\rangle = |1\rangle\,|\psi_N\rangle$ for $\lambda = -1$.
5. $|\psi_5\rangle = |\psi_N\rangle$, and $s_i = 0$ or $s_i = 1$.

Therefore, if the error $E \in \mathcal{E}$ that affects the encoded state commutes with the generator M_i, the value of λ will be +1, and the syndrome measured at the output of the quantum circuit will be $s_i = 0$. On the contrary, if the error E anticommutes with M_i, the value of λ will be -1, and the syndrome measured at the output of this quantum circuit will be $s_i = 1$.

3.4.4 Correcting errors in stabilizer codes

Once the syndrome vector $\overline{s}$ is obtained, the correction operator shown in equation (3.4) needs to be applied to the codeword. Note that $\hat{E}(\overline{s})$ is the estimated error given the syndrome vector $\overline{s}$.

$$|\hat{\psi}\rangle = \hat{E}(\overline{s})^{\dagger} |\psi_N\rangle \tag{3.4}$$

A code is said to be *non-degenerate* if that code has a different syndrome for each error E_a in the error set $\mathcal{E}$, and therefore it is capable of distinguishing each error. On the other hand, a code is said to be *degenerate* if the error set $\mathcal{E}$ contains errors that cannot be distinguished between each other because they share the same error syndrome; however, degenerate errors (errors that share the same syndrome) can be corrected by the same operator.

In the case of nondegenerate codes, since each error E_a in $\mathcal{E}$ has a different syndrome, measuring the $n-k$ generators will diagnose the error completely [16]. This means that *any* error in the error set $\mathcal{E}$ is correctable in nondegenerate codes.

In the case of degenerate codes, a uniquely identifiable error syndrome is not always required for an error to be correctable. Many errors that share the same syndrome can be corrected by the same operator. However, problems arise when an error $E_a \neq I_n$ commutes with all the generators M_i of the stabilizer S. The syndrome vector of this error will be a zero vector, which is usually associated with $E_a = I_n$, where the codeword is not corrupted by the error at all. Two different cases should be considered here:

1. If $E_a \in S$, then we do not need to worry because E_a will not corrupt the codeword at all. Remember that all the elements of S must fulfill equation (3.2). Hence, the correction operator applied to this kind of errors will be I_n (qubits are not modified) and the output will be correct.

2. If $E_a \notin S$, then this error is not correctable because it does corrupt the codeword but it has a zero vector syndrome and it is undetectable for the code.

3.5 Relationship between quantum and classical codes

This section presents how quantum error-correcting codes can be created from classical error correcting codes.

3.5.1 Stabilizer codes and classical quaternary codes

One of the greatest things of stabilizer codes is that they can be constructed from classical quaternary codes. A classical quaternary code over a finite field *GF(4)* is defined by its parity-check matrix H_4 and contains the elements $0, 1, \omega$, and $\overline{\omega}$. These elements obey the multiplication properties summarized in table 3.1.

Let $\tilde{H}_4$ be a new matrix defined by equation (3.5), and consider a map between the elements of GF(4) and the Pauli operators shown in table 3.2. If we apply this map from GF(4) to the Pauli operators and substitute the quaternary elements of $\tilde{H}_4$ with the corresponding Pauli operators, the rows of the matrix $\tilde{H}_4$ will represent a set of generators M_i for a quantum stabilizer code [10].

$$\tilde{H}_4 = \begin{pmatrix} \omega H_4 \\ \overline{\omega} H_4 \end{pmatrix} \tag{3.5}$$

x	0	1	ω	$\overline{\omega}$
0	0	0	0	0
1	0	1	ω	$\overline{\omega}$
ω	0	ω	$\overline{\omega}$	1
$\overline{\omega}$	0	$\overline{\omega}$	1	ω

Table 3.1: Summation properties of the elements in GF(4)

Note that, as it is stated in section 3.4.1, all the generators M_i of a stabilizer must commute with each other. So, in principle, not every classical quaternary code is suitable to be transfromed into a quantum stabilizer code. The initial classical quaternary code must be a dual-containing code (that is, the rows of its parity-check matrix must be orthogonal with each other, including with themselves) so that it can be used in order to get a stabilizer code [8].

Element in GF(4)	Pauli operator
0	I
1	X
ω	Y
$\overline{\omega}$	Z

Table 3.2: Map between the Pauli operators and the elements in GF(4)

By following this procedure, a classical quaternary $[n, k, d]$ code can be turned into a non-degenerate $[[n, 2k - n, d]]$ stabilizer quantum code [16]. Furthermore, the better the classical quaternary code is, the better stabilizer code will be obtained [10].

Similarly, binary codes can also be used in order to construct stabilizer quantum codes [8].

3.5.2 Entanglement Assisted Quantum Error Correction Codes

As seen in section 3.5.1, the problem of finding a good quantum error-correcting code is reduced to finding a good classical quaternary code. However, the parity-check matrix of such quaternary code must be self-orthogonal, which means that we need to find dual-containing quaternary codes. This restriction could have relevant implications because there might be a case where the best quaternary code could not be used to obtain a quantum code. Nevertheless, as it will be explained in this section, a shared entanglement between transmitter and receiver can overcome this obstacle. The codes that use entanglement in order to overcome the dual-containing constraint are called *Entanglement Assisted Quantum Error Correction Codes (EAQECCs)*.

An EAQECC encodes k logical qubits into n physical qubits, with the help of $n-k-c$ ancilla qubits and c *ebits* or entangled pair of qubits.

The c ebits are assumed to be pre-shared by the transmitter and the receiver, that it, the receiver has received c halves of the EPR pairs before the transmission of the other qubits, and those ebit halves are assumed to have been transmitted without errors.

Roughly explained, the introduction of the c ebits modifies the group $\mathcal{B}$ that stabilizes the overall quantum state $|\varphi\rangle$ and expands the generators of such group [16]. As a consequence of this expansion, we find a new group that is abelian [8]. This means that if we follow the

procedure defined in section 3.5.1, we will get a set of M_i that commute with each other, even when the original quaternary code is not dual-containing.

The amount of ebits that need to be introduced in each quantum code depends on the parity-check matrix of the classical quaternary code it is based on [18], as it can be seen in equation (3.6). Note that if the quaternary code is dual-containing, c is 0 because no entanglement is needed.

$$c = rank(H_4 {H_4}^{\dagger}) \tag{3.6}$$

In conclusion, stabilizer codes can be constructed based on any classical binary or quaternary code thanks to entanglement assistance. Moreover, it can be said that it is enough to find a good classical code in order to get a good non-degenerate quantum code [10]. Moreover, pre-shared entanglement allows us to import modern codes such as Turbo codes to quantum error correction, as we will see in the following chapter.

4 Analysis of Quantum Turbo Codes

Following the possibility of constructing quantum error correcting codes based on already existing classical codes (thanks to the stabilizer codes and entanglement assistance, as it is explained in chapter 3), the next step in quantum error correction would be to choose the "best" classical codes, that is, near Shannon limit error-correcting codes, and import them into the quantum world.

In that sense, classical turbo codes were first published in 1993 and they were the first practical codes to closely approach the channel capacity [19]. Turbo codes are used in UMTS and LTE standards, as well as in deep space satellite communications.

The initial Quantum Turbo Codes (QTCs) proposed by David Poulin, Jean-Pierre Tillich and Harold Ollivier were based on classical serial turbo codes [20], but these QTCs failed to be recursive and non-catastrophic at the same time. Quantum Turbo Codes need to be simultaneously recursive and non-catastrophic so that the codes have a minimum distance growing with the blocklength, and the iterative decoding algorithm converges, respectively. Entanglement assistance in QTCs can be used in order to overcome such simultaneity problem [21].

In this chapter, Quantum Convolutional Codes (QCCs) are explained first, since they are used as building blocks for the construction of QTC codes [10]. Then, the actual Quantum Turbo Codes are analysed, assessing the construction, decoding and performance of such codes.

4.1 Entanglement Assisted Quantum Convolutional Codes

Quantum Convolutional Codes (QCCs) can be defined as stabilizer codes [20] with an encoding matrix U and a convolutional structure [21]. QCCs are often provided with entanglement, obtaining Entanglement Assisted Quantum Convolutional Codes (EAQCCs). The entanglement in EAQCCs allows the construction of both recursive and non-catastrophic encoders [10].

The unitary encoder U of a EAQCC encodes m memory qubits, k logical qubits of a quantum state $|\varphi\rangle$, a ancilla qubits and c halves of ebits into m new memory qubits and $n = k + a + c$ physical qubits, as shown in equation (4.1).

$$|\psi\rangle = U\left(|m\rangle \otimes |\varphi\rangle \otimes |0\rangle^{\otimes a} \otimes (|\Phi\rangle_{AB}^{+})^{\otimes c}\right) \tag{4.1}$$

In equation (4.1), $|\psi\rangle$ represents the resulting encoded state of m memory qubits and $n = k + a + c$ physical qubits. The state $|m\rangle$ represents the m memory qubits of the initial state, and $|\Phi\rangle_{AB}^{+}$ represents the EPR pairs that work as entanglement to the code.

The transformation that the unitary U produces on binary representations of Pauli operators acting on the m, k and a qubits and c halves of ebits is shown in equation (4.2).

$$(M' : P) = (M : L : S : E)U \tag{4.2}$$

In equation (4.2), M' acts on the m output memory qubits, P acts on the n output physical qubits, M acts on the m input memory qubits, L acts on the k input logical qubits, S acts on the a input ancilla qubits, and E acts on the c halves of input ebits [21]. Note that S can be decomposed as $S = S^x + S^z$. The i^{th} component S_i^x of S is the Pauli operator X if the syndrome $s_i = 1$, and $S_i^x = I$ if $s_i = 0$.

Quantum convolutional codes work similarly to classical convolutional codes. The quantum data is divided in periodic blocks of information qubits, ancilla qubits and halves of ebits. The

overall encoding operation of the code is the iterative application of the unitary transformation matrix U to each block, where the m output memory qubits of one transformation are fed into the next transformation as its m input memory qubits [21], as shown in figure 4.1. The total number of identical repetition of the transformation U is called the *length of the code* [20], and is denoted as N.

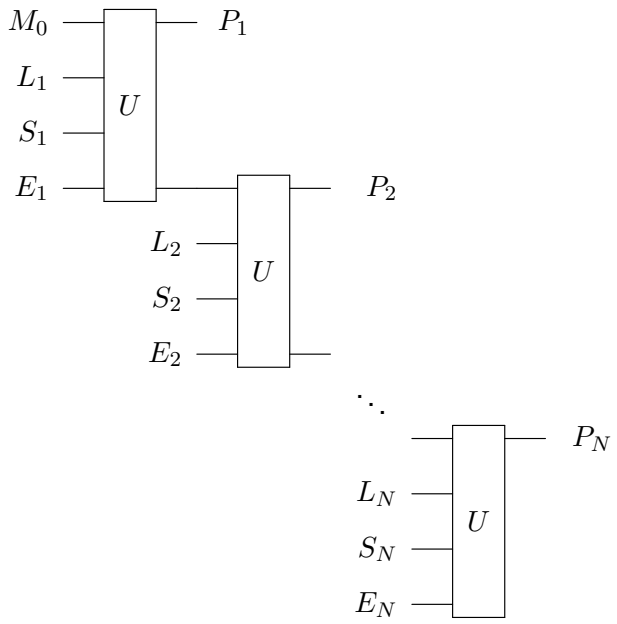


Figure 4.1: Circuit diagram of a quantum convolutional encoder with unitary U

The main advantage of QCCs is that the complexity of the overall encoding increases only linearly with the length of the code for a fixed m, and the complexity of the decoding increases linearly with the length of the code by using a local maximum likelihood decoder along with a belief propagation algorithm [21]. When N is large compared to n, the quantum communication rate of a quantum convolutional code is k/n, and the entanglement consumption rate is c/n.

4.1.1 State Diagram

The *state diagram* of a quantum convolutional code is a very important tool in the analysis of such code. The *state diagram* is defined as a directed multi-graph with 4^m vertices (called *memory states*), where each vertix is labeled with an m-qubit Pauli operator M. Two vertices M and M' are linked by a directed edge ($M \to M'$) with a label (L, P), if there exists a k-qubit Pauli operator L, an n-qubit Pauli operator P, and an a-qubit Pauli operator $S^z \in \{I, Z\}$ such that

$$(M' : P) = (M : L : S^z : I_c)U.$$

The labels L and P are referred to as the logical label and the physical label of the edge, respectively. Figure 4.2 shows an example of a seed transformation U, and figure 4.3 shows the state diagram of such transformation seed.

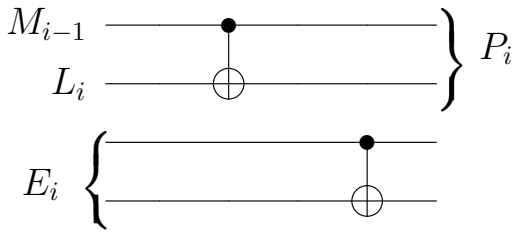


Figure 4.2: Example of a transformation seed U

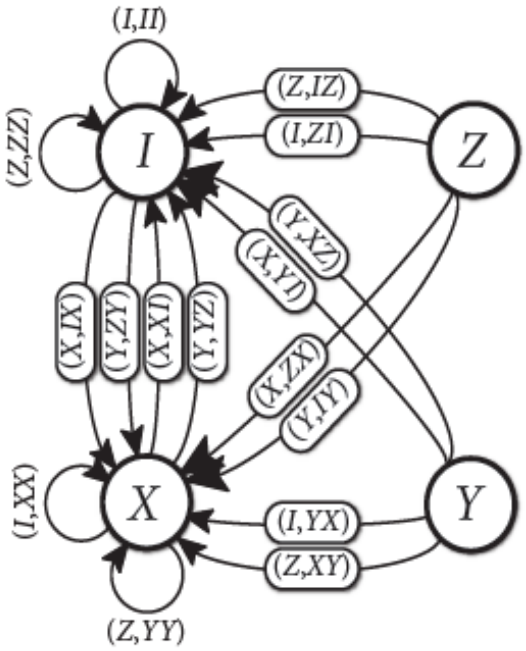


Figure 4.3: State diagram of the transformation seed in figure 4.2

The state diagram of a quantum convolutional code is used in order to determine if such QCC is *non-catastrophic* and *recursive*.

4.1.2 Non-catastrophity

It is very important for a quantum convolutional encoder to be *non-catastrophic* because otherwise a finite-weight error could turn into an error with infinite weight [10]. We can tell if a QCC is catastrophic or non-catastrophic by taking a look at the state diagram of such code. However, some concepts need to be introduced first before explaining how to find non-catastrophic codes by looking at the state diagram.

A *path* in the state diagram is a sequence of vertices $M_1, M_2, ..., M_t$ such that the edge $M_i \rightarrow M_{i+1}$ belongs to that sequence, for $i \in 1, 2, ..., t$. Each logical operator belonging to the code C corresponds to a path in the state diagram, which corresponds to the memory states that are visited while encoding the logical operator [20]. A closed path is also called a *cycle*.

The *weight* of a Paul group is the number of terms in the tensor product which are not equal

to the identity I. Then, the *physical* and *logical* weights of a logical operator are defined as the sums of the corresponding weights of the edges traversed in a path that encodes such logical operator [10].

Now, a QCC encoder is *non-catastrophic* if and only if the only cycles in its state diagram with a physical weight equal to zero have also a logical weight equal to zero [20].

If we observe the state diagram in figure 4.3, we will see that there is no zero physical-weight cycle with a non-zero logical weight, and therefore, such encoder is non-catastrophic.

4.1.3 Recursiveness

Recursiveness is also desirable for QCC encoders, specially when the QCC encoder is used as the inner encoder of a quantum turbo code. Using a recursive QCC as the inner encoder of a quantum turbo code ensures that the distance of such QTC, on average, will grow almost linearly with the length of the code [21].

The concept of *admissible path* is used to determine the recursiveness of an encoder. An *admissible path* is a path in the state diagram such that its first edge is not part of a zero pysical-weight cycle.

A *recursive encoder* is such that any admissible path with logical weight 1 starting from a vertex belonging to a zero physical-weight loop does not contain a zero physical-weight loop [20].

As an example, we can see that the encoder of the state diagram in figure 4.3 is not recursive. The vertex I belongs to a zero physical-weight loop; this loop is formed by the edge (I, II). The path formed by the edges (Z, ZZ) and (I, II) is an admissible path with a logical weight equal to one. This is path starts from vertex I, and it contains the zero physical-loop (I, II). Therefore, by definition, the encoder of this example is not recursive.

4.2 Quantum Turbo Codes

This section presents the construction of quantum turbo codes as an interleaved serial concatenation of the quantum convolutional codes explained in section 4.1. This section also presents the decoding process of QTCs, and the performance analysis of these codes depending on the estimated error probability of the depolarizing channel.

4.2.1 Construction: Interleaved serial concatenation

Quantum turbo codes are obtained from a particular form of interleaved serial concatenation of quantum convolutional codes. In classical communication, it is possible to construct classical turbo codes from interleaved *parallel* concatenation of classical convolutional codes too; however, the no-cloning theorem explained in section 2.2.3 makes interleaved parallel concatenation impossible in the quantum world [10]. Therefore, the only possible way to construct quantum turbo codes is from an interleaved serial concatenation of QCCs.

So, the encoder of a QTC is formed as the interleaved serial concatenation of QCCs and it has three basic components:

- Outer code: A QCC that encodes k^{out} qubits into n^{out} qubits with encoder V^{out}.
- Inner code: A QCC that encodes $k^{in} = n^{out}$ qubits into n^{in} qubits with encoder V^{in}.

• Interleaver: A quantum interleaver of size $N = n^{out} = k^{in}$.

A *quantum interleaver* Π of size N is an N-qubit symplectic transformation, composed of a permutation π of N qubts and a tensor product of single-qubit symplectic transformation [20]. The overall transformation generated by a quantum interleaver is shown in equation (4.3), where $K_1, ..., K_N$ are some fixed symplectic matrices acting on the Pauli group G_1.

$$(P_1, ..., P_N) \leftarrow (P_{\pi(1)}K_1, ..., P_{\pi(N)}K_N) \tag{4.3}$$

Figure 4.4 shows a graphical representation of a quantum turbo encoder. The interleaver Π is constructed by a combination of *SWAP gates* and Pauli gates, and the convolutional encoders V^{in} and V^{out} are constructed by unitaries that can be formed by combinations of Hadamard, phase and CNOT gates. The encoder of a quantum turbo code works as follows: first, the outer encoder encodes the information stream, then the interleaver performs a transformation to all the qubits, and finally the qubits at the output of the interleaver are encoded again by the inner encoder.

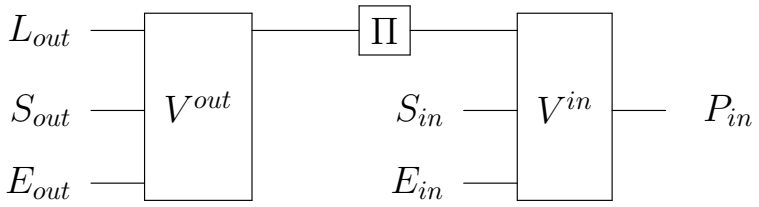


Figure 4.4: Circuit diagram of a quantum turbo encoder

Therefore, the resulting encoding matrix of the interleaved concatenated code is

$$V = V^{out}\Pi V^{in}.$$

It is considered that the best combination in order to build a quantum turbo code is to choose a recursive, non-catastrophic QCC as the inner code, and a non-recursive, non-catastrophic QCC as the outer code of such quantum turbo code. This combination ensures that the resulting quantum turbo code is recursive and non-catastrophic. Furthermore, this combination minimizes the entanglement consumption of the code, because the outer code is not an entanglement assisted code.

The communication rate of the resulting turbo code is k^{out}/n^{in}, and the entanglement consumption rate is $(c^{out} + c^{in})/n^{in}$, where c^{out} and c^{in} are the number of ebits consumed by the outer code and the inner code, respectively.

4.2.2 Iterative decoding

Section 4.2.1 explained how encoders are constructed in quantum turbo codes. This section shows how these codes are decoded.

The decoding of classical turbo codes, such as in the sum-product algorithm, is based on finding the codeword that is most probable to have been sent by the transmitter based on the syndrome. However, this method cannot be applied in quantum communication because quantum states are continuous. Instead, the Pauli error (which is discrete) that has affected the quan-

tum state is considered in quantum decoding. In quantum error correction, maximum-likelihood decoding estimates the most probable error coset that may have happened to the information during transmission [20], so that the corresponding correcting operator can be applied to the received state.

The "Soft-Input-Soft-Output" (SISO) decoding algorithm for QTCs was proposed by David Poulin, Jean-Pierre Tillich and Harold Ollivier [20]. This algorithm consists in the exchange of *a posteriori* information between the constituent quantum convolutional decoders of the QTC, and was later improved by introducing the concept of extrinsic information transfer [21].

Figure 4.5 shows a schematic representation of the SISO decoder. The decoding algorithm [10] is detailed below:

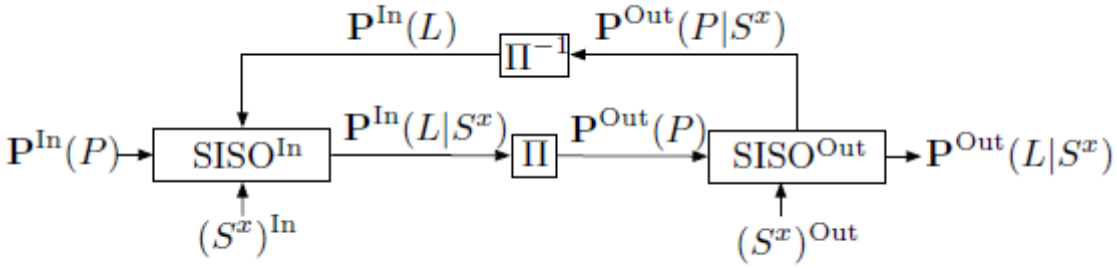


Figure 4.5: Circuit diagram of the SISO decoder for quantum turbo codes

1. The inner SISO decoder $SISO^{In}$ first decodes the inner code, based on the channel information, $P^{In}(P)$, and the measured error syndrome, $(S^x)^{In}$ corresponding to such inner code. The inner code is decoded by $(V^{in})^{\dagger}$, and the syndrome measurement is done as explained in section 3.4.3.

2. Then, the obtained probabilities of error $P^{In}(L/S^x)$ that might have happened during the transmission (based on the measured syndrome $(S^x)^{In}$) are passed through the interleaver and sent to the outer SISO decoder as its input information, $P^{Out}(P)$.

3. The outer SISO decoder, $SISO^{Out}$, decodes the outer code similarly as in step 1: by $(V^{Out})^{\dagger}$, and based on the input information $P^{Out}(P)$ and syndrome $(S^x)^{Out}$ of the outer code.

4. The output probability of the outer SISO decoder $P^{Out}(P/S^x)$ (that is, the information about the probability of the physical operator P depending on the syndrome) is used as the input of the inner SISO decoder. In order to do that, $P^{Out}(P/S^x)$ is deinterleaved before it is sent back to $SISO^{In}$, in the form of the probability $P^{In}(L)$ that the inner code has suffered some logical error L.

5. The inner SISO decoder $SISO^{In}$ decodes the inner code again with the input $P^{In}(L)$ and syndrome $(S^x)^{In}$. Note that for the first iteration, the probability $P^{In}(L)$ is taken as equiprobable.

6. The whole procedure is repeated an arbitrary number of times. The process can be stopped after a fixed number of iterations, or when the decoding in the inner and outer decoders match.

7. The eventual output of the overall SISO decoder is the probability distribution $P^{Out}(L/S^x)$, which tells us the most probable error coset given the syndrome S^x, and the corresponding error correction operation can be now applied to the decoded quantum state.

The algorithm described above passes along *a posteriori* information between the outer and inner decoder, and as a consequence, the iterations on each decoder depend on one another. This is translated into a harmful positive feedback effect that prevents the decoding algorithm from achieving the desired performance results usually obtained in iterative decoding [21].

In order to overcome this undesired effect, it is necessary that the *a priori* information directly related to a given information qubit is not used again in the other decoder. This can be achieved by designing the decoders in a way such that they remove *a priori* information from the *a posteriori* information before they feed it to the other decoder. This way, decoders do not exchange *a posteriori* information, but only *extrinsic* information instead, which is new and unkown to the other decoder.

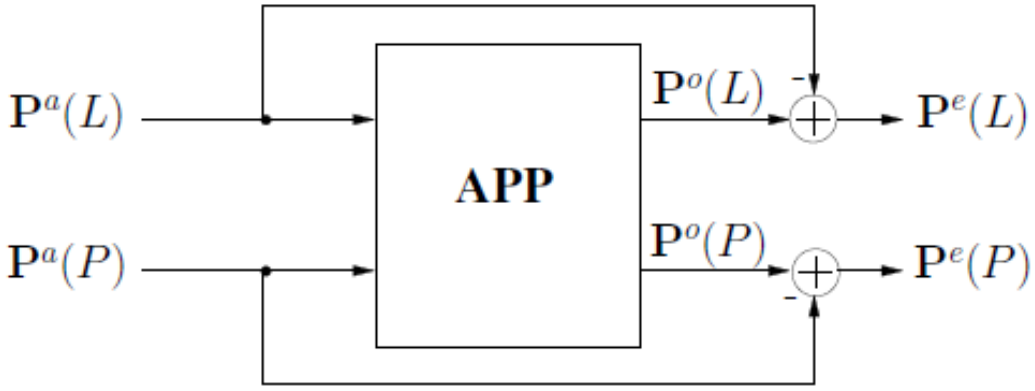


Figure 4.6: Circuit diagram of a SISO decoder that removes *a priori* information from *a posteriori* information in order to obtain extrinsic information. Note that the probabilities are in logarithmic scale.

Figure 4.6 shows a four-port SISO decoder that exploits an *A Posteriori Probability* (APP) module. The inputs are the *a priori* information $P^a(L)$ and $P^a(P)$, and the outputs are the *a posteriori* information $P^o(L)$ and $P^o(P)$. The *extrinsic* probabilities $P^e(L_i^j)$ and $P^e(P_i^j)$, for the j^{th} qubit and time instant i, are obtained by discarding the *a priori* information from the *a posteriori* information, as shown in equation (4.4).

$$\begin{aligned} P^e(L_i^j) &= N_{L^j} \frac{P^o(L_i^j)}{P^a(L_i^j)}, \\ P^e(P_i^j) &= N_{P^j} \frac{P^o(P_i^j)}{P^a(P_i^j)}. \end{aligned} \tag{4.4}$$

Note that N_{L^j} and N_{P^j} in equation (4.4) are normalization factors. Moreover, in order to reduce the computational complexity, log-domains are considered, so that multiplications and fractions are simplified into additions and substractions, as shown in equation (4.5).

$$\begin{aligned} \ln[P^e(L_i^j)] &= \ln[N_{L^j}] + \ln[P^o(L_i^j)] + \ln[P^a(L_i^j)], \\ \ln[P^e(P_i^j)] &= \ln[N_{P^j}] + \ln[P^o(P_i^j)] + \ln[P^a(P_i^j)]. \end{aligned} \tag{4.5}$$

Therefore, the inputs and outputs of the SISO decoders, such as the one if figure 4.6, are the logarithmic *a priori* and extrinsic probabilities, respectively [10].

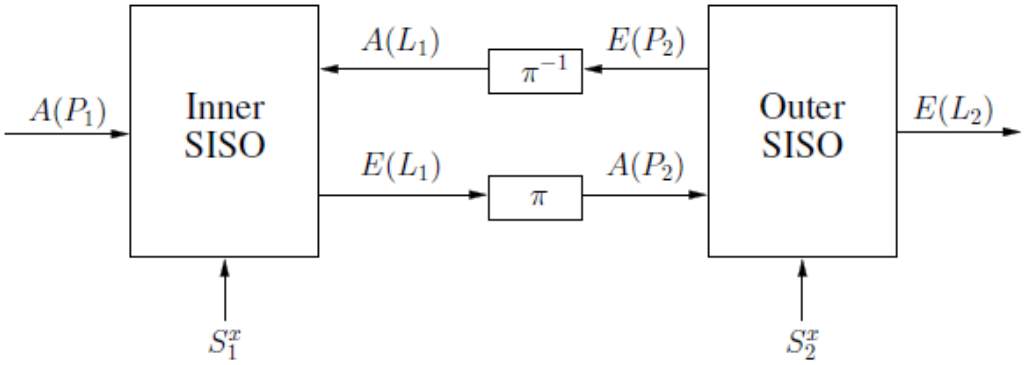


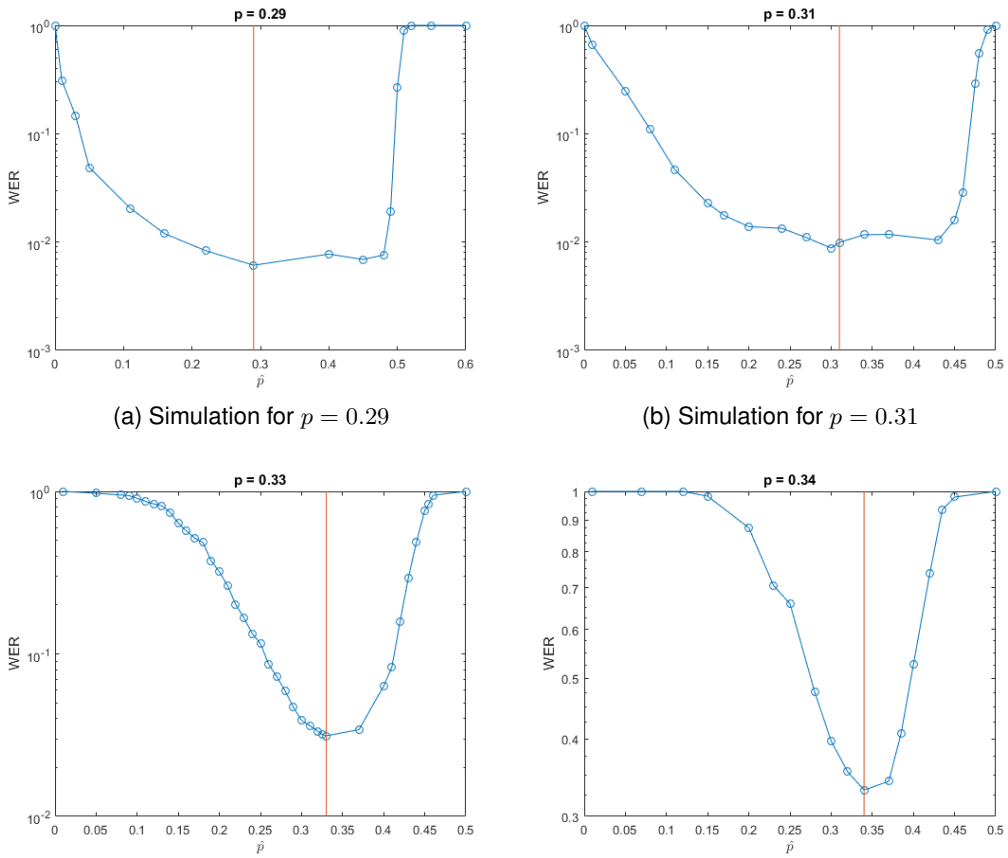
Figure 4.7: Circuit diagram of the updated SISO decoder for quantum turbo codes

Finally, the introduction of the extrinsic information modifies the decoding algorithm slightly, and the circuit diagram of such decoder changes too. Figure 4.7 shows an updated circuit diagram of the decoding algorithm in SISO decoders, in which the inner and outer decoders exchange extrinsic information (in contrast to figure 4.5, where *a posteriori* information is exchanged). If figure 4.7, $A(y)$ and $E(y)$ represent the logarithmic *a priori* and extrinsic probabilities of y, respectively, with $y \in \{L_1, L_2, P_1, P_2\}$.

4.2.3 Performance of iterative decoding algorithm with channel mismatch

In section 4.2.2 we saw that the channel information is given to the SISO decoder as an input. However, in many cases it is impossible to obtain an exact determination of a noisy quantum channel. This section analyses the impact of a channel mismatch on the error correction performance, that is, how strong SISO decoders are when the estimated probability error of the depolarizing channel is not accurate.

The quantum error correcting systems were simulated by the Matlab programs authored by Mark M. Wilde, Min-Hsiu Hsieh, and Zunaira Babar [22]. This software measures the word error rate (WER) of a quantum turbo code, and it is available under the GNU General Public License v3. The QTC of this software is optimized with EXIT charts. In these original Matlab codes, the depolarizing channel and the channel information fed to the SISO decoder were characterized by the same variables. Therefore, a little modification had to be done so these codes so that a channel mismatch could be simulated (new variables had to be defined).



(a) Simulation for $p = 0.29$

(b) Simulation for $p = 0.31$

(c) Simulation for $p = 0.33$

(d) Simulation for $p = 0.34$

Figure 4.8: Simulations of the channel mismatch in a QTC with a random interleaver. The graphics show the WER on the y axis, and the estimated probability $\hat{p}$ on the x axis

When it comes to the interleaver of the quantum turbo encoder, simulations were performed with three different interleavers: a random interleaver, an S-random interleaver and a JPL interleaver.

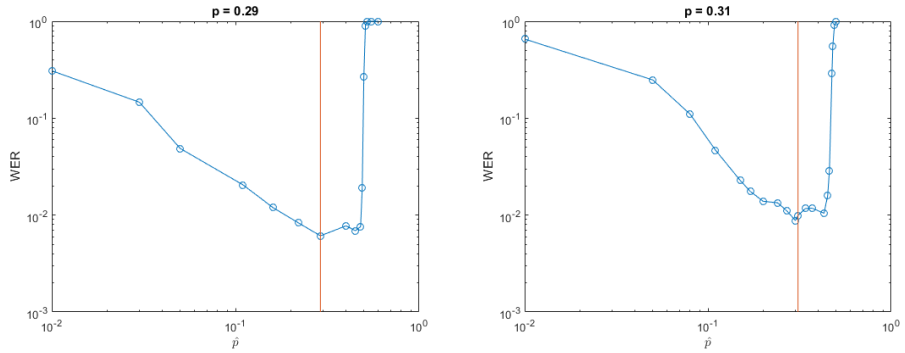
A *random interleaver* is a pattern π that permutes the symbols of the code in a completely arbitrary way. It reorders the input symbols using a random permutation. The random interleaver chosen for the simulations in this project had a with a blocklength of $N = 3000$.

With the random interleaver explained before, four different scenarios were simulated. In each scenario, the depolarizing channel of the quantum communication had a different error probability p, and the WER of the QTC was estimated for a set of estimated depolarizing probabilities $\hat{p}$. The value $\hat{p}$ is the channel information that is given to the SISO decoder, even when this probability does not match with the actual depolarizing probability p of the depolarizing channel used in the quantum communication. These four simulations were done for $p = 0.29$, $p = 0.31$, $p = 0.33$ and $p = 0.34$, and in each simulation, the WER was calculated for a set of estimated probabilities $\hat{p} \in [0, 0.5]$.

Figure 4.8 summarizes the results obtained in such simulations. Subfigure 4.8 a) shows the

WER obtained depending on the estimated depolarizing probability $\hat{p}$ when the actual depolarizing probability is $p = 0.29$; subfigure 4.8 b) shows the WER depending on $\hat{p}$ when $p = 0.31$; subfigure 4.8 c) shows the WER depending on $\hat{p}$ when $p = 0.33$; and subfigure 4.8 d) shows the WER depending on $\hat{p}$ when $p = 0.34$. The vertical lines in subfigures 4.8 a), 4.8 b), 4.8 c) and 4.8 d) represent the value $\hat{p} = p$.

Note that the channel mismatch does affect the error correction of quantum turbo codes. The SISO decoders present the lower WER when the estimated error probability $\hat{p}$ matches the actual error probability p, and the WER increases as the value of $\hat{p}$ moves away from p. Overall, the WER curves presented in figure 4.8 are quite similar to the WER curves corresponding to a channel mismatch in classical turbo codes [23], so it seems that the mismatch results in the quantum world are coherent with the results in the classical world.



(a) Simulation for $p = 0.29$

(b) Simulation for $p = 0.31$

(c) Simulation for $p = 0.33$

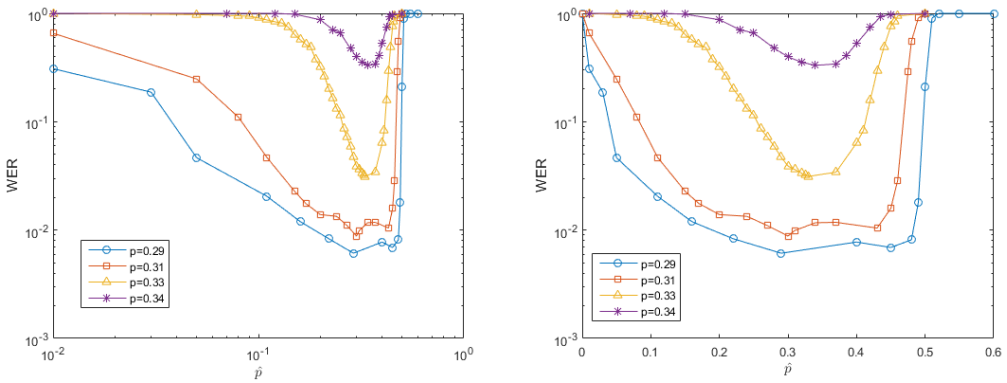
(d) Simulation for $p = 0.34$

Figure 4.9: Simulations of channel mismatch in a QTC with a random interleaver. The graphics show the WER on the y axis, and $\hat{p}$ on the x axis in logarithmic scale

If we analyse the graphics more meticulously, we can deduce a few more interesting conclusions. Even though both the underestimation and overestimation of the depolarizing probability affects quantum error correction negatively, a big overestimation of the channel is slightly more disadvantageous than a big underestimation. This can be seen more clearly in figure 4.9, where the x axis of the plots are shown in logarithmic scale. In the subfigures of figure 4.9, the slope of the WER is greater at the right of the vertical line (where $\hat{p} > p$) than at the left of the vertical

line (where $\hat{p} < p$).

Moreover, there is an area around $\hat{p} = p$ where the WER seems to be quite stable in every subfigure. This is shown more clearly in figure 4.10, where all the WER curves are drawn in the same graphic. Among the four simulations, the narrowest stable area is shown in the case where the depolarizing probability of the channel is $p = 0.34$; and the widest stable area is shown in the case where $p = 0.29$. It seems like the lower the depolarizing probability of the channel is, the wider the stable area in the WER curve will be.



(a) All simulations with the x axis in logarithmic scale

(b) All simulations with the x axis in linear scale

Figure 4.10: All the simulations of channel mismatch in QTC

Appart from the random interleaver, other kinds of interleavers were simulated too, in order to compare how channel mismatch affects different interleavers. The other simulated interleavers are the S-random interleaver and the JPL interleaver.

On the one hand, an *S-random interleaver* is an interleaving pattern π which permutes elements randomly with the following condition:

$$|\pi(i) - \pi(j)| > S \text{ for } i \text{ and } j \text{ such that } |i - j| \leq S.$$

In order to choose the value of S it is recommended that the condition $S = \sqrt{\frac{N}{2}}$ is satisfied, where N is the blocklength of the interleaver. This way, S-random interleavers can usually be produced in reasonable time by generating random integers repeatedly until such condition is fulfulled [24]. The S-random interleaver that was simulated in this project was an interleaver with length $N = 3000$ and $S = 25$. Note that these two parameters fulfill the condition recommended before. This S-random interleaver was simulated for a depolarizing channel with probability $p = 0.33$.

The results of this simulation are shown in figure 4.11. Figure 4.11 shows the performance analysis of the S-random interleaver compared to the results of the random interleaver for $p = 0.33$. We can see that the performance of the S-random interleaver is analogous to the performance of the random interleaver, in the sense that both graphics have a similar shape in figure 4.11. The S-random interleaver, however, shows a lower WER especially when the

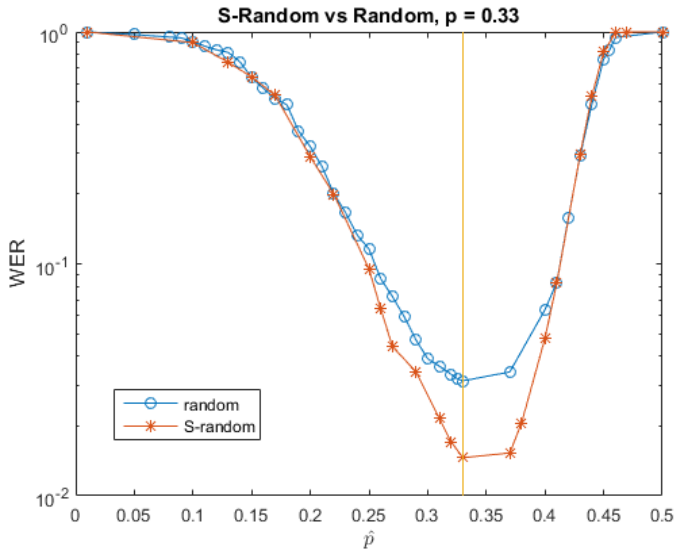


Figure 4.11: Simulation of the S-random interleaver compared to the random interleaver for $p = 0.33$

estimated $\hat{p}$ is close to the actual p. These results are coherent with the simulations of the interleavers without mismatch [24], where it is shown that S-random interleavers present a lower error floor than random interleavers.

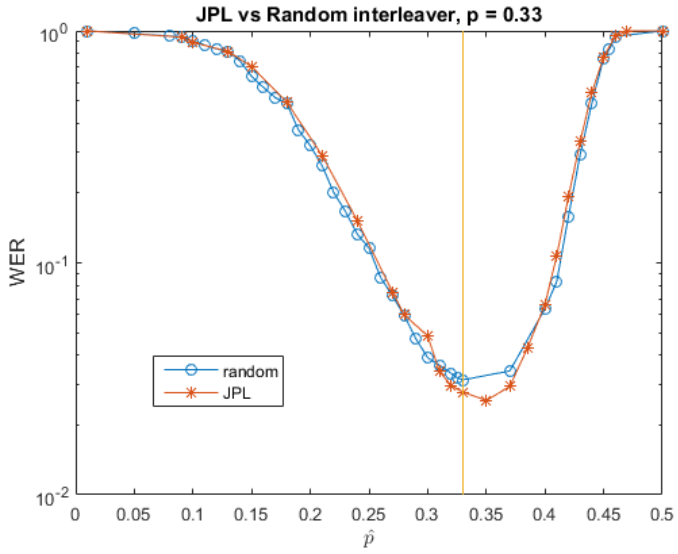


Figure 4.12: Simulation of the JPL interleaver compared to the random interleaver for $p = 0.33$

On the other hand, a *JPL interleaver* is an interleaving pattern that, unlike the random or the S-random interleaver, permutes the symbols based on a deterministic algorithm. This specific

algorithm consists of factorizing the length N into two integers and performing a few operations for each position s from $s = 1$ to $s = N$.

The JPL interleaver used for the simulation has a blocklength of $N = 3000$, just like the random and S-random interleavers used in the previous simulations. This simulation was made for a depolarizing channel with error probability $p = 0.33$, in order to compare the performance of the JPL interleaver with the performances of the previous interleavers. The obtained results are depicted in figure 4.12. Note that the WER curve for the JPL interleaver has a similar shape to the WER curve for the random interleaver; the only difference is that the JPL interleaver presents a slightly lower WER when the estimated depolarizing probability $\hat{p}$ is close to p.

The results of the three interleavers for a depolarizing channel with $p = 0.33$ are plotted in figure 4.13. This figure shows that for $\hat{p}$ values that are close to p, the S-random interleaver is the one with the best performance in terms of word error rate, followed by the JPL interleaver and the random interleaver. These results coincide with the simulations of the simulations of the interleavers with no mismatch [24], where it is shown that the S-random interleaver has a lower error floor than the JPL interleaver, and that the JPL has a lower error floor than the random interleaver.

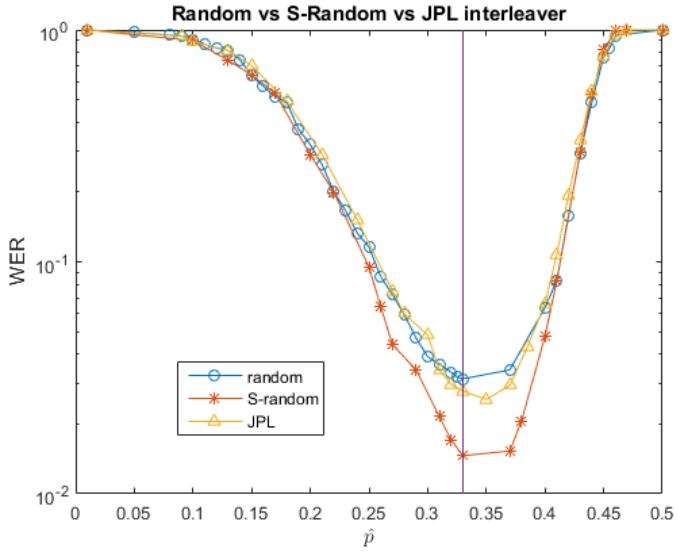


Figure 4.13: Simulations of the random, S-random and JPL interleaver for a depolarizing channel with $p = 0.33$

5 Conclusion

Section 4.2.3 showed that channel mismatch does affect the decoding in quantum turbo codes. That is, when the SISO decoder is fed with a channel information that is not accurate, the decoder of the QTC will present a higher word error rate than in the case where the channel information was accurate. On the one hand, when the estimated error probability $\hat{p}$ of the depolarizing channel is close to the actual error probability p of the channel, the QTC presents a relatively low WER. On the other hand, when the estimated $\hat{p}$ is far from probability p, the WER in QTC increases considerably. Overall, the behavior of quantum turbo codes with channel mismatch is coherent with the behavior of classical turbo codes with channel mismatch.

Section 4.2.3 also discloses the influence of the interleaver choice when there is a channel mismatch. The three interleavers simulated in this project presented a similar WER when the estimated $\hat{p}$ was far away from the actual p. However, when the estimation of $\hat{p}$ was quite close to the actual value of the error probability of the channel, noticeable differences emerged. Some interleavers had a lower WER than others when $\hat{p}$ was close to p. Specifically, when the estimation $\hat{p}$ did not vary much from p, the interleavers with a lower error floor [24] performed better than the other interleavers with a higher error floor.

These results indicate how important it is to feed the SISO decoder of a QTC with the correct channel information. Even when the exact determination of a quantum communication channel is unavailabe, it is crucial to find a moderately accurate estimation of the depolarizing probability in order to get a low WER. The figures in section 4.2.3 demonstrate that the deviation of a few tenths, or even a few hundredths, may results in a huge increase of the WER.

Future work on the study of channel mismatch may touch upon the influence of the interleaver choice on different depolarizing channels. In this document, the performances of the random, S-random and JPL interleaver were compared for a depolarizing channel of $p = 0.33$ only. The three interleavers could be simulated for other values of p too, in order to check if the conclusions deduced in this document can be extracted to other depolarizing channels with a different error probability. Besides, many other interleavers could be simulated as well, such as the *Welch-Costas* interleaver.

6 Project Budget

The budget of the development of this Final Degree Project is detailed below. The whole budget is divided in the following headings:

• Inmobilized material: The sets of products and components bought to third parties and were used during the realization of the project. The budget of the inmobilized material is explained in table 6.1.

• Consumable material: The set of every consumable material consumed during the realization of the project. The budget of the consumable material can be seen in table 6.2.

• Equipment: The costs corresponding to the use of every machine during the development of the project. These costs were calculated by taking into account the average amortization cost of each machine and the amount of time each machine was used. The budget of the equipment is shown in table 6.3.

• Software: The software used for the development of the project. The amortization of the licenses are taken into account. Table 6.4 shows the budget of the software.

• Workforce: The costs corresponding to the human resources involved in every phase of the project. The budget of the workforce is shown in figure 6.5.

Description	Quantity	Unit Price (€)	Total price (€)
Paper sheets	12	0.01	0.12
Total Inmobilized Material			0.12

Table 6.1: Budget of the inmobilized material

Description	Quantity	Unit Price (€)	Total price (€)
Pens	2	2.20	4.40
Pencils	2	0.90	1.80
Eraser	1	1.30	1.30
Total Consumable Material			7.50

Table 6.2: Budget of the consumable material

Equipment	Acquisiton quota (€)	Amortization time (years)	Amortization quota (€/ hour)	Time of use (hours)	Amortization (€)
Laptop	876.00	5	0.02	220	4.40
Desktop computer	4380.00	5	0.10	672	67.20
Total Equipment					71.60

Table 6.3: Budget of the equipment

Software	Acquisiton quota (€)	Amortization time (years)	Amortization quota (€/ hour)	Time of use (hours)	Amortization (€)
Windows 10	139.00	1	0.02	220	4.40
Matlab	800.00	1	0.09	672	60.48
Total Software					64.88

Table 6.4: Budget of the software

Position	Task	Duration (hours)	Fee (€/ hour)	Total cost (€)
Junior engineer	Working on the Final Degree Project	240	10.00	2,400.00
Total Workforce				2,400.00

Table 6.5: Budget of the workforce

Budget Summary: The budget summary of the Final Degree Project is detailed in table 6.6. It can be seen that the whole budget ascends to 3,386.20 €.

Heading	Amount (€)
Inmobilized material	0.12
Consumable material	7.50
Equipment	71.60
Software	64.88
Workforce	2,400.00
Sum	2,544.10
Indirect costs (10%)	254.11
Total without VAT	2,798.51
Total with VAT	3,386.20
TOTAL BUDGET	3,386.20

Table 6.6: Overall budget

References

[1] Preskill, John, "Quantum Computing and the Entanglement Frontier", *arXiv:1203.5813v3*. Nov. 2012.

[2] "Making the World's First Integrated Quantum System", *IBM*, 5th June 2019, <https://www.research.ibm.com/ibm-q/system-one/>

[3] "Intel Drives Development of Quantum Cryoprober with Bluefors and Afore to Accelerate Quantum Computing", *Intel*, 5th June 2019, <https://newsroom.intel.com/news/intel-drives-development-quantum-cryoprober-bluefors-afore-accelerate-quantum-computing/#gs.h10jb2>

[4] "Microsoft Quantum Network", *Microsoft*, 5th June 2019, <https://www.microsoft.com/en-us/quantum/quantum-network>

[5] "NASA Quantum Artificial Intelligence Laboratory (QuAIL)", *National Aeronautics and Space Administration*, 5th June 2019, <https://ti.arc.nasa.gov/tech/dash/groups/physics/quail/>

[6] Yiu, Yuen, "Is China the Leader in Quantum Communications?", *Inside Science*. 19 Jan 2018. 5 June 2019, <https://www.insidescience.org/news/china-leader-quantum-communications>

[7] Shannon, Claude, "A Mathematical Theory of Communication", *The Bell System Technical Journal*. Vol. 27, July 1948: 79-423.

[8] Brun, Todd, and Min-Hsiu Hsieh, "Entanglement-Assisted Quantum Error-Correcting Codes," *arXiv:1610.04013v1.* Oct. 2016.

[9] Nielsen, Michael A., and Isaac L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition*. New York, NY, USA: Cambridge University Press, 10th ed., 2011.

[10] Etxezarreta Martínez, Josu, *Quantum Error Correction: stabilizer coding and beyond*. Graduate thesis. University of Navarra, 2018.

[11] Mastriani, Mario, "Simplified Protocol of Quantum Teleportation," *Journal of Quantum Information Science.* Sept. 2018: 107-120.

[12] Preskill, John, "Lecture Notes for Ph219/CS2019: Quantum Information. Chapter 3". California Institute of Technology. Oct. 2018: 24-25.

[13] Xie, Yixuan, *Quantum Error Correction and Stabilizer Codes*. PhD thesis. University of New South Wales, 2016.

[14] Raussendorf, Robert, "Key ideas in quantum error correction," *Philosophical Transactions of the Royal Society A.* Sept. 2012: 4541-4565.

[15] Albouy, Olivier, "Discrete algebra and geometry applied to the Pauli group and mutually unbiased bases in quantum information theory". Université Claude Bernard - Lyon I. 2009: 35-36.

[16] Brun, Todd, Igor Devetak, and Min-Hsiu Hsieh, "Correcting Quantum Errors with Entanglement," *Science.* Oct. 2006: 436-438.

[17] Gottesman, Daniel, "The Heisenberg Representation of Quantum Computers", *International Colloquium Group THeoretical Methods in Physics*, Cambridge, MA, International Press, July 1998: 32-43.

[18] Brun, Todd, Igor Devetak, and Min-Hsiu Hsied, "General entanglement-assisted quantum error-correcting codes", *Proc. IEEE Int. Symp. Inf. Theory*. Jun. 2007: 2101-2105.

[19] Berrou, Claude, Alain Glavieux, and Punya Thitimajshima, "Near Shannon limi error-correcting coding and decoding: Turbo codes", *IEEE Proceeding of the International Conference on Communications.* May 1993: 1064-1070.

[20] Poulin, David, Jean-Pierre, Tillich, and Harold Ollivier, "Quantum serial turbo-codes", *IEEE Transactions on Information Theory.* Vol.55, Jun. 2009: 2776-2798.

[21] Wilde, Mark M., Min-Hsiu Hsieh, and Zunaira Babar, "Entanglement-assisted quantum turbo codes", *IEEE Transactions of Information Theory.* Vol. 60, Feb. 2014: 1203-1222.

[22] Wilde, Mark M., Min-Hsiu Hsieh, and Zunaira Babar, "EA-Turbo". 12th June 2019, <https://code.google.com/archive/p/ea-turbo/>

[23] Ho, Mark S. C., and Steven S. Pietrobon, "A variance mismatch study for serial concatenated turbo codes", *2nd International Symposium on Turbo Codes & Related Topics*, Sept. 2000: 483-486.

[24] Etxezarreta, Josu, Pedro M. Crespo, and Javier Garcia-Frías, *On the performance of interleavers for Quantum Turbo Codes*. May 2019.