UNIVERSITÀ DEGLI STUDI DI PADOVA

FACOLTÀ DI INGEGNERIA

Corso di Laurea in Ingegneria Meccanica e Meccatronica

Tesi di Laurea Triennale

DISCRETE - TIME FEEDBACK CONTROL STRATEGIES FOR QUANTUM STATES PREPARATION

Laureando

GIACOMO BAGGIO

Matricola

601802

Relatore

PROFESSOR FRANCESCO TICOZZI

A.A. 2010/2011

Abstract

Preparation of quantum states in physical systems, and in particular entangled states, is a central task in many experimental settings, including control of molecular dynamics, cooling of nano-mechanical resonators, and the full spectrum of quantum information processing applications. In this work we focus on entangled state preparation for multiple quantum bits. We consider control capabilities and strategies that are readily available for trapped-ion systems, one of the most promising physical support for quantum computing.

After a brief introduction on quantum information theory and its applications, in Chapter 2 we discuss some mathematical preliminaries about Quantum Mechanics. Chapter 3 is a short presentation of recent results on discrete-time feedback control of quantum systems provided in [1, 4]. In Chapter 4, the core of the work, we analyze the solution proposed by J. T. Barreiro *et al.* in [2] concerning Bell states and GHZ state preparation and, exploiting the tools described in Chapter 3, we study an alternative solution based on a elementary quantum feedback control. Most interestingly, we show that the proposed approach can offer significant advantages in terms of simplicity of the circuit and total implementation time if fast measurement are employed.

Sommario

La preparazione di stati quantistici in sistemi fisici, ed in particolare la preparazione di stati entangled, è una tematica di ricerca centrale in molti contesti sperimentali, riguardanti, ad esempio, il controllo della dinamica molecolare, la realizzazione di risonatori nano-meccanici, e l'intero spettro di applicazioni concernenti l'elaborazione di informazione di natura quantistica. Questo lavoro tratta in particolare della preparazione di stati entangled per più bit quantistici. Verranno prese in considerazione capacità e strategie di controllo che sono attualmente disponibili nella pratica in architetture ion-traps, una delle più promettenti nel campo della computazione quantistica.

Dopo una breve introduzione sulla teoria dell'informazione quantistica e su alcune delle sue applicazioni, nel capitolo 2 verranno discussi alcuni preliminari matematici di Meccanica Quantistica. Il capitolo 3 consiste in una breve presentazione dei recenti risultati forniti negli articoli [1, 4] riguardanti il controllo feedback a tempo discreto di sistemi quantistici. Nel capitolo 4, il corpo centrale del lavoro, verrà analizzata la strategia proposta da J. T. Barreiro *et al.* in [2], adottata per la preparazione dello stato di Bell $|\Psi^-\rangle$ e dello stato entangled GHZ a 4 qubit. Quindi, sfruttando gli strumenti descritti nel Capitolo 3, verrà studiata una soluzione alternativa basata su una strategia di controllo a feedback. Tale approccio è in grado di offrire significativi vantaggi in termini di semplicità del circuito e di tempi di implementazione totale se si utilizzano operazioni di misura abbastanza rapide.

Contents

1

Introduction and Overview

In the beginning of the twentieth century two new revolutionary theories shook the well-established field of physics: Einstein's theory of relativity and quantum mechanics. Nowadays these theories have found many practical applications even in the everyday life, for example to achieve accuracy requirements the GPS (Global Positioning System) uses principles of Einstein's general theory of relativity and the operation of the transistor, the fundamental device of modern electronics invented by Bardeen, Brattain and Shockley in 1948, can be described by laws of quantum mechanics.

In the field of computation and information theory is taking place a similar revolution. In particular, in the near future, quantum laws will become fundamental for computation because of technological miniaturization. The electronic industry for computers grows in accordance with the decrease in size of integrated circuits. This miniaturization is necessary to increase computational power and memory. In 1965 Gordon Moore observed that the number of transistors that may be placed on a single integrated-circuit chip doubles approximately every 18-24 months. This observation has been verified in practice and became a law: Moore's law. The exponential growth, predicted by Moore, has not yet saturated and is still valid. At the present time the limit is approximately 10^8 transistors per chip and the typical size of circuit components is of the order of 100 nanometres. Extrapolating Moore's law, one would estimate that around the year 2020 it will be reached the atomic size for storing a single bit of information. At that point, quantum effects will become unavoidably relevant. For these and other reasons the study and the control of quantum systems as a means for exchange of information has become a field of ongoing research.

1.1 Quantum Information Theory

Quantum information Theory (QIT) is concerned with using the special features of quantum physics for the processing and trasmission of information. QIT is closely linked to Quantum Computation science and provides in particular the theoretical tools needed for analyzing and addressing challenging tasks such as preserving and storing this new kind of information, two fundamental problems in the physical realization of a quantum computer.

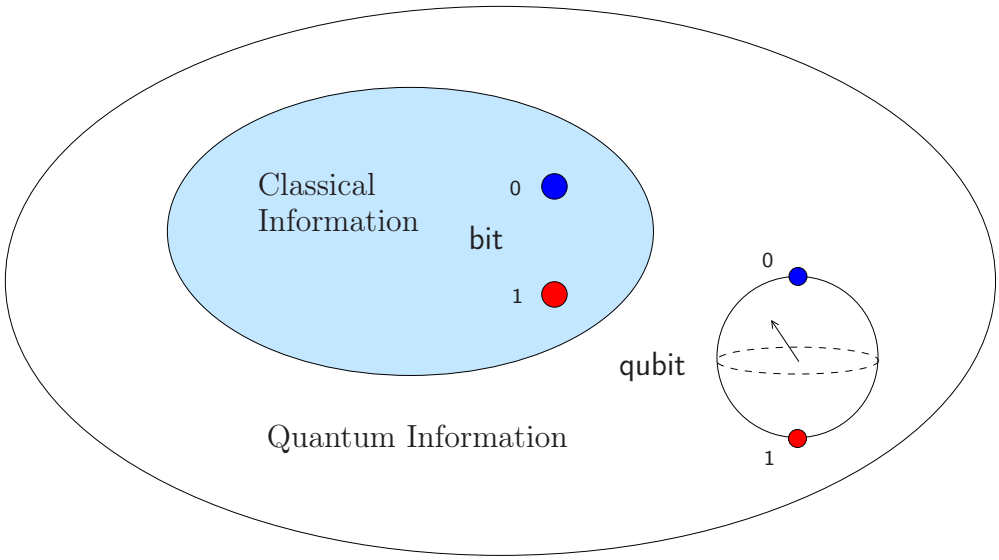


Figure 1.1: When information is represented as a quantum state (qubit) rather than in terms of classical bits, quantum information theory is described as being generalization or extension of classical information theory. The well-established theory of classical information and computation is thus a subset of a much larger field, the emerging theory of quantum information and computation

The pioneer of this new research field can be considered the Nobel prize-winning physicist Richard Feynman. Indeed in 1982 he thought up the idea of a quantum computer, a computer that uses the effects of quantum mechanics to its advantage. For some time, the notion of a quantum computer was primarily of theoretical interest only, but recent developments have brought the idea to everybody's attention. One such development is related to the field of cryptanalysis and was the invention of an algorithm to factor large numbers on a quantum computer, the well-known *Shor's*

algorithm, proposed in 1994 by Peter Shor. By using this algorithm, a quantum computer would be able to crack codes much more quickly than any classical computer could. In fact a quantum computer capable of performing Shor's algorithm would be able to break current cryptography techniques (based on RSA algorithm[1]) in a matter of seconds. With the motivation provided by this algorithm, the topic of quantum computing has gathered momentum, supported also by many national government and military funding agencies, and researchers around the world are racing to be the first to create a practical quantum computer.

A large number of different proposal to build experimental quantum computers have been put forward. They range from NMR (nuclear magnetic resonance) quantum processor to cold ion traps (see also Appendix B), superconducting tunnel-junction circuits and spin in semiconductors, to name but a few. Even though in some cases elementary quantum gates have been realized (e.g. Cirac-Zoller CNOT gate and Mølmer-Sørensen gate in ion traps quantum computing) and quantum algorithms with small number of qubits demonstrated, it is too early to say what type of implementation will be the most suitable to build a scalable piece of quantum hardware.

A fundamental obstacle to the practical realization of a quantum computer is *decoherence*, i.e. the decay of the quantum information stored in a quantum computer, due to the inevitable interaction of the quantum computer with the environment.

The unit of quantum information is known as a *qubit* (the quantum counterpart of the classical bit) and a quantum computer may be viewed as a many-qubit system. The fundamental difference between a bit and a qubit is that the first one takes the value 0 or 1 insted a qubit can also take all intermediate values between these two. This property is known as *superposition principle* of quantum states. The two states in which a qubit may be measured are known as basis states. Dirac, or bra-ket notation, is used to represent them (see section §2.1). This means that the two computational basis states are conventionally written as $|0\rangle$ and $|1\rangle$. As noted earlier a pure qubit state is a linear superposition of the basis states. This means that the qubit can be represented as a linear combination of

[1]In cryptography, RSA (which stands for Rivest, Shamir and Adleman who first publicly described it) is an algorithm for public-key cryptography. It is the first algorithm known to be suitable for signing as well as encryption, and was one of the first great advances in public key cryptography. RSA is widely used in electronic commerce protocols, and is believed to be sufficiently secure given sufficiently long keys and the use of up-to-date implementations.

and $|0\rangle$ and $|1\rangle$:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle,$$

where α and β are probability amplitudes and can in general both be complex numbers. When we measure this qubit in the standard basis, the probability of outcome $|0\rangle$ is $|\alpha|^2$ and the probability of outcome $|1\rangle$ is $|\beta|^2$. Because the absolute squares of the amplitudes equate to probabilities, it follows that α and β must satisfy the equation:

$$|\alpha|^2 + |\beta|^2 = 1.$$

The possible states for a single qubit can be visualised using *Bloch sphere* (Fig. 1.2). Represented on such a sphere, a classical bit could only be in the locations where $|0\rangle$ and $|1\rangle$ are respectively. The rest of the surface of the sphere is inaccessible to a classical bit, but a pure qubit state can be represented by any point on the surface.

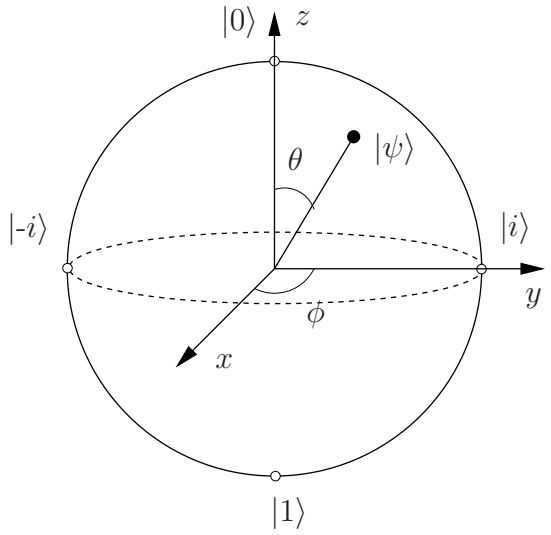


Figure 1.2: Bloch sphere representation of a qubit: probability amplitudes are given by $\alpha = \cos(\theta/2)$ and $\beta = \sin(\phi/2)$

1.2 The Importance of Entanglement

The second fundamental property, after superposition, on which quantum computing is based is called *entanglement*. Using the words of one of the father of quantum mechanics, Erwin Schrödinger:

> "Entanglement is not *one* but rather *the* characteristic trait of quantum mechanics".

Entanglement is the most spectacular and counter-intuitive manifestation of quantum mechanics, observed in composite quantum systems: it signifies the existence of non-local correlations, which Einstein called "spooky action at a distance", between measurements performed on well-separated particles. After two classical systems have interacted, they are in well-defined states. In contrast, after two quantum particle have interacted, in general, they can no longer be described independently of each other. There will be quantum correlations between two such particles, independently of their spacial separation. This is the content of the famous EPR paradox, a thought experiment proposed by Einstein, Podolsky and Rosen in 1935 that shook the not yet stable foundations of newborn quantum mechanics. In this famous paper ([9]) the authors showed that quantum theory leads to a contradiction, provided that we accept the two, seemingly natural, principles of realism and locality. In 1964 John Bell proved that the local realism lead to predictions, Bell's inequalities, that are in contrast with quantum theory. Aspect's experiment (1982), performed with pairs of entangled photons, exhibited an evident violation of Bell's inequality and an impressive agreement with quantum mechanics. More recently, other experiments have come closer to the requirements of the ideal EPR scheme. These results show that entanglement is a fundamentally new resource, beyond the realm of classical physics, and that it is possible to experimentally manipulate entangled states.

Quantum entanglement is central to many quantum-communication protocols. Of particular importance are *quantum dense coding*, which permits transmission of two bits of classical information through the manipulation of only one of the two entangled qubits, and *quantum teleportation*, which allows the transfer of the state of one quantum system to another over an arbitrary distance. Moreover another application of entanglement can be found in *quantum cryptography* where entangled particles are used to transmit signals that cannot be eavesdropped upon without leaving a trace.

2 Quantum Mechanics

In this chapter we describe the principal features and notions of Quantum Mechanics. In the final section we present the formalism of quantum circuits and the main quantum gates that will be used in the work. For a detailed intoduction on Quantum Mechanics we refer the reader to specific literature, as for example [25, 7, 21] to name just a few.

2.1 States and Operators

2.1.1 Hilbert space

Postulate 1. The state of a quantum mechanical system is represented by a vector in a separable Hilbert space.

Definition 2.1 (Hilbert space)**.** Hilbert space $\mathcal{H}$ is a complex vector space with an inner product that is an operation $(\cdot,\cdot) : \mathcal{H} \times \mathcal{H} \to \mathbb{C}$, which satisfies the following properties:

1. $$(\vec{x},\vec{y}) = (\vec{y},\vec{x})^*, \tag{2.1}$$
   where a^* denotes the complex conjugated of a,
2. $$(\vec{x}+\vec{y},\vec{z}) = (\vec{x},\vec{z}) + (\vec{y},\vec{z}), \tag{2.2}$$
3. $$(\alpha\vec{x},\vec{y}) = \alpha^*(\vec{x},\vec{y}), \tag{2.3}$$
   for any complex scalar α, and,

4.
$$(\vec{x}, \vec{x}) \geq 0, \tag{2.4}$$
where equality holds if and only if $\vec{x} = 0$.

The *norm* of a vector $\vec{x}$, $\|\vec{x}\|$, is defined as $\|\vec{x}\| := \sqrt{(\vec{x}, \vec{x})}$. The *metric* of $\mathcal{H}$ is defined using this norm, the distance between the vectors $\vec{x}$ and $\vec{y}$ being given by $\|\vec{x} - \vec{y}\|$. A Hilbert space is required to be *complete*[1] with respect to this metric.

Definition 2.2 (Separable Hilbert space)**.** A Hilbert space is called *separable* if there exists a dense countable set $\{\vec{e_j}\}$, with j in a set J which forms an orthonormal basis, i.e., it is such that[2]:
$$(\vec{e_j}, \vec{e_k}) = \delta_{jk}, \tag{2.5}$$
and every vector $\vec{x} \in \mathcal{H}$ can be written in a unique way as:
$$\vec{x} = \sum_{j \in J} \vec{e_j}(\vec{e_j}, \vec{x}), \tag{2.6}$$

2.1.2 Dirac bra-ket notation

In quantum mechanics, the state vectors are denoted using Dirac's notation, that is, the state ψ is denoted as $|\psi\rangle$ and called *ket*. This state ket is postulated to contain complete information about the *physical state*. Corresponding to every ket $|\psi\rangle$ there exist a *bra*, denoted by $\langle\psi|$ contained in a vector space (bra space) "dual to" the ket space. The bra dual to $\alpha|\psi\rangle$, with α a complex scalar, is $\alpha^*\langle\psi|$.

The inner product two kets $|\phi\rangle$ and $|\psi\rangle$ is denoted by $\langle\phi|\psi\rangle$. Using this notation the property (2.1) can be replaced by:
$$\langle\phi|\psi\rangle = \langle\psi|\phi\rangle^*. \tag{2.7}$$

Kets $|\psi\rangle$ and $\alpha|\psi\rangle$, for any $\alpha \in \mathbb{C}, \alpha \neq 0$, represent the same physical state. For this reason, it is often more appropriate to speak about *rays* or

[1] In mathematical analysis, a metric space $\mathcal{M}$ is called complete (or Cauchy) if every Cauchy sequence of points in $\mathcal{M}$ has a limit that is also in $\mathcal{M}$ or, alternatively, if every Cauchy sequence in $\mathcal{M}$ converges in $\mathcal{M}$.

[2] Symbol δ_{jk} denotes Kronecker's delta function defined as:
$$\delta_{jk} = \begin{cases} 1 & j = k \\ 0 & j \neq k \end{cases}$$

directions in the Hilbert space $\mathcal{H}$ as representing the state of a quantum system. It will be often assumed that the vector representing the state of the system has norm equal to one, i.e., for every ket $|\psi\rangle$, $\langle\psi|\psi\rangle = \||\psi\rangle\|^2 = 1$. Physical states can be viewed as points on a complex sphere with radius one in a Hilbert space, with points differing by a phase factor treated as the same state.

2.1.3 Linear operators

Definition 2.3 (Linear operator)**.** Given two Hilbert spaces $\mathcal{H}_1$ and $\mathcal{H}_2$ a linear operator A , $A : \mathcal{H}_1 \to \mathcal{H}_2$, is a linear map from $\mathcal{H}_1$ to $\mathcal{H}_2$. It is called bounded if $\|A\| := \sup\{\|A\|\vec{x} : \|\vec{x}\| = 1\}$ is finite. If $\mathcal{H}_1$ and $\mathcal{H}_2$ are both finite dimensional every linear operator A is bounded. From now on we denote the set of bounded operators acting on the Hilbert space $\mathcal{H}$ by the symbol $\mathfrak{M}(\mathcal{H})$.

The bra $\langle\psi| : \mathcal{H} \to \mathbb{C}$ associate to the ket $|\psi\rangle$ is an example of a linear operator which, when applied to the vector $|\phi\rangle$, gives the number $\langle\psi|\phi\rangle$.

Another important linear operator is the *outer product* of a bra $\langle\phi|$ and a ket $|\psi\rangle$, which is defined as $|\psi\rangle\langle\phi| : \mathcal{H} \to \mathcal{H}$. It maps the ket $|\vartheta\rangle$ to the ket $|\psi\rangle\langle\phi|\vartheta\rangle$, i.e. the ket $|\psi\rangle$ multiplied by the scalar $\langle\phi|\vartheta\rangle$.

We list here some basic properties concerning composition of linear operators:

1. Given two (or more) linear operators A and B in $\mathfrak{M}(\mathcal{H})$ we can construct a new sum operator $A + B \in \mathfrak{M}(\mathcal{H})$ which acts on a ket $|\psi\rangle$ as:
$$(A + B)|\psi\rangle = A|\psi\rangle + B|\psi\rangle. \tag{2.8}$$

2. If α is a scalar, $\alpha \in \mathbb{C}$, and $B \in \mathfrak{M}(\mathcal{H})$, the operator $\alpha B \in \mathfrak{M}(\mathcal{H})$ is defined as:
$$(\alpha B)|\psi\rangle = \alpha(B|\psi\rangle). \tag{2.9}$$

3. If A and B are two linear operators in $\mathfrak{M}(\mathcal{H})$, the linear operator $AB \in \mathfrak{M}(\mathcal{H})$, i.e. the *product* of operators A and B, is defined as:
$$AB|\psi\rangle = A(B|\psi\rangle). \tag{2.10}$$

Definition 2.4 (Adjoint of a linear operator)**.** Let $A \in \mathfrak{M}(\mathcal{H})$, there exists a unique linear operator, denoted by $A^\dagger$, acting on $\mathcal{H}$, such that for every $|\psi\rangle$ and $|\phi\rangle$:
$$\langle A\psi|\phi\rangle = \langle\psi|A^\dagger|\phi\rangle. \tag{2.11}$$
$A^\dagger$ is called the adjoint of A .

Definition 2.5 (Hermitian operator)**.** A linear operator $A \in \mathfrak{M}(\mathcal{H})$ satisfying the relation $A = A^\dagger$ is called *Hermitian operator*. We use the symbol $\mathfrak{H}(\mathcal{H})$ to denote the set of Hermitian operators in $\mathfrak{M}(\mathcal{H})$.

Definition 2.6 (Unitary operator)**.** A *unitary operator* U is defined by the relation $U^\dagger U = I$, where I denotes identity operator. This implies that, for every state $|\psi\rangle$,

$$\langle U\psi|U|\psi\rangle = \langle\psi|U^\dagger U\psi\rangle = \langle\psi|\psi\rangle = I. \tag{2.12}$$

We use notation $\mathfrak{U}(\mathcal{H})$ to define the set of unitary operators in $\mathfrak{M}(\mathcal{H})$.

2.1.4 State of composite system and tensor product

Postulate 2. Consider two systems Σ_1 and Σ_2 with states represented by vectors in the Hilbert spaces $\mathcal{H}_1$ and $\mathcal{H}_2$, respectively. The state of the composite system (Σ_1 together with Σ_2) is represented by a vector in a Hilbert space which is the *tensor product* of $\mathcal{H}_1$ and $\mathcal{H}_2$, denoted by $\mathcal{H}_1 \otimes \mathcal{H}_2$.

In particular suppose $\mathcal{H}_1$ and $\mathcal{H}_2$ are Hilbert spaces of finite dimension m and n respectively. The their tensor product $\mathcal{H}_1 \otimes \mathcal{H}_2$ is a mn dimensional vector space which satisfies the following properties:

1. For an arbitrary scalar α and elements $|\psi\rangle$ of $\mathcal{H}_1$ and $|\phi\rangle$ of $\mathcal{H}_2$,

$$\alpha(|\psi\rangle \otimes |\phi\rangle) = (\alpha|\psi\rangle) \otimes |\phi\rangle. \tag{2.13}$$

2. For arbitrary $|\psi_1\rangle$ and $|\psi_2\rangle$ in $\mathcal{H}_1$ and $|\phi\rangle$ in $\mathcal{H}_2$,

$$(|\psi_1\rangle + |\psi_2\rangle) \otimes |\phi\rangle = |\psi_1\rangle \otimes |\phi\rangle + |\psi_2\rangle \otimes |\phi\rangle. \tag{2.14}$$

3. For arbitrary $|\psi\rangle$ in $\mathcal{H}_1$ and $|\phi_1\rangle$ and $|\phi_2\rangle$ in $\mathcal{H}_2$:

$$|\psi\rangle \otimes (|\phi_1\rangle + |\phi_2\rangle) = |\psi\rangle \otimes |\phi_1\rangle + |\psi\rangle \otimes |\phi_2\rangle. \tag{2.15}$$

A similiar reasoning can be extended to operators. Suppose $|\psi\rangle$ and $|\phi\rangle$ are vectors in $\mathcal{H}_1$ and $\mathcal{H}_2$, and $A \in \mathfrak{M}(\mathcal{H}_1)$, $B \in \mathfrak{M}(\mathcal{H}_2)$. Then we can define a linear operator $A \otimes B \in \mathfrak{M}(\mathcal{H}_1 \otimes \mathcal{H}_2)$ by the equation:

$$(A \otimes B)(|\psi\rangle \otimes |\phi\rangle) \equiv A|\psi\rangle + B|\phi\rangle. \tag{2.16}$$

In a matrix representation of operators A and B this rather abstract operation is known as the *Kronecker product*. Suppose A is a $m \times n$ matrix and B a $p \times q$ one. Then $A \otimes B$ can be written as:

$$A \otimes B = \overbrace{\left.\begin{bmatrix} A_{11}B & A_{12}B & \cdots & A_{1n}B \\ A_{21}B & A_{22}B & \cdots & A_{2n}B \\ \vdots & \vdots & \vdots & \vdots \\ A_{m1}B & A_{m2}B & \cdots & A_{mn}B \end{bmatrix}\right\} m \times p.}^{n \times q} \tag{2.17}$$

2.1.5 The density operator

The density operator provides a convenient means for describing a ensemble of quantum states, i.e. a large number of identical systems in different states. More precisely, suppose a quantum system is in one of states $|\psi_i\rangle$ with respective probabilities p_i. We call $\{p_i, |\psi_i\rangle\}$ an ensemble of pure states. The density operator for the system is defined as:

$$\rho \equiv \sum_i p_i |\psi_i\rangle\langle\psi_i|. \tag{2.18}$$

The density operator, also known as the *density matrix*, is a linear operator $\mathcal{H} \to \mathcal{H}$ as it is a linear combination of outer products. It completely describes the state of the ensemble. Special cases are *pure ensembles* or *pure states* which are such that $p_j = 1$, for some index j . They are described by density operators of the form:

$$\rho = |\psi_j\rangle\langle\psi_j|. \tag{2.19}$$

On the contrary, *mixed ensembles* (or *mixed states*) are described by density matrices with more than one p_i different from zero.

Below we list the fundamental properties characterizing density operator which can be proved directly using the definition (2.18):

1. ρ is Hermitian and positive semidefinite[3].

2. The trace[4] of ρ must be unitary:

$$\mathrm{tr}(\rho) = 1. \tag{2.20}$$

[3]A Hermitian operator $A : \mathcal{H} \to \mathcal{H}$ is called positive semidefinite if $\langle\psi|A|\psi\rangle \geq 0$, for every $|\psi\rangle \in \mathcal{H}$.

[4]The trace of a square matrix A is defined as the sum of the elements on the main

3.
$$\rho^2 = \rho \tag{2.21}$$
if and only if ρ represents a pure ensemble.

4.
$$0 < \mathrm{tr}(\rho^2) < 1. \tag{2.22}$$
if ρ represents a mixed ensemble.

Before proceeding we introduce some useful nomenclature. We indicate with symbol $\mathfrak{D}(\mathcal{H})$ the set of density operator acting on Hilbert space $\mathcal{H}$. The subset $\mathfrak{P}(\mathcal{H}) \subset \mathfrak{D}(\mathcal{H})$ denotes the set of pure ensembles or states. Note that $\mathfrak{D}(\mathcal{H})$ is a convex set, whose extreme points are the pure states $\mathfrak{P}(\mathcal{H})$ and its border $\delta\mathfrak{D}(\mathcal{H})$ contains all the states that are not full rank ([1]).

2.1.6 The reduced density operator

One of the most important uses for the density operator formulation is as a tool for describing the state of a *subsystem* of a composite system. Consider a two-qubit pure state $|\psi_{12}\rangle \in \mathcal{H}_1 \otimes \mathcal{H}_2$. The general state of such a system may be entangled, and so it may not be possible to factor out the state vector $|\psi_1\rangle \in \mathcal{H}_1$ for the state of the first qubit. However, the state of the first qubit can in general be described as a mixed state. This means that it can be described by a density operator $\rho_1 \in \mathfrak{D}(\mathcal{H}_1)$, called *reduced density operator*. The mathematical operation for calculating the reduced density operator is the *partial trace*.

Definition 2.7 (Partial trace)**.** Let $|\psi\rangle$ and $|\phi\rangle$ two generic states in $\mathcal{H}_1$ and $\mathcal{H}_2$ respectively. Now consider the composite system with Hilbert space $\mathcal{H}_1 \otimes \mathcal{H}_2$. Then the partial trace over system $\mathcal{H}_2$ can be defined for elementary operators as:

$$\mathrm{tr}_{\mathcal{H}_2} \underbrace{(|\psi_1\rangle\langle\psi_2| \otimes |\phi_1\rangle\langle\phi_2|)}_{\text{operator in } \mathcal{H}_1 \otimes \mathcal{H}_2} = |\psi_1\rangle\langle\psi_2|\mathrm{tr}(|\phi_1\rangle\langle\phi_2|) = \underbrace{|\psi_1\rangle\langle\psi_2|}_{\text{operator in } \mathcal{H}_1} \underbrace{(\langle\phi_1|\phi_2\rangle)}_{\in \mathbb{C}}. \tag{2.23}$$

diagonal of A. Hence for a bounded linear operator A we can write:

$$\mathrm{tr}(A) = \sum_i \langle i|A|i\rangle,$$

where kets $\{|i\rangle\}$ are an orthonormal basis of finite-dimensional Hilbert space $\mathcal{H}$ in which A acts.

Since elementary operators contain a basis for the whole space of operators, the definition can be extended by linearity.

The operation of computing the partial trace over the i-ith system is sometimes referred to as *tracing-out system i*.

Now we have the tools to properly define the reduced density operator.

Definition 2.8 (Reduced density operator)**.** Suppose we have a composite system with Hilbert space $\mathcal{H}_1 \otimes \mathcal{H}_2$ and let it be described by a density operator $\rho \in \mathfrak{D}(\mathcal{H}_1 \otimes \mathcal{H}_2)$. The reduced density operator of $\mathcal{H}_1$ (resp. $\mathcal{H}_2$) is given by:

$$\rho_1 = \mathrm{tr}_{\mathcal{H}_2}\rho, \qquad \rho_2 = \mathrm{tr}_{\mathcal{H}_1}\rho. \tag{2.24}$$

2.2 Observables and Measurement

2.2.1 Observables and spectral theorem

In general an *observable* is any dynamical variable that can be measured. In quantum mechanics, observables are associated with Hermitian operators on the Hilbert space $\mathcal{H}$, i.e. with the set $\mathfrak{H}(\mathcal{H})$.

For the characterization of an observable we must recall a fundamental result in linear algebra: the *spectral theorem* (reported here for a finite-dimensional case).

Theorem 2.1 (Spectral theorem)**.** *Let V be an n-dimensional inner product space (real or complex) with the standard Hermitian inner product. Let A be an $n \times n$ Hermitian matrix. Then there exists an orthonormal basis of V consisting of eigenvectors of A and all corresponding eigenvalues are real.*

Proof. See, for example, [12, Section 9.5 - Theorem 9].

Hence a finite-dimensional observable $\mathcal{O} \in \mathfrak{H}(\mathcal{H})$ has a spectral decomposition:

$$\mathcal{O} = \sum_m \lambda_m \Pi_m, \tag{2.25}$$

where Π_m are the eigenvectors of observable $\mathcal{O}$ and λ_m the corresponding real eigenvalues. The measurement described by such an operator is called *projective or von Neumann's* measurement and the eigenvectors Π_m *projectors* or *projections* onto the eigenspace of $\mathcal{O}$ with eigenvalues λ_m. So the possible outcomes of the measurement corresponds to eigenvalues of the observable.

2.2.2 The measurement postulate

Now we are ready to introduce a fundamental postulate of quantum mechanics, the *meaurement postulate*, a generalization of what we describe previously:

Postulate 3. Quantum measurements are described by a collection $\{M_m\}$ of *measurement operators* in $\mathfrak{M}(\mathcal{H})$. These are operators acting on the state space of the system being measured. The measurement operators satisfy the completeness equation, i.e.,

$$\sum_m M_m^\dagger M_m = I, \tag{2.26}$$

the index m in refers to the measurement outcome that may occur in the experiment. If the state of the system is $|\psi\rangle$ immediately before the measurement then the probability that result m occurs is given by

$$p(m) = \langle\psi|M_m^\dagger M_m|\psi\rangle, \tag{2.27}$$

and the state of the system right after the measurement is

$$\frac{M_m|\psi\rangle}{\sqrt{\langle\psi|M_m^\dagger M_m|\psi\rangle}}. \tag{2.28}$$

This postulate can be reformulated in the density operator picture by replacing the probability of outcome (2.27) with:

$$p(m) = \mathrm{tr}(M_m^\dagger M_m \rho), \tag{2.29}$$

and the corresponding state collapse equation (2.28) with:

$$\frac{M_m \rho M_m^\dagger}{\mathrm{tr}(M_m^\dagger M_m \rho)}. \tag{2.30}$$

2.3 Dynamics of Quantum Systems

2.3.1 Closed quantum systems

Postulate 4 (Continuous-time evolution)**.** The continuous time evolution of the state of a closed quantum system is described by the *Schrödinger equation*,

$$i\hbar\frac{d|\psi\rangle}{dt} = H|\psi\rangle, \tag{2.31}$$

where $\hbar$ is a the *Planck's constant* (in practice, it is common to absorb this factor into H setting $\hbar = 1$) and $H \in \mathfrak{H}(\mathcal{H})$ is a fixed operator known as the *Hamiltonian* of the closed system. Just as the Schrödinger equation describes how pure states evolve in time, the *Liouville-von Neumann equation* describes how a density operator evolves in time:

$$i\hbar\frac{d\rho}{dt} = [H, \rho], \tag{2.32}$$

where the brackets denote a *commutator*[5].

Since in the following chapters we we will restrict our analysis only to the discrete-time case it is worth presenting the evolution postulate for the discrete-time case.

Postulate 4′ (Discrete-time evolution)**.** The discrete time evolution of a closed quantum systems, i.e. a system that it is not interacting in any way with other systems such the environment, is described by an unitary linear transformation (2.12). That is the state $|\psi\rangle$ at the time t_1 is related to the state $|\psi'\rangle$ of the system at time t_2 by a unitary operator $U \in \mathfrak{U}(\mathcal{H})$ and this operator depends only on the times t_1 and t_2:

$$|\psi'\rangle = U|\psi\rangle \tag{2.33}$$

Similiarly in the density operator picture the state ρ at time t_1 is related to the state ρ' of the system at time t_2 by a unitary transformation:

$$\rho' = U\rho U^\dagger \tag{2.34}$$

The unitarity of discrete-time evolution can be demonstrated through integration of the Schrödinger equation (2.31) in the time interval $[t_1, t_2]$, which gives:

$$|\psi(t_2)\rangle = \exp\left(\frac{-iH(t_2 - t_1)}{\hbar}\right)|\psi(t_1)\rangle = U(t_1, t_2)|\psi(t_1)\rangle, \tag{2.35}$$

where $U(t_1, t_2)$ is defined as follows:

$$U(t_1, t_2) = \exp\left(\frac{-iH(t_2 - t_1)}{\hbar}\right). \tag{2.36}$$

[5]The commutator between two operators A and B in $\mathfrak{M}(\mathcal{H})$ is defined to be:

$$[A, B] = AB - BA.$$

If it equals zero, that is $AB = BA$, then we say A commutes with B.

This operator can be easily proven to be unitary. In fact, since $H \in \mathfrak{H}(\mathcal{H})$, i.e. $H^\dagger = H$, it satisfies the definition 2.6 characterizing a unitary operator:

$$U(t_1, t_2)^\dagger U(t_1, t_2) = \exp\left(\frac{iH(t_2 - t_1)}{\hbar}\right) \exp\left(\frac{-iH(t_2 - t_1)}{\hbar}\right) = I. \quad (2.37)$$

Some important properties of a unitary transformation $U \in \mathfrak{U}(\mathcal{H})$ which we have not presented yet are:

1. The rows of U form an orthonormal basis.
2. The colums of U form an orthonormal basis.
3. U preserves inner products, i.e. $\langle\phi|\psi\rangle = \langle U\phi|U\psi\rangle$. Indeed, $(U|\phi\rangle)^\dagger U|\psi\rangle = \langle\phi|U^\dagger U|\psi\rangle = \langle\phi|\psi\rangle$. Therefore, U can be seen as a rotation in the Hilbert space which preserves norms and angles (up to sign).
4. The eigenvalues of U are all of the form $e^{i\theta}$ (since U is norm-preserving, i.e., $\langle\psi|\psi\rangle = \langle U\psi, U\psi\rangle$).
5. U can be diagonalized into the form:

$$\begin{bmatrix} e^{i\theta_1} & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & e^{i\theta_n} \end{bmatrix} \quad (2.38)$$

2.3.2 Open quantum systems

In real-world applications all systems interact at least weakly with other systems, so unitary evolutions are in fact difficult to obtain in practice. Nevertheless every open system can be described as part of a larger closed quantum system which evolves unitarily.

Hence the full discrete-time dynamics of an open quantum system S coupled to an environment or bath E is described by the unitary transformation:

$$\rho_{SE} \mapsto U\rho_{SE}U^\dagger, \quad (2.39)$$

with ρ_{SE} the joint density matrix of the composite system $S + E$.

To obtain the reduced state of the system S alone we perform a partial trace (definition 2.7) over the environment E (see Fig. 2.1 for an intuitive representation of this operation). Thus the discrete-time dynamics of the system, ρ_S, will evolve as

$$\rho_S \mapsto \mathrm{tr}_E(U\rho_{SE}U^\dagger). \quad (2.40)$$

The resulting time evolution for the reduced system is then associated to a completely positive trace preserving (CPTP) *Kraus map*[6]:

$$\rho_S \mapsto \mathcal{E}(\rho_S) = \sum_k E_k \rho_S E_k^\dagger, \tag{2.41}$$

with $\{E_k\}$ operation elements satisfying *completeness relation* $\sum_k E_k^\dagger E_k = I$.

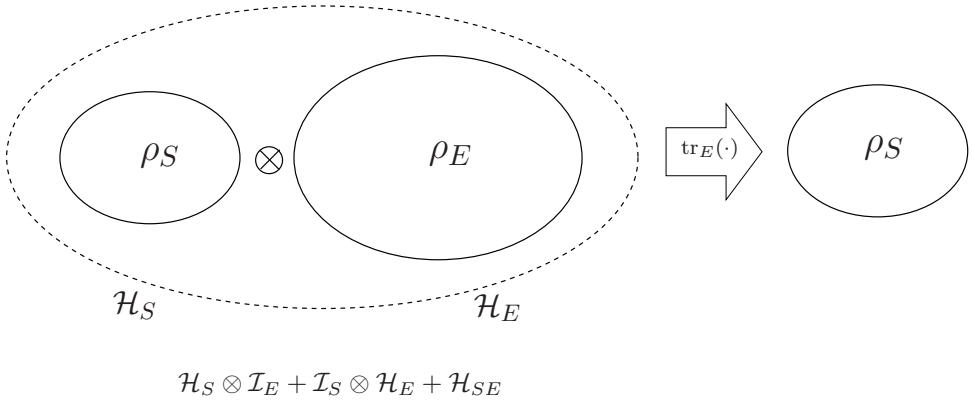


Figure 2.1: Reduced density operator, $\rho_S = \mathrm{tr}_E(\rho_{SE})$.

2.4 Quantum Circuits Formalism

Here we describe the main quantum gates acting on one or more qubits. Some gates have their classical counterpart, e.g. the X gate corresponds to the NOT classical logic gate, others, instead, are completely new, such as the Hadamard gate. Moreover, in this section, we introduce some useful circuit notations that we will use in the rest of the work. For further information on quantum computation theory and quantum circuit theory see [20, 3, 13, 22].

2.4.1 Single qubit gates

Operations on a single qubit are described by 2-dimensional unitary matrices. Unitarity is the only constraint that a quantum gate must satisfy. The *Pauli gates* are some of the most important single qubit gates and for

[6]Kraus representation is also known as *Operator-Sum Representation* (OSR).

this importance they will be widely used in the following sections. We list them below:

$$\begin{aligned} \sigma_0 \equiv I \equiv \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad & \sigma_1 \equiv \sigma_x \equiv X \equiv \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \\ \sigma_2 \equiv \sigma_y \equiv Y \equiv \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \qquad & \sigma_3 \equiv \sigma_z \equiv Z \equiv \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \end{aligned} \tag{2.42}$$

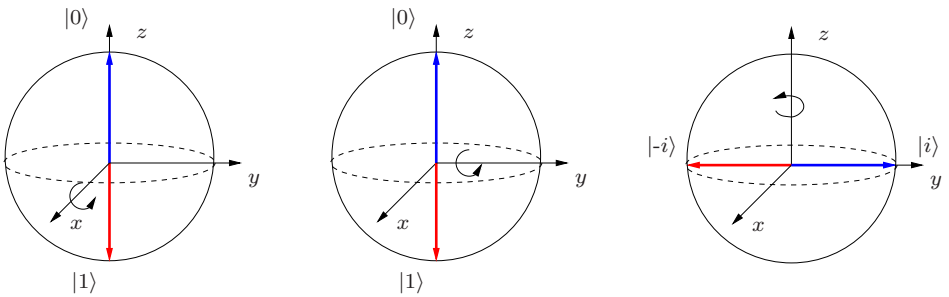
Fig. 2.2 shows on Bloch sphere some examples of action of Pauli gates.

Pauli X,Y and Z gates give rise to three useful classes of unitary matrices when they are exponentiated, the *rotation gates*, about the x, y and z axis, defined as:

$$R_x(\theta) \equiv e^{-i\theta X/2} = \cos\frac{\theta}{2} I - i \sin\frac{\theta}{2} X = \begin{bmatrix} \cos\frac{\theta}{2} & -i\sin\frac{\theta}{2} \\ -i\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{bmatrix}, \tag{2.43}$$

$$R_y(\theta) \equiv e^{-i\theta Y/2} = \cos\frac{\theta}{2} I - i \sin\frac{\theta}{2} Y = \begin{bmatrix} \cos\frac{\theta}{2} & \sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{bmatrix}, \tag{2.44}$$

$$R_z(\theta) \equiv e^{-i\theta Z/2} = \cos\frac{\theta}{2} I - i \sin\frac{\theta}{2} Z = \begin{bmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{bmatrix}. \tag{2.45}$$



(a) X gate **(b)** Y gate **(c)** Z gate

Figure 2.2: Action of Pauli gates (the initial state is represented in blue, the final in red): (a) Pauli X gate flips the state $|0\rangle$ to $|1\rangle$, (b) Pauli Y gate performs the operation: $|0\rangle \mapsto i|1\rangle$, (c) Pauli Z gate flips the state $|i\rangle \equiv (|0\rangle + i|1\rangle)/\sqrt{2}$ to state $|\text{-}i\rangle \equiv (|0\rangle - i|1\rangle)/\sqrt{2}$

Other useful single qubit gates are the *Hadamard gate*, the *phase gate* and $\pi/8$ *gate*. This gate turns the computational basis $\{|0\rangle, |1\rangle\}$ into the new basis $\{|+\rangle, |-\rangle\}$, whose states are a superposition of the states of the computational basis:

$$\begin{aligned} H|0\rangle = |+\rangle \equiv \frac{|0\rangle + |1\rangle}{\sqrt{2}}, \\ H|1\rangle = |-\rangle \equiv \frac{|0\rangle - |1\rangle}{\sqrt{2}}. \end{aligned} \tag{2.46}$$

Hence, with respect to the computational basis, Hadamard gate has the following matrix representation:

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{2.47}$$

The action of Hadamard gate on the state $|+\rangle$ is visualized in Fig. 2.3.

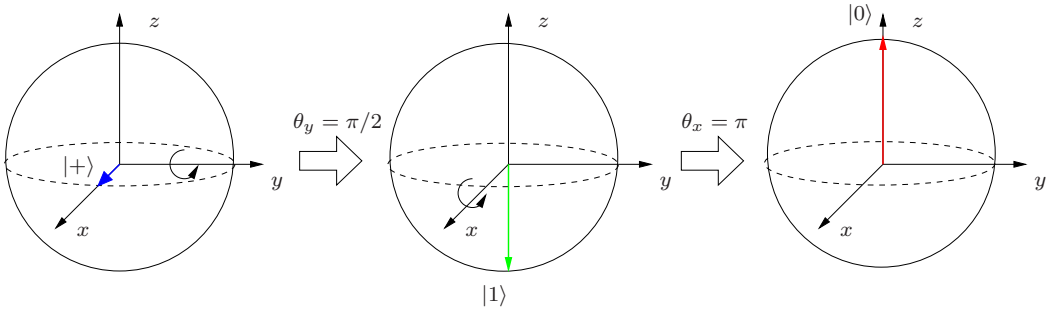


Figure 2.3: Action of Hadamard gate on input state $|+\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$

Phase gate S is defined as:

$$S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}. \tag{2.48}$$

This gate turns $|0\rangle$ into $|0\rangle$ and $|1\rangle$ into $i|1\rangle$. Since global phases have no physical meaning, the states of the computational basis remain unchanged.

Finally $\pi/8$ gate has the form:

$$T = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}. \tag{2.49}$$

Note that T is equivalent to:

$$T = \begin{bmatrix} e^{-i\pi/8} & 0 \\ 0 & e^{i\pi/8} \end{bmatrix} \tag{2.50}$$

(up to global phase), which is why it is called $\pi/8$ gate.

In Fig. 2.4 we report the symbolic circuit representation for the analyzed single qubit gates.

2.4.2 Controlled gates

Controlled gates act on two or more qubits, where one or more qubits act as a control for some operation. A fundamental controlled gate acting on

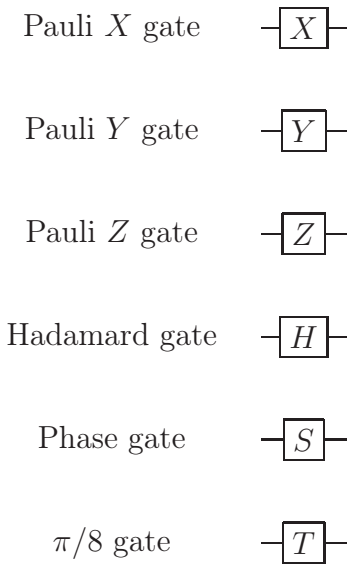


Figure 2.4: Circuit symbols for the most important single qubit gates

two input qubits is the controlled-NOT or briefly CNOT. In terms of the computational basis the action of this gate is given by: $|c\rangle|t\rangle \to |c\rangle|t \oplus c\rangle$, where $|t\rangle$ denotes the *target qubit* while $|c\rangle$ the *control qubit*; that is, it flips the state of the target qubit if the control qubit is in state $|1\rangle$ and does nothing otherwise. The matrix representation for the CNOT gate is

$$\text{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \tag{2.51}$$

and its circuit symbol is shown in Fig. 2.5.

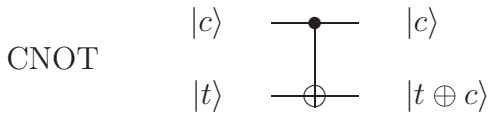


Figure 2.5: Circuit symbol for the controlled-NOT gate

A more general two-qubit controlled gate is the controlled-U gate. In this case if the control qubit is set to $|1\rangle$ then operator $U \in \mathfrak{U}(\mathbb{C}^2)$ is applied to the target qubit, otherwise the target qubit is left alone, in mathematical terms $|c\rangle|t\rangle \to |c\rangle U^c|t\rangle$. In matrix form controlled-U gate is

given by:

$$\mathrm{C}(U) = \left[\begin{array}{c|c} I & O \\ \hline O & U \end{array}\right], \tag{2.52}$$

where matrix block O denotes a 2×2 matrix of zeros and block I a 2-dimensional identity matrix. Fig. 2.6 shows the circuit representation of this operation.

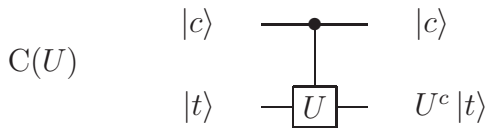


Figure 2.6: Circuit symbol for the controlled-U gate

2.4.3 Measurements

In quantum circuits we denote a projective measurement in the computational basis using the symbol shown in Fig. 2.7. For more general measurements it is conventional to not use special circuit symbols to denote them, since they can always be represented by unitary transformations with auxiliary ancilla qubits followed by projective measurements.

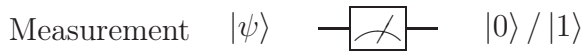


Figure 2.7: Circuit symbol for measurement in the computational basis

In the work we will often use quantum measurements to conditionally control subsequent quantum gates. Hence for these cases we represent with a double line a classical bit used to perform controlled operations. This procedure is illustrated in Fig. 2.8.

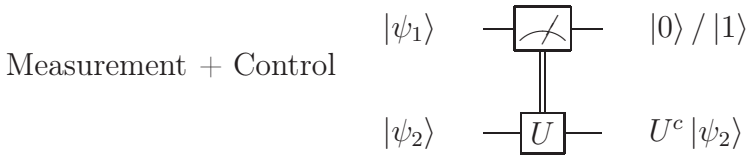


Figure 2.8: Circuit symbol for a classical control based on measurement outcome

3

Control of Quantum Systems

In this chapter we present some useful notions and theorems provided recently by Albertini and Ticozzi in [1] and by Bolognani and Ticozzi in [4] for the discrete-time feedback control of quantum systems. First we introduce the basic notions on controllability of quantum systems which evolve discretely in time. Then we describe the feedback control model and report the main results on controllability and stability of such a model. Finally we analyze an approach based on Lyapunov stabilization for the state inizialization of a quantum system.

3.1 Controllability Definitions

The controlled discrete-time evolution of a quantum system is described by:

$$\rho(t+1) = \mathcal{E}(\rho(t), \vec{u}(t)), \tag{3.1}$$

with, recalling notations given in chapter 2, $\rho(\cdot) \in \mathfrak{D}(\mathcal{H})$, $\vec{u}(t) \in \mathcal{U}$, where $\mathcal{U}$ denotes the set of control actions, and $\mathcal{E} : \mathfrak{D}(\mathcal{H}) \to \mathfrak{D}(\mathcal{H})$ a Completely Positive and Trace Preserving (CPTP) map which admits a Kraus representation of the form (2.41). Let $\mathfrak{R}_T(\rho)$ the reachable set from ρ in T steps.

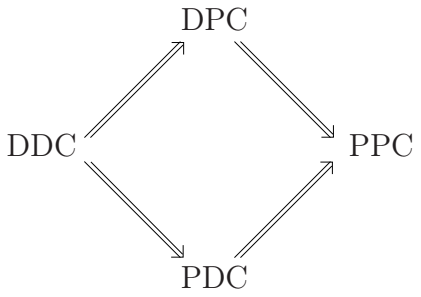
Definition 3.1 (PPC)**.** The system is said to be *Pure state to Pure state Controllable (PPC) in T steps* if $\forall \rho_0 = |\psi\rangle\langle\psi| \in \mathfrak{P}(\mathcal{H})$, $\mathfrak{R}_T(\rho) \supseteq \mathfrak{P}(\mathcal{H})$.

Definition 3.2 (DDC)**.** The system is said to be *Density Operator to Density Operator Controllable (DDC) in T steps* if $\forall \rho_0 \in \mathfrak{D}(\mathcal{H})$, $\mathfrak{R}_T(\rho) = \mathfrak{D}(\mathcal{H})$.

Definition 3.3 (PDC)**.** The system is said to be *Pure state to Density Operator Controllable (PDC) in T steps* if $\forall \rho_0 = |\psi\rangle\langle\psi| \in \mathfrak{P}(\mathcal{H})$, $\mathfrak{R}_T(\rho) = \mathfrak{D}(\mathcal{H})$.

Definition 3.4 (DPC)**.** The system is said to be *Density Operator to Pure state Controllable (DPC) in T steps* if $\forall \rho_0 \in \mathfrak{D}(\mathcal{H})$, $\mathfrak{R}_T(\rho) \supseteq \mathfrak{P}(\mathcal{H})$.

Note that, since $\mathfrak{P}(\mathcal{H}) \subset \mathfrak{D}(\mathcal{H})$, we have the following relations:



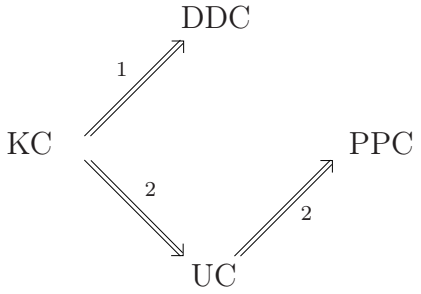
Definition 3.5 (UC)**.** The system is said to be *Unitary Controllable (UC) in T steps* if $\forall U \in \mathfrak{U}(\mathcal{H})$ there exist a choice of T controls $\vec{u}_i$, such that,

$$U\rho U^\dagger = \mathcal{E}_T \circ \cdots \circ \mathcal{E}_1(\rho), \qquad \forall \rho \in \mathfrak{D}(\mathcal{H}),$$

where $\mathcal{E}_i(A) = \mathcal{E}(A, \vec{u}_i)$.

Definition 3.6 (KC)**.** The system is said to be *Kraus Controllable (KC) in T steps* if for any CPTP map $\mathcal{E}$ there exists a choice of T controls $\vec{u}_i$, such that $\mathcal{E} = \mathcal{E}_T \circ \cdots \circ \mathcal{E}_1$.

It can be demonstrated that the following implications hold:



Indeed implication 1 can be easily proven considering a constant map $\mathcal{K} : \mathfrak{D}(\mathcal{H}) \to \mathfrak{D}(\mathcal{H})$, which maps the initial state ρ to the target state ρ_f. $\mathcal{K}$ can be extended to a linear CPTP map on $\mathfrak{M}(\mathcal{H})$, and hence it admits a Kraus representation.

3.2 Discrete-Time Feedback Control

3.2.1 The model

We present here a discrete-time feedback model for controlling dynamics of open quantum systems. The model is based on the application of a unitary operator $U_k \in \mathfrak{U}(\mathcal{H})$ chosen in the set $\{U_k\}$ conditioned to the outcome $M_k \in \mathfrak{M}(\mathcal{H})$ of a finite set of generalized measurements $\{M_k\}$. We recall that the probability of measuring the k-th outcome is given by:

$$\mathbb{P}_\rho(k) = \mathrm{tr}(M_k \rho M_k^\dagger), \tag{3.2}$$

and the conditioned state after the measurement takes the form:

$$\rho|_k = \frac{M_k \rho M_k^\dagger}{\mathbb{P}_\rho(k)}. \tag{3.3}$$

Hence if the state at time t was $\rho(t) \in \mathfrak{D}(\mathcal{H})$, the state at time $t+1$ conditioned to the k-th outcome of the generalized measurement is[1]:

$$\rho(t+1)|_k = \frac{U_k M_k \rho M_k^\dagger U_k^\dagger}{\mathbb{P}_\rho(k)}. \tag{3.4}$$

By averaging over the possible outcomes we get the CPTP map:

$$\rho(t+1) = \sum_k U_k(t) M_k(t) \rho(t) U_k(t)^\dagger M_k(t)^\dagger. \tag{3.5}$$

A scheme of the control model is shown in Fig. 3.1.

3.2.2 Results on controllability and stabilization

First we report some results regarding generic aymptotic controllability of systems which evolve under feedback control dynamics (3.5).

Theorem 3.1. *Given a measurement with Kraus operators $\{M_k\}$ and arbitrary conditional control actions $\{U_k\} \subset \mathfrak{U}(\mathcal{H})$, the system (3.5) is asymptotically DPC if and only if there is a k such $M_k \neq qV_k$, $\forall q \in \mathbb{C}$, and $V_k \in \mathfrak{U}(\mathcal{H})$.*

[1] In writing equation (3.4) we used the cyclic property of the trace,that is, for three operators $A, B, C \in \mathfrak{M}(\mathcal{H})$:

$$\mathrm{tr}(ABC) = \mathrm{tr}(CAB) = \mathrm{tr}(BCA).$$

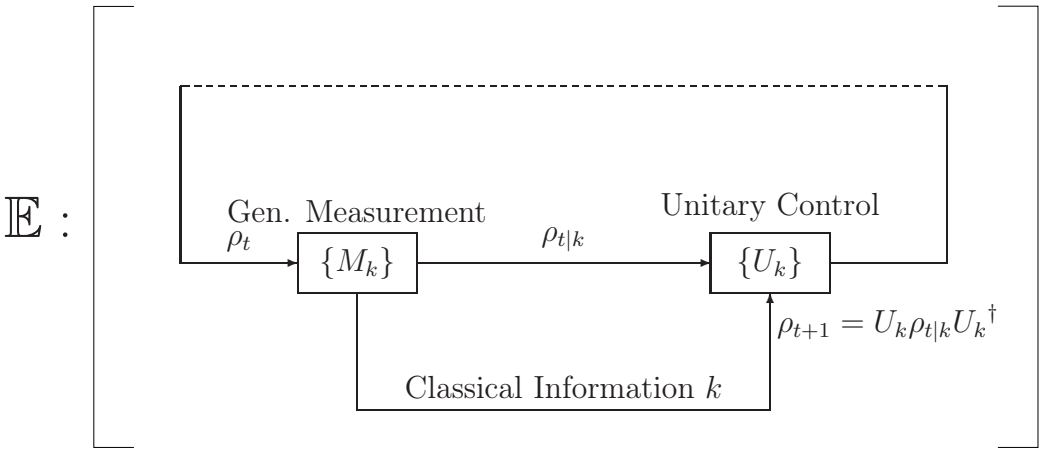


Figure 3.1: Schematic representation of the feedback control model. If the measurement-control loop is iterated the average over the measurement results at each step yields to the CPTP map given by (3.5)

Proof. See [1, Theorem 3.2].

Corollary 3.1. *If there is a k such $M_k \neq qV_k$, with $q \in \mathbb{C}$, and $V_k \in \mathfrak{U}(\mathcal{H})$, and in addition we can pick a control action at random from a finite set $\{\hat{U}_j\}$ with an arbitrary probability distribution p_j, then the system is approximately DDC.*

Proof. See [1, Corollary 3.1].

Other useful results, provided in [1], concern the case of a given generalized measurements with only two outcomes and associated operators M_1 and M_2 satisfying the completeness relation. This is the case that we will consider in chapter 4.

These results are based on the following intial assumptions:

1. Both matrices M_1 and M_2 are diagonal;

2. Both matrices M_1 and M_2 are singular;

Assumption 1 is not restricitive using the feedback control model analyzed previously since the following Lemma help us.

Lemma 3.1. *Consider two operators $\tilde{M}_1, \tilde{M}_2 \in \mathfrak{M}(\mathcal{H})$ that satisfy completeness relation $\tilde{M}_1^\dagger \tilde{M}_1 + \tilde{M}_2^\dagger \tilde{M}_2 = I$. Then there exist unitaries $U_0, U_1, U_2 \in \mathfrak{U}(\mathcal{H})$ such that $M_k = U_k \tilde{M}_k U_0$ is diagonal for $k = 1, 2$.*

Proof. See [1, Lemma 4.1].

Under these assumptions, the following proposition states that a N-level system can be made DPC in finite-time through a sequence of controlled operations of length N.

Proposition 3.1. *There exists a sequence* $U_1(0), \cdots, U_1(N-2)$ *and* $U_2(0)$, $\cdots$, $U_2(N-2)$, *such that* $\forall \rho_0 \in \mathfrak{D}(\mathcal{H})$, $\rho(N)$ *is a pure state.*

Proof. See [1, Proposition 4.1].

The converse is also true, that is, the system is finite-time PDC.

Proposition 3.2. *Assume that* ρ_0 *is a pure state, then* $\forall \rho_f \in \mathfrak{D}(\mathcal{H})$ *there exists a sequence of controls of length* N *that steers* ρ_0 *to* ρ_f.

Proof. See [1, Proposition 4.2].

So from propositions 3.1 and 3.2 it can be deduced that a N-level system is also finite-time DDC and the (maximum) number of feedback steps needed to obtain any desired state-to-state transition is equal to $2N$.

3.3 Lyapunov Stabilization

In Lyapunov stabilization of quantum systems, one specifies a function of the state, the *Lyapunov function* $V(\rho)$, where $\rho \in \mathfrak{D}(\mathcal{H})$ is the density operator of the system, and designs the control so that the value of the $V(\rho)$ decreases to a desired value ([6]).

In the following we describe the approach given in [4] for the choice of a possible Lyapunov function $V(\rho)$. This approach is useful when applied to control the discrete-time dynamics of open quantum systems and specially for the case of quantum states preparation, that is the issue we address in this work. First we define what is a *quantum subspace*.

Definition 3.7. A quantum subspace $\mathcal{S}$ of a system $\mathcal{I}$ with associated Hilbert space $\mathcal{H}_\mathcal{I}$ is a quantum system whose Hilbert space is a subspace $\mathcal{H}_\mathcal{S}$ of $\mathcal{H}_\mathcal{I}$:

$$\mathcal{H}_\mathcal{I} = \mathcal{H}_\mathcal{S} \oplus \mathcal{H}_\mathcal{R} \tag{3.6}$$

for some remainder space $\mathcal{H}_\mathcal{R}$. The set of linear operators on $\mathcal{S}$, $\mathfrak{M}(\mathcal{H}_\mathcal{S})$, is isomorphic to the algebra on $\mathcal{H}_\mathcal{I}$ with elements of the form $\mathcal{H}_\mathcal{I} = \mathcal{H}_\mathcal{S} \oplus \mathbb{O}_R$, with $\mathbb{O}$ the zero operator.

Let $n = \dim(\mathcal{H}_\mathcal{I})$, $m = \dim(\mathcal{H}_\mathcal{S})$, and $r = \dim(\mathcal{H}_\mathcal{R})$, and let $\{|s\rangle_j\}_{j=1}^m$, $\{|r\rangle_k\}_{k=1}^r$ denote orthonormal bases for $\mathcal{H}_\mathcal{S}$ and $\mathcal{H}_\mathcal{R}$, respectively. Decomposition (3.6) is then associated with the following basis for $\mathcal{H}_\mathcal{I}$:

$$\{|i\rangle_l\}_{l=1}^n = \{|s\rangle_j\}_{j=1}^m \cup \{|r\rangle_k\}_{k=1}^r$$

This basis induces a block structure for matrices representing operators in $\mathfrak{M}(\mathcal{H}_\mathcal{I})$:

$$X = \left[\begin{array}{c|c} X_S & X_P \\ \hline X_Q & X_R \end{array}\right]. \tag{3.7}$$

We call $\Pi_\mathcal{S}$ and $\Pi_\mathcal{R}$ the projection operators over the subspaces $\mathcal{H}_\mathcal{S}$ and $\mathcal{H}_\mathcal{R}$, respectively.

Now we give the definition of *state inizialization*, i.e. the preparation of a certain quantum state $\rho_S \in \mathfrak{D}(\mathcal{H}_S)$.

Definition 3.8. The system $\mathcal{I}$ with state $\rho \in \mathfrak{D}(\mathcal{H}_\mathcal{I})$ is initialized in $\mathcal{S}$ with state $\rho_S \in \mathfrak{D}(\mathcal{H}_\mathcal{S})$ if ρ is of the form:

$$\rho = \left[\begin{array}{c|c} \rho_S & O \\ \hline O & O \end{array}\right]. \tag{3.8}$$

We use symbol $\mathfrak{I}_\mathcal{S}(\mathcal{H}_\mathcal{I})$ to denote the set of state of the form (3.8) for some $\rho_S \in \mathfrak{D}(\mathcal{H}_\mathcal{S})$.

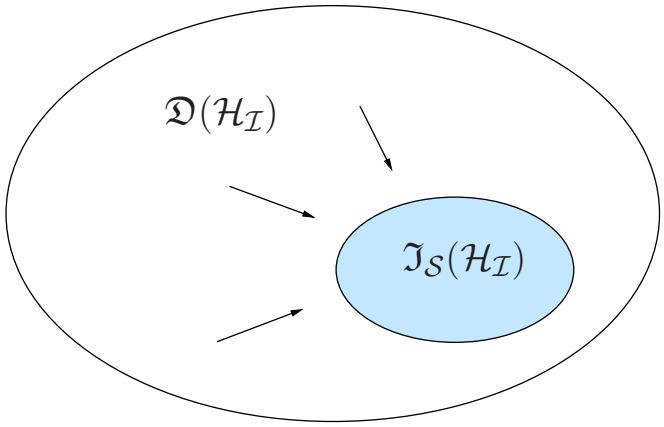


Figure 3.2: Representation of the set $\mathfrak{D}(\mathcal{H}_\mathcal{I})$ and the target subset $\mathfrak{I}_\mathcal{S}(\mathcal{H}_\mathcal{I})$

Next, following the analysis given in [4], we characterize *invariance* and *attractivity* of $\mathfrak{I}_\mathcal{S}(\mathcal{H}_\mathcal{I})$.

Definition 3.9. Let $\mathcal{I}$ evolve under iterations of a CPTP map $\mathcal{T} : \mathcal{H}_\mathcal{I} \to \mathcal{H}_\mathcal{I}$. The set $\mathfrak{I}_\mathcal{S}(\mathcal{H}_\mathcal{I})$ is *invariant* if the evolution of any initialized $\rho \in \mathfrak{I}_\mathcal{S}(\mathcal{H}_\mathcal{I})$ obeys:

$$\mathcal{T}[\rho(t)] = \left[\begin{array}{c|c} \mathcal{T}_S[\rho(t)] & O \\ \hline O & O \end{array}\right] \in \mathfrak{I}_\mathcal{S}(\mathcal{H}_\mathcal{I}), \quad \forall t \geq 0. \tag{3.9}$$

Definition 3.10. Let $\mathcal{I}$ evolve under iterations of a CPTP map $\mathcal{T} : \mathcal{H}_\mathcal{I} \to \mathcal{H}_\mathcal{I}$. The set $\mathfrak{I}_\mathcal{S}(\mathcal{H}_\mathcal{I})$ is *attractive* if $\forall \rho \in \mathfrak{D}(\mathcal{H}_\mathcal{I})$ we have:

$$\lim_{t\to\infty} \|\mathcal{T}[\rho] - \Pi_S \mathcal{T}[\rho]\Pi_S\| = 0. \tag{3.10}$$

Definition 3.11. Let $\mathcal{I}$ evolve under iterations of a CPTP map $\mathcal{T} : \mathcal{H}_\mathcal{I} \to \mathcal{H}_\mathcal{I}$. The set $\mathfrak{I}_\mathcal{S}(\mathcal{H}_\mathcal{I})$ is *Globally Asymptotically Stable* (GAS) if it is invariant and attractive.

The value of the state $\rho \in \mathfrak{D}(\mathcal{H}_\mathcal{I})$ will tend to a limit set which can be characterized using *La Salle's invariance principle* for discrete-time systems.

Theorem 3.2. *Consider a discrete-time system $x(t+1) = \mathcal{T}[x(t)]$. Suppose $V \in \mathcal{C}^1(\mathbb{R}^n)$, bounded below and satisfying:*

$$\Delta V(x) = V(\mathcal{T}[x]) - V(x) \leq 0, \quad \forall x, \tag{3.11}$$

that is, $V(x)$ is non-increasing along forward trajectories of the plant dynamics. Then any bounded trajectory converges to the largest invariant subset W contained in the locus $\Omega = \{x : \Delta V(x) = 0\}$.

Proof. A proof can be found in [14, Theorem 3.4] for continuous-time systems, which can be extended to the discrete-time case.

Hence a possible candidate for the Lyapunov function is:

$$V(\rho) = \mathrm{tr}(\Pi_\mathcal{R}\rho). \tag{3.12}$$

$V(\rho)$ represents the probability of the event $\Pi_\mathcal{R}$, i.e., the probability that the system is found in the reminder subspace $\mathcal{H}_\mathcal{R}$ after the measurement. Moreover can be proved that this function satisfies the conditions imposed by La Salle's theorem when $\mathfrak{I}_\mathcal{S}(\mathcal{H}_\mathcal{I})$ is invariant.

Now we present the main results concerning conditions for invariance and attractivity of $\mathfrak{I}_\mathcal{S}(\mathcal{H}_\mathcal{I})$.

Proposition 3.3. *Consider the Kraus representation* $\mathcal{T}[\rho] = \sum_k E_k \rho E_k^\dagger$ *of the CPTP map* $\mathcal{T} : \mathcal{H}_\mathcal{I} \to \mathcal{H}_\mathcal{I}$*. Let the matrices* E_k *be expressed in the block form:*

$$E_k = \left[\begin{array}{c|c} E_{k,S} & E_{k,P} \\ \hline E_{k,Q} & E_{k,R} \end{array}\right], \tag{3.13}$$

according to the space decomposition (3.7). Then the set $\mathfrak{I}_\mathcal{S}(\mathcal{H}_\mathcal{I})$ *is invariant if and only if:*

$$E_{k,Q} = 0 \quad \forall k. \tag{3.14}$$

Proof. See [4, Section IV - Proposition 1].

Theorem 3.3. *Consider the Kraus representation* $\mathcal{T}[\rho] = \sum_k E_k \rho E_k^\dagger$ *of the CPTP map* $\mathcal{T} : \mathcal{H}_\mathcal{I} \to \mathcal{H}_\mathcal{I}$*. Let* $\mathcal{H}_\mathcal{S} \oplus \mathcal{H}_\mathcal{R}$ *an orthogonal subset decomposition of* $\mathcal{H}_\mathcal{I}$ *with* $\mathfrak{I}_\mathcal{S}(\mathcal{H}_\mathcal{I})$ *invariant. Consider the block form expression of matrices* E_k*:*

$$E_k = \left[\begin{array}{c|c} E_{k,S} & E_{k,P} \\ \hline O & E_{k,R} \end{array}\right]. \tag{3.15}$$

Then the set $\mathfrak{I}_\mathcal{S}(\mathcal{H}_\mathcal{I})$ *is GAS if and only if there are no invariant states with support*[2] *on* $\bigcap_k \ker(E_{k,P})$*.*

Proof. See [4, Section IV - Theorem 2].

[2]We recall some mathematical definitions. The *support* of a complex function f (on any topological space) is the closure of $\{x : f(x) \neq 0\}$ ([24]). The *kernel* of a matrix $A \in \mathbb{C}^{m\times n}$ is the set: $\ker(A) = \{\vec{x} \in \mathbb{C}^n : A\vec{x} = \vec{0}\}$.

4

Entanglement Generation

In this chapter we first analyze the approach proposed in [2] concerning Bell-state $|\Psi^-\rangle$ and GHZ-state preparation, then we study the experimental quantum circuits engineered for this purpose through numerical simulations. In the second section we introduce a new approach based on an elementary quantum feedback control paradigm and using this approach we engineer two alternative quantum circuits implementing both Bell-state $|\Psi^-\rangle$ and GHZ-state preparation. Finally in the last section, for each case, we list pros and cons of the two analyzed solutions.

4.1 J. T. Barreiro *et al.* approach

In this section we study the approach used in [2] for entanglement generation. First we focus our attention on the Bell-state $|\Psi^-\rangle$ generation case study and on the relative solution offered by the authors of the article. Then we analyze the solution given for the GHZ-state generation.

4.1.1 Article review

In [2] the authors present a toolbox for simulating and controlling dynamics of an open quantum system with up to five qubits. To do so experimentally they use a quantum computing architecture with trapped ions, combining multi-qubit gates with optical pumping to implement coherent operations and dissipative processes. In a string of trapped ions, each ion encoding a qubit, the qubits are divided into "system" and "environment". The system-environment coupling is engineered through the uni-

versal set of quantum operations available in ion-trap quantum computers and a dissipative mechanism based on optical pumping. In particular, The Mølmer-Sørensen (MS) entangling gate plays a key role in the experimental quantum circuits realization and for its importance it will be analyzed in detail in paragraph §4.2.2. Specifically Barreiro *et al.* study how to engineer the process of dissipative preparation of entangled states. In the first part of the article is illustrated a process for the Bell-state $|\Psi^-\rangle$ "cooling" by pumping the initially mixed system state in appropriate eigenspaces of stabilizer operators[1] of Bell-states. Then, in the second part, the same approach used for Bell-states generation is applied to dissipatively prepare a 4-qubit GHZ-state[2].

4.1.2 Bell-state $|\Psi^-\rangle$ generation

Process implementation

The system, described by density matrix ρ_S, is prepared in the Bell-state $|\Psi^-\rangle$ realizing the quantum operator: $\rho_S \mapsto |\Psi^-\rangle\langle\Psi^-|$. This is achieved by constructing two dissipative maps and exploiting the useful properties of stabilizer operators Z_1Z_2 and X_1X_2 of the Bell-states[3]. First of all we recall the definitions of the four Bell-states:

$$|\Phi^\pm\rangle = \frac{1}{\sqrt{2}}(|00\rangle \pm |11\rangle),\ |\Psi^\pm\rangle = \frac{1}{\sqrt{2}}(|01\rangle \pm |10\rangle). \tag{4.1}$$

It is worth noting that Bell-states are *stabilizer states*: for instance $|\Phi^+\rangle$ is said to be stabilized by the two stabilizers Z_1Z_2 and X_1X_2 as it is the only two-qubit state being simultaneously an eigenstate of eigenvalue $+1$ of these two commuting operators ($Z_1Z_2|\Phi^+\rangle = +|\Phi^+\rangle$, $X_1X_2|\Phi^+\rangle = +|\Phi^+\rangle$). Moreover each of the four Bell-states (4.1) is uniquely determined as an eigenstate with eigenvalues ± 1 with respect to Z_1Z_2 and X_1X_2.

[1] Suppose S is a subgroup of the *Pauli group* G_n on n qubits and define V_S to be the set of n qubit states which are fixed by every element of S. V_S is the *vector space stabilized* by S and S is said to be the *stabilizer* of the space V_S, since every element of V_S is stable under the action of elements in S. For further details on *stabilizer formalism* refer to [20].

[2] *Greenberger-Horne-Zeilinger* (GHZ) *state* state is a certain type of entangled quantum state of M qubits that is defined as:

$$|GHZ\rangle = \frac{|0\rangle^{\otimes M} + |1\rangle^{\otimes M}}{\sqrt{2}},$$

where $M > 2$ denotes the number of qubits entangled.

[3] $X_i,\ Y_i,\ Z_i$ stand for X, Y, Z Pauli gates repectively acting on the i-th system qubit.

Noting that $|\Psi^-\rangle$ state is an eigenstate with eigenvalue -1 of these two stabilizers, the key idea to prepare the initial system state in the $|\Psi^-\rangle$ state is to engineer two maps under which the two system qubits are irreversibly transferred from the $+1$ into the -1 eigenspace of Z_1Z_2 and X_1X_2. The first map is performed by two dissipative Kraus operators E_1, E_2:

$$\rho_S \mapsto \mathcal{E}'(\rho_S) = \rho'_S = E_1\rho_S E_1^\dagger + E_2\rho_S E_2^\dagger \tag{4.2}$$

$$E_1 = \sqrt{p}\; X_2\, \frac{1}{2}(1 + Z_1Z_2) \tag{4.3}$$

$$E_2 = \frac{1}{2}(1 - Z_1Z_2) - \sqrt{1-p}\, \frac{1}{2}(1 + Z_1Z_2) \tag{4.4}$$

The map implements the process of "pumping" the initial random system state, described by the density matrix ρ_S, from eigenspace with eigenvalue $+1$ of stabilizer operator Z_1Z_2 of the Bell-states, into the -1 eigenspace. The operation element E_1 can be interpreted as follows: first, apply the projector $\frac{1}{2}(1 + Z_1Z_2)$ onto the $+1$ eigenspace of Z_1Z_2; then, the spin flip X_2 converts $+1$ into -1 eigenstates of Z_1Z_2 , e.g. $|\Phi^+\rangle \mapsto |\Psi^-\rangle$.The process can be visualized intuitively in Fig. 4.1.

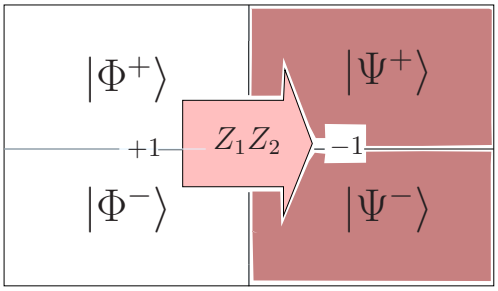


Figure 4.1: Schematic representation of Z_1Z_2 map

Then a similar process is achieved by the two operators E'_1, E'_2:

$$\rho'_S \mapsto \mathcal{E}(\rho_S) = E'_1\rho_S E_1'^\dagger + E'_2\rho_S E_2'^\dagger \tag{4.5}$$

$$E'_1 = \sqrt{p}\; Z_2\, \frac{1}{2}(1 + X_1X_2) \tag{4.6}$$

$$E'_2 = \frac{1}{2}(1 - X_1X_2) - \sqrt{1-p}\, \frac{1}{2}(1 + X_1X_2) \tag{4.7}$$

The process implemented by E_1' and E_2' is similar to the previous one but in this case it implements the pumping of the system state ρ_S' from eigenspace with eigenvalue $+1$ of stabilizer operator X_1X_2 of the Bell states, into the -1 with a probability p. The map action is represented in Fig. 4.2.

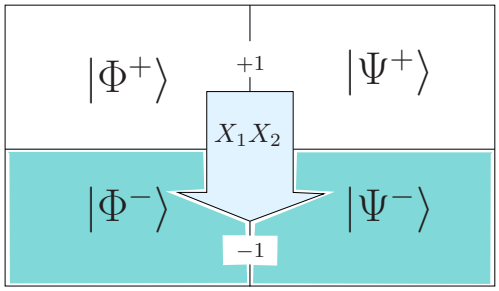


Figure 4.2: Schematic representation of X_1X_2 map

So the action of the two maps described above implements the projection or pumping of the initial unknown state, ρ_S, into the target $|\Psi^-\rangle$ Bell-state, as it is summarized in Fig. 4.3.

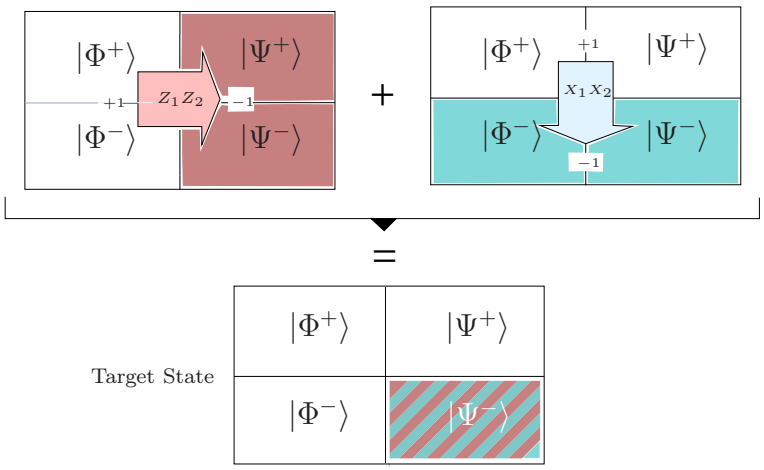


Figure 4.3: Action of $Z_1Z_2 + X_1X_2$ map

Experimental quantum circuit

To implement experimentally the two maps described in the previous paragraph the authors suggest to use the quantum circuit shown in Fig 4.4, which consists of three unitary operations, $(i), (ii), (iii)$ and a dissipative one (iv). Both maps act on the two system qubits, denoted by subscript S, and an ancilla which plays the role of the environment, with subscript E.

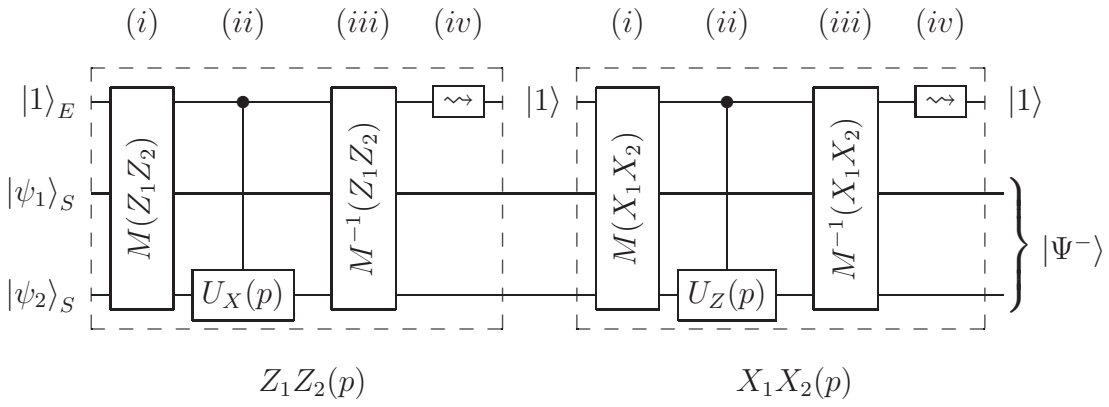


Figure 4.4: Quantum circuit used to generate Bell-state $|\Psi^-\rangle$

The first map implementation proceeds as follows:

(i) Information about whether the system is in the +1 or -1 eigenspace of Z_1Z_2 is mapped by unitary $M(Z_1Z_2)$ onto the logical states $|0\rangle$ and $|1\rangle$ of the ancilla (initially in $|1\rangle$).

(ii) A controlled gate, $C(p)$ in figure, converts +1 into -1 eigenstates by flipping the state of the second qubit with probability p:

$$C(p) = |0\rangle\langle 0|_E \otimes U_{X_2}(p) + |1\rangle\langle 1|_E \otimes I, \tag{4.8}$$

where $U_{X_2}(p) = e^{i\alpha X_2}$ and $p = \sin^2(\alpha)$.

(iii) The initial mapping is inverted by $M^{-1}(Z_1Z_2)$. At this stage, in general, the ancilla and system qubits are entangled.

(iv) The ancilla is dissipatively reset to $|1\rangle$, which carries away entropy to "cool" the two system qubits. This reinitialization is performed through optical pumping technique.

The second map for cooling into the -1 eigenspace of X_1X_2 is obtained by interchanging the roles of X and Z above.

Indeed the real circuit used in the experiment is a little more different from the one analyzed above, as can be seen in Fig 4.5.

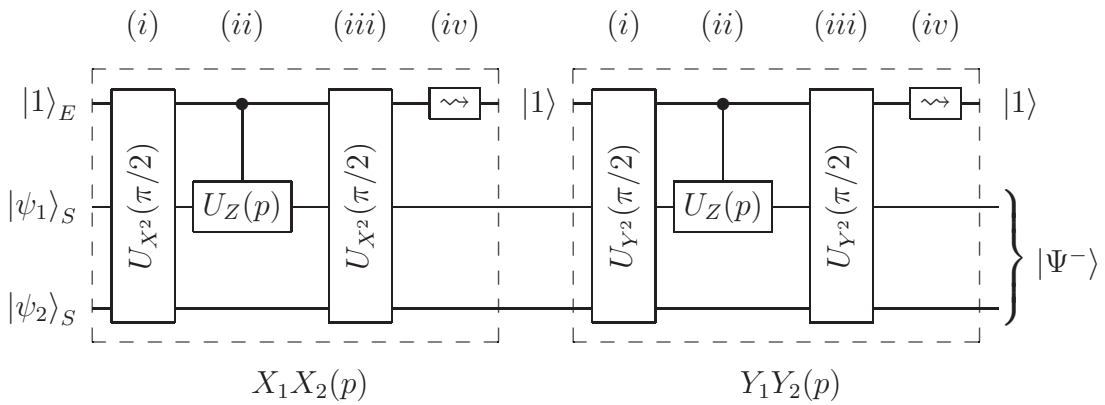


Figure 4.5: Quantum circuit used to experimentally generate Bell-state $|\Psi^-\rangle$

In this new inplementation the authors exploit the fact (see [2], supplementary informations) that the Bell-state $|\Psi^-\rangle$ is not only uniquely determined as the simultaneous eigenstate with eigenvalue -1 of the two stabilizer operators X_1X_2 and Z_1Z_2, but also by X_1X_2 and Y_1Y_2.
So equations (4.2), (4.3), (4.4), describing map Z_1Z_2 action, can be replaced respectively by:

$$\rho_S \mapsto \mathcal{E}'(\rho_S) = \rho'_S = E_1\rho_S E_1^\dagger + E_2\rho_S E_2^\dagger \tag{4.9}$$

$$E_1 = \sqrt{p}\, Y_1\, \frac{1}{2}(1 + X_1X_2) \tag{4.10}$$

$$E_2 = \frac{1}{2}(1 - X_1X_2) - \sqrt{1-p}\, \frac{1}{2}(1 + X_1X_2) \tag{4.11}$$

The map generates pumping from intial system state into the -1 eigenspace of X_1X_2 stabilizer and its action, similar to that described in the previous section, is shown in Fig 4.2.

In the same way equations (4.5), (4.6), (4.7), describing map X_1X_2 action, can be replaced equivalently by:

$$\rho'_S \mapsto \mathcal{E}(\rho_S) = E'_1\rho_S E_1'^\dagger + E'_2\rho_S E_2'^\dagger \tag{4.12}$$

$$E'_1 = \sqrt{p}\, X_1\, \frac{1}{2}(1 + Y_1Y_2) \tag{4.13}$$

$$E'_2 = \frac{1}{2}(1 - Y_1Y_2) - \sqrt{1-p}\, \frac{1}{2}(1 + Y_1Y_2) \tag{4.14}$$

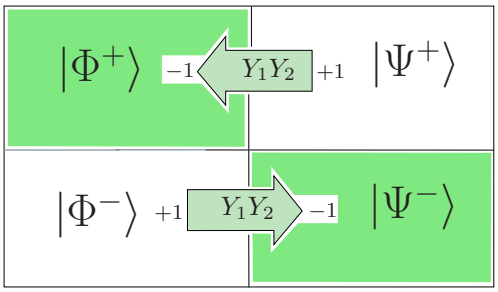


Figure 4.6: Schematic representation of Y_1Y_2 map

The map implements the pumping of the system state ρ'_S from eigenspace with eigenvalue $+1$ of stabilizer operator Y_1Y_2, into the -1 with a probability p. The map action is represented in Fig. 4.6.

The overall action of the two maps can be seen in Fig. 4.7.

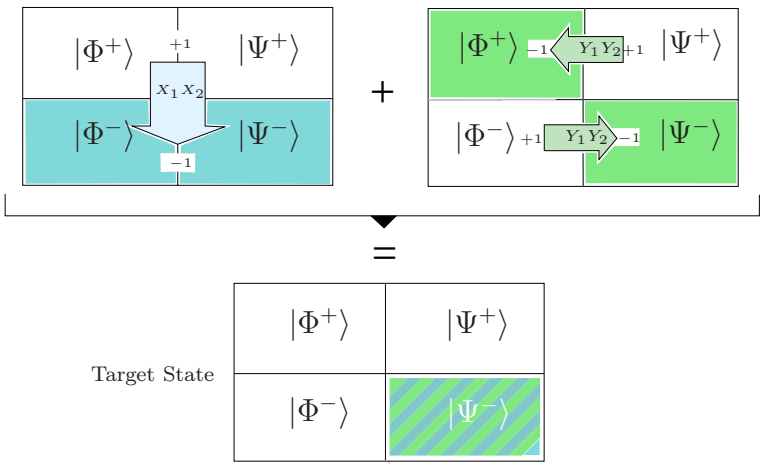


Figure 4.7: Action of $X_1X_2 + Y_1Y_2$ map

The great advantage of this substitution is that the mapping and unmapping steps, shown as (i) and (iii) in Fig 4.4, now can be simply realized by a single Mølmer-Sørensen gate $U_{X^2}(\pi/2)$ and $U_{Y^2}(\pi/2)$[4], respectively. The great benefit in using such a gate lies in its experimental availability.

[4]Mølmer-Sørensen gates $U_{X^2}(\theta)$ and $U_{Y^2}(\theta)$ are defined as: $U_{X^2}(\theta) = \exp(-i\frac{\theta}{4}{S_x}^2)$ and $U_{Y^2}(\theta) = \exp(-i\frac{\theta}{4}{S_y}^2)$, where S_x and S_y denote collective spin operators: $S_x = \sum_{i=0}^{n} X_i$ and $S_y = \sum_{i=0}^{n} Y_i$.

The main steps of this experimental quantum circuit are listed below:

(*i*) Information about whether the system is in the +1 or -1 eigenspace of X_1X_2 is mapped by MS gate $U_{X^2}(\pi/2)$ onto the logical states $|0\rangle$ and $|1\rangle$ of the ancilla.

(*ii*) A controlled gate performs a conversion from the +1 eigenvalue of the stabilizer X_1X_2 to -1 exchanging status of the first system qubit $|\psi_1\rangle_S$ with a certain probability p:

$$C(p) = |0\rangle\langle 0|_E \otimes U_{Z_1}(p) + |1\rangle\langle 1|_E \otimes I, \tag{4.15}$$

where $U_{Z_1}(p) = e^{i\alpha Z_1}$ and $p = \sin^2(\alpha)$ [5].

(*iii*) The initial MS gate is inverted to perform successfully the projection onto -1 eigenspace.

(*iv*) The ancilla qubit is then reinitialized dissipatively in state $|1\rangle$.

Numerical simulation

For the numerical simulation of the circuit described above (Fig. 4.5) we wrote a code in **Matlab** language. In this code the user can enter the desired probability p of pumping and the program calculates the final system density matrix after 20 iterations of the process. The video output of the program by setting a probability $p = 0.5$ is shown below. In particular Fig. 4.8 shows the evolution of the initial system described by a random density matrix, ρ_S, increasing the number of iterations of the process. Moreover in Fig. 4.9 is reported the evolution of the system after sequentially pumping the stabilizers X_1X_2, Y_1Y_2 for the first iteration of the process.

```
Insert a probability of pumping: 0.5
Initial density matrix:

psi =

   0.4092                -0.0938 - 0.1104i    0.1617 + 0.2033i   -0.0089 - 0.0807i
  -0.0938 + 0.1104i       0.2395             -0.2224 - 0.0501i    0.0805 + 0.0298i
   0.1617 - 0.2033i      -0.2224 + 0.0501i    0.2978             -0.0903 - 0.0231i
  -0.0089 + 0.0807i       0.0805 - 0.0298i   -0.0903 + 0.0231i    0.0535

Final density matrix desired:

PSIm =

         0         0         0         0
         0    0.5000   -0.5000         0
         0   -0.5000    0.5000         0
         0         0         0         0
```

[5] In the experiment the gate is realized through the following sequence of operators: $C(p) = U_{Z_1}(\alpha)U_Y(\pi/2)U^{(0,1)}_{X^2}(-\alpha)U_Y(-\pi/2)$ con $U^{(0,1)}_{X^2}(-\alpha) = exp(i(\alpha/2)X_0X_1)$. In the following sections for simplicity we will refer to equation (4.15).

```
System density matrix after 1 iteration:

phi =

   0.2162 - 0.0000i   -0.0800 - 0.0933i    0.1041 + 0.1261i   -0.0045 - 0.0403i
  -0.0800 + 0.0933i    0.3582 + 0.0000i   -0.3612 - 0.0251i    0.0466 + 0.0035i
   0.1041 - 0.1261i   -0.3612 + 0.0251i    0.3873 + 0.0000i   -0.0501 - 0.0011i
  -0.0045 + 0.0403i    0.0466 - 0.0035i   -0.0501 + 0.0011i    0.0383 + 0.0000i

System density matrix after 20 iterations:

phi =

   0.0000 - 0.0000i   -0.0001 - 0.0001i    0.0001 + 0.0001i   -0.0000 - 0.0000i
  -0.0001 + 0.0001i    0.5000 + 0.0000i   -0.5000 - 0.0000i    0.0000 - 0.0000i
   0.0001 - 0.0001i   -0.5000 + 0.0000i    0.5000 + 0.0000i   -0.0000 + 0.0000i
  -0.0000 + 0.0000i    0.0000 + 0.0000i   -0.0000 - 0.0000i    0.0000 - 0.0000i
```

If the probability of pumping is unitary the results change as shown below. It is worth noticing that $|\Psi^-\rangle$ "cooling" is successfully performed after one iteration (as can be seen in Fig. 4.11).
The full **Matlab** code is listed in Appendix A.

```
Insert a probability of pumping: 1
Initial density matrix:

psi =

   0.1896                0.0108 + 0.0106i   -0.0990 - 0.0642i    0.1418 + 0.0765i
   0.0108 - 0.0106i      0.2172              0.0309 + 0.0222i   -0.0516 + 0.0273i
  -0.0990 + 0.0642i      0.0309 - 0.0222i    0.3409            -0.0837 + 0.1232i
   0.1418 - 0.0765i     -0.0516 - 0.0273i   -0.0837 - 0.1232i    0.2523

Final density matrix desired:

PSIm =

         0         0         0         0
         0    0.5000   -0.5000         0
         0   -0.5000    0.5000         0
         0         0         0         0

System density matrix after 1 iteration:

phi =

   0.0000 - 0.0000i    0.0000 + 0.0000i   -0.0000 - 0.0000i    0.0000 - 0.0000i
   0.0000 - 0.0000i    0.5000 - 0.0000i   -0.5000 - 0.0000i   -0.0000 + 0.0000i
        0 - 0.0000i   -0.5000 - 0.0000i    0.5000             0.0000 - 0.0000i
   0.0000 - 0.0000i    0.0000 + 0.0000i   -0.0000 - 0.0000i   -0.0000 - 0.0000i

System density matrix after 20 iterations:

phi =

  -0.0000 + 0.0000i         0                   0            -0.0000 - 0.0000i
         0              0.5000 - 0.0000i   -0.5000 + 0.0000i         0
         0             -0.5000 + 0.0000i    0.5000 - 0.0000i         0
  -0.0000 - 0.0000i         0                   0            -0.0000 + 0.0000i
```

4.1.3 GHZ-state generation

Process implementation

The process implementing the dissipative preparation of a four-qubit *Greenberger - Horne - Zeilinger* (GHZ) state $(|0000\rangle + |1111\rangle)/\sqrt{2}$, described in [2], can be seen as an extension to a four-qubit system of the method used

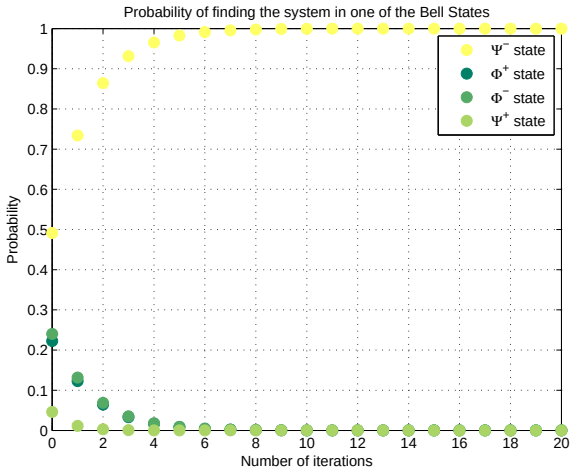


Figure 4.8: Probability of finding the system in one of the Bell-states for 20 iterations of the process ($p = 0.5$)

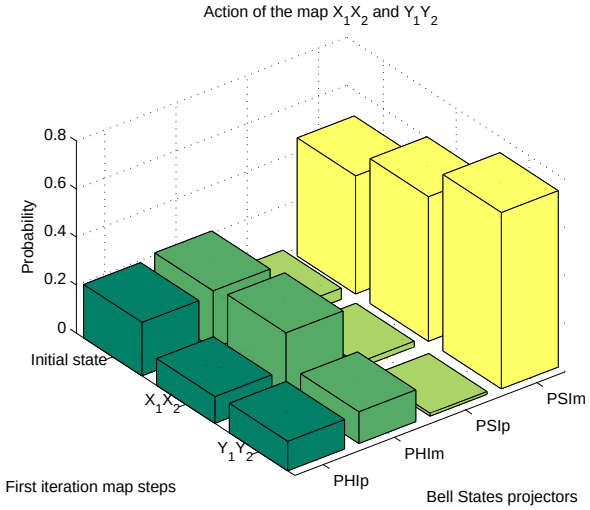


Figure 4.9: Action of the maps X_1X_2 and Y_1Y_2 on the system density matrix after the first iteration ($p = 0.5$)

for the Bell-states cooling, analyzed in section §4.1.2. The aim here is to achieve the projective operator: $\rho_S \mapsto |GHZ\rangle\langle GHZ|$ through stabilizers pumping. First of all, for convenience, we rename the four-qubit system

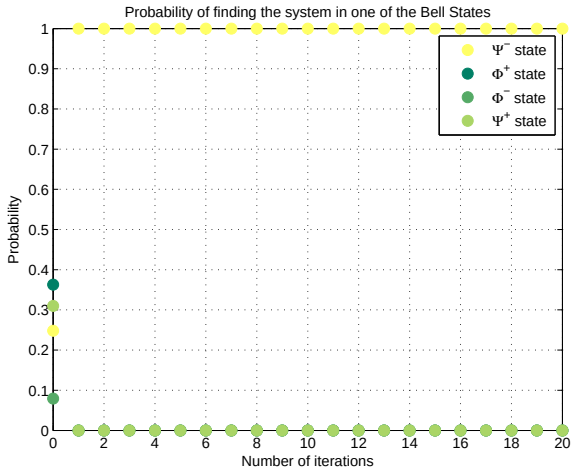


Figure 4.10: Probability of finding the system in one of the Bell-states for 20 iterations of the process ($p = 1$)

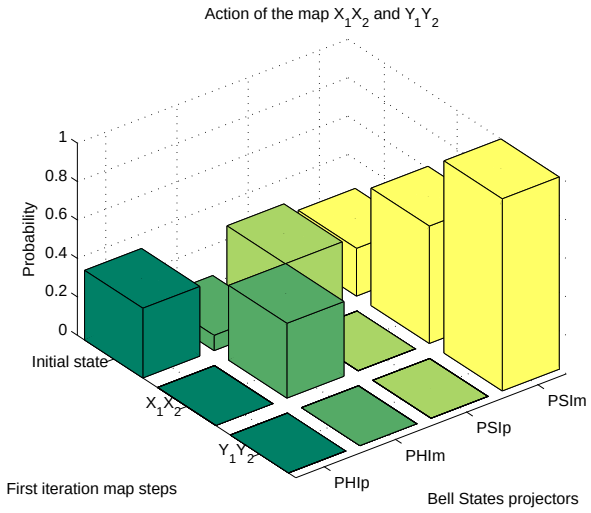


Figure 4.11: Action of the maps X_1X_2 and Y_1Y_2 on the system density matrix after the first iteration ($p = 1$)

orthonormal basis states containing GHZ-state:

$$
\begin{aligned}
|b_{0/1}\rangle &:= \frac{(|0000\rangle \pm |1111\rangle)}{\sqrt{2}}, \quad |b_{2/3}\rangle := \frac{(|0001\rangle \pm |1110\rangle)}{\sqrt{2}}, \\
|b_{4/5}\rangle &:= \frac{(|0010\rangle \pm |1101\rangle)}{\sqrt{2}}, \quad |b_{6/7}\rangle := \frac{(|0011\rangle \pm |1100\rangle)}{\sqrt{2}}, \\
|b_{8/9}\rangle &:= \frac{(|0100\rangle \pm |1011\rangle)}{\sqrt{2}}, \quad |b_{10/11}\rangle := \frac{(|0101\rangle \pm |1010\rangle)}{\sqrt{2}}, \\
|b_{12/13}\rangle &:= \frac{(|0110\rangle \pm |1001\rangle)}{\sqrt{2}}, \quad |b_{14/15}\rangle := \frac{(|0111\rangle \pm |1000\rangle)}{\sqrt{2}}.
\end{aligned}
\tag{4.16}
$$

In this notation $|GHZ\rangle = |b_0\rangle$.

GHZ-state is uniquely characterized as the simultaneous eigenstate of the four stabilizers Z_1Z_2, Z_2Z_3, Z_3Z_4 and $X_1X_2X_3X_4$, all with eigenvalue $+1$. Hence the pumping dynamics into the GHZ-state are realized by four consecutive dissipative steps, each pumping the system into the $+1$ eigenspaces of the four stabilizers. In the table below we list the eigenvalues of each state (4.16) for stabilizer operators Z_1Z_2, Z_2Z_3, Z_3Z_4 and $X_1X_2X_3X_4$.

State	Z_1Z_2	Z_2Z_3	Z_3Z_4	$X_1X_2X_3X_4$
$\|b_0\rangle$=$\|\mathrm{GHZ}\rangle$	$+1$	$+1$	$+1$	$+1$
$\|b_1\rangle$	$+1$	$+1$	$+1$	-1
$\|b_2\rangle$	$+1$	$+1$	-1	$+1$
$\|b_3\rangle$	$+1$	$+1$	-1	-1
$\|b_4\rangle$	$+1$	-1	-1	$+1$
$\|b_5\rangle$	$+1$	-1	-1	-1
$\|b_6\rangle$	$+1$	-1	$+1$	$+1$
$\|b_7\rangle$	$+1$	-1	$+1$	-1
$\|b_8\rangle$	-1	-1	$+1$	$+1$
$\|b_9\rangle$	-1	-1	$+1$	-1
$\|b_{10}\rangle$	-1	-1	-1	$+1$
$\|b_{11}\rangle$	-1	-1	-1	-1
$\|b_{12}\rangle$	-1	$+1$	-1	$+1$
$\|b_{13}\rangle$	-1	$+1$	-1	-1
$\|b_{14}\rangle$	-1	$+1$	$+1$	$+1$
$\|b_{15}\rangle$	-1	$+1$	$+1$	-1

Table 4.1: Eigenvalues of eigenstates $|b_i\rangle$ $(i = 0, ..., 15)$ of stabilizers Z_1Z_2, Z_2Z_3, Z_3Z_4, $X_1X_2X_3X_4$

Returning to the implementation of the process, the first three pumping steps are described by the Kraus map:

$$\rho_S \mapsto \mathcal{E}'(\rho_S) = \rho'_S = E_1\rho_S E_1^\dagger + E_2\rho_S E_2^\dagger \tag{4.17}$$

$$E_1 = \frac{1}{2}(1 + Z_iZ_j) \tag{4.18}$$

$$E_2 = X_j\frac{1}{2}(1 - Z_iZ_j) \tag{4.19}$$

for $i, j = 12, 23, 34$.

The Kraus maps are constructed such that the $+1$ eigenspace of Z_iZ_j is left invariant, whereas a spin flip X_j on the second spin (index j) converts with unitary probability -1 into $+1$ eigenstates. The action of the maps Z_1Z_2, Z_2Z_3 ans Z_3Z_4 on the four-qubit system are shown in Fig 4.12, 4.13 and 4.14 respectively.

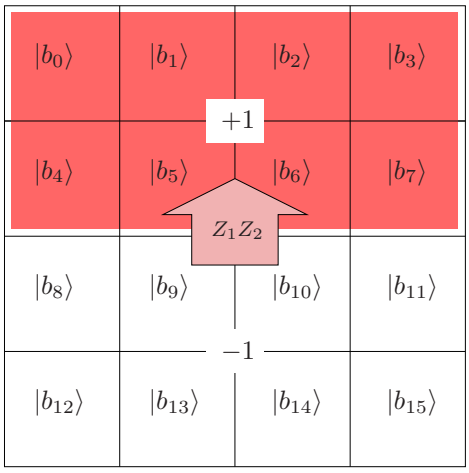


Figure 4.12: Schematic representation of Z_1Z_2 map action

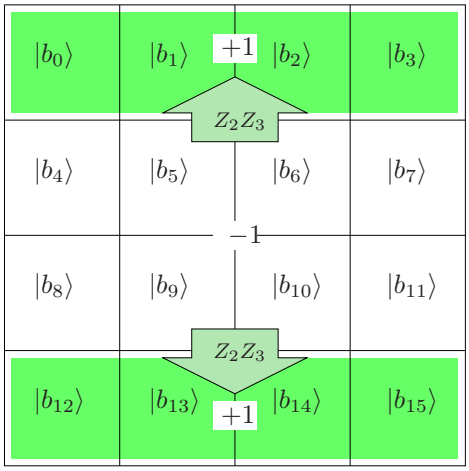


Figure 4.13: Schematic representation of Z_2Z_3 map action

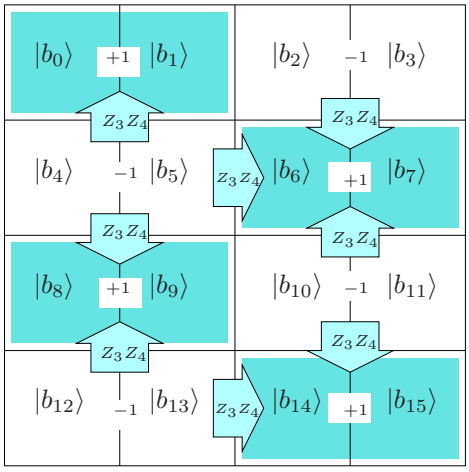


Figure 4.14: Schematic representation of Z_3Z_4 map action

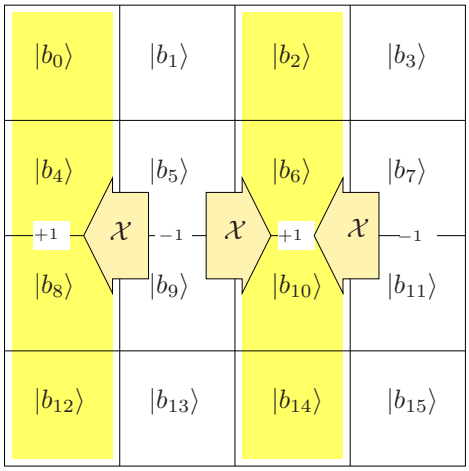


Figure 4.15: Schematic representation of $\mathcal{X} = X_1X_2X_3X_4$ map action

The fourth dissipative step, which realizes pumping into the $+1$ eigenspace of $X_1X_2X_3X_4$, is described by the Kraus map (see also Fig. 4.15):

$$\rho'_S \mapsto \mathcal{E}(\rho_S) = E_1\rho_S E_1^\dagger + E_2\rho_S E_2^\dagger \tag{4.20}$$

$$E_1 = \frac{1}{2}(1 + X_1X_2X_3X_4) \tag{4.21}$$

$$E_2 = Z_4\frac{1}{2}(1 - X_1X_2X_3X_4) \tag{4.22}$$

Fig. 4.16 shows the action of the four maps on the system state.

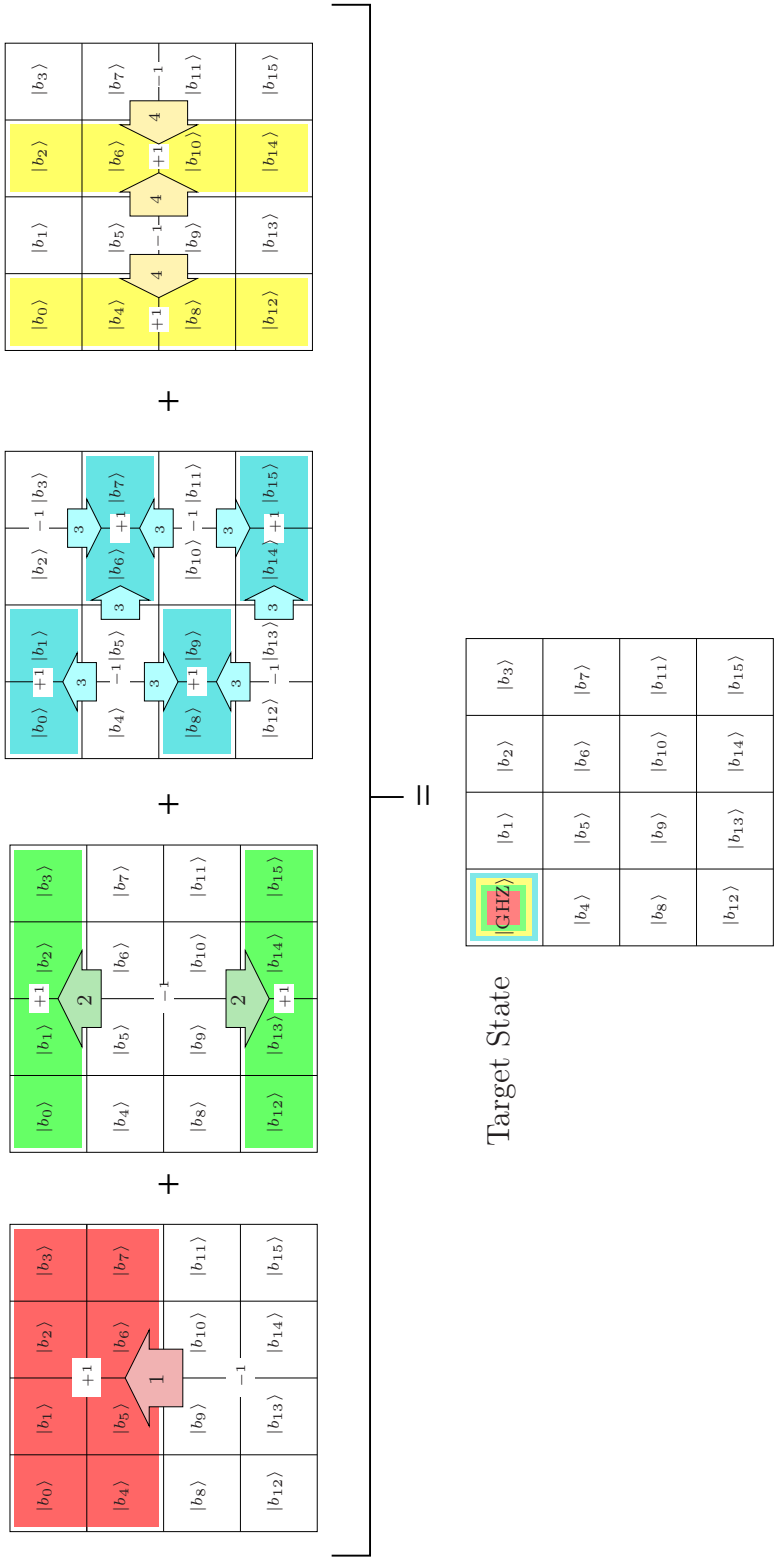


Figure 4.16: Overall action of $Z_1Z_2 + Z_2Z_3 + Z_3Z_4 + X_1X_2X_3X_4$ map

Experimental quantum circuit

The experimental quantum circuit of GHZ state pumping has been engineered using an ancilla qubit and this universal set of operators (experimentally realized through laser techniques):

1. a Mølmer-Sørensen (MS) entangling gate:

$$U_{X^2}(\theta) = \exp\Big(-i\frac{\theta}{4}(\sum_i X_i)^2\Big), \tag{4.23}$$

2. a collective single-qubit rotation gate[6]:

$$U_X(\theta) = \exp\Big(-i\frac{\theta}{2}\sum_i X_i\Big), \tag{4.24}$$

3. a single-qubit rotation Z-gate:

$$U_{Z_i}(\theta) = \exp\Big(-i\frac{\theta}{2}Z_i\Big). \tag{4.25}$$

The experimental implementation of the four dissipative maps described in the previous section is given schematically in Fig. 4.17.

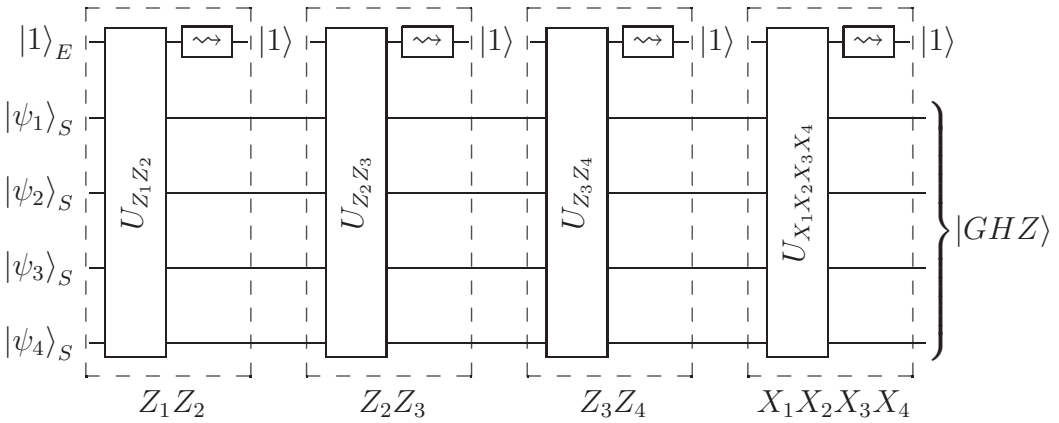


Figure 4.17: Quantum circuit used to generate GHZ-state

Here unitary operators denoted by $U_{Z_1Z_2}$, $U_{Z_2Z_3}$, $U_{Z_3Z_4}$ and $U_{X_1X_2X_3X_4}$ consist of a well-defined sequence of operators (4.23), (4.24) and (4.25).

[6]Shifting the optical phase of the bichromatic light field by $\pi/2$ exchanges X_i by Y_i in these operations.

These gate sequences are explicitly given below[7]:

Step 1: $U_{Z_1Z_2}$ (pumping into the $+1$ eigenspace of Z_1Z_2):

$$\begin{aligned} &U_Y(-\pi/2)U_{Z_2}(-\pi/2)\\ &U_X(-\pi/2)U_{Z_2}(-\pi/2)U_X(-\pi/2)\\ &U_{Z_1}(\pi)U_{X^2}(\pi/4)U_{Z_2}(\pi)U_{Z_0}(\pi)U_{X^2}(\pi/4)\\ &U_X(-\pi/2)U_{Z_2}(-\pi/2)U_{Z_0}(-\pi/2)U_X(\pi/2)\\ &U_{X^2}(\pi/4)U_{Z_4}(\pi)U_{Z_3}(\pi)U_{X^2}(\pi/4)\\ &U_Y(\pi/2)U_X(-\pi/2)U_{Z_0}(-\pi/2)U_X(\pi/2) \end{aligned} \tag{4.26}$$

Step 2: $U_{Z_2Z_3}$ (pumping into the $+1$ eigenspace of Z_2Z_3):

$$\begin{aligned} &U_Y(-\pi/2)U_{Z_3}(-\pi/2)\\ &U_X(-\pi/2)U_{Z_3}(-\pi/2)U_X(-\pi/2)\\ &U_{Z_2}(\pi)U_{X^2}(\pi/4)U_{Z_3}(\pi)U_{Z_0}(\pi)U_{X^2}(\pi/4)\\ &U_X(-\pi/2)U_{Z_3}(-\pi/2)U_{Z_0}(-\pi/2)U_X(\pi/2)\\ &U_{X^2}(\pi/4)U_{Z_4}(\pi)U_{Z_1}(\pi)U_{X^2}(\pi/4)\\ &U_Y(\pi/2)U_X(-\pi/2)U_{Z_0}(-\pi/2)U_X(\pi/2) \end{aligned} \tag{4.27}$$

Step 3: $U_{Z_3Z_4}$ (pumping into the $+1$ eigenspace of Z_3Z_4):

$$\begin{aligned} &U_Y(-\pi/2)U_{Z_4}(-\pi/2)\\ &U_X(-\pi/2)U_{Z_4}(-\pi/2)U_X(-\pi/2)\\ &U_{Z_3}(\pi)U_{X^2}(\pi/4)U_{Z_4}(\pi)U_{Z_0}(\pi)U_{X^2}(\pi/4)\\ &U_X(-\pi/2)U_{Z_4}(-\pi/2)U_{Z_0}(-\pi/2)U_X(\pi/2)\\ &U_{X^2}(\pi/4)U_{Z_2}(\pi)U_{Z_1}(\pi)U_{X^2}(\pi/4)\\ &U_Y(\pi/2)U_X(-\pi/2)U_{Z_0}(-\pi/2)U_X(\pi/2) \end{aligned} \tag{4.28}$$

Step 4: $U_{X_1X_2X_3X_4}$ (pumping into the $+1$ eigenspace of $X_1X_2X_3X_4$):

$$\begin{aligned} &U_X(-\pi/2)\\ &U_{Z_4}(-\pi/2)U_X(\pi/2)U_{Z_4}(-\pi/2)\\ &U_{X^2}(\pi/4)U_{Z_4}(\pi)U_{Z_0}(\pi)U_{X^2}(\pi/4)\\ &U_{Z_4}(-\pi/2)U_X(-\pi/2)U_{Z_0}(-\pi/2)U_X(\pi/2)\\ &U_{X^2}(\pi/4)U_{X^2}(\pi/4) \end{aligned} \tag{4.29}$$

[7]Local rotations of the system ions at the end of a pumping step, which would be compensated at the beginning of the subsequent pumping step, were omitted when several dissipative maps were applied in a row. The corresponding gate operations of the sequences are displayed in blue in Steps 1-3.

Hence the complete circuit decomposition of one pumping cycle involves 16 five-ion entangling operations (4.23), 28 (20) collective unitaries (4.24) and 36 (34) single-qubit operations (4.25), for a total of 80 (70) quantum operations with (without) optional operations in blue.

Numerical simulation

We wrote a **Matlab** program to test the experimental quantum circuit. The complete **Matlab** code can be found in Appendix A.
Here we report the output of the program for a completely mixed initial state and in Fig. 4.18 and 4.19 the evolution of the system after each pumping step.

```
Initial density matrix:

  Columns 1 through 8

    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625
    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625
    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625
    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625
    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625
    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625
    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625
    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625
    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625
    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625
    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625
    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625
    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625
    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625
    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625
    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625

  Columns 9 through 16

    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625
    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625
    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625
    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625
    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625
    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625
    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625
    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625
    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625
    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625
    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625
    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625
    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625
    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625
    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625
    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625

Final density matrix:

    0.5    0    0    0    0    0    0    0    0    0    0    0    0    0    0  0.5
      0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
      0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
      0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
      0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
      0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
      0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
      0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
      0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
      0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
      0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
      0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
      0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
      0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
      0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0.5    0    0    0    0    0    0    0    0    0    0    0    0    0    0  0.5
```

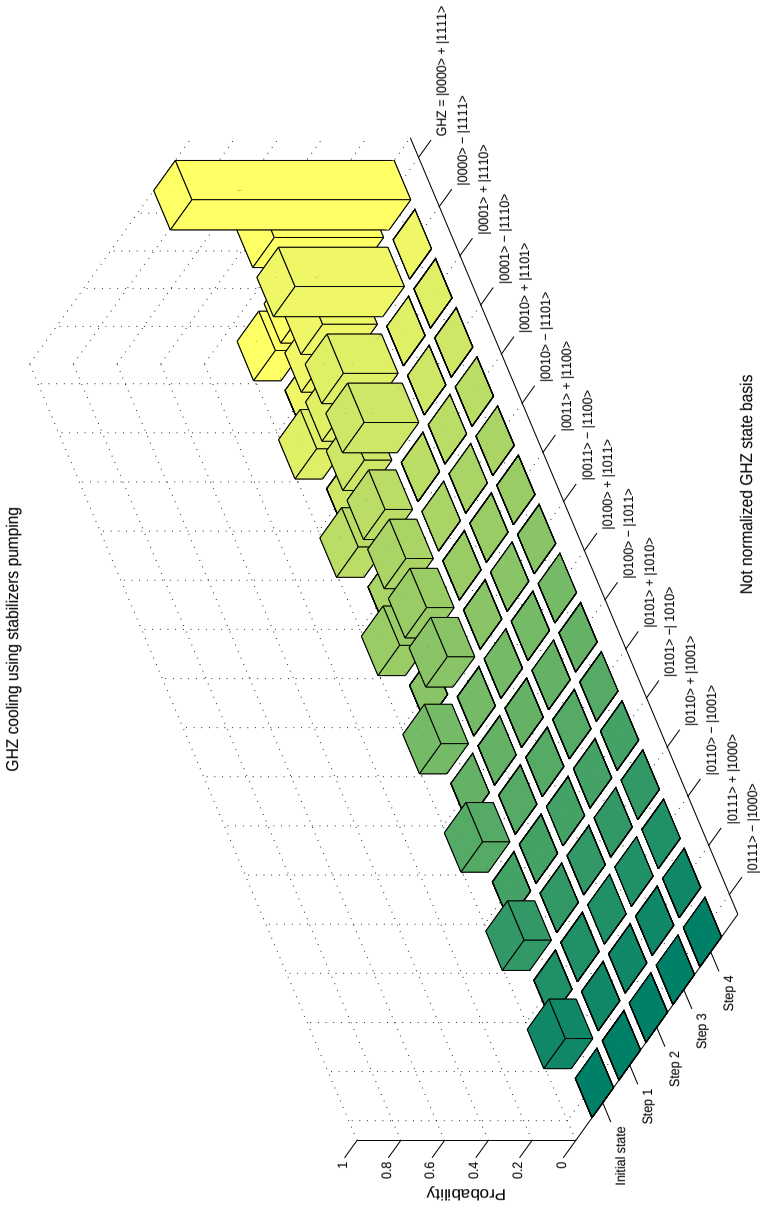
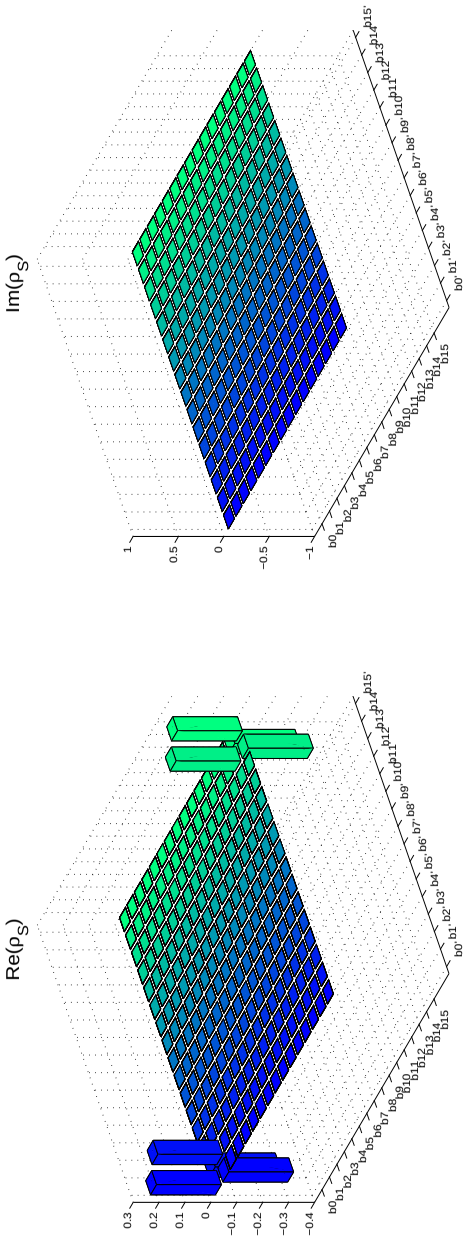


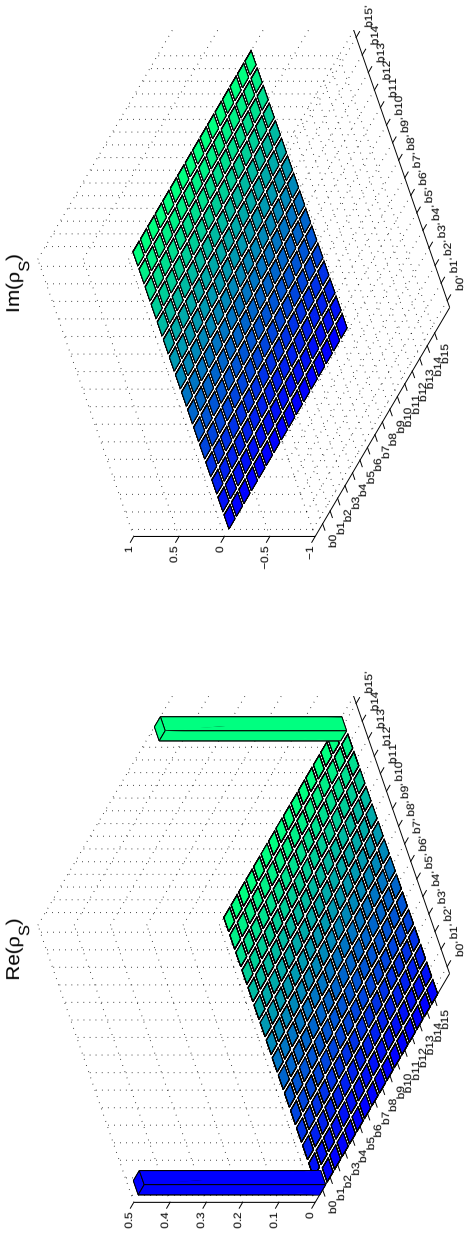
Figure 4.18: Probability of finding the system in one of the GHZ states basis after each pumping step

(a) Initial density matrix

(b) First step



(c) Second step



(d) Third step

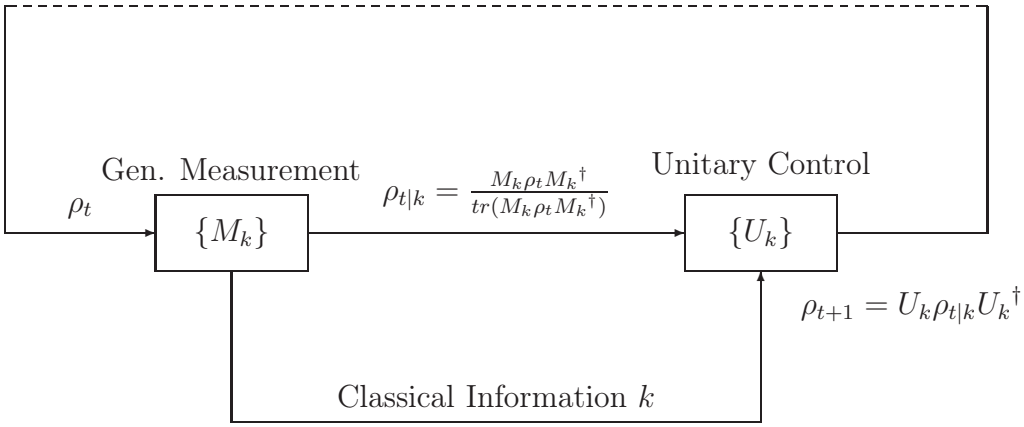
(e) Fourth step

Figure 4.19: Evolution of real and imaginary part of system density matrix: (a) initial completely mixed state, (b) pumping into the $+1$ eigenspace of Z_1Z_2, (c) pumping into the $+1$ eigenspace of Z_2Z_3, (d) pumping into the $+1$ eigenspace of Z_3Z_4, (e) pumping into the $+1$ eigenspace of $X_1X_2X_3X_4$

4.2 An elementary quantum feedback control approach

4.2.1 Description of the control strategy

The goal of this section is to illustrate an alternative implementation of the solutions proposed in [2] that employs the simple feedback control circuit previously analyzed in chapter 3. A scheme of this circuit can be seen below.



According to the result obtained with a set of generalized measurements $\{M_k\}$[8], the corresponding unitary operator $\{U_k\}$ is applied to the system, resulting in a discrete-time evolution of the system ρ as described by:

$$\rho(t+1) = \sum_k U_k M_k \rho(t) {M_k}^\dagger {U_k}^\dagger \tag{4.30}$$

In the next paragraphs we try to find two sets of operators $\{M_k\}$ and $\{U_k\}$ that implement the maps described in the previous section for both Bell-state and GHZ-state "cooling" processes.

[8]We recall that a set of operators $\{M_1, M_2, .., M_m\}$ on a Hilbert space $\mathcal{H}$ is called *generalized measurement operators* if they satisfy the condition: $\sum_{i=1}^{m} {M_i}^\dagger M_i = I$.
A general measurement on the system for these operators is a process such that when the system is in state $|\psi\rangle$:
(i) the result i is obtained with probability $p_i = \langle\psi|M_i^\dagger M_i|\psi\rangle$,
(ii) the state collapses to $\frac{1}{\sqrt{p}} M_i|\psi\rangle$.

4.2.2 Bell-state $|\Psi^-\rangle$ generation

Application of the control strategy:

(i) A trivial case: $p = 1$

For a unitary probability of pumping it's trivial to find the set of generalized measurements M_1 and M_2. In fact, in this case the maps (4.10), (4.11) can be rewritten as:

$$E_1 = Y_1 \frac{1}{2}(1 + X_1X_2) = Y_1 \cdot \Pi_{+1} \tag{4.31}$$

$$E_2 = \frac{1}{2}(1 - X_1X_2) = \Pi_{-1} \tag{4.32}$$

where Π_{+1} denotes the "bad" projective operator for the $+1$ eigenspace of X_1X_2 and Π_{-1} is the "good" one. So in this simple case M_1,M_2 are the projective or von Neumann's mesurements[9] Π_{+1} and Π_{-1}. The unitary operators U_1,U_2 in this case are respectively Y_1 and I, where I is the identity operator. For the maps (4.13), (4.14) the reasoning is analogous and we can write: $\{M_1', M_2'\} = \{\Pi_{+1}', \Pi_{-1}'\}$ and $\{U_1', U_2'\} = \{X_1, I\}$, where we define $\Pi_{+1}' = 1 + Y_1Y_2$ and $\Pi_{-1}' = 1 - Y_1Y_2$. The implemented control process can be described as a *dead-beat control*[10].

A possible experimental quantum circuit that implements the feedback control strategy is shown in fig 4.20.

In step (i) we apply the operator $X \otimes \Pi_{+1} + I \otimes \Pi_{-1}$, so that the information about which of the two eigenspaces the system is in is stored in an ancillary qubit (i.e. if the system state is in $+1$ eigenspace of X_1X_2, or Y_1Y_2 for the second map, then the ancilla qubit is flipped).

Then in (*ii*) the ancilla qubit is measured and according to the result a Y gate (or a X gate for second map) is applied to the first qubit of the system.

[9] We recall that (see paragraph §2.2) a *Projective or von Neumann's measurement* is a set of mutually orthogonal projection operators $\{\Pi_1, \Pi_2, ..., \Pi_m\}$ which complete to identity, i.e., $\Pi_i\Pi_j = \delta_{i,j}\Pi_j$ and $\sum_{i=1}^m \Pi_i{}^\dagger\Pi_i = I$.
When this measurement is carried out on a system with state $|\psi\rangle$ then:
(i) the result i is obtained with probability $p_i = \langle\psi|\Pi_i^\dagger\Pi_i|\psi\rangle$,
(ii) the state collapses to $\frac{1}{\sqrt{p}}\Pi_i|\psi\rangle$.

[10] The dead-beat controller aims for the best response possible to a set-point change. Qualitatively, this means that following a set-point change, and after a time period equal to the system time-delay, the output should be at set-point and remain there.

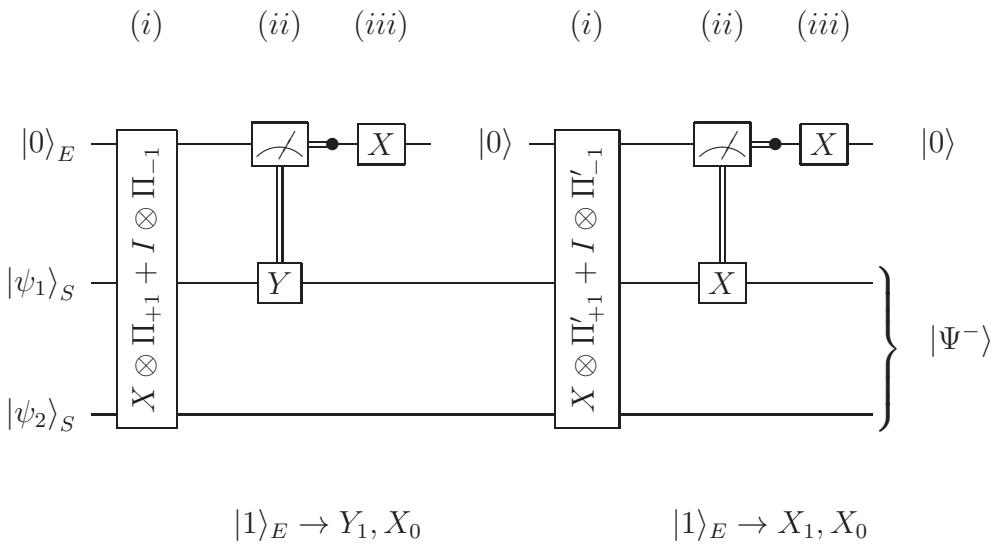


Figure 4.20: Scheme of a quantum circuit realizing Bell-state $|\Psi^-\rangle$ generation through feedback control strategy

Finally in (*iii*) the ancilla qubit is reset to $|0\rangle$ through a X gate conditioned on the ancilla measurement result.

We make use of an ancilla qubit, instead of simply making the projective measurement on the system and then acting according to the result, because in this case it's possible to engineer the first quantum gate using a MS gate.

(ii) The general case: $p \neq 1$

If the probability of pumping is not unitary, in choosing a suitable set of measurement $\{M_1, M_2\}$ we must include the value of the desired probability p.

First of all we rewrite (4.10), (4.11) in the new notation:

$$E_1 = \sqrt{p} \cdot Y_1 \cdot \Pi_{+1} \tag{4.33}$$

$$E_2 = \Pi_{-1} + \sqrt{1-p} \cdot \Pi_{+1} \tag{4.34}$$

For the map $X_1X_2(p)$ we can choose $M_1 = \sqrt{p} \cdot \Pi_{+1}$ and $M_2 = \Pi_{-1} + \sqrt{1-p} \cdot \Pi_{+1}$, as they satisfy the completeness relation. In this way the set of unitary operators is still the same: $\{U_1, U_2\} = \{Y_1, I\}$.

For the map $Y_1Y_2(p)$ we have:

$$E'_1 = \sqrt{p} \cdot X_1 \cdot \Pi'_{+1} \tag{4.35}$$
$$E'_2 = \Pi_{-1} + \sqrt{1-p} \cdot \Pi'_{+1}. \tag{4.36}$$

So, in analogy with the previous case, we can write $\{M'_1, M'_2\} = \{\sqrt{p} \cdot \Pi'_{+1}, \Pi'_{-1} + \sqrt{1-p} \cdot \Pi'_{+1}\}$ and $\{U'_1, U'_2\} = \{X_1, I\}$.

In this general case, the problem of finding an experimental circuit which performs the measurements M_1, M_2 and M'_1, M'_2 gets trickier and the availability of such an experimental implementation depends on the state of-the-art in the field of experimental quantum computing and simulation.

MS gate analysis

In this paragraph we try to study the Mølmer-Sørensen gate (MS for simplicity) in detail.

The MS gate is often used in the experimental field of quantum information processing and computation with trapped ions.

The MS entangling gate (see also [18]) is based on pairwise two-ion interaction terms and can be parametrized by two angles θ and ϕ:

$$U_{MS}(\theta, \phi) = \exp\Big(- i\frac{\theta}{4}(\cos\phi S_x + \sin\phi S_y)^2\Big), \tag{4.37}$$

The sum in the collective spin operators $S_x = \sum_{i=0}^{n} X_i$ and $S_y = \sum_{i=0}^{n} Y_i$ with X, Y the usual Pauli matrices, is understood to be performed over all ions involved in the gate.
On the rest of the work we suppose $\phi = 0$, in this way the relation (4.37) can be rewritten as:

$$U_{X^2}(\theta) = \exp\Big(- i\frac{\theta}{4}\Big(\sum_{i=0}^{n} X_i\Big)^2\Big). \tag{4.38}$$

In the experimental quantum circuit proposed in [2] it's used an MS gate with a phase angle $\theta = \pi/2$. In this case the $U_{X^2}(\pi/2)$ operator can be decomposed in a more explicit form as:

$$U_{X^2}\Big(\frac{\pi}{2}\Big) = U'_X(I \otimes \Pi_{-1} + X \otimes \Pi_{+1}), \tag{4.39}$$

where U'_X is a unitary 8×8 matrix of the form:

$$U'_X = \frac{1}{2} e^{-i\pi/8} \begin{bmatrix} 1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 \\ -1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 \\ -1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & -1 & -1 \\ 0 & 0 & 0 & 0 & -1 & 1 & -1 & -1 \\ 0 & 0 & 0 & 0 & -1 & -1 & 1 & -1 \\ 0 & 0 & 0 & 0 & -1 & -1 & -1 & 1 \end{bmatrix}. \tag{4.40}$$

In particular U'_X can be rewritten as:

$$U'_X = I \otimes \frac{1}{2} e^{-i\pi/8} \begin{bmatrix} 1 & -1 & -1 & -1 \\ -1 & 1 & -1 & -1 \\ -1 & -1 & 1 & -1 \\ -1 & -1 & -1 & 1 \end{bmatrix} = I \otimes U''_X. \tag{4.41}$$

The action of the unitary operator U''_X on the Bell states basis (4.1) is described as:

$$\begin{aligned} U''_X|\Phi^+\rangle &= -\frac{1}{\sqrt{2}} e^{-i\pi/8} \cdot |\Psi^+\rangle \\ U''_X|\Phi^-\rangle &= \frac{1}{\sqrt{2}} e^{-i\pi/8} \cdot |\Phi^-\rangle \\ U''_X|\Psi^+\rangle &= -\frac{1}{\sqrt{2}} e^{-i\pi/8} \cdot |\Phi^+\rangle \\ U''_X|\Psi^-\rangle &= \frac{1}{\sqrt{2}} e^{-i\pi/8} \cdot |\Psi^-\rangle \end{aligned}$$

Then U''_X is not harmful for our purpose since it swaps the Bell-states in $+1$ eigenspace of X_1X_2 and does not change, except for a constant, the other Bell-states in the -1 eigenspace. So In the Bell U'' basis, $\mathcal{B}_{Bell} = \{|\Phi^+\rangle, |\Psi^+\rangle, |\Phi^-\rangle, |\Psi^-\rangle\}$, can be written as:

$$U''_{X,Bell} = \frac{1}{\sqrt{2}} e^{-i\pi/8} \left[\begin{array}{c|c} X & O \\ \hline O & I \end{array}\right], \tag{4.42}$$

where symbol O denotes a 2×2 matrix of zeros.

Finally we have obtained a decomposition of the MS gate $U_{X^2}(\pi/2)$ in a form that includes two terms:

1. The operator $U_\bigstar = I \otimes \Pi_{-1} + X \otimes \Pi_{+1}$ that stores in the ancilla qubit the information about in which of the two eigenspaces the system is.

2. An additional operator U'_X: a unitary matrix not harmful for our purposes.

The MS gate $U_{Y^2}(\pi/2)$, used to implement the dissipative map Y_1Y_2, can be similiarly decomposed in the form:

$$U_{Y^2}\left(\frac{\pi}{2}\right) = U'_Y(Z \otimes \Pi'_{-1} + X \otimes \Pi'_{+1}), \tag{4.43}$$

where in this case unitary operator U'_Y stands for:

$$U'_Y = \frac{1}{2}e^{-i\pi/8}\begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ -1 & 1 & -1 & 1 & 0 & 0 & 0 & 0 \\ -1 & -1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & -1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 \\ 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 \\ 0 & 0 & 0 & 0 & -1 & 1 & 1 & -1 \end{bmatrix}, \tag{4.44}$$

U'_Y can be rewritten in the form:

$$U'_Y = Z \otimes \frac{1}{2}e^{-i\pi/8}\begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & 1 & -1 & 1 \\ -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 \end{bmatrix} = Z \otimes U''_Y. \tag{4.45}$$

Even in this case the two operators Z, acting on the ancilla qubit, and U''_Y, acting on system qubits, do not affect the desired result as Z gate does not change the outcome of the measurement in the standard basis of the ancilla qubit and the unitary operator U''_Y swaps the $+1$ Bell eigenstates of the stabilizer Y_1Y_2 and does not change the -1 Bell eigenstates.

In conclusion we have found a decomposition of the MS gate $U_{Y^2}(\pi/2)$ which involves:

1. The operator $U_\bigstar = Z \otimes \Pi'_{-1} + X \otimes \Pi'_{+1}$ that stores in the ancilla qubit the information about in which of the two eigenspaces of stabilizer Y_1Y_2 the system is.

2. An additional operator U'_Y: a unitary matrix that does not affect the relevant action of $U_\bigstar$ operator, since it maps the $+1$ and -1 eigenspaces into themselves.

An alternative quantum circuit implementation

A quantum circuit that implements the elementary quantum feedback control for the Bell-state $|\Psi^-\rangle$ "cooling" process with a probability of pumping $p = 1$ using MS gates $U_{X^2}(\pi/2)$ and $U_{Y^2}(\pi/2)$ is shown in Fig. 4.21.

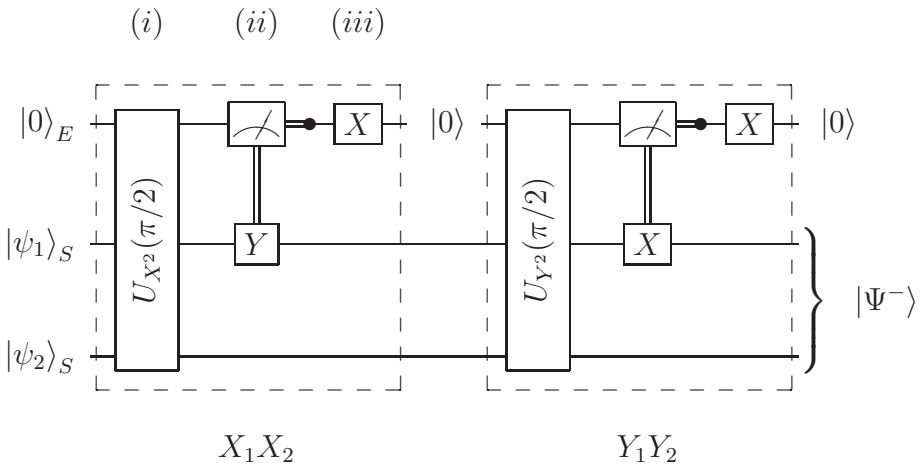


Figure 4.21: Quantum circuit for Bell-state $|\Psi^-\rangle$ generation using feedback control strategy

Here the main steps related to X_1X_2 map:

(i) Information about whether the system is in the "good" (-1) or "bad" (+1) eigenspace of X_1X_2 is mapped by the MS gate onto the logical states 0,1 of the ancilla qubit.

(ii) A measurement is made on the ancilla qubit: if it collapses on the $|1\rangle$ state then an Y_1 operator is applied to the system, otherwise nothing is made.

(iii) The ancilla qubit is reset to $|0\rangle$ through the application, which depends on the outcome of the measurement, of a X gate on the ancilla.

By comparing the experimental quantum circuit engineered by Barreiro *et al.* in [2] (Fig. 4.5) with the circuit analyzed here (Fig. 4.21), the main difference is that using an elementary feedback control strategy we save two MS gates. Furthermore, if we consider that the real circuit used in the experiment consists of 18 unitary operations and 2 dissipative steps (see supplementary informations of [2]) this difference rises to 14 gates[11].

[11] In the count of gates we assumed the dissipative step of ancilla qubit resetting

4.2.3 GHZ-state generation

Application of the control strategy

In this case the Kraus operators of the first three dissipative maps (4.18), (4.19) can be rewritten as:

$$E_1 = \Pi_{ij,+1}, \tag{4.46}$$
$$E_2 = X_j \cdot \Pi_{ij,-1} \quad ij = 12, 23, 34, \tag{4.47}$$

where the projective operators $\Pi_{ij,+1}$ and $\Pi_{ij,-1}$ stand for $\frac{1}{2}(1 + Z_i Z_j)$ and $\frac{1}{2}(1 - Z_i Z_j)$. Hence the two sets of operators $\{M_k\}$ and $\{U_k\}$, $k = 1, 2$, are given respectively by: $\{\Pi_{ij,+1}, \Pi_{ij,-1}\}$ and $\{I, X_j\}$, for $ij = 12, 23, 34$.

For the fourth pumping step $X_1X_2X_3X_4$, described by Kraus operators (4.21) and (4.22), we can write:

$$E_1 = \Pi'_{+1}, \tag{4.48}$$
$$E_2 = Z_4 \cdot \Pi'_{-1}, \tag{4.49}$$

here the von Neumann's operators Π'_{+1} and Π'_{-1} are the projections onto +1 and -1 eigenspaces of stabilizer $X_1X_2X_3X_4$: $\frac{1}{2}(1 + X_1X_2X_3X_4)$ and $\frac{1}{2}(1 - X_1X_2X_3X_4)$. So, similarly to previous cases, we can choose: $\{M_1, M_2\} = \{\Pi'_{+1}, \Pi'_{-1}\}$ and $\{U_1, U_2\} = \{I, Z_4\}$.

An alternative quantum circuit implementation

The measurement steps for the first three maps, described by the set of projective operators $\{\Pi_{+1,ij}, \Pi_{-1,ij}\}$, for $ij = 12, 23, 34$, can be engineered through MS gates acting on the system qubit i, j and on an ancilla qubit[12] and a few more unitary gates. More specifically MS gate $U_{Y^2,0ij}(\pi/2)$[13] can be decomposed as:

$$U_{Y^2,0ij}(\pi/2) = U'_{0ij} \cdot (I \otimes \Pi_{+1,ij} + X \otimes \Pi_{-1,ij}), \tag{4.50}$$

equivalent, in term of experimental implementation times, to a unitary gate acting on the ancilla.

[12]MS gate acting on 3 of 5 ions can be realized through different ways : (i) by focusing the driving laser of the MS gate operation only onto the relevant subset of ions, or (ii) by hiding the electronic population of these ions in uncoupled electronic states for the duration of the gate sequence, or (iii) by means of refocusing techniques. For more details see [18].

[13]Subscripts $0, i, j$ stands for qubits (0 is the ancilla) on which the MS operator acts.

unitary operator U'_{0ij} can be written as a sequence of gates:

$$U'_{0ij} = U''_{0ij} \cdot U_{X^2,0ij}(-\pi/2)U_{Y^2,0ij}(-\pi/4)U_{Z,0}(\pi)U_{Y^2,0ij}(\pi/4) \\ U_{X^2,0ij}(\pi/4)U_{Z,0}(\pi)U_{X^2,0ij}(\pi/4) \tag{4.51}$$

where U''_{0ij} written in matrix form is:

$$U''_{0ij} = \frac{1}{2}e^{-i\pi/8}\begin{bmatrix} 0 & 0 & 0 & 0 & -1 & i & i & 1 \\ 0 & 0 & 0 & 0 & i & -1 & 1 & i \\ 0 & 0 & 0 & 0 & i & 1 & -1 & i \\ 0 & 0 & 0 & 0 & 1 & i & i & -1 \\ -1 & i & i & 1 & 0 & 0 & 0 & 0 \\ i & -1 & 1 & i & 0 & 0 & 0 & 0 \\ i & 1 & -1 & i & 0 & 0 & 0 & 0 \\ 1 & i & i & -1 & 0 & 0 & 0 & 0 \end{bmatrix} = X \otimes U'''_{ij}, \tag{4.52}$$

Finally decomposition of operator U'''_{ij} gives:

$$U'''_{ij} = \frac{1}{2}e^{-i\pi/8}\begin{bmatrix} -1 & i & i & 1 \\ i & -1 & 1 & i \\ i & 1 & -1 & i \\ 1 & i & i & -1 \end{bmatrix} = \\ = e^{-i\pi/8} \cdot U_{X^2,ij}(\pi/2)U_{X,ij}(\pi/2)U_{X^2,ij}(-\pi/2). \tag{4.53}$$

In conclusion, after some rearrangements of equations (4.50), (4.51) and (4.53), the projective operator $I \otimes \Pi_{+1,ij} + X \otimes \Pi_{-1,ij} = {U'_{0ij}}^{\dagger} \cdot U_{Y^2,0ij}(\pi/2)$ can be replaced, for example, by the sequence[14] (see also Fig. 4.22):

$$U_{X^2,0ij}(-\pi/4)U_{Z,0}(-\pi)U_{X^2,0ij}(-\pi/4)U_{Y^2,0ij}(-\pi/4) \\ U_{Z,0}(-\pi)U_{Y^2,0ij}(\pi/4)U_{X^2,0ij}(\pi/2)U_{X,0}(\pi) \\ U_{X^2,ij}(\pi/2)U_{X,ij}(-\pi/2)U_{X^2,ij}(-\pi/2)U_{Y^2,0ij}(\pi/2) \tag{4.54}$$

The measurement step of the fourth map $X_1X_2X_3X_4$ can be engineered through a MS gate $U_{X^2,01234}(\pi/2)$. In fact it has a decomposition:

$$U_{X^2,01234}(\pi/2) = U'_{01234} \cdot (I \otimes \Pi'_{+1} + X \otimes \Pi'_{-1}), \tag{4.55}$$

where the operator $U'_{01234} = I \otimes U''_{1234}$ and it can be demonstrated that U''_{1234} does not affect +1 and -1 eigenstates of stabilizer $X_1X_2X_3X_4$.

A scheme of the final circuit is shown in Fig. 4.23.

[14] In writing the final sequence we use matrix properties: (i) $(AB)^{\dagger} = B^{\dagger}A^{\dagger}$, (ii) $U(\theta)^{\dagger} = U(-\theta)$.

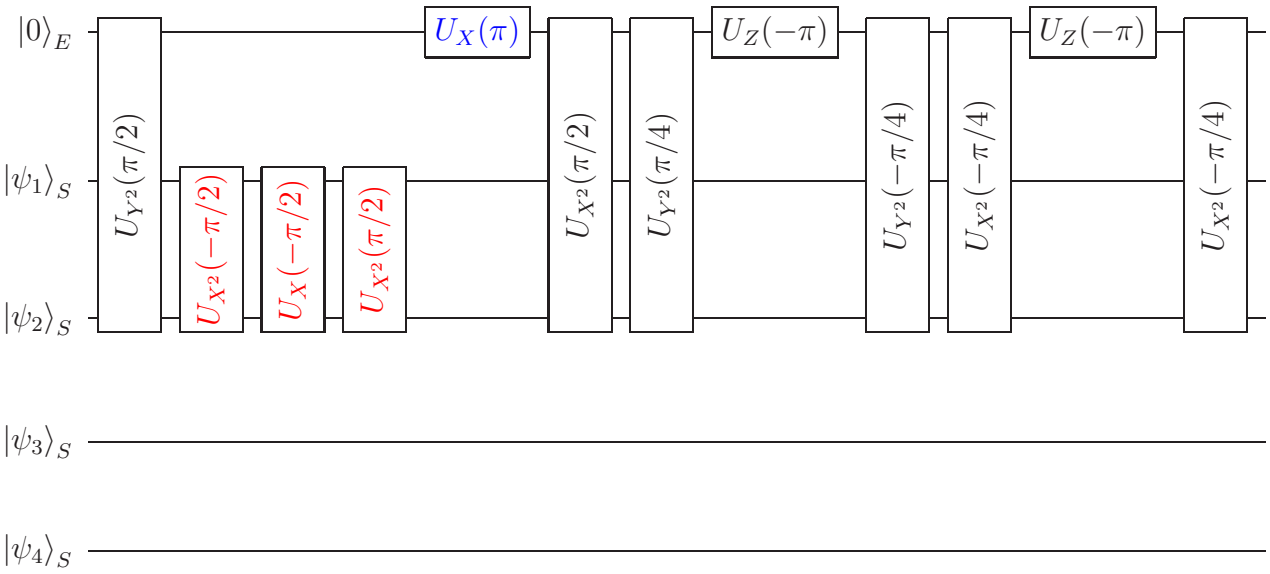


Figure 4.22: Quantum circuit implementing operator $U_{Z_1Z_2} = I \otimes \Pi_{+1,12} + X \otimes \Pi_{-1,12}$

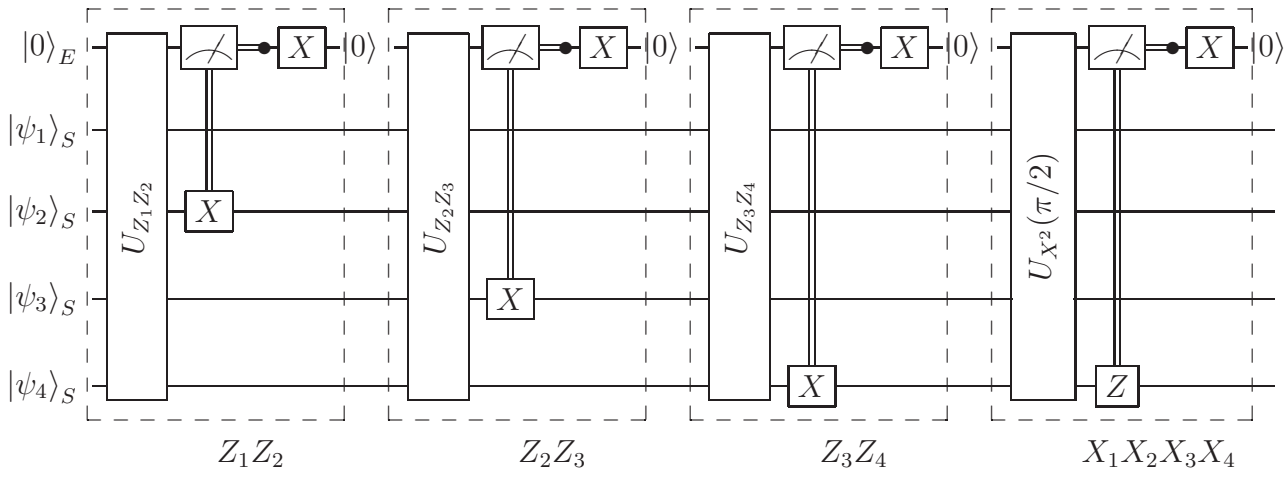


Figure 4.23: Quantum circuit for GHZ-state generation using feedback control strategy (gate $U_{Z_iZ_j}$ stands for gate sequence (4.54))

Assuming that unitary operations used in sequence (4.54) are experimentally available[15], in this new formulation of the circuit we use 41 gates with a saving of 39 (29 after simplifications) gates in comparison to the circuit engineered by Barreiro *et al.* (Fig. 4.17).

4.3 Experimental implementation times: a brief analysis

In this section we focus on the implementation times required for the experimental realization of the previously described operations in a $^{40}Ca^{+}$-ion traps architecture (see also Appendix B). This analysis does not pretend to be accurate or rigorous, but it must be viewed as an attempt to compare the two control strategies described in the work and, consequently, to provide some reasons regarding why the solution we study can be of potential interest in practice. In particular in the rest of the analysis we refer to the two strategies concerning Bell state preparation.

First of all we list the experimental times for the operations used in the two strategies, most of these informations can be found in the scientific works of the *Quantum Optics and Spectroscopy Group* directed by Rainer Blatt of the Innsbruck University:

- 3-qubit Mølmer-Sørensen gate $U_{X^2}(\pi/2)$ and $U_{Y^2}(\pi/2)$ (fidelity[16] 99%): $t_{MS} \simeq 100\ \mu s$ ([15]);
- Single qubit gates (Rabi flops, fidelity >99%): $t_S \simeq 1 - 10\ \mu s$ ([11]);
- State inizialization:
  - optical pumping (fidelity 99%): $t_{init} \simeq 70\ \mu s$ ([15]);
  - frequency resolved optical pumping (fidelity >99.8%): additional 500 μs ([15]);

[15]Probably this hypothesis fails because some gates used in the sequence, e.g. $U_{X^2,ij}$, could not be available in practice or may require additional operations. Nevertheless it is also possible that exist a better experimental implementation of operators $U_{Z_iZ_j}$. In any case our aim here is not to exactly calculate the number of required gates but to show a possible (maybe better) alternative implementation of the circuit.

[16]Fidelity can be seen as a measure of the agreement between the actual outcome of a quantum operation and the desired state, and it will in general depend on the initial state on which the operation is applied. Since the operation may generally be applied to any arbitrary state, it is natural to average the fidelity over all pure initial states, chosen uniformly in the system Hilbert space.

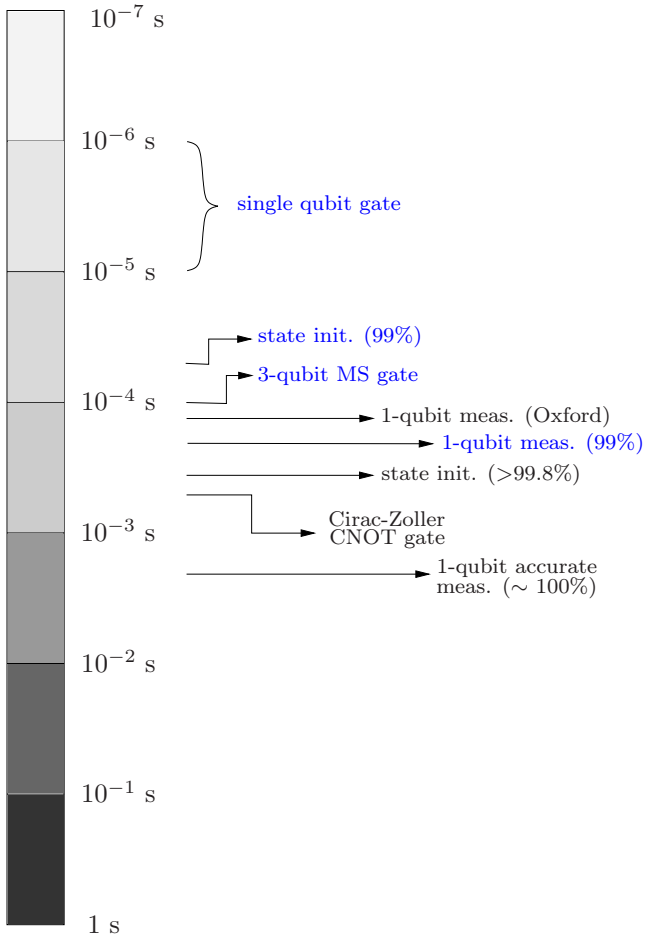


Figure 4.24: Schematic view of quantum operations implementation times in ion-traps architecture (operations in blue are going to be used in the following analysis)

- Single qubit measurement (fluorescence detection):
  - accurate measurements (used in final steps, fidelity $\sim$ 100%): $t_{m1} \simeq 3.5\ ms$ ([23]);
  - fast measurements (if the result has to be used for immediately following operations, as in our case; fidelity 99%): $t_{m2} \simeq 300\ \mu s$ ([23]);
  - in June 2008 the *Ion Trap Quantum Computing Group* of the University of Oxford developed a technique for fast high-fidelity readout of trapped ion qubits (fidelity 99.991%): $t_{m3} = 145\ \mu s$ ([11, 19]).

In Fig. 4.24 it is shown a schematic summary of currently achieved implementation times for these and other operations in trapped $^{40}Ca^{+}$-ion quantum computing settings.

The experimental quantum circuit described in the supplementary informations section of [2] consists of: 4 MS gate $U_{X^2}(\pi/2)$, 2 collective single-qubit gates $U_Y(\pi/2)$, 3 single-qubit Z gates and 1 ancilla qubit resetting operation to cool the system. Using the informations listed above and considering, in first approximation, negligible the times required for single-qubit gates, we obtain a total process time: $t_{tot,1} \simeq 4 \cdot t_{MS} + t_{init} \simeq 470\ \mu s$.

In the elementary feedback control strategy we use: 1 MS gate $U_{X^2}(\pi/2)$, 1 ancilla qubit measurement, 2 single-qubit conditioned X gates. Neglecting, as previously, times of single-qubit rotations and assuming a measurement time $t_{m2} = 300\ \mu s$, we achieve a process time: $t_{tot,2} \simeq t_{MS} + t_{m2} \simeq 400\ \mu s$.

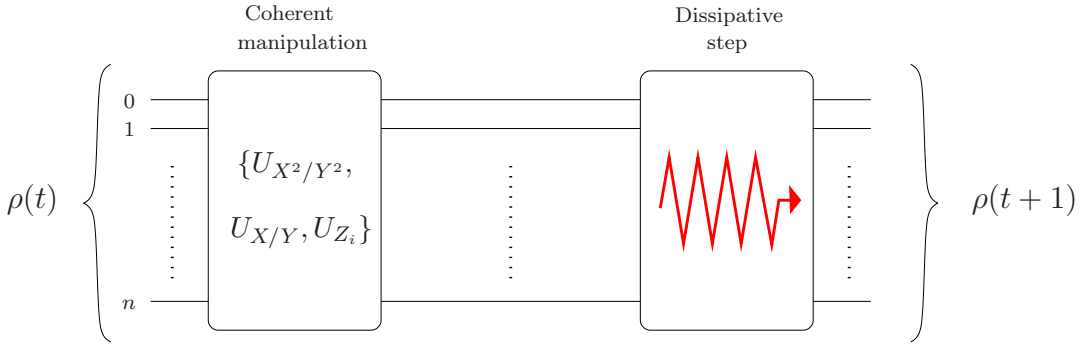
In conclusion we can state that the total process times, $t_{tot,1}$ and $t_{tot,2}$, are approximately the same in both cases. Furthermore if fast measurements are employed the feedback control strategy requires even less time.

4.4 Conclusions

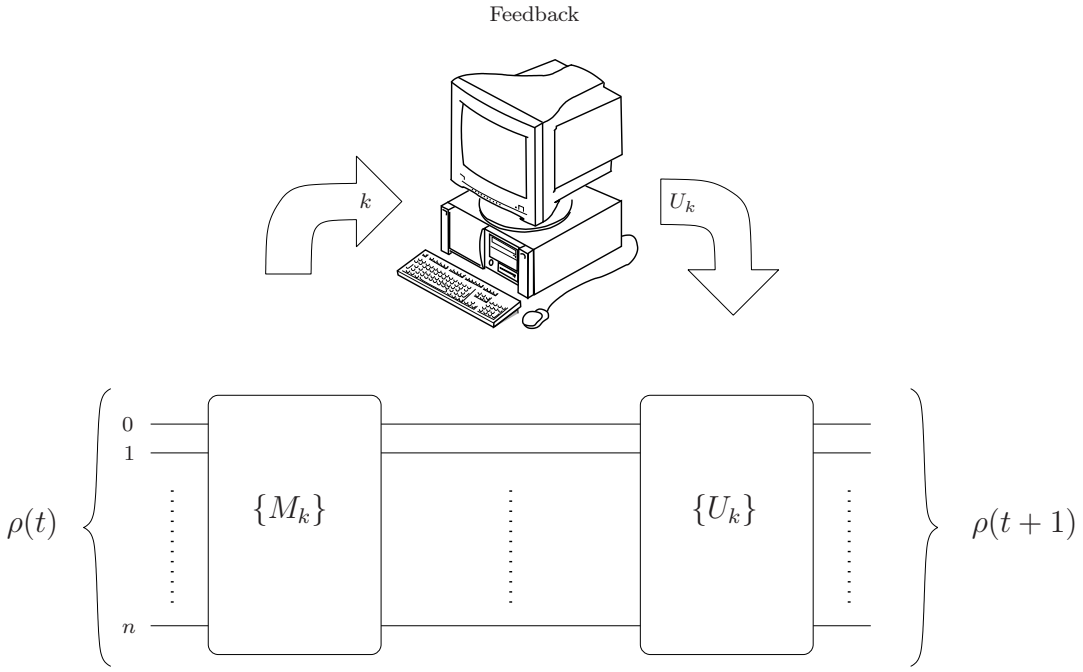
In this chapter we have analyzed and compared two control strategies for quantum state preparation: the first one used in [2] can be called "coherent" control, the second one feedback control.

The first strategy involves the following steps: (i) "coherent" manipulation of the system using unitary gates, (ii) final dissipative step to "cool" the system qubits and obtain the desired discrete-time evolution. The second one instead has as a first step the measurement of the system realized through a set of Kraus operators $\{M_k\}$, then accordingly to the outcome of this measurement it's applied to the system the appropriate unitary gate in the set $\{U_k\}$. These two strategies are summarized in Fig. 4.25.

In the cases analyzed (Bell-state and GHZ-state generation) we have found two alternative quantum circuits implementing a feedback control strategy. The great advantage of our implementation is in the number of operations (unitary gates, measurements and state reinitializations) required: 8 instead of 20 for Bell-state generation and 45 instead of 84 (74 after circuit simplifications) for GHZ-state generation. Nevertheless, even though implementation times of required operations for Bell-state preparation seem to confirm the positive result, this advantage must be verified in term of current experimental capabilities.



(a) "Coherent" control



(b) Feedback control

Figure 4.25: Schematic representation of the two control strategy analyzed in this work: (a) "coherent" control strategy, (b) feedback control strategy.

A

Matlab Codes

We list here the **Matlab** codes realized to simulate quantum circuits described in chapter 4.

It is worth noticing that in builiding the programs we used the auxiliary functions[1]:

1. `TrX()` that calculates the partial trace of a matrix;
2. `randRho()` that returns a random density matrix of dimension specified in the arguments;
3. `killtiny()` that sets a scalar or a matrix that have very small magnitude to exactly zero;
4. `tensor()` that returns tensor product of two or more arguments (a generalization of built-in **Matlab** function `kron()`);
5. `MSGate(pauli,alpha)` that calculates the 5-qubits Mølmer- Sørensen gate $U_{pauli^2}(alpha/4)$;
6. `UGate(pauli,alpha)` that calculates the 5-qubits collective single-qubit rotation gate $U_{pauli}(alpha/2)$;
7. `UOneGate(pauli,i,alpha)` that calculates the single-qubit rotation gate $U_{pauli_i}(alpha/2)$;

[1]Functions 1,2,3,4 can be found in the **Matlab** package: `QuantInf package (version 0.4)` (link `http://www.dr-qubit.org/matlab.php#quantinf`). We implement functions 5,6,7: their codes are listed in the end of the appendix.

Bell State $|\Psi^-\rangle$ Generation

```
clear all;
clc;

% Probability of pumping
p = input ('Insert a probability of pumping: ');

% Phase gate angles
theta = pi/2;
alpha = asin(sqrt(p));

% Pauli gates
I = eye(2);
X = [0 1;1 0];
Y = [0 -i;i 0];
Z = [1 0;0 -1];
X1 = kron(X,kron(I,I));
X2 = kron(I,kron(X,I));
X3 = kron(I,kron(I,X));
Y1 = kron(Y,kron(I,I));
Y2 = kron(I,kron(Y,I));
Y3 = kron(I,kron(I,Y));


% Bell States density matrices
PHIp = ([1 0 0 1]/sqrt(2))'*([1 0 0 1]/sqrt(2));
PHIm = ([1 0 0 -1]/sqrt(2))'*([1 0 0 -1]/sqrt(2));
PSIp = ([0 1 1 0]/sqrt(2))'*([0 1 1 0]/sqrt(2));
PSIm = ([0 1 -1 0]/sqrt(2))'*([0 1 -1 0]/sqrt(2));


% MS X gate
MS_X = expm(-i*theta/4*(X1 + X2 + X3)^2);

% U_Z gate
U_Z = expm(i*alpha*kron(Z,I));

% Controlled gate
C = kron([1 0;0 0],U_Z) + kron([0 0;0 1],eye(4));

% Initial random system density matrix
psi = randRho(4);

% MS_Y gate
MS_Y = expm(-i*theta/4*(Y1 + Y2 + Y3)^2);


color = summer(4);

% Number of iterations
k = 0;
% Plot of probability of finding the initial system in one of the Bell
% States
figure(1)
plot(k,killtiny(trace(PSIm*psi*PSIm')),'ow','MarkerFaceColor',color(4,:),...
'MarkerEdgeColor',color(4,:),'MarkerSize',7.5);
hold on;

plot(k,killtiny(trace(PHIp*psi*PHIp')),'ow','MarkerFaceColor',color(1,:)...
,'MarkerEdgeColor',color(1,:),'MarkerSize',7.5);
hold on;

plot(k,killtiny(trace(PHIm*psi*PHIm')),'ow','MarkerFaceColor',color(2,:)...
,'MarkerEdgeColor',color(2,:),'MarkerSize',7.5);
hold on;

plot(k,killtiny(trace(PSIp*psi*PSIp')),'ow','MarkerFaceColor',color(3,:),...
'MarkerEdgeColor',color(3,:),'MarkerSize',7.5);
hold on;

y0 = [killtiny(trace(PHIp*psi*PHIp')) killtiny(trace(PHIm*psi*PHIm'))...
killtiny(trace(PSIp*psi*PSIp')) killtiny(trace(PSIm*psi*PSIm'))];


% --- First iteration ---
% First Map:
% density matrix system+environment
phi = kron([0 0;0 1],psi);
% step (i)
phi = MS_X*phi*MS_X';
% step (ii)
phi = C*phi*C';
```

```
% step (iii)
phi = MS_X*phi*MS_X';
% step (iv)
phi = TrX(phi,1,[2,2,2]);
y1 = [killtiny(trace(PHIp*phi*PHIp')) killtiny(trace(PHIm*phi*PHIm'))...
killtiny(trace(PSIp*phi*PSIp')) killtiny(trace(PSIm*phi*PSIm'))];
% Second Map:
% ancilla qubit reset to 1
phi = kron([0 0; 0 1],phi);
% step(i)
phi = MS_Y*phi*MS_Y';
% step (ii)
phi = C*phi*C';
% step (iii)
phi = MS_Y*phi*MS_Y';
% step (iv)
phi = TrX(phi,1,[2,2,2]);
y2 = [killtiny(trace(PHIp*phi*PHIp')) killtiny(trace(PHIm*phi*PHIm'))...
killtiny(trace(PSIp*phi*PSIp')) killtiny(trace(PSIm*phi*PSIm'))];

% Display results after 1 iteration
disp('Initial density matrix: ');
psi
disp('Final density matrix desired: ');
PSIm
disp('System density matrix after 1 iteration: ');
phi


% 20 iterations system simulation
for(k=1:20)

    % Plot of probability of finding the system in one of the Bell States
    % after k iterations
    plot(k,killtiny(trace(PSIm*phi*PSIm')),'o','MarkerFaceColor',color(4,:),...
'MarkerEdgeColor',color(4,:),'MarkerSize',7.5);
    hold on;

    plot(k,killtiny(trace(PHIp*phi*PHIp')),'o','MarkerFaceColor',color(1,:),...
'MarkerEdgeColor',color(1,:),'MarkerSize',7.5);
    hold on;

    plot(k,killtiny(trace(PHIm*phi*PHIm')),'o','MarkerFaceColor',color(2,:),...
'MarkerEdgeColor',color(2,:),'MarkerSize',7.5);
    hold on;

    plot(k,killtiny(trace(PSIp*phi*PSIp')),'o','MarkerFaceColor',color(3,:),...
'MarkerEdgeColor',color(3,:),'MarkerSize',7.5);
    hold on;


    % Iteration process
    phi = kron([0 0;0 1],phi);
    phi = MS_X*phi*MS_X';
    phi = C*phi*C';
    phi = MS_X*phi*MS_X';
    phi = TrX(phi,1,[2,2,2]);
    phi = kron([0 0; 0 1],phi);
    phi = MS_Y*phi*MS_Y';
    phi = C*phi*C';
    phi = MS_Y*phi*MS_Y';
    phi = TrX(phi,1,[2,2,2]);

end

% Display results after 20 iterations
disp('System density matrix after 20 iterations: ');
phi

% Plot configuration

title('Probability of finding the system in one of the Bell States');
xlabel('Number of iterations');
ylabel('Probability');
legend('\Psi^- state','\Phi^+ state','\Phi^- state','\Psi^+ state');
grid on;


figure(2)
y = [y0
     y1
     y2];
```

```
bar3(y);
colormap summer;
grid on;
text('Interpreter','latex');
title('Action of the map X_1X_2 and Y_1Y_2');
xlabel('Bell States projectors');
ylabel('First iteration map steps');
zlabel('Probability');
[hx,hy] = format_ticks(gca,{},{'Initial state', 'X_1X_2','Y_1Y_2'});
set(gca,'Xtick',1:4,'XTickLabel',{'PHIp', 'PHIm', 'PSIp', 'PSIm'});
```

GHZ State Generation

```
clear all;
clc;


% Pauli gates
I = eye(2);
X = [0 1;1 0];
Y = [0 -i;i 0];
Z = [1 0;0 -1];


% GHZ state basis
GHZ = (1/sqrt(2))*(tensor([1;0],[1;0],[1;0],[1;0]) + tensor([0;1],[0;1],[0;1],[0;1]));%0000
b1 = (1/sqrt(2))*(tensor([1;0],[1;0],[1;0],[1;0]) - tensor([0;1],[0;1],[0;1],[0;1]));
b2 = (1/sqrt(2))*(tensor([1;0],[1;0],[1;0],[0;1]) + tensor([0;1],[0;1],[0;1],[1;0]));%0001
b3 = (1/sqrt(2))*(tensor([1;0],[1;0],[1;0],[0;1]) - tensor([0;1],[0;1],[0;1],[1;0]));
b4 =  (1/sqrt(2))*(tensor([1;0],[1;0],[0;1],[1;0]) + tensor([0;1],[0;1],[1;0],[0;1]));%0010
b5 = (1/sqrt(2))*(tensor([1;0],[1;0],[0;1],[1;0]) - tensor([0;1],[0;1],[1;0],[0;1]));
b6 = (1/sqrt(2))*(tensor([1;0],[1;0],[0;1],[0;1]) + tensor([0;1],[0;1],[1;0],[1;0]));%0011
b7 = (1/sqrt(2))*(tensor([1;0],[1;0],[0;1],[0;1]) - tensor([0;1],[0;1],[1;0],[1;0]));
b8 = (1/sqrt(2))*(tensor([1;0],[0;1],[1;0],[1;0]) + tensor([0;1],[1;0],[0;1],[0;1]));%0100
b9 = (1/sqrt(2))*(tensor([1;0],[0;1],[1;0],[1;0]) - tensor([0;1],[1;0],[0;1],[0;1]));
b10 = (1/sqrt(2))*(tensor([1;0],[0;1],[1;0],[0;1]) + tensor([0;1],[1;0],[0;1],[1;0]));%0101
b11 = (1/sqrt(2))*(tensor([1;0],[0;1],[1;0],[0;1]) - tensor([0;1],[1;0],[0;1],[1;0]));
b12 = (1/sqrt(2))*(tensor([1;0],[0;1],[0;1],[1;0]) + tensor([0;1],[1;0],[1;0],[0;1]));%0110
b13 = (1/sqrt(2))*(tensor([1;0],[0;1],[0;1],[1;0]) - tensor([0;1],[1;0],[1;0],[0;1]));
b14 = (1/sqrt(2))*(tensor([1;0],[0;1],[0;1],[0;1]) + tensor([0;1],[1;0],[1;0],[1;0]));%0111
b15 = (1/sqrt(2))*(tensor([1;0],[0;1],[0;1],[0;1]) - tensor([0;1],[1;0],[1;0],[1;0]));


% projectors
p0 = GHZ*GHZ';
p1 = b1*b1';
p2 = b2*b2';
p3 = b3*b3';
p4 = b4*b4';
p5 = b5*b5';
p6 = b6*b6';
p7 = b7*b7';
p8 = b8*b8';
p9 = b9*b9';
p10 = b10*b10';
p11 = b11*b11';
p12 = b12*b12';
p13 = b13*b13';
p14 = b14*b14';
p15 = b15*b15';


% Completely mixed system density matrix
psi = ones(16); psi = normalise(psi);
psi0 = psi;
y0 = [killtiny(trace(p15*psi*p15')) killtiny(trace(p14*psi*p14'))...
killtiny(trace(p13*psi*p13')) killtiny(trace(p12*psi*p12'))...
killtiny(trace(p11*psi*p11')) killtiny(trace(p10*psi*p10'))...
killtiny(trace(p9*psi*p9')) killtiny(trace(p8*psi*p8')) killtiny(trace(p7*psi*p7'))...
killtiny(trace(p6*psi*p6')) killtiny(trace(p5*psi*p5')) killtiny(trace(p4*psi*p4'))...
killtiny(trace(p3*psi*p3')) killtiny(trace(p2*psi*p2')) killtiny(trace(p1*psi*p1'))...
killtiny(trace(p0*psi*p0'))];


% Z1Z2 Map:
% density matrix system+environment
phi = kron([0 0;0 1],psi);

phi = UGate(Y,pi/2)*UGate(X,-pi/2)*UOneGate(Z,0,-pi/2)*UGate(X,pi/2)*phi*UGate(X,pi/2)'*...
UOneGate(Z,0,-pi/2)'*UGate(X,-pi/2)'*UGate(Y,pi/2)';
phi = MSGate(X,pi/4)*UOneGate(Z,4,pi)*UOneGate(Z,3,pi)*MSGate(X,pi/4)*phi*...
```

```
MSGate(X,pi/4)'*UOneGate(Z,3,pi)'*UOneGate(Z,4,pi)'*MSGate(X,pi/4)';
phi = UGate(X,-pi/2)*UOneGate(Z,2,-pi/2)*UOneGate(Z,0,-pi/2)*UGate(X,pi/2)*phi*...
UGate(X,pi/2)'*UOneGate(Z,0,-pi/2)'*UOneGate(Z,2,-pi/2)'*UGate(X,-pi/2)';
phi = UOneGate(Z,1,pi)*MSGate(X,pi/4)*UOneGate(Z,2,pi)*UOneGate(Z,0,pi)*...
MSGate(X,pi/4)*phi*MSGate(X,pi/4)'*UOneGate(Z,0,pi)'*UOneGate(Z,2,pi)'*MSGate(X,pi/4)'*...
UOneGate(Z,1,pi)';
phi = UGate(X,pi/2)*UOneGate(Z,2,-pi/2)*UGate(X,-pi/2)*phi*UGate(X,-pi/2)'*...
UOneGate(Z,2,-pi/2)'*UGate(X,pi/2)';
phi = UGate(Y,-pi/2)*UOneGate(Z,2,-pi/2)*phi*UOneGate(Z,2,-pi/2)'*UGate(Y,-pi/2)';

psi = killtiny(TrX(phi,1,[2,2,2,2,2]));

psi1=psi;

y1 = [killtiny(trace(p15*psi*p15')) killtiny(trace(p14*psi*p14')) killtiny(trace(p13*psi*p13'))...
killtiny(trace(p12*psi*p12')) killtiny(trace(p11*psi*p11')) killtiny(trace(p10*psi*p10'))...
killtiny(trace(p9*psi*p9')) killtiny(trace(p8*psi*p8')) killtiny(trace(p7*psi*p7'))...
killtiny(trace(p6*psi*p6')) killtiny(trace(p5*psi*p5')) killtiny(trace(p4*psi*p4'))...
killtiny(trace(p3*psi*p3')) killtiny(trace(p2*psi*p2')) killtiny(trace(p1*psi*p1'))...
killtiny(trace(p0*psi*p0'))];

%Z2Z3 Map:
phi = kron([0 0;0 1],psi);

phi = UGate(Y,pi/2)*UGate(X,-pi/2)*UOneGate(Z,0,-pi/2)*UGate(X,pi/2)*phi*UGate(X,pi/2)'*...
UOneGate(Z,0,-pi/2)'*UGate(X,-pi/2)'*UGate(Y,pi/2)';
phi = MSGate(X,pi/4)*UOneGate(Z,4,pi)*UOneGate(Z,1,pi)*MSGate(X,pi/4)*phi*...
MSGate(X,pi/4)'*UOneGate(Z,1,pi)'*UOneGate(Z,4,pi)'*MSGate(X,pi/4)';
phi = UGate(X,-pi/2)*UOneGate(Z,3,-pi/2)*UOneGate(Z,0,-pi/2)*UGate(X,pi/2)*phi*...
UGate(X,pi/2)'*UOneGate(Z,0,-pi/2)'*UOneGate(Z,3,-pi/2)'*UGate(X,-pi/2)';
phi = UOneGate(Z,2,pi)*MSGate(X,pi/4)*UOneGate(Z,3,pi)*UOneGate(Z,0,pi)*...
MSGate(X,pi/4)*phi*MSGate(X,pi/4)'*UOneGate(Z,0,pi)'*UOneGate(Z,3,pi)'*MSGate(X,pi/4)'*...
UOneGate(Z,2,pi)';
phi = UGate(X,pi/2)*UOneGate(Z,3,-pi/2)*UGate(X,-pi/2)*phi*UGate(X,-pi/2)'*...
UOneGate(Z,3,-pi/2)'*UGate(X,pi/2)';
phi = UGate(Y,-pi/2)*UOneGate(Z,3,-pi/2)*phi*UOneGate(Z,3,-pi/2)'*UGate(Y,-pi/2)';

psi = killtiny(TrX(phi,1,[2,2,2,2,2]));

psi2=psi;

y2 = [killtiny(trace(p15*psi*p15')) killtiny(trace(p14*psi*p14')) killtiny(trace(p13*psi*p13'))...
killtiny(trace(p12*psi*p12')) killtiny(trace(p11*psi*p11')) killtiny(trace(p10*psi*p10'))...
killtiny(trace(p9*psi*p9')) killtiny(trace(p8*psi*p8')) killtiny(trace(p7*psi*p7'))...
killtiny(trace(p6*psi*p6')) killtiny(trace(p5*psi*p5')) killtiny(trace(p4*psi*p4'))...
killtiny(trace(p3*psi*p3')) killtiny(trace(p2*psi*p2')) killtiny(trace(p1*psi*p1'))...
killtiny(trace(p0*psi*p0'))];

%Z3Z4 Map:
phi = kron([0 0;0 1],psi);

phi = UGate(Y,pi/2)*UGate(X,-pi/2)*UOneGate(Z,0,-pi/2)*UGate(X,pi/2)*phi*UGate(X,pi/2)'...
*UOneGate(Z,0,-pi/2)'*UGate(X,-pi/2)'*UGate(Y,pi/2)';
phi = MSGate(X,pi/4)*UOneGate(Z,2,pi)*UOneGate(Z,1,pi)*MSGate(X,pi/4)*phi*MSGate(X,pi/4)'...
*UOneGate(Z,1,pi)'*UOneGate(Z,2,pi)'*MSGate(X,pi/4)';
phi = UGate(X,-pi/2)*UOneGate(Z,4,-pi/2)*UOneGate(Z,0,-pi/2)*UGate(X,pi/2)...
*phi*UGate(X,pi/2)'*UOneGate(Z,0,-pi/2)'*UOneGate(Z,4,-pi/2)'*UGate(X,-pi/2)';
phi = UOneGate(Z,3,pi)*MSGate(X,pi/4)*UOneGate(Z,4,pi)*UOneGate(Z,0,pi)*MSGate(X,pi/4)...
*phi*MSGate(X,pi/4)'*UOneGate(Z,0,pi)'*UOneGate(Z,4,pi)'*MSGate(X,pi/4)'*UOneGate(Z,3,pi)';
phi = UGate(X,pi/2)*UOneGate(Z,4,-pi/2)*UGate(X,-pi/2)*phi*UGate(X,-pi/2)'*UOneGate(Z,4,-pi/2)'...
*UGate(X,pi/2)';
phi = UGate(Y,-pi/2)*UOneGate(Z,4,-pi/2)*phi*UOneGate(Z,4,-pi/2)'*UGate(Y,-pi/2)';

psi = killtiny(TrX(phi,1,[2,2,2,2,2]));

psi3=psi;

y3 = [killtiny(trace(p15*psi*p15')) killtiny(trace(p14*psi*p14')) killtiny(trace(p13*psi*p13'))...
killtiny(trace(p12*psi*p12')) killtiny(trace(p11*psi*p11')) killtiny(trace(p10*psi*p10'))...
killtiny(trace(p9*psi*p9')) killtiny(trace(p8*psi*p8')) killtiny(trace(p7*psi*p7'))...
killtiny(trace(p6*psi*p6')) killtiny(trace(p5*psi*p5')) killtiny(trace(p4*psi*p4'))...
killtiny(trace(p3*psi*p3')) killtiny(trace(p2*psi*p2')) killtiny(trace(p1*psi*p1'))...
killtiny(trace(p0*psi*p0'))];

%X1X2X3X4 Map:
phi = kron([0 0;0 1],psi);
```

```
phi = MSGate(X,pi/4)*MSGate(X,pi/4)*phi*MSGate(X,pi/4)'*MSGate(X,pi/4)';
phi = UOneGate(Z,4,-pi/2)*UGate(X,-pi/2)*UOneGate(Z,0,-pi/2)*UGate(X,pi/2)*...
phi*UGate(X,pi/2)'*UOneGate(Z,0,-pi/2)'*UGate(X,-pi/2)'*UOneGate(Z,4,-pi/2)';
phi = MSGate(X,pi/4)*UOneGate(Z,4,pi)*UOneGate(Z,0,pi)*MSGate(X,pi/4)*phi*MSGate(X,pi/4)'...
*UOneGate(Z,0,pi)'*UOneGate(Z,4,pi)'*MSGate(X,pi/4)';
phi = UOneGate(Z,4,-pi/2)*UGate(X,pi/2)*UOneGate(Z,4,-pi/2)*phi*UOneGate(Z,4,-pi/2)'...
*UGate(X,pi/2)'*UOneGate(Z,4,-pi/2)';
phi = UGate(X,-pi/2)*phi*UGate(X,-pi/2)';


psi = killtiny(TrX(phi,1,[2,2,2,2,2]));

psi4 =psi;

y4 = [killtiny(trace(p15*psi*p15')) killtiny(trace(p14*psi*p14')) killtiny(trace(p13*psi*p13'))...
killtiny(trace(p12*psi*p12')) killtiny(trace(p11*psi*p11')) killtiny(trace(p10*psi*p10'))...
killtiny(trace(p9*psi*p9')) killtiny(trace(p8*psi*p8')) killtiny(trace(p7*psi*p7'))...
killtiny(trace(p6*psi*p6')) killtiny(trace(p5*psi*p5')) killtiny(trace(p4*psi*p4'))...
killtiny(trace(p3*psi*p3')) killtiny(trace(p2*psi*p2')) killtiny(trace(p1*psi*p1'))...
killtiny(trace(p0*psi*p0'))];


disp('Initial density matrix: ');
psi0
disp('Final density matrix: ');
psi4


figure(1);
y = [y0 y1 y2 y3 y4];
bar3(y);
colormap summer;
grid on;
text('Interpreter','latex');
title('GHZ cooling using stabilizers pumping');
xlabel('Not normalized GHZ state basis');
ylabel('Map steps');
zlabel('Probability');
set(gca,'Xtick',1:16,'XTickLabel',{'|0111> - |1000>','|0111> + |1000>','|0110> -...
 |1001>', '|0110> + |1001>','|0101> -| 1010>','|0101> + |1010>','|0100> - |1011>',...
'|0100> + |1011>','|0011> - |1100>','|0011> + |1100>','|0010> - |1101>','|0010> +...
 |1101>','|0001> - |1110>','|0001> + |1110>', '|0000> - |1111>','GHZ = |0000> +...
 |1111>'},'FontSize',8);
set(gca,'Ytick',1:5,'YTickLabel',{'Initial state', 'Step 1','Step 2','Step 3',...
'Step 4'});


figure(2);
subplot(1,2,1),
y = real(psi0);
bar3(y);
colormap winter;
grid on;
title('Re(\rho_S)','FontSize',14);

set(gca,'Xtick',1:16,'XTickLabel',{'b0''','b1''','b2''', 'b3''','b4''','b5''','b6''',...
'b7''','b8''','b9''','b10''','b11''','b12''','b13''', 'b14''','b15'''},'FontSize',8);
set(gca,'Ytick',1:16,'YTickLabel',{'b0','b1','b2', 'b3','b4','b5','b6',...
'b7','b8','b9','b10','b11','b12','b13', 'b14','b15'},'FontSize',8);

subplot(1,2,2),
y = imag(psi0);
bar3(y);
colormap winter;
grid on;
title('Im(\rho_S)','FontSize',14);

set(gca,'Xtick',1:16,'XTickLabel',{'b0''','b1''','b2''', 'b3''','b4''','b5''','b6''',...
'b7''','b8''','b9''','b10''','b11''','b12''','b13''', 'b14''','b15'''},'FontSize',8);
set(gca,'Ytick',1:16,'YTickLabel',{'b0','b1','b2', 'b3','b4','b5','b6',...
'b7','b8','b9','b10','b11','b12','b13', 'b14','b15'},'FontSize',8);


figure(3);
subplot(1,2,1),
y = real(psi1);
bar3(y);
colormap winter;
grid on;
title('Re(\rho_S)','FontSize',14);

subplot(1,2,2),
```

```
y = imag(psi1);
bar3(y);
colormap winter;
grid on;
title('Im(\rho_S)','FontSize',14);

figure(4);
subplot(1,2,1),
y = real(psi2);
bar3(y);
colormap winter;
grid on;
title('Re(\rho_S)','FontSize',14);

subplot(1,2,2),
y = imag(psi2);
bar3(y);
colormap winter;
grid on;
title('Im(\rho_S)','FontSize',14);

figure(5);
subplot(1,2,1),
y = real(psi3);
bar3(y);
colormap winter;
grid on;
title('Re(\rho_S)','FontSize',14);

subplot(1,2,2),
y = imag(psi3);
bar3(y);
colormap winter;
grid on;
title('Im(\rho_S)','FontSize',14);

figure(6);
subplot(1,2,1),
y = real(psi4);
bar3(y);
colormap winter;
grid on;
title('Re(\rho_S)','FontSize',14);

subplot(1,2,2),
y = imag(psi4);
bar3(y);
colormap winter;
grid on;
title('Im(\rho_S)','FontSize',14);
```

Auxiliary functions:

Mølmer-Sørensen gate

```
function  U = MSGate(pauli,alpha)

I = eye(2);

U = expm(-i*alpha/4*(tensor(pauli,I,I,I)+tensor(I,pauli,I,I)...
+tensor(I,I,pauli,I)+tensor(I,I,I,pauli)+tensor(I,I,I,pauli))^2);

end
```

Collective single-qubit rotation gate

```
function  U = UGate(pauli,alpha)

I = eye(2);

U = expm(-i*alpha/2*(tensor(pauli,I,I,I)+tensor(I,pauli,I,I)...
+tensor(I,I,pauli,I)+tensor(I,I,I,pauli)+tensor(I,I,I,pauli)));
```

```
8  end
```

Single-qubit rotation gate

```
1  function  U = UOneGate(pauli,i,alpha)

3  I = eye(2);

5  if(i == 0)
       U = expm(-i*alpha/2*(tensor(pauli,I,I,I)));
7  elseif(i == 1)
       U = expm(-i*alpha/2*(tensor(I,pauli,I,I)));
9  elseif(i == 2)
       U = expm(-i*alpha/2*(tensor(I,I,pauli,I)));
11 elseif(i == 3)
       U = expm(-i*alpha/2*(tensor(I,I,I,pauli)));
13 elseif(i == 4)
       U = expm(-i*alpha/2*(tensor(I,I,I,I,pauli)));
15 else
   end
17
   end
```

B

Quantum Information Processing with Ion Traps

Trapped ion quantum computer system is one of the most promising architectures for a scalable, universal quantum computer. The ion trap quantum information processor consists of a number N of controlled ions which are confined in a region by an appropriate potential. Typically, the ions are placed along a line (linear Paul trap) as shown in Fig. B.1. Without going into technical details about the physical realization we can say that each ion is used to implement a qubit (two of the energy levels of each ion are used to represent states 0 and 1) and can be controlled by laser fields and pulses.

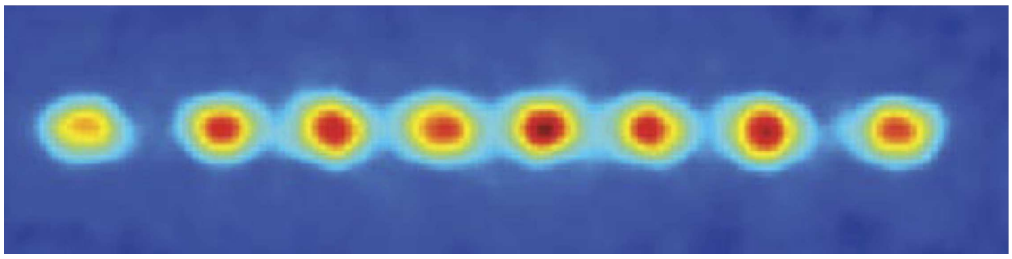


Figure B.1: Levitated string of eight Calcium ions are confined in a vacuum chamber and laser-cooled to be nearly at rest: such a string can perform quantum calculations

The first proposal of such an architecture was given by P. Zoller and I. Cirac in 1995 ([5]). The key idea of the proposal is to use laser pulses to

mediate an effective interaction between the electronic states of individual ions. Currently, experimental ion trap quantum information processing is pursued by about 20 research groups worldwide and recently, in the Innsbruck group, coherent manipulation of up to 14 qubits has been achieved ([17]).

The ion-traps system satisfies in principle all *DiVincenzo criteria* ([8]), which are the requirements that have to be met to actually build a quantum computer. These criteria are listed below along with the current achievements in trapped ions implementation ([11]):

I. **A scalable physical system with well characterized qubits**: there are two ways to form a qubit using the electronic states of an ion:

   i. two ground state hyperfine levels (these are called *hyperfine qubits*),

   ii. a ground state level and an excited level (these are called *optical qubits*).

   Hyperfine qubits are extremely long-lived (decay time of the order of thousands to millions of years). Optical qubits are also relatively long-lived, compared to the logic gate operation time. Scalability can be obtained in principle using a qubit register formed by strings of ions in a Paul trap.

II. **The ability to initialize the state of the qubits**: ions can be prepared in a specific qubit state using a process called *optical pumping*, which can be performed with extremely high fidelity.

III. **A coherence time much longer than the operational time**: in current quantum computing experiments, typically coherence times of a few milliseconds are achieved which are about one to two orders of magnitude longer than the time scale for quantum operations.

IV. **A universal set of quantum gates**:

   i. Single qubit gates are implemented by driving *Rabi oscillations*[1] between the two qubit levels with resonant laser pulses.

[1] If an electromagnetic wave is resonant with an atom's transition frequency, it excites the atom (e.g. from the ground state to the first excited state). This means the probability amplitude of finding the atom in the excited state increases over time. However, at some point when the atom is completely in its excited state, the wave actually goes on to de-excite the atom again. This cycle of absorption-emission is called Rabi oscillations, and it proceeds at a frequency that is proportional to the strength of the electric field.

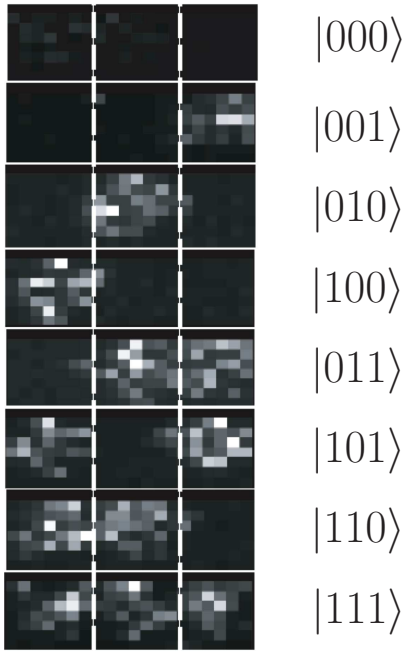


Figure B.2: Example of images acquired by CCD for state detection on a three ions system

ii. High-fidelity experimental two-qubit entangling gates have been realized, such as: the Cirac-Zoller CNOT gate, the Mølmer-Sørensen gate and the so-called geometric phase gate.

V. **A qubit-specific measurement**: measuring the state of a qubit stored in a ion proceeds as follows:

- a laser is applied to the ion that couples only one of the qubit state and two cases may occur:
  i. the ion collapsed into this state during the measurement process, the laser will excite it, resulting in a photon being released when the ion decays from the excited state;
  ii. the ion collapsed into the other qubit state, then it does not interact with the laser and no photon will be emitted.
- In case (i), after decay, the ion is continually excited by the laser and repeatedly emitting photons.
- These photons can be collected by a photomultiplier tube (PMT) or a charge-coupled device (CCD) camera.

- By counting the collected photons, it is easy to determine which state the ion is in with very high fidelity (as illustrated in Fig. B.2).

Additional DiVincenzo requirements are:

VI. **The ability to interconvert stationary and flying qubits**: for converting stationary (ion) qubits into flying (photon) qubits, the techniques of cavity quantum electrodynamics (CQED) are used and several experiments are currently under way.

VII. **The ability to faithfully transmit flying qubits between specified locations**: faithful transmission of photonic qubits between two quantum computer nodes was theoretically shown to be feasible; a transfer protocol is available, however, at this time no experimental work is carried out yet.

In conclusion it can be inferred that, at present, quantum information processing with ion traps provides most of the requirements for quantum computation experiments and maybe in the near future further improvements can lead to the ultimate target: the physical realization of the first quantum computer.

Bibliography

[1] F. Albertini and F. Ticozzi. Discrete-time controllability for feedback quantum dynamics. 2010. preprint: arXiv:1012.1397v1 [quant-ph].

[2] J. T. Barreiro, M. Muller, P. Schindler, D. Nigg, T. Monz, M. Chwalla, M. Hennrich, C. F. Roos, P. Zoller, and R. Blatt. An open-system quantum simulator with trapped ions. *Nature 470 486-491*, 2011.

[3] G. Benenti, G. Casati, and G. Strini. *Principles of Quantum Computation and Information - Vol.1: Basic Concepts*. World Scientific Publishing Company, 2004.

[4] S. Bolognani and F. Ticozzi. Engineering stable discrete-time quantum dynamics via a canonical qr decomposition. *IEEE Transactions on Automatic Control, VOL. 55, NO. 12*, 2010.

[5] J. I. Cirac and P. Zoller. Quantum computations with cold trapped ions. *Phys. Rev. Lett.*, 74(20):4091–4094, May 1995.

[6] D. D'Alessandro. *Introduction to quantum control and dynamics*. Chapman & Hall/CRC, 2007.

[7] P. A. M. Dirac. *The Principles of Quantum Mechanics*. Oxford University Press, 1982.

[8] David P. Divincenzo. The physical implementation of quantum computation. *Fortschr. Phys*, 48:2000, 2000.

[9] A. Einstein, B. Podolsky, and N. Rosen. Can quantum-mechanical description of physical reality be considered complete? *Phys. Rev.*, 47:777–780, May 1935.

[10] H. O. Everitt. *Experimental Aspects of Quantum Computing*. Springer, 2005.

[11] H. Haeffner, C. F. Roos, and R. Blatt. Quantum computing with trapped ions. Phys. Rep. 469, 155-203 (2008), 2008.

[12] K. Hoffman and R. Kunze. *Linear Algebra (Second Edition)*. Prentice Hall, 1971.

[13] P. Kaye, R. Laflamme, and M. Mosca. *An Introduction to Quantum Computing*. Oxford University Press, 2007.

[14] H. K. Khalil. *Nonlinear Systems (second edition)*. Prentice Hall, 1996.

[15] G. Kirchmair. *Quantum non-demolition measurements and quantum simulation*. PhD thesis, Faculty of Mathematics, Computer Science and Physics of the Leopold-Franzens University of Innsbruck, 2010.

[16] S. Lloyd and L. Viola. Engineering quantum dynamics. *Phys. Rev. A*, 65(1):010101, Dec 2001.

[17] T. Monz, P. Schindler, J. T. Barreiro, M. Chwalla, D. Nigg, W. A. Coish, M. Harlander, W. Haensel, M. Hennrich, and R. Blatt. 14-qubit entanglement: creation and coherence. Phys. Rev. Lett. 106, 130506 (2011), 2010.

[18] M. Muller, K. Hammerer, Y. L. Zhou, C. F. Roos, and P. Zoller. Simulating open quantum systems: from many-body interactions to stabilizer pumping. 2011. preprint: arXiv:1104.2507v1 [quant-ph].

[19] A. H. Myerson, D. J. Szwer, S. C. Webster, D. T. C. Allcock, M. J. Curtis, G. Imreh, J. A. Sherman, D. N. Stacey, A. M. Steane, and D. M. Lucas. High-fidelity readout of trapped-ion qubits. *Phys. Rev. Lett.*, 100:200502, May 2008.

[20] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

[21] A. Peres. *Quantum Theory: Concepts and Methods*. Springer, 1995.

[22] J. Preskill. Lecture notes on quantum computation and information, 2001. Available on line at `http://theory.caltech.edu/~preskill/ph229/#lecture`.

[23] M. Riebe. *Preparation of entangled states and quantum teleportation with atomic qubits*. PhD thesis, Faculty of Mathematics, Computer Science and Physics of the Leopold-Franzens University of Innsbruck, 2005.

[24] W. Rudin. *Functional Analysis*. McGraw-Hill, 1973.

[25] J. J. Sakurai. *Modern Quantum Mechanics*. Addison Wesley, 1993.