Doctoral Dissertation
Doctoral Program in Electrical, Electronics and Communications Enginering
(34.th cycle)

Engineering quantum computing technologies

from compact modelling to applications

Giovanni Amedeo Cirillo

* * * * *

Supervisors
Prof. Mariagrazia Graziano, Supervisor

Doctoral Examination Committee:
Prof. Robert Wille, Referee, Technische Universität München
Prof. Paolo Olivero, Referee, Università degli Studi di Torino

Politecnico di Torino
July 22, 2022

I hereby declare that, the contents and organisation of this dissertation constitute my own original work and does not compromise in any way the rights of third parties, including those relating to the security of personal data.

. . . . . .

Giovanni Amedeo Cirillo

Turin, July 22, 2022

Summary

Quantum computing could impact Society and Industry in an unprecedented way, especially in critical domains, such as optimization problem solving, computational chemistry and artificial intelligence. Properties of quantum mechanics, such as superposition and entanglement, can be exploited to define problem-specific quantum algorithms with lower computational complexity than their best corresponding classical counterparts. A common limitation of quantum computers is the presence of hardware non-idealities, such as decoherence, dramatically affecting performance and reliability. In any case, efforts in improving hardware and investigating short and long-term quantum algorithms can be already done.
The goal of this doctorate thesis is to explore different levels of quantum computing engineering, from devices to applications. In particular, the attention was focused on two activities: the definition of a compact-model-based simulation methodology for quantum technologies for the general-purpose quantum circuit model and the proposal of an approach for solving the users scheduling optimization problem in Joint-Transmission-based wireless communication systems, exploiting the quantum-compliant Quadratic Unconstrained Binary Optimization (QUBO) formalism.
Classical simulation of quantum computers allows a preliminary estimation of their capabilities in specific applications. Canonical simulation of qubits affected by non-ideality phenomena requires solving the computationally expensive Lindblad master equation. A compact model of a quantum technology, obtained through fitting operations or specific mimic algorithms, is thought to describe qubits evolution under quantum gates, taking into account the device-characteristic physical parameters and not numerically solving Lindblad equation, thus permitting faster and more scalable simulations. Compact models of two families of spin-based molecular quantum computers, switch-based and Nuclear Magnetic Resonance, were developed, both involving two other general non-ideality models: decoherence and off-resonance of untargeted qubits. A quantum circuits simulator, involving all these compact models and interfaceable with OpenQASM 2.0 language, was entirely developed in MATLAB. The compact model approximations were compared with either results already available in the state of art or through numerical simulations with consolidated formalisms, at both Hamiltonian and circuit levels. The achieved results give evidence that the proposed simulation methodology provides

results fast and close to the references, thus paving the way for fast and reliable technology-dependent quantum circuits simulation.
Joint Transmission (JT) is a modern mechanism for Multiple-Input-Multiple-Output (MIMO) wireless communication systems, where multiple antennas co-operate to transmit signals to users with minimized interference. An optimization problem, looking for the best users subset servable in JT, can be defined. In a practical scenario, this cannot be efficiently solved with a brute-force-only approach, since it could imply a too high number of combinations. This thesis presents a strategy for solving the JT optimization problem, based on a QUBO model, which is responsible of a first selection of users among all available. The subset selected with QUBO is then brute-force explored, to find the best solution. In all executed tests, the obtained results permit to say that the proposed method achieves a quite good compromise between solution quality and total execution time.
Even though the obtained results in both contexts can still be further expanded and improved, they are quite encouraging and prove that these research activities, if continued in the future, can help in improving quantum computing engineering, at both hardware and software levels, with the hope that it can concretely solve problems of interest for Society and Industry.

Contents

List of Tables x

List of Figures xii

I Introductory chapters 1

1 Introduction 3
- 1.1 Background . . . 3
  - 1.1.1 Perceiving quantum computing by intuition . . . 4
- 1.2 Paradigms . . . 6
- 1.3 Quantum computers in NISQ era . . . 9
- 1.4 Defining a toolchain for a quantum computing ecosystem . . . 12
- 1.5 Thesis motivations and organization . . . 14
  - 1.5.1 Noisy simulation of quantum computers . . . 14
  - 1.5.2 Joint Transmission . . . 16
  - 1.5.3 Thesis organization . . . 16

2 Basic Concepts of Quantum Mechanics 18
- 2.1 Linear Algebra of Quantum Mechanics . . . 18
  - 2.1.1 Hilbert spaces and the Dirac notation . . . 18
  - 2.1.2 Operators . . . 19
  - 2.1.3 Tensor product . . . 21
  - 2.1.4 Hermitian and unitary operators . . . 23
  - 2.1.5 Eigenvalues and eigenvectors . . . 23
  - 2.1.6 Trace . . . 25
  - 2.1.7 Operator exponentiation . . . 25
  - 2.1.8 Pauli matrices . . . 26
- 2.2 Postulates of quantum mechanics . . . 28
  - 2.2.1 Observables . . . 28
  - 2.2.2 State representation . . . 29
  - 2.2.3 The Schrödinger equation . . . 31

2.2.4 Measurement . . . 32
2.3 Additional concepts . . . 34
2.3.1 Mixed states and density matrix . . . 34
2.3.2 Distance metrics . . . 35
2.3.3 No-cloning theorem . . . 36
2.3.4 Relaxation and decoherence . . . 38

II Quantum circuit model 43

3 Introduction to the quantum circuit model 45
3.1 Quantum gates . . . 46
3.1.1 Single-qubit gates . . . 47
3.1.2 Multi-qubit gates . . . 49
3.1.3 Universal set of quantum gates . . . 52
3.2 Formalizing quantum circuits description . . . 53
3.3 Technology overview . . . 55
3.3.1 DiVincenzo criteria . . . 55
3.3.2 Overview of technologies for quantum circuit model . . . 56
3.3.3 Magnetic resonance . . . 61

4 Compact-modeling-based simulation of quantum computing technologies 71
4.1 Overview of simulation infrastructure . . . 73
4.2 Organization of a compact model . . . 74
4.3 Compact-model-based simulation methodology . . . 78
4.4 Transversal sub-models . . . 84
4.5 Fourier model for off-resonance description . . . 84
4.5.1 Analytical descrption of the phenomenon . . . 85
4.5.2 Fourier model . . . 87
4.5.3 Validation . . . 89
4.6 Relaxation and Decoherence . . . 90
4.6.1 Proposed model . . . 91
4.6.2 Results . . . 94

5 Molecular technologies for quantum computing 98
5.1 Switch-based molecules . . . 99
5.1.1 Vanadyl moieties . . . 102
5.1.2 Cr_7Ni supramolecular complexes . . . 104
5.1.3 Compact model . . . 106
5.1.4 Modelling results . . . 110
5.2 Nuclear Magnetic Resonance . . . 117

5.2.1 Compact model . . . 121
5.2.2 Results . . . 122
5.2.3 Optimization of single-qubit-gate duration . . . 122
5.3 Concluding remarks . . . 128

III QUBO modelling 129

6 Introduction to binary optimization problems 131
6.1 QUBO model . . . 131
6.2 Ising model . . . 135
6.3 Problems overview . . . 138
6.4 Approaches to QUBO problem solving . . . 140
6.4.1 Simulated annealing . . . 141
6.4.2 Quantum annealing . . . 144

7 Quantum-inspired techniques for Joint Transmission allocation problem 147
7.1 Theoretical model . . . 148
7.1.1 MIMO model . . . 148
7.1.2 Joint transmission mechanism . . . 150
7.1.3 Choice of power attenuation factors . . . 153
7.2 Scheduling algorithms . . . 154
7.2.1 Applying quantum computing to Joint Transmission . . . 154
7.2.2 QUBO formulation for Joint Transmission . . . 156
7.2.3 Software development . . . 159
7.2.4 Results . . . 160
7.3 Concluding remarks . . . 169

IV Conclusions and Future Perspectives 170

8 Future perspectives 172
8.1 Simulation based on compact models . . . 172
8.1.1 Improving and extending compact models . . . 172
8.1.2 Methodology for simulator software development . . . 173
8.2 Joint Transmission . . . 174

9 Conclusions 176

V Appendices **178**

A Understanding the Rotating Wave Approximation with a simple scalar integral example 180

B Organization of the simulator 182
B.1 `simulation_exec/quantum_computing` 183
B.2 `quantum_information_utils/` 189
B.3 `simulation_config/quantum_computing` 189

Bibliography 195

List of Tables

3.1 Comparison of some quantum computing technologies. 61
4.1 Relaxation and decoherence time constants of two the NMR quantum computers employed in the simulations for validating the proposed model. 95
4.2 Comparison of the most probable outcomes on a cytosine quantum computer, according to the thermal relaxation model of Qiskit and the simulation strategy based on the proposed relaxation model. . . 97
5.1 Comparison of the switch-based molecules examined in this thesis. . 102
5.2 Resonance angular frequencies with the two technologies. 107
5.3 Fidelities for some benchmark circuits, employing the same simulation setup of [58] (fixed pulse duration). 115
5.4 Physical parameters of the crotonic acid: the isotropic chemical shifts are reported in [70, 71], the J-coupling constants (properly $\frac{J}{2\pi}$) are taken from [71] and the time constants $T_{1,2}$ are available in [71, 72]. 120
5.5 Physical parameters of the crotonic acid: the isotropic chemical shifts are reported in [70], the J-coupling constants (properly $\frac{J}{2\pi}$) are taken from [73], such as $T_{1,2}$. 120
5.6 Physical parameters of the ^{13}C-labeled diethyl fluoromalonate. All the parameters are taken from [74] and the magnetic static field is expected to be $B = 14.09\,\mathrm{T}$. 120
5.7 Comparison of the most probable outcomes on a cytosine quantum computer, according to the thermal relaxation model of Qiskit and two simulation strategies based on the proposed relaxation model. Virtual-Z gates are assumed to be employed. 127
5.8 Comparison of the most probable outcomes on a crotonic acid quantum computer, according to the thermal relaxation model of Qiskit and two simulation strategies based on the proposed relaxation model. Virtual-Z gates are assumed to be employed. 127
6.1 Known QUBO penalty functions for constraints evaluation, arranged from [11]. 135
7.1 QUBO building parameters. 161
7.2 Optimal solution rate with random H matrices and $n_{\text{repetitions}} = 100$. 163

7.3 Users selection done by QUBO on subchannel 2. 167

List of Figures

1.1 Overview of superposition and entanglement principles. 4
1.2 Comparison of classical and quantum computing for a search problem. 5
1.3 Quantum computation models 7
1.4 Qualitative representation of relaxation and decoherence phenomena. 10
1.5 Conceptual representation of a gate error which does not ensure an equal superposition of states encoding 0 and 1. 10
1.6 Schematic representation of temperature lowering in a generic *quantum chandelier*. 11
1.7 Partially-connected (left) and fully-connected (right) quantum computers. 12
1.8 Organization of a generic toolchain for quantum computing technologies 13
2.1 Bloch sphere in the Cartesian space. The north and south poles correspond to basis states $|0\rangle$ and $|1\rangle$, while $|\pm\rangle = \frac{|0\rangle \pm |1\rangle}{\sqrt{2}}$. 30
2.2 Quantum teleportation overview. 37
2.3 Visual demonstration of the experiments for qubit spectroscopy. . . 41
2.4 Characteristic qubit transients, obtained with different experiments. 42
3.1 Quantum circuits overview. 45
3.2 Comparison of the two quantum circuit notations. 46
3.3 Circuit representation of the CNOT gate. 49
3.4 Circuit representations of Cϕ and CZ gates. 50
3.5 Construction of CNOT and CZ gates with $R_{zz}(\theta)$. 51
3.6 CNOT gate constructed with Hadamard and CZ gates. 52
3.7 Quantized LC circuits. Anharmonicity given by Josephson Junction (lower image) ensures that $|\Delta\omega_{|n\rangle,|n+1\rangle} - \Delta\omega_{|n+1\rangle,|n+2\rangle}|$ is not constant anymore. 58
3.8 Overview of quantum computing with trapped ions. 59
3.9 Qubit encodings in trapped ions. 59
3.10 Electron spin-$\frac{1}{2}$ overview. 60
3.11 Representation of Virtual-Z principle on the Bloch sphere. The final state $|\phi_f\rangle = R_z\left(\frac{\pi}{2}\right)|\psi_i\rangle$ can be achieved by rotating the z-axis by $\frac{\pi}{2}$ or rotating the x and y axes by $-\frac{\pi}{2}$. 67

4.1 Methodology for analysis and validation of compact models. 72
4.2 Block scheme of the complete simulation infrastructure. 73
4.3 Overview of a quantum computing system, where the device model has the role of properly manipulating the state vector and the density matrix. 75
4.4 Overview of a single-qubit gate simulation. The garnet-red circles indicate the operations ordering. 79
4.5 Possible generation mechanisms of U_{real} in the available compact models. 80
4.6 Flow chart of measurement operation. 83
4.7 Visual description of off-resonance mechanism. 85
4.8 Effective rotation angles with three different port precession signals. 88
4.9 Functional tests of the Fourier model. 89
4.10 Equal local baths model, where each qubit interacts thermodynamically or electro-magnetically with a specfic environment. 91
4.11 Simulation of spectroscopic experiments for estimating the relaxation and decoherence time constants $T_1 = 17\,\text{ţs}$ and $T_2 = 680\,\text{ns}$. 94
5.1 Overview of switch-based molecules 100
5.2 Conceptual scheme of the quantum computer based on vanadyl moieties. 102
5.4 Circuit representation of the effective Cϕ of vanadyl moieties. . . . 103
5.3 Zeeman diagram for a vanadyl system involving two qubits and one switch. 104
5.5 Zeeman diagram for a Cr_7Ni system involving two qubits and one switch. 104
5.6 Fidelity of Cϕ gate for a two-qubit Cr_7Ni molecule with respect to different degrees of freedom. 112
5.7 Analysis of the residual qubit-qubit interaction on the fidelity of the system. The values of the J tensor are those in [65]. 112
5.8 Fidelity reduction due to simultaneous decoherence of multiple qubits subjected to Markovian dynamics. 113
5.9 Fidelity of two quantum circuits on two and three-qubit Cr_7Ni molecules, with respect to different degrees of freedom. 116
5.10 CZ gate with NMR native gates. 117
5.11 Examples of NMR quantum computers. In cytosine and diethyl fluoromanolate cases, nuclei encoding qubits are marked with green ellipses. 119
5.12 Dependency of fidelity on pulse amplitude and duration for a two-qubit cytosine molecule. 123
5.13 Dependency of fidelity on pulse amplitude and duration for a four-qubit crotonic acid. 124

5.14 Probability distribution resulting from the simulation of the execution of Grover's quantum algorithm (binary solution equal to 10) on a cytosine molecule, adopting the direct integration, the proposed model and Qiskit `QasmSimulator`. . . . 126
5.15 Probability distribution resulting from the simulation of the quantum circuit of Bernstein-Vazirani's algorithm (binary string to be found equal to 101) on a crotonic acid molecule, adopting the direct integration, the proposed model and Qiskit's `QasmSimulator`. Only the eigenstates with non-null probabilities are reported. . . . 126
6.1 Profile of the cost function of Equation 6.3. . . . 132
6.2 Example of Ising system. . . . 136
6.3 Energy profile of the Ising model equivalent to QUBO one in Equation 6.3. . . . 138
6.4 Overview of problems which can be formulated with QUBO model. 139
6.5 Comparison of simulated and quantum annealing techniques. . . . 141
6.6 Solving an Ising problem with simulated annealing: effects of temperature. . . . 142
6.7 Acceptance rate, for different temperatures. . . . 143
6.8 Conceptual scheme of quantum annealing evolution with spins and the bathtub filled with water. . . . 145
7.1 Conceptual scheme of a wireless MIMO system. . . . 149
7.2 Conceptual scheme of a MIMO system with Joint Transmission. . . 151
7.3 Block scheme of a hybrid solver for Joint Transmission allocation problem. . . . 155
7.5 Total number QUBO variables, with different total number of users. 161
7.6 Response of the proposed algorithm to random channel matrices, with $N_{\mathrm{UE}} = 30$. . . . 164
7.7 Base stations involved in TIM S.p.A. simulation. Cells with the same colour form a BS triplet. . . . 165
7.8 **Left column plots**: Reduction of the total number of *brute-force* iterations with the number of selected users. **Right column plots**: Reduction of the total number of *brute-force* iterations with G. . . . 168
7.9 Transmission rate cumulative distributions for four base stations triplets. Values μ e σ are the mean value and standard deviation of transmission rate. . . . 169
B.1 `simulation_exec/quantum_computing` content. . . . 183
B.2 Mechanism for creating the simulation netlist from an OpenQASM 2.0 file. . . . 187
B.3 Testbench workflow. . . . 188
B.4 Overview of the main directories of QuanTO simulator. . . . 193
B.5 Contents of `devices` sub-directory. . . . 194

Part I

Introductory chapters

Chapter 1

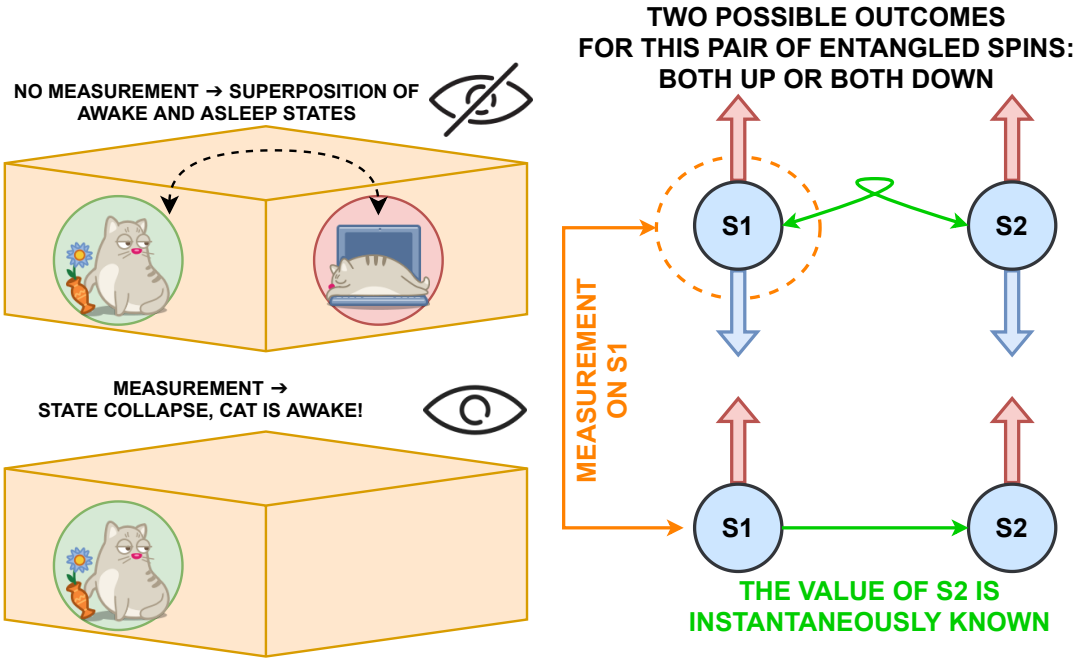
Introduction

The emerging of quantum technologies could significantly change computation and communication. The exploitation of characteristic properties of quantum mechanics, such as superposition and entanglement, can provide a computational advantage in solving hard problems with respect to classical computers and an increase of data security in communication systems. Quantum information is expected to impact Society in an unprecedented way, especially for what concerns research in critical fields, ranging from optimization problem solving [1] to simulation of molecules [2], from cryptography [3] to artificial intelligence [4].
This chapter is going to introduce some important features of quantum computation (first three paragraphs, arranging some contents presented in the informative article [5]) and of the research activities at the basis of this thesis (last three), which are detailed in the following chapters. Even though this thesis is focused on quantum computation part, the concepts introduced in the following are expected to be adaptable and arrangeable to quantum communication. In the following of this thesis, the Copenhagen interpretation of quantum mechanics is going to be dogmatically assumed valid for understanding the mechanisms behind quantum computing.

1.1 Background

In this first section, an intuitive description of the main actors of quantum computing is provieded, with a historical timeline with the milestones of this computational paradigm.

1.1.1 Perceiving quantum computing by intuition



(a) Schrödinger's cat paradox. (b) Entanglement of two spin states.

Figure 1.1: Overview of superposition and entanglement principles.

Superposition and **entanglement** are the main fundamental principles of quantum mechanics providing to quantum computers computational advantages in some specific problems.
According to the first one, a quantum state is generally described by a linear combination of two or more measurable basis states, each one with a potential non-null probability of being measured. Superposition state is kept by a quantum system until a measurement operation is done and its time evolution can be evaluated by linearly combining those of single basis states. A very famous paradox describing the superposition principle is that of Schrödinger's cat, reported in Figure 1.1a in its *nonviolent* version. The cat is located in a closed box an it is not known if it is awake or asleep (*i.e.* its state is described by a superposition of the two states) until box is opened, *i.e.* a measurement operation is done.
On the other hand, entanglement is a property of quantum systems made of more subsystems. These are entangled when they are intrinsically correlated in such a way that it is impossible to evaluate their state individually. This means that a measurement operation on a subsystem varies the state of the others entangled with it. An example of entanglement in a two-spin system is reported in Figure 1.1b: the global state is characterized by a superposition of the two states where spins

are both up or down. If a measurement operation is done on the right spin and it results spin-up(spin-down), the state of the other is immediately known and equal to spin-up(spin-down) too.
In the context of quantum information, the qubit state is described by the superposition of two states encoding 0 and 1 values and consequently the N-qubit state is described by the superposition of 2^N basis states. This implies that, given an N-qubit computer and assuming that each element of an input dataset is encoded onto a basis state, an operation — implemented in terms of an unitary evolution — can be simultaneously evaluated on all 2^N input values/data. Moreover, entanglement can be exploited to reduce, in some problems, the number of operations required for obtaining a solution, thanks to the mechanism according to which measurement on a qubit immediately affects the value of other qubits.

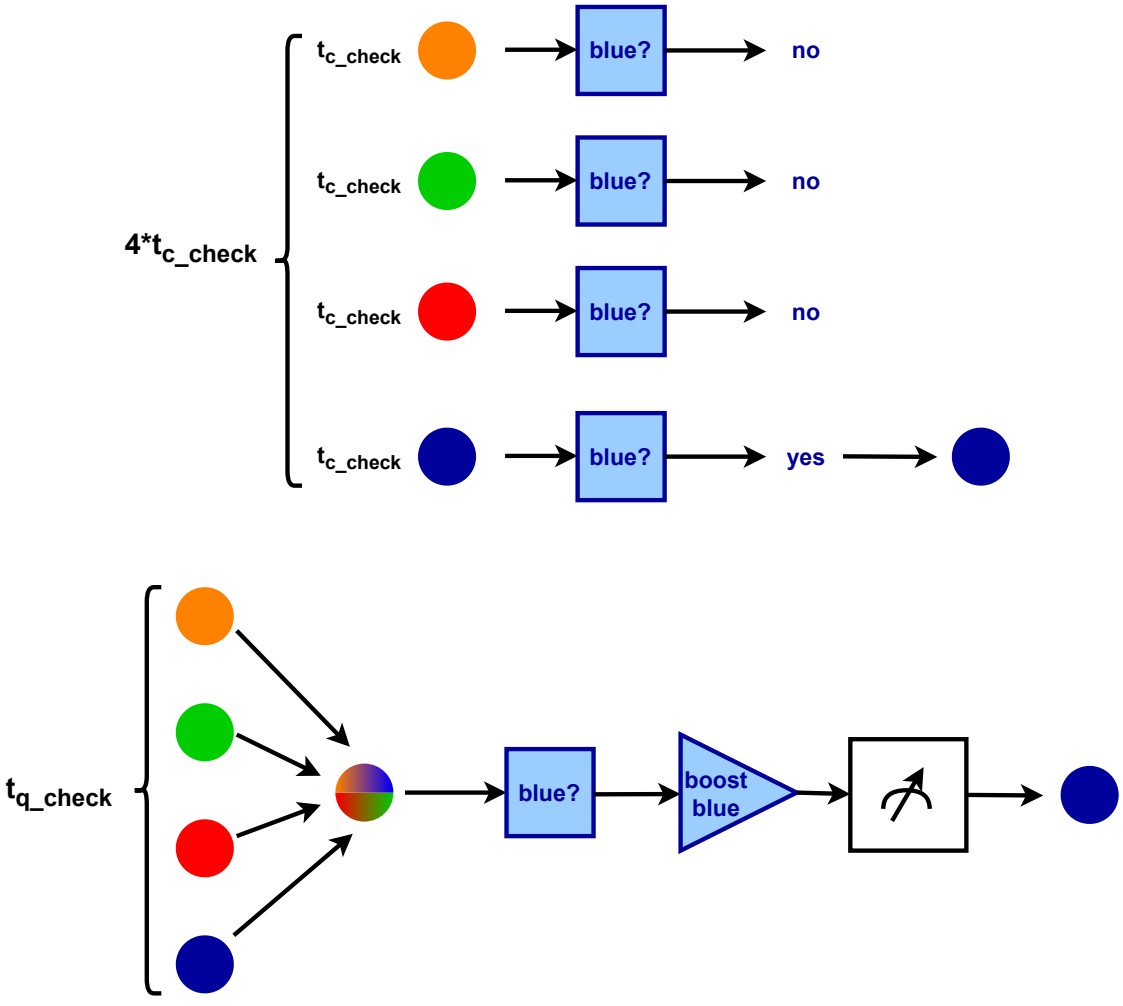


Figure 1.2: Comparison of classical and quantum computing for a search problem.

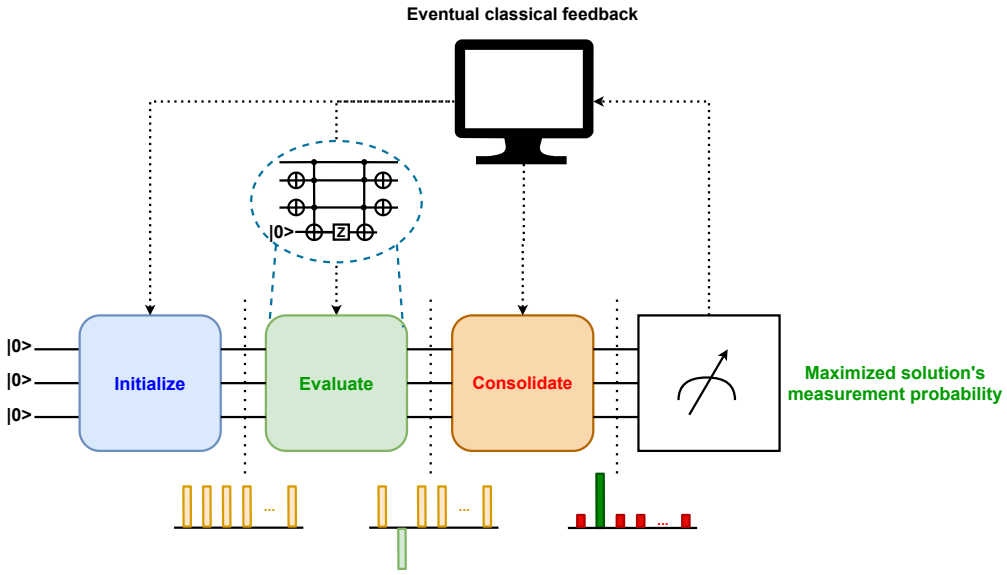
Even though superposition and entanglement are fundamental resources for implementing an efficient quantum algorithm, they are not sufficient for achieving a computational advantage. In fact, considering that measurement will force the qubit state only to a single basis state, the equivalent parallel processing of quantum computers can be practically exploited in problems where it is possible to identify specific solutions, whose measurement probabilities must be maximized through **interference** mechanisms. An example is shown in Figure 1.2: four data, each one associated with a different colour, are available in a dataset and the blue element must be found. While in a classical computing worst-case scenario a check for each element must be executed — thus implying a total execution time equal to four times the duration of each check $t_{\text{c check}}$ (when the operation on the blue input is the last one to be processed) — in a quantum computing scenario a qubits superposition state, involving all data in the dataset, is created to execute a simultaneous check, then the probability of measuring the blue element is enhanced. In this case, differently from the previous one, the total number of checks is expected to increase sub-linearly with the number of elements in the dataset. Moreover, if the check time of the quantum procedure $t_{\text{q check}} \simeq t_{\text{c check}}$, the total execution time is expected to be lower. This is a qualitative example of the well-known problem of searching an item in an unsorted database, which can be solved by the quantum Grover's algorithm [6] with a lower computational complexity than its classical counterpart. In particular, if the total number of elements in the dataset is N, Grover's algorithm can find the target with $O(\sqrt{N})$ queries (or checks), while classically the number of queries is linear with the set size ($O(N)$).

1.2 Paradigms

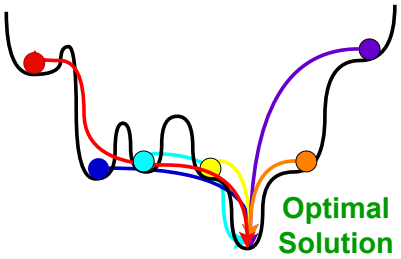
Figure 1.3 shows the two most consolidated quantum computing paradigms: quantum circuit model and quantum annealing.
A quantum computer based on the so-called circuit model [7] is characterized by the execution of operations in terms of quantum gates, which are a sort of extension to qubits of logic gates employed in classical computing. It is important to precise this paradigm is general-purpose, *i.e.* a quantum computer based on it can theoretically implement[1] any classical or quantum operation. Figure 1.3a shows a schematic and generic organization of a quantum algorithm to be run on a computer based on the circuit model:

1. **Initialization**, which consists in two steps:

[1]It would be better to say that any operation can be executed with a higher or lower degree of approximation on a quantum-circuit-model-based computer.



(a) Quantum circuit model



(b) Quantum annealing

Figure 1.3: Quantum computation models

(a) initialization of the quantum computer to a fiducial state, which is usually its ground, onto which $|0\cdots0\rangle$ is encoded;

(b) application of a set of quantum gates which forces the quantum state to a superposition of basis states, each one associated with an information to be processed.

2. **Evaluation**, which consists in applying a sequence of quantum gates, associated with a specific operation, to the superposition state, thus permitting the simultaneous evaluation of the same operation on multiple data, each one encoded onto a basis state. At the end of this step, the solution of the problem to be solved is expected to be already available in some way.

3. **Consolidation** of the solution.

4. **Measurement** of the superposition state.

5. **Classical post-processing**, consisting in some additional calculations which can be efficiently done by classical computers. This can be also employed to update quantum computer operations in iterative algorithms.

Considering that measurement will force the collapse of the superposition state onto a single basis state, this parallel evaluation cannot allow a computationally cheap storage of all processed data. For this reason, parallel computation with quantum circuit model is usually declined in finding a subset of input data satisfying a given task or obtaining a variation of the state of an auxiliary qubit due to the characteristics of input. For this reason, evaluation could include a labelling operation of the problem's solution and consolidation is usually declined in the maximization of the solution's measurement probability.
Since quantum hardware is nowadays limited, in terms of performance and number of qubits, a consolidated approach consists in defining hybrid quantum-classical algorithms, where quantum and classical computers cooperate executing specific tasks. The variational algorithms are the most employed hybrid quantum-classical procedures with the quantum circuit model, mainly exploited for solving optimization problems [8, 9]. These are iterative procedures where a quantum circuit with parametric gates is run to explore the solutions space. Gates are tuned by a classical optimizer, according to the problem's cost function, associated with the measurement outcomes of the quantum circuit. The result is conceptually the same, in sense that the probability of measuring the optimal solution is going to be maximized.
Quantum annealing [10] is a special-purpose paradigm for solving optimization problems. A cost function, which can be equivalently seen as an energy profile of a physical system, is usually associated with an optimization problem. A quantum system can be engineered in such a way that it can encode an optimization problem onto its energy profile, so its minimum-energy state can be associated with the problem's optimal solution, which can be achieved through quantum properties, like superposition, tunneling and adiabatic evolution. Current quantum annealing systems permit to encode and solve Quadratic Unconstrained Binary Optimization (QUBO) or Ising problems, characterized by the involving of binary variables with constraints on single variables or pairs. Further details can be found in Sections 6.1 and 6.2.
A complete description of the formalism of the problems solvable with these paradigms can be found in [7] and [11], for quantum circuit model and quantum annealing respectively.

Quantum supremacy and quantum advantage

Quantum supremacy is a term employed to describe an operating regime where quantum computers could effectively execute tasks practically impossible for classical computers. This term can be extremely misleading, since it seems that classical

computers would become useless in a quantum supremacy regime.
First of all, it is important to remind that contemporary quantum computing hardware has limited computational capabilities. Moreover, contour infrastructures for its monitoring are extremely expensive, thus inserting an additional limitation to fabrication of quantum computers with many qubits. This means that time required to achieve quantum supremacy at hardware level — *i.e.* that a quantum computer can operate with many thousands of fault-tolerant qubits, thus ensuring a total amount of processable data much larger than the corresponding one of a classical computer — is expected to be still too much.
On the other hand, classical computers do not have to be excessively underestimated. In 2019 a research article published by Google on *Nature* declared the achievement of quantum supremacy [12], in terms of execution of a random quantum circuit which was effectively run on a fifty-three-qubit quantum computer and resulted too large for being simulated on a classical computer. In particular, a simulation of 10000 years was estimated on the Summit supercomputer. However, IBM team demonstrated that a reasonable classical hardware and software monitoring of the simulation permitted to execute the task in a couple of days [13], thus proving that an effective quantum supremacy regime has not been achieved.
In conclusion, considering the possibility to stress optimization at classical level and also taking into account the current limits of quantum computation at hardware level (in terms of computational capabilities and infrastructures for their monitoring), it would be convenient that classical and quantum computers do not fight for supremacy, but they co-operate to find solution of complex problems for Society and Industry, exploiting their current strengths. In particular, quantum computation should be employed as a sort of **accelerator**, capable of providing a **computational advantage** with respect to their classical counterpart in solving some specific problems. For this reason, the term **quantum advantage** can be considered more effective to refer to the use-specific computational complexity reduction of a quantum algorithm with respect to its best classical counterpart.

1.3 Quantum computers in NISQ era

The current era of quantum computing is called Noisy Intermediate-Scale (**NISQ**). The term was coined by John Preskill in 2018 [14] and it is used to describe the current state of the art in the fabrication of quantum processors, whose leading devices contain at most few hundred qubits (in the circuit model case), a value not sufficient for reaching fault-tolerance and for achieving a generalized quantum advantage, at least for what concerns the total amount of processable data.
Quantum computers based on both computational paradigms have been already fabricated with different technologies. Details about these are reported in Sections 3.3.2 and 6.4 for quantum circuit model and quantum annealing respectively. In

this context, some information about their generic limitations are provided here, starting from the words constituting the acronym NISQ (except for *Quantum*).
Noisy is related to the fact that current quantum processors — due to the high sensitivity to thermodynamic and radiation-related interactions with the external environment — are affected by noise, which can force a loss of information due to decoherence phenomena [7] and errors in the executed operations. The two most critical decoherence phenomena are relaxation and decoherence. They are both dynamic phenomena which tend to be more significant for $t \to \infty$: the first one is associated with the gradual approaching of the quantum state to its ground, *i.e.* the lowest-energy state, while the other one is associated with the gradual impossibility to properly distinguish quantum states in superposition, thus obtaining a mixture, represented by the white ball in Figure 1.4.

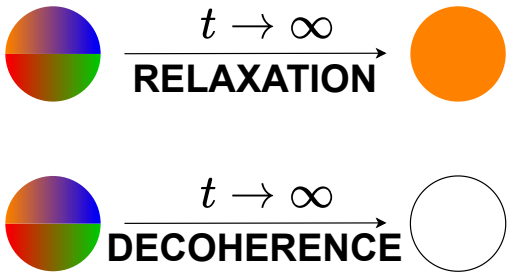


Figure 1.4: Qualitative representation of relaxation and decoherence phenomena.

In order to have an idea of a possible execution error, Figure 1.5 shows a conceptual representation of a possible gate error in the quantum circuit model: the unbalance of the measurement probabilities of 0 and 1 for a gate, which is expected to force an uniform probability distribution when applied to the qubit state associated with 0. Errors like these are to the fact that each operation executed on a quantum computer can be slightly different from the expected one.

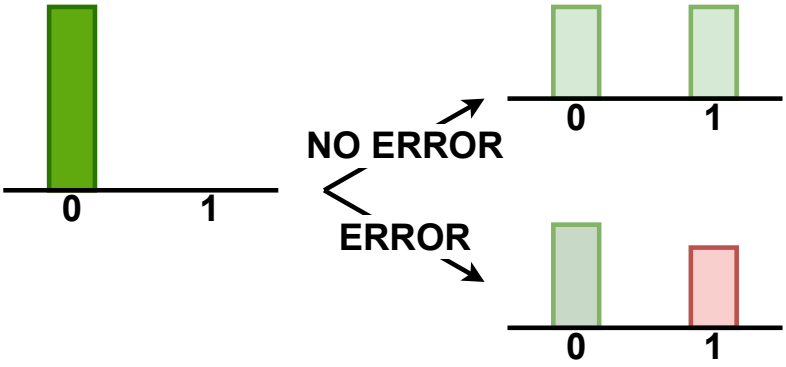


Figure 1.5: Conceptual representation of a gate error which does not ensure an equal superposition of states encoding 0 and 1.

Intermediate-Scale refers to the fact that current quantum computers have a limited number of qubits (usually less than one hundred in those based on the quantum circuit model and some thousand in quantum annealers), thus limiting the amount of data processable by them. Noise and scaling are clearly related. In fact, it is convenient to limit the number of qubits on current quantum computers, for having a better control of the available qubits and keeping the decoherence effects as negligible as possible. Moreover, sophisticated and very expensive contour infrastructures — for monitoring temperature, shielding from radiations, manipulating qubits state, *etc.* — are required, thus also implying high costs also for hardware with few qubits. In order to have an idea of the complexity of a quantum computer architecture, Figure 1.6 shows a schematic representation of the so-called *quantum chandelier*, which is the typical infrastructure for controlling superconducting qubits of both computational paradigms discussed in section 1.2. Temperature is gradually reduced up to about 15 mK with machinery having costs in the order of millions of dollars. Another limitation of current quantum computers is connectiv-

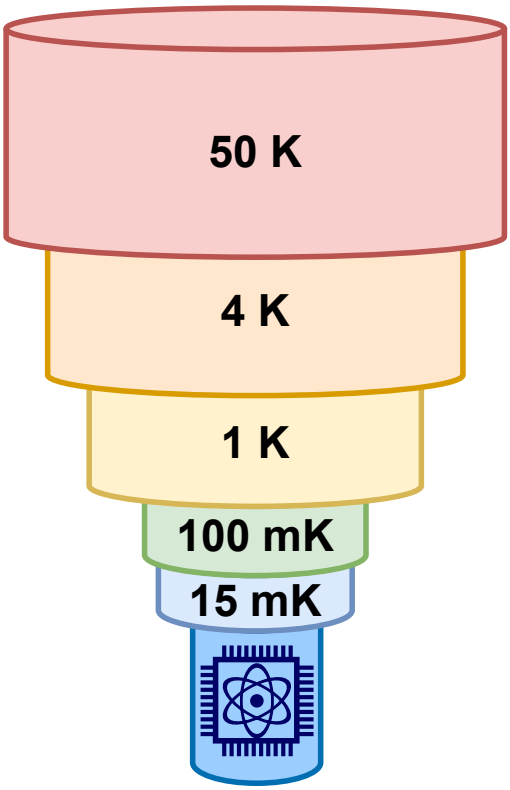


Figure 1.6: Schematic representation of temperature lowering in a generic *quantum chandelier*.

ity, common to both paradigms and consists in the unavailability, in some current

quantum computer architectures (such as the left one in Figure 1.7), of direct connections between each pair of qubits. Even though current quantum computers

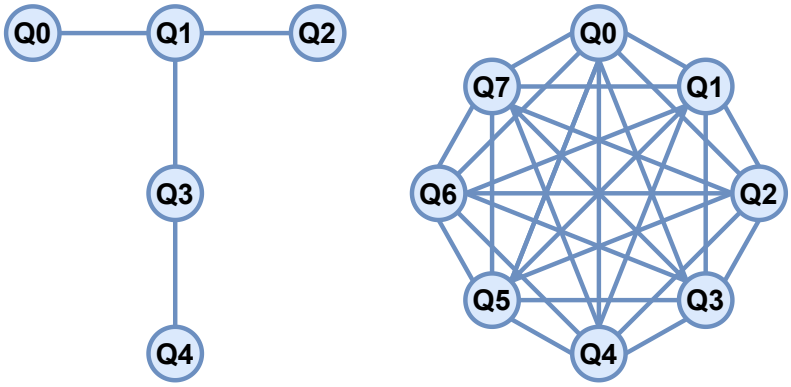


Figure 1.7: Partially-connected (left) and fully-connected (right) quantum computers.

show these limitations and others more characteristic, some manufacturers already permit to users a cloud access to their hardware, on either dedicated domains or common environments (*e.g.* Amazon Web Services). These manufacturers are both private companies — such as Honeywell and IonQ, fabricating trapped ions, quantum computers and IBM and Rigetti, which focus their attention on superconducting qubits — or academic research groups, such as QuTech at TU Delft, which permits a access to a quantum processor with two silicon quantum-dot qubits. Access can be both free or subjected to fees and it is possible to ascertain that the free one is generally limited to devices involving no more than fifteen highly-noisy qubits and in the available tools routines for optimizing the qubits physical and control parameters are not generally available.

1.4 Defining a toolchain for a quantum computing ecosystem

Even though the characteristic hardware limitations of NISQ era affect the applications which can be effectively run on real quantum computers, an attempt of organizing quantum computation in a perspective view can be already done, considering that fundamental theory is substantially consolidated.
Different research contributions in quantum computing can be provided and these can be organized on multiple layers. Figure 1.8 shows a schematic representation of a quantum computing toolchain, sufficiently generic for being arranged to both quantum circuit model and quantum annealing paradigms.
On the left, the formulation of quantum-compliant models is shown. Quantum

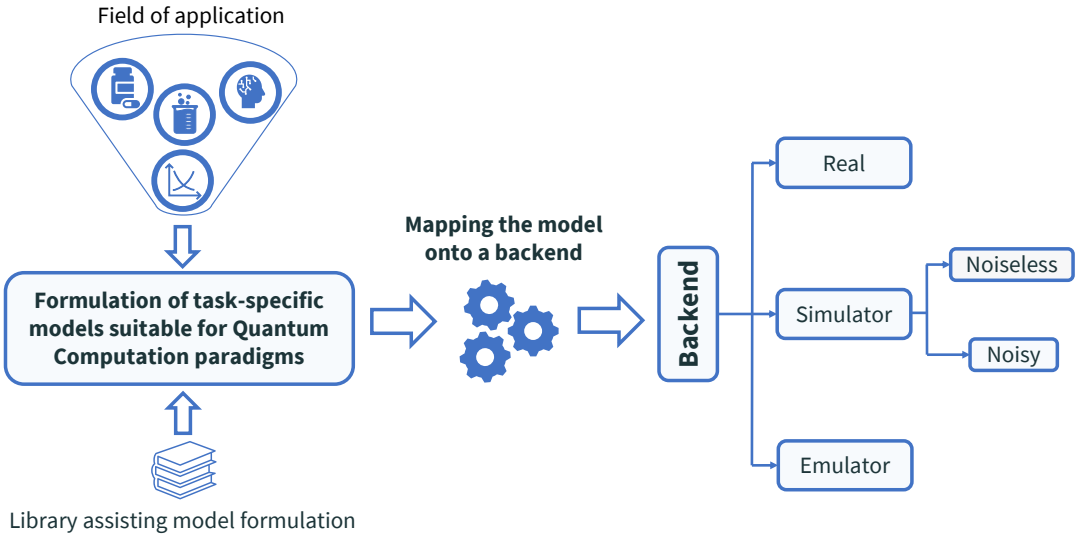


Figure 1.8: Organization of a generic toolchain for quantum computing technologies

computing can be employed in different contexts, which are not only those reported in the introduction of this thesis, but also others more *exotic*, such as music and performing arts [15], [16]. According to the application, a quantum circuit or an Ising/QUBO model can be formulated. Libraries available in the state of art or developed *ad-hoc* can be exploited for this task. Then, the model can pass for an intermediate layer, which is responsible for properly implementing a model-to-backend mapping. In the case of quantum circuit model, this operation is called compilation [17], while in the case of quantum annealing it is called embedding [18]. Finally, the model can be provided to a backend to obtain a solution. Three different backends are shown:

- **real quantum computers**, which can be implemented with different technologies, *e.g.* those introduced in Section 1.3;
- software **simulators**, which try to estimate the response of a quantum computer through functional or physical models, even including non-ideality phenomena (also called noise);
- hardware or software **emulators**, which try to mimic a noiseless quantum computer to obtain its ideal results.

In order to properly implement and formalize the described toolchain, multiple competencies are required. In fact, the development of libraries assisting the models formulation requires a good knowledge of the mathematical formalism behind

them — which is mainly a prerogative of computer scientists, mathematicians, theoretical physicists and logic circuits designers — the mapping phase is expected to require people with an expertise for circuits optimization and routing procedures (in general in the so-called Design Automation [19]), while the backend design and fabrication clearly depend on the backend itself. Experimental physicists and engineers with different backgrounds (from microwave to cryogenics, from embedded systems for data post-processing to fabrication) can co-operate to build a real quantum computer, while simulators and emulators can be effectively implemented by hardware or software engineers — for what concerns the development of computationally efficient simulation or emulation platforms — and by experts of electron or quantum devices, for what concerns the definition of simulation models.
The proposed toolchain is substantially supported by multiple frameworks for quantum computing, which are usually declined into Python libraries, having all the routines for developing a quantum algorithm and for executing it on different backends, available locally o remotely through a cloud access. These libraries are developed by either quantum hardware — such as Qiskit by IBM Q, Cirq by Google, Forest by Rigetti and Strawberry Fields by Xanadu for quantum circuit model and Leap by D-Wave Systems for quantum annealing — - or by startups developing software for quantum computation — such as Orquestra by Zapata Computing — or by Academia, such as ProjectQ by ETH Zürich and the open-source toolset for design automation for quantum computing JKQ developed at the Johannes Kepler University of Linz (Austria).

1.5 Thesis motivations and organization

This thesis reports the activities done during the three years research as PhD student at the VLSI Lab at the Department of Electronics and Telecommunications of Politecnico di Torino. In the toolchain shown in Figure 1.8, these belong to the **noisy simulation** and **model formulation** sections. In the first case, these are strictly related to the quantum circuit model, while in the other they are declined into the construction of a QUBO model for solving the scheduling of users in a Multiple-Input-Multiple-Output (MIMO) wireless telecommunication system characterized by the **Joint Transmission** mechanism. The first activity was internal to Politecnico di Torino, while the second one was done in a research contract between the Department of Electronics and Telecommunications of Politecnico di Torino and TIM S.p.A.

1.5.1 Noisy simulation of quantum computers

Physical-dependent classical simulation of quantum computers can help to evaluate the effectiveness of an algorithm on a real device, taking into account its physical

degrees freedom and its native operations. Moreover, this simulative approach can assist the design of real hardware, since it permits to predict the effects of some implementation choices, without fabricating any device. The canonical approach for the simulation of quantum mechanical systems requires the resolution of differential equations involving matrices. However, this could be extremely heavy from a computational point of view and could limit the scalability of the simulation. For this reason, the availability of approximated models which mimic the physical behaviour can speed-up the simulation, without loosing significantly the accuracy of the results (at least in a fiducial operating area). In this thesis, a model like this is called compact; it lives at a higher level of abstraction than a physical one and is thought for the simulation of quantum circuits, taking into account the characteristic physical parameters of the devices to be described. In other words, it can be employed for application-specific quantum computing engineering, working on different levels: from devices (affecting model parameters) to applications themselves, through circuit optimization. From a conceptual point of view, they can be associated with the device backend noise model available in Qiskit [20]. The expected workflow with compact models is the following:

1. Using physical simulators for obtaining the parameters to be employed in compact models.

2. Building compact models.

3. Simulating quantum circuits with compact models.

4. Improving compact models, after the comparison of outcomes provided by it with consolidated references in the state of the art.

5. Determining how physical quantities (or more abstract features, such as technology-dependent compilation) should be arranged to improve quantum circuits execution reliability.

It is generally an iterative procedure for improving quantum computation on multiple levels. In this thesis, a methodology for the definition of compact models and the construction of a complete quantum circuit simulation infrastructure based on these is proposed and all the obtained results are reported and properly commented, thus also permitting to define the validity limits of the proposed models, which are related to both specific technologies (molecular spins) and phenomena transversal to all quantum circuit model technologies (relaxation, decoherence and off-resonance).
As already written, the technologies explored in this thesis are molecular; these are the Nuclear Magnetic Resonance (NMR) and the so-called switch-based and belong to the more general family encoding quantum information onto spins. The choice of these among the other families of quantum computers (*e.g.* superconducting and

trapped-ions) can be traced back to multiple reasons. The first to be mentioned is the simplicity of the quantum systems associated with qubits, which permitted to define a methodology for the construction of compact models to be exploited also for more complex devices. Another one is that spin qubits have aroused the interest of Industry and Academia for the possibility of fabricating scalable devices, potentially cheaper and integrable on classical computing devices. Among these, spin qubits in semiconductor quantum dots [21] or in diamond [22] — fabricated at QuTech in Delft and Fujitsu — could be mentioned. Moreover, the Chinese company SpinQ started the purchase of desktop NMR quantum computers [23]. According to all these industrial and commercial positive perspectives, the study of spin qubits was found to be interesting from a scientific point of view and the choice of starting from NMR, which is the simplest spin technology, was considered reasonable.

1.5.2 Joint Transmission

Joint Transmission (JT) is a technique proposed in the last ten years to improve wireless communication, especially in mobile networks. It permit to increase the quality of signal received by users, removing interference effects. This mechanism also increases the transmission rate. An optimization problem, looking for the best users subset served in JT in terms of data transmission rate, can be defined to improve the effectiveness of the technique. The solution space tends to increase significantly with the total number of potentially served users, thus making unfeasible a *brute-force* approach, since it could imply a too high execution time, which is important in this context since it should be solved continuously at *run-time*, according to the time variation of users. For these reasons, heuristic approaches should be employed for problem solving. Since, in general, solutions provided by classical solvers are often not completely satisfactory, research is moving towards quantum solvers, which are expected to achieve — in the future — better solutions in less time [24]. A users scheduling algorithm, based on a QUBO model, is described in this thesis. This choice of this model can be traced back to the interest in having a future compatibility with quantum annealers, which are nowadays the most advanced quantum computers suitable for optimization problems. In this context, it is already possible to say that the proposed QUBO is responsible of a first selection of users among all available and this subset is then explored with *brute-force*, to find the best solution.

1.5.3 Thesis organization

Thesis is based on five parts:

1. Introduction (chapter 1-chapter 2).

2. Contributions on the quantum circuit model area (chapter 3-chapter 5).
3. Contributions on QUBO modelling (chapter 6-chapter 7).
4. General conclusions and future perspectives (chapter 9-chapter 8).
5. Appendices.

More details about chapters are reported in the following.

Chapter 2 reports a recap of quantum mechanics concepts which will be useful in the following of the thesis.

Chapter 3 is an introduction to the quantum circuit model, in terms of both formalism and technologies.

Chapter 4 describes the simulation methodology based on compact models, with details about models transversally employable with different technologies. This chapter is strictly related to Appendix B.

Chapter 5 describes the simulation methodology based on compact models, with details about models transversally employable with different technologies.

Chapter 6 introduces QUBO modelling and some solvers of these optimization problems: simulated annealing, quantum annealing and emulators of quantum annealers.

Chapter 7 reports the research activity about the definition of a QUBO-based algorithm for solving the Joint Transmission problem in Multiple Input Multiple Output (MIMO) telecommunication systems. This activity belonged to a research grant with TIM S.p.A.

Chapter 9 contains general conclusions concerning the research activity of this thesis and some hints for future perspectives are provided.

Chapter 2

Basic Concepts of Quantum Mechanics

Quantum mechanics describes the behavior of matter and its interactions with energy at atoms and subatomic scales. The qubit, *i.e.* the quantum unit of information, is encoded onto a quantum physical system, such as the spin state of an electron, atomic energy levels, polarization states of single photons, or paths in an interferometer.
In order to understand all the mechanisms behind quantum information — in terms of protocols, routines and hardware, no matter if the application field is computing, communication or sensing applications — the knowledge of the fundamental formalism of quantum mechanics is required. This chapter is going to provide the concepts of quantum mechanics which can help in understanding the following of this thesis. It is important to clarify that it is not thought for replacing any textbook on quantum mechanics and takes mainly inspiration from [7] and [25].

2.1 Linear Algebra of Quantum Mechanics

According to the so-called **Dirac notation**, the state of a quantum system and its time evolution is described in terms of vectors and matrices. For this reason, a review of some fundamental concepts of linear algebra is reported in the following.

2.1.1 Hilbert spaces and the Dirac notation

A **Hilbert space** is a generalization of Euclidean space, characterized by the possibility of having any number of dimensions and involving complex numbers. Every vector belonging to this space can be multiplied by a complex scalar or by another vector of the same dimension, since an inner complex scalar product is available.

The results of scalar product, differing from the order of vectors, are complex conjugate, so the result depends on the order in which the vectors are taken:

$$\mathbf{v} \cdot \mathbf{u} = (\mathbf{u} \cdot \mathbf{v})^* = \alpha \in \mathbb{C} \,. \tag{2.1}$$

Equivalently to the Euclidean space, the inner product of a vector belonging to a Hilbert space with itself $\mathbf{u} \cdot \mathbf{u}$ is real and it is the square value of its norm $|\mathbf{u}|^2$, a quantity that can be interpreted as the vector's length.
Each vector can be written as linear combination of orthonormal basis vectors:

$$\mathbf{v} = \sum_i \alpha_i \mathbf{u}_i \,, \tag{2.2}$$

where $\mathbf{u}_i \cdot \mathbf{v} = \alpha_i$ and $\mathbf{u}_i \cdot \mathbf{u}_j = \delta_{ij}$ (Kronecker delta).
In the context of quantum mechanics and Dirac notation, quantum states are described by column vectors called **kets**:

$$|\psi\rangle = \begin{bmatrix} a_{\psi_0} \\ a_{\psi_1} \\ \vdots \\ a_{\psi_{n-1}} \end{bmatrix} = \sum_{i=0}^{n-1} a_{\psi_i} |\psi_i\rangle \,, \tag{2.3}$$

while their transposed conjugate are called **bras**:

$$\langle\psi| = [a_0^*, a_1^*, \cdots, a_{n-1}^*] = \sum_{i=0}^{n-1} a_{\psi_i}^* \langle\psi_i| = (|\psi\rangle^T)^* \,. \tag{2.4}$$

The inner product of a *bra* and a *ket* is represented by the **bra-ket** notation

$$\langle\phi \,|\, \psi\rangle = \langle\phi| \cdot |\psi\rangle = \sum_{i=0}^{n-1} a_{\phi_i}^* a_{\psi_i} \,. \tag{2.5}$$

It is important to further clarify that, apart from the fact that complex numbers are involved and that the transposition operation requires also the complex conjugation, Hilbert spaces behave exactly as the Euclidean one, so all theory of linear algebra and geometry courses [26], typically involving real-number vectors, can be easily arranged for Hilbert spaces. Details about the physical meaning of the elements of ket vectors are reported in subsection 2.2.2.

2.1.2 Operators

In linear algebra, an operator is a linear mapping operation which acts on the elements of a space vector V to produce those on another one W. Linearity implies the **effects superposition principle**, so the result of the application of an operator

on a linear combination of vectors $A\left(\sum_i a_i \mathbf{v}_i\right)$ is equal to the linear combination of the results obtained applying the same operator on each vector $\mathbf{v}_i$ independently on others:

$$A\left(\sum_i a_i \mathbf{v}_i\right) = \sum_i a_i A(\mathbf{v}_i)\,. \tag{2.6}$$

Moreover, linearity ensures that the sum of two operators applied on the same vector is equal to:

$$(A+B)\mathbf{v} = A\mathbf{v} + B\mathbf{v}\,. \tag{2.7}$$

From a practical point of view, linear operators are represented as matrices that map a vector $\mathbf{v}_j$ belonging to the basis of V into a linear combination of vectors $\mathbf{w}_i$ belonging to the basis of W.

$$A\mathbf{v}_j = \sum_i A_{ij}\mathbf{w}_i\,. \tag{2.8}$$

It is important to precise that, in the context of quantum computing, the initial and final vector spaces are the same, so the effect of an operator on a state vector consists in changing the linear combination of the basis states describing it. In other words, according the Dirac notation, a linear operator transforms kets into other kets according to

$$A\,|\psi\rangle = |\psi'\rangle\ . \tag{2.9}$$

An equivalent transformation on bras can be done

$$\langle\psi|\,A = \langle\psi'|\ . \tag{2.10}$$

Given an operator A, its Hermitian conjugate or **adjoint** $A^\dagger$ is an operator such that

$$\langle\psi|\,A^\dagger = A\,|\psi\rangle\ . \tag{2.11}$$

or equivalently

$$\left\langle\psi|A^\dagger|\phi\right\rangle = (\langle\phi|A|\psi\rangle)^*\ , \tag{2.12}$$

so the Hermitian conjugate is nothing but the transposed conjugate of the matrix A

$$A^\dagger_{ij} = A^*_{ji}\,. \tag{2.13}$$

Moreover, it is possible to prove that $(A^\dagger)^\dagger = A$.
All these concepts can currently seem too abstract and generic. However they will result extremely useful for understanding quantum gates in chapter 3.

2.1.3 Tensor product

Tensor product permits to put vector spaces together to form larger spaces. It is fundamental for studying the evolution of multiparticle quantum systems. Given two vector spaces V and W of dimension m and n respectively, the tensor product $V \otimes W$ combines the elements of V and W, thus obtaining a $m \cdot n$ dimensional vector space. A practical representation of the tensor product is the **Kronecker product**. Given two matrices A $(m \times n)$ and B $(p \times q)$, the Kronecker product is

$$A \otimes B = \begin{bmatrix} A_{11}B & \cdots & A_{1n}B \\ A_{21}B & \cdots & A_{2n}B \\ \vdots & \vdots & \vdots \\ A_{m1}B & \cdots & A_{mn}B \end{bmatrix}. \tag{2.14}$$

An example is reported, with two column vectors of dimension 3 and 2 respectively.

$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \otimes \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} a_1 b_1 \\ a_1 b_2 \\ a_2 b_1 \\ a_2 b_2 \\ a_3 b_1 \\ a_3 b_2 \end{bmatrix}.$$

Linearity of tensor product ensures that the it satisfies the following basic properties:

1. Given a scalar z and two vectors $\mathbf{v}$ and $\mathbf{w}$ belonging to V and W respectively:

$$z(\mathbf{v} \otimes \mathbf{w}) = (z\mathbf{v}) \otimes \mathbf{w} = \mathbf{v} \otimes (z\mathbf{w}). \tag{2.15}$$

2. Given two vectors $\mathbf{v}_1$ and $\mathbf{v}_2$ in V and $\mathbf{w}$ in W:

$$(\mathbf{v}_1 + \mathbf{v}_2) \otimes \mathbf{w} = (\mathbf{v}_1 \otimes \mathbf{w}) + (\mathbf{v}_2 \otimes \mathbf{w}). \tag{2.16}$$

3. Given $\mathbf{v}$ in V and two vectors $\mathbf{w}_1$ and $\mathbf{w}_2$ in W:

$$\mathbf{v} \otimes (\mathbf{w}_1 + \mathbf{w}_2) = (\mathbf{v} \otimes \mathbf{w}_1) + (\mathbf{v} \otimes \mathbf{w}_2). \tag{2.17}$$

Tensor product can be also applied on linear operators. In fact, given two linear operators A and B applied to $\mathbf{v}$ and $\mathbf{w}$ belonging V and W respectively, it is possible to define an operator $A \otimes B$ on the $V \otimes W$ vector space

$$(A \otimes B)(\mathbf{v} \otimes \mathbf{w}) \equiv A\mathbf{v} \otimes B\mathbf{w}. \tag{2.18}$$

Linearity permits to represent an arbitrary linear operator mapping $V \otimes W$ to $V' \otimes W'$ as a linear combination of tensor products of operators mapping V to V' and W to W':

$$C = \left(\sum_i c_i A_i \otimes B_i \right) \mathbf{v} \otimes \mathbf{w} = \sum_i (c_i A_i \mathbf{v} \otimes B_i \mathbf{w}) \,. \tag{2.19}$$

These properties are exploited in the definition of quantum gates and quantum computing algorithms.
In conclusion, it is important to precise that tensor product does not have a *reciprocal* operation. This means that it is not always possible, given a vector $\mathbf{v} \in V \otimes W$, to obtain two sub-vectors $\mathbf{v} \in V$ and $\mathbf{w} \in W$. Two examples are reported in the following, involving a larger and two smaller spaces, with vectors having dimensions equal to four (larger) and two (smaller) respectively. Before showing them, it is important to remind that:

$$\begin{gathered} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} \otimes \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} a_1 b_1 \\ a_1 b_2 \\ a_2 b_1 \\ a_2 b_2 \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix} \\ \begin{cases} a_1 b_1 = c_1 \\ a_1 b_2 = c_2 \\ a_2 b_1 = c_3 \\ a_2 b_2 = c_4 \end{cases} . \end{gathered} \tag{2.20}$$

The *decomposition* of a tensor product convolves a system of equations, each one with a product between two variables, each one associated with one of the two subspaces. The two problems are finally reported:

$$\begin{gathered} \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\ \frac{1}{\sqrt{2}} \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} \rightarrow \text{Not decomposable} \,. \end{gathered} \tag{2.21}$$

It is possible to prove that the system of equations associated with the lower vector does not have common solutions, so decomposition cannot be done.
While the higher four-element vector can be written as the tensor product of two smaller ones, the second one cannot be decomposed. In the context of quantum mechanics, the impossibility to separate a larger vector in smaller ones is a *mathematical signature* of **entanglement**, introduced in subsection 1.1.1.

2.1.4 Hermitian and unitary operators

Most processes of quantum mechanics are described by **Hermitian** or **unitary** operators. The main feature of Hermitian operators, to which the Hamiltonian ones belong, is that their adjoint is equal to themselves:

$$H = H^\dagger \,. \tag{2.22}$$

An example of Hermitian matrix is reported in the following:

$$\left(\begin{bmatrix} 1 & -2\mathrm{i} \\ 2\mathrm{i} & -3 \end{bmatrix}\right)^\dagger = \left(\begin{bmatrix} 1 & 2\mathrm{i} \\ -2\mathrm{i} & -3 \end{bmatrix}\right)^* = \begin{bmatrix} 1 & -2\mathrm{i} \\ 2\mathrm{i} & -3 \end{bmatrix} .$$

Real symmetric matrices, and consequently the sub-cases block diagonal and anti-diagonal matrices

$$\begin{bmatrix} A & B \\ B & C \end{bmatrix}$$

are trivially Hermitian. In the complex case, Hermitian matrix condition is quite harder to be satisfied. Important particular cases are diagonal anti anti-diagonal block matrices:

$$\begin{bmatrix} A & 0 \\ 0 & A^\dagger \end{bmatrix} , \begin{bmatrix} 0 & A \\ A^\dagger & 0 \end{bmatrix} \tag{2.23}$$

Unitary operators have their inverse equal to their adjoint, so that

$$UU^\dagger = U^\dagger U. \tag{2.24}$$

The **identity** operator is the most evident example of unitary operator ($\mathbb{I}^\dagger = \mathbb{I}$). Hermitian and unitary operators are related by the fact that the complex exponential of a Hermitian matrix is unitary:

$$\begin{aligned} U &= \mathrm{e}^{\mathrm{i}\alpha A} \,, \\ UU^\dagger &= \mathrm{e}^{\mathrm{i}\alpha A}\mathrm{e}^{-\mathrm{i}\alpha A} = \mathbb{I} \,, \end{aligned} \tag{2.25}$$

with α generic scalar real number.

2.1.5 Eigenvalues and eigenvectors

An **eigenvector** of a linear operator A on a vector space is a non-zero column vector $\mathbf{v}$ such that

$$A\mathbf{v} = \lambda\mathbf{v} \,, \tag{2.26}$$

where λ is a scalar named **eigenvalue**. If Equation 2.26 is interpreted in a Euclidean way, the application of an operator to an eigenvector of its does not change the orientation of the vector, but only scales its length by an amount λ.

The problem of finding eigenvalues and eigenvector of a matrix A is equivalent to the diagonalization of A

$$A = S\Lambda S^{-1}\,, \tag{2.27}$$

where Λ is a diagonal matrix with the eigenvalues of A and S is a matrix formed by the eigenvectors of A, organized in columns.

$$\begin{aligned} \Lambda &= \mathrm{diag}\left([\lambda_1, \lambda_2, \cdots, \lambda_n]\right)\,, \\ S &= [\mathbf{v}_1 \mathbf{v}_2 \cdots \mathbf{v}_n]\,. \end{aligned} \tag{2.28}$$

From a practical point of view, the calculation of eigenvalues and eigenvectors is done by rewriting Equation 2.27 as:

$$\begin{aligned} AS &= S\Lambda\,, \\ A\,[\mathbf{v}_1 \mathbf{v}_2 \cdots \mathbf{v}_n] &= [\lambda_1 \mathbf{v}_1 \lambda_2 \mathbf{v}_2 \cdots \lambda_n \mathbf{v}_n]\,. \end{aligned} \tag{2.29}$$

Considering the lower expression in Equation 2.29, a linear system can be associated with each column i of S:

$$\begin{aligned} A\mathbf{v}_i &= \lambda_i \mathbf{v}_i\,, \\ (A - \lambda_i \mathbb{I})\mathbf{v}_i &= 0\,. \end{aligned} \tag{2.30}$$

Eigenvalues λ_i can be computed all together as roots of the n^{th} polynomial given by the determinant of $A - \lambda\mathbb{I}$. According to the fundamental theorem of algebra, every polynomial has at least one complex root, thus ensuring the existence of one eigenvalue and a corresponding eigenvector for a linear operator A. When the eigenvalue polynomial presents repeated roots, the condition of **degenerate** eigenvalues takes place.
Once the eigenvalues λ_i have been determined, it is possible to find their corresponding eigenvectors solving the linear systems of equations associated with the columns of Equation 2.30. For each linear system of equation, a characteristic polynomial root is assigned to λ_i, with coefficients ψ_j as solutions.
It is possible to prove that an **Hermitian operator has real eigenvalues and mutually orthogonal eigenvectors** (their inner product is null), so for any Hermitian operator in an n-dimensional Hilbert space n eigenvalues always exist. On the other hand, **unitary operators have complex eigenvalues with magnitude one and their corresponding eigenvectors are orthogonal**, so they constitute an orthonormal basis.
The previous concepts are employed in the same way in quantum mechanics. From a practical point of view, the eigenvalue problem is associated with a Hamiltonian operator and is exploited to find the energy states of a quantum system (usually called eigenstates).

2.1.6 Trace

The **trace** of a matrix A is a function, defined to be the sum of the diagonal elements of A

$$\mathrm{tr}(A) = \sum_i A_{ii}. \tag{2.31}$$

It is possible to prove that trace is cyclic

$$\mathrm{tr}(AB) = \mathrm{tr}(BA) \tag{2.32}$$

and linear

$$\mathrm{tr}(aA + bB) = a\mathrm{tr}(A) + b\mathrm{tr}(B) \tag{2.33}$$

where A and B are two arbitrary matrices and a and b are two scalars.
Trace can be extended to the matrix representation of any operator described by matrix A. Moreover, the invariance of the trace under unitary transformations $A \to UAU^\dagger$

$$\mathrm{tr}(UAU^\dagger) = \mathrm{tr}(A) \tag{2.34}$$

ensures the well-definition of the trace of an operator.
Trace is practically of interest with density matrix, discussed in subsection 2.3.1.

2.1.7 Operator exponentiation

As already introduced in subsection 2.1.4, an important link between unitary and Hermitian operators is given by matrix exponentiation. It is possible to prove that, given a unitary operator U, there is a Hermitian operator A such that

$$U = \mathrm{e}^{-\mathrm{i}A}\,. \tag{2.35}$$

Reminding that the eigenvalues λ_i of a unitary operator have magnitude equal to 1, they can be written as e^{-ia_i}, where a_i are real numbers corresponding to the eigenvalues of a Hermitian operator A. The **exponentiation of a matrix** A is generally computed through a series expansion

$$\mathrm{e}^{\mathrm{i}A} = \mathbb{I} + \mathrm{i}A + \frac{1}{2!}(\mathrm{i})^2 AA + \frac{1}{3!}(\mathrm{i})^3 AAA + \cdots = \sum_{k=0}^{+\infty} \frac{(\mathrm{i})^k A^k}{k!}\,, \tag{2.36}$$

or exploiting the equivalent **Euler's formula for matrices**:

$$\mathrm{e}^{\mathrm{i}A} = \cos(A) + \mathrm{i}\sin(A)\,, \tag{2.37}$$

with

$$\begin{aligned} \cos(A) &= I - \frac{A^2}{2!} + \frac{A^4}{4!} + \cdots = \sum_{k=0}^{+\infty} \frac{(-1)^k A^{2k}}{(2k)!}\,, \\ \sin(A) &= A - \frac{A^3}{3!} + \frac{A^5}{5!} + \cdots = \sum_{k=0}^{+\infty} \frac{(-1)^k A^{2k+1}}{(2k+1)!}\,. \end{aligned} \tag{2.38}$$

Unfortunately, calculations of series involving matrices could be hard from a computational point of view. However, there are some simpler particular cases which can be exploited for simplifying the calculation. In particular, since the exponentiation of a diagonal matrix is:

$$\exp\left(\begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix}\right) = \begin{bmatrix} \mathrm{e}^a & 0 \\ 0 & \mathrm{e}^b \end{bmatrix}, \tag{2.39}$$

and:

$$\mathrm{e}^{S\Lambda S^{-1}} = S\mathrm{e}^{\Lambda}S^{-1}, \tag{2.40}$$

the exponential of a generic matrix can be obtained in a simpler way by first diagonalizing the matrix and then exploiting Equation 2.40.
Matrix exponentiation does not properly satisfy the well-known property of scalar exponentiation $\mathrm{e}^{a+b} = \mathrm{e}^a\mathrm{e}^b$. In fact, the equivalence is only valid in the following limit:

$$\mathrm{e}^{A+B} = \lim_{n\to\infty}\left(\mathrm{e}^{\frac{A}{n}}\mathrm{e}^{\frac{B}{n}}\right)^n, \quad \text{if } [A,B] = AB - BA = 0\,. \tag{2.41}$$

In Equation 2.41, $\mathrm{e}^{\frac{A}{n}}$ is the n^{th} rooth of e^A and $[A,B]$ is the commutator, which must provide a null matrix as result. This property is exploited for approximating the unitary evolution of an Hamiltonian given by the sum of multiple terms, according to the so-called Trotter-Suzuki expansion.

2.1.8 Pauli matrices

The three 2×2 complex **Pauli matrices** can be reasonably considered the most employed mathematical instruments of quantum information theory. They are reported in the following:

$$\sigma_x = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \ \sigma_y = \begin{bmatrix} 0 & -\mathrm{i} \\ \mathrm{i} & 0 \end{bmatrix}, \ \sigma_z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \tag{2.42}$$

An important feature of these matrices is that they are Hermitian and unitary at the same time. The proof for σ_y is reported in the following:

$$\begin{aligned} \sigma_y^\dagger &= \left\{\left(\begin{bmatrix} 0 & -\mathrm{i} \\ \mathrm{i} & 0 \end{bmatrix}\right)^T\right\}^* = \begin{bmatrix} 0 & \mathrm{i}^* \\ -\mathrm{i}^* & 0 \end{bmatrix} = \begin{bmatrix} 0 & -\mathrm{i} \\ \mathrm{i} & 0 \end{bmatrix} = \sigma_y\,, \\ \sigma_y\sigma_y^\dagger = \sigma_y^2 &= \begin{bmatrix} 0 & -\mathrm{i} \\ \mathrm{i} & 0 \end{bmatrix}\begin{bmatrix} 0 & -\mathrm{i} \\ \mathrm{i} & 0 \end{bmatrix} = \begin{bmatrix} -\mathrm{i}^2 & 0 \\ 0 & -\mathrm{i}^2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \mathbb{I}\,. \end{aligned} \tag{2.43}$$

This property has an important practical consequence: Pauli matrices can be exploited for both describing the Hamiltonian and the unitary evolution of a quantum system.

The Pauli matrices have eigenvalues ± 1 and their unit eigenvectors are:

$$\begin{aligned} |\psi\rangle_{+,x} &= \frac{1}{\sqrt{2}}\begin{bmatrix}1\\1\end{bmatrix}, \qquad |\psi\rangle_{-,x} = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\-1\end{bmatrix}, \\ |\psi\rangle_{+,y} &= \frac{1}{\sqrt{2}}\begin{bmatrix}1\\\mathrm{i}\end{bmatrix}, \qquad |\psi\rangle_{-,y} = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\-\mathrm{i}\end{bmatrix}, \\ |\psi\rangle_{+,z} &= \begin{bmatrix}1\\0\end{bmatrix}, \qquad |\psi\rangle_{-,z} = \begin{bmatrix}0\\1\end{bmatrix}. \end{aligned} \tag{2.44}$$

The Pauli matrices are related by the following equations:

$$\begin{aligned} \sigma_x\sigma_y &= \mathrm{i}\sigma_z\,, \\ \sigma_y\sigma_x &= -\mathrm{i}\sigma_z\,, \\ \sigma_y\sigma_z &= \mathrm{i}\sigma_x\,, \\ \sigma_z\sigma_y &= -\mathrm{i}\sigma_x\,, \\ \sigma_z\sigma_x &= \mathrm{i}\sigma_y\,, \\ \sigma_x\sigma_z &= -\mathrm{i}\sigma_y\,. \end{aligned} \tag{2.45}$$

An important property of Pauli matrices concerns their complex exponentiation:

$$\mathrm{e}^{-\mathrm{i}\frac{\theta}{2}\sigma_{x,y,z}} = \cos\left(\frac{\theta}{2}\right)\mathbb{I} - \mathrm{i}\sin\left(\frac{\theta}{2}\right)\sigma_{x,y,z} = R_{x,y,z}\left(\theta\right). \tag{2.46}$$

where $\mathbb{I}$ is the identity matrix and $R_{x,y,z}$ refers to the fact, detailed in subsection 2.2.2, that the complex exponential of Pauli matrices can be associated with a vector rotation around one of the three Cartesian axes.
A compact representation of Pauli matrices is given by the Pauli vector:

$$\overrightarrow{\sigma} = \begin{bmatrix}\sigma_x\\\sigma_y\\\sigma_z\end{bmatrix}, \tag{2.47}$$

which permits to write a matrix given by their linear combination as:

$$\overrightarrow{a}\cdot\overrightarrow{\sigma} = \begin{bmatrix}a_x & a_y & a_z\end{bmatrix}\begin{bmatrix}\sigma_x\\\sigma_y\\\sigma_z\end{bmatrix} = a_x\sigma_x + a_y\sigma_y + a_z\sigma_z = \begin{bmatrix}a_z & a_x - \mathrm{i}a_y\\ a_x + \mathrm{i}a_y & -a_z\end{bmatrix}. \tag{2.48}$$

The eigenvalues of matrix $\overrightarrow{a}\cdot\overrightarrow{\sigma}$ are $\pm|\overrightarrow{a}|$ and eigenvectors are

$$\begin{aligned} |\psi\rangle_{+} &= \frac{1}{\sqrt{2|\overrightarrow{a}|(a_z + |\overrightarrow{a}|)}}\begin{bmatrix}a_z + |\overrightarrow{a}|\\ a_x + \mathrm{i}a_y\end{bmatrix}, \\ |\psi\rangle_{-} &= \frac{1}{\sqrt{2|\overrightarrow{a}|(a_z + |\overrightarrow{a}|)}}\begin{bmatrix}a_x + \mathrm{i}a_y\\ a_z + |\overrightarrow{a}|\end{bmatrix}. \end{aligned} \tag{2.49}$$

A relation analogous to that reported in Equation Equation 2.46 is valid for Pauli vector exponentiation, when $\overrightarrow{a}$ is an unit vector $\hat{\mathbf{n}}$:

$$e^{-\mathrm{i}\frac{\theta}{2}\hat{\mathbf{n}}\cdot\overrightarrow{\sigma}} = \cos\left(\frac{\theta}{2}\right)\mathbb{I} - \mathrm{i}\sin\left(\frac{\theta}{2}\right)\hat{\mathbf{n}}\cdot\overrightarrow{\sigma}\,. \tag{2.50}$$

This property is useful in some models, *e.g.* in the calculation of the unitary evolution of spin-$\frac{1}{2}$ under magnetic resonance, described in subsection 3.3.3.

2.2 Postulates of quantum mechanics

According to the so-called Copenhagen interpretation, quantum mechanics is founded on some non-provable postulates, which are accepted from experimental results. There is not a unique adopted set of postulates of quantum mechanics. In this context, they are going to be reported in the following in a form compatible with quantum computing theory and in an order which could be slightly different from that reported in textbooks, but that could simplify their understanding for the purposes of this thesis.

2.2.1 Observables

As already introduced in subsection 2.1.2 and in subsection 2.1.4, quantum mechanics is massively described in terms of operators. According to one of the postulates of quantum mechanics, a Hermitian operator can be associated with each physical **observable**, *i.e.* a measurable physical quantity.
The undoubtedly most considered operator in this thesis is **Hamiltonian**, which gives the total energy of the system, adding all kinetic and potential energy contributions. The eigenstates of a Hamiltonian operator $\mathcal{H}$ are all possible energy states of the quantum system associated with $\mathcal{H}$. All technologies for quantum computing are described starting from their Hamiltonian, since its analysis permits to uniquely understand the states which can encode quantum information and the mechanisms for implementing gates, as it will more clear after subsection 2.2.3.
Another family of operators to be mentioned is the **spin**, consolidated in Magnetic Resonance spectroscopy and quantum computing:

$$\begin{aligned} S_x &= \frac{\hbar}{2}\sigma_x\,, \\ S_y &= \frac{\hbar}{2}\sigma_y\,, \\ S_z &= \frac{\hbar}{2}\sigma_z\,, \end{aligned} \tag{2.51}$$

where $\hbar = \frac{h}{2\pi} = \frac{6.626\,070\,15 \times 10^{-34}\,\mathrm{J\,Hz^{-1}}}{2\pi} = 1.054\,571\,817 \times 10^{-34}\,\mathrm{J\,s}$ is the **reduced Planck constant** [27], [28] and $\sigma_{x,y,z}$ are the Pauli operators. In all the Hamiltonians in the following of this thesis, these operators will be always written in an *unrolled* way, *i.e.* $\frac{\hbar}{2}$ and a Pauli operator will be always explicitly written and will be not contracted in their corresponding spin operators.

2.2.2 State representation

A quantum state can be described by a column vector, such as that in Equation 2.3. The complex elements of a ket vector a_i are called **probability amplitudes**. Each one associated with one state $|\psi_i\rangle$ belonging to the vector basis, whose magnitude square values $|a_i|^2$ give the probability for the quantum state of being in their corresponding basis state

$$|a_i|^2 = P(|\psi\rangle = |\psi_i\rangle) \tag{2.52}$$

Since every vector in a vector space is a linear combination of the corresponding basis set, it is possible to say that the sum of the probabilities of being in these states must be equal to 1

$$\sum_{i=1}^{n} |a_i|^2 = 1. \tag{2.53}$$

In quantum computing and information theories, the qubit is abstracted from physical details. The main difference between a qubit and a classical bit is that the first is not confined to one of the states $|0\rangle$ and $|1\rangle$ (forming the so-called **computational basis**), but can be found in arbitrary **superposition** states

$$|\psi\rangle \triangleq c_0\,|0\rangle + c_1\,|1\rangle = c_0 \cdot \begin{bmatrix}1\\0\end{bmatrix} + c_1 \cdot \begin{bmatrix}0\\1\end{bmatrix} = \begin{bmatrix}c_0\\c_1\end{bmatrix} \tag{2.54}$$

where $c_0, c_1 \in \mathbb{C}$ are the probability amplitudes, each one associated with one qubit basis state. The square magnitude of each probability amplitude is equal to the probability of finding the qubit in the corresponding state, thus $|c_0|^2 = P(|\psi\rangle = |0\rangle)$, $|c_1|^2 = P(|\psi\rangle = |1\rangle)$ and $|c_0|^2 + |c_1|^2 = 1$. For these reasons, the qubit can be in two different states (encoding 0 and 1) at the same time. The one-qubit model can be generalized to a **quantum register** of N qubits with 2^N basis states $|\psi\rangle = \sum_{i=0}^{2^N-1} c_i\,|\psi_i\rangle$, with $|c_i|^2 = P(|\psi\rangle = |\psi_i\rangle)$ and $\sum_{i=0}^{2^N-1} |c_i|^2 = 1$. Superposition is the property permitting to quantum bits of being more powerful than their classical equivalents, since a higher degree of computing parallelism can be exploited by employing qubits, given the same number N of information units. However, qubit's superposition is suppressed when the qubit value is measured; for this reason, measurement must be executed only at the end of the sequence of operations required by a certain quantum algorithm.
Any state of the qubit can be represented as a unit vector in the Cartesian space connecting the origin and a point on the surface of the Bloch sphere (Figure 2.1).

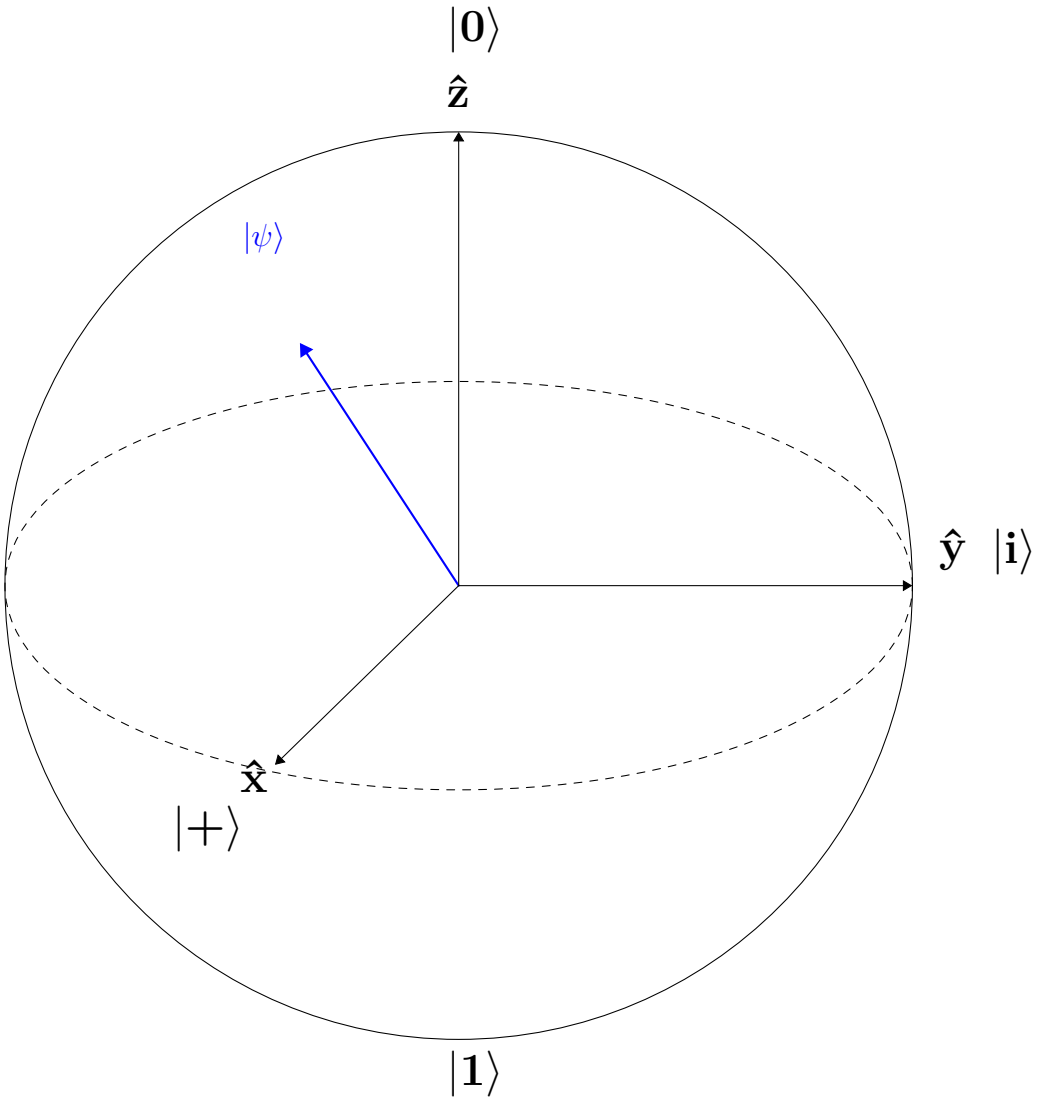


Figure 2.1: Bloch sphere in the Cartesian space. The north and south poles correspond to basis states $|0\rangle$ and $|1\rangle$, while $|\pm\rangle = \frac{|0\rangle \pm |1\rangle}{\sqrt{2}}$.

The state vector can be written as $|\psi\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\phi}\sin\left(\frac{\theta}{2}\right)|1\rangle$, where $0 \leq \theta \leq \pi$ is the co-latitude angle between $|\psi\rangle$ and $\hat{z}$ axis, while $0 \leq \phi \leq 2\pi$ is the azimuth angle between the projection of $|\psi\rangle$ on the xy-plane and $\hat{x}$ axis. The two basis states $|0\rangle$ ($\theta = 0$) and $|1\rangle$ ($\theta = \pi$), corresponding to the states 0 and 1 of a classical bit, lie at the poles of the sphere; in these cases the value of ϕ is irrelevant. **Points on the same latitude have the same probabilities** $|c_0|^2$ and $|c_1|^2$.
Operations on a qubit correspond to moving the state vector, thus eventually changing its probabilities of being $|0\rangle$ or $|1\rangle$, through **rotations of the Cartesian axes** described by matrices $R_x(\theta)$, $R_y(\theta)$, $R_z(\theta)$ [7]. This concept will be more clear when quantum gates of the so-called quantum circuit model will be presented in chapter 3.

2.2.3 The Schrödinger equation

The time evolution of a closed quantum system is described by the **Schrödinger equation**

$$i\hbar\frac{\partial}{\partial t}|\psi\rangle = \mathcal{H}(t)|\psi\rangle \tag{2.55}$$

where $\hbar$ is the **Planck's constant** divided by 2π and $\mathcal{H}$ is the Hamiltonian of the closed system. The general solution is

$$|\psi(t)\rangle = e^{-i\cdot\frac{1}{\hbar}\cdot\int_0^t \mathcal{H}(t')dt'}|\psi(0)\rangle \ . \tag{2.56}$$

Since $\mathcal{H}$ is Hermitian, the expression is equivalent to

$$|\psi(t)\rangle = U(t)|\psi(0)\rangle \ , \tag{2.57}$$

so the **evolution** of a quantum state can be described by a unitary operator U usually called **propagator**.
Equation 2.56 requires two operations: Hamiltonian time integration and exponentiation. In order to simplify the analysis, it is customary to define a reference frame where Hamiltonian does not have explicit time dependence (an example will be reported in subsection 3.3.3). This implies that Hamiltonian is constant during the time interval of integration and the evolution can be written as:

$$|\psi(t)\rangle = e^{-i\frac{\mathcal{H}}{\hbar}t}|\psi(0)\rangle \ , \tag{2.58}$$

and the only operation to be done is exponentiation. In the context of quantum circuits, quantum gates applied to single qubits are unitary. It has been assumed that the Hamiltonian is time-independent, but complex systems are controlled by varying the Hamiltonian, so the previous assumption is not valid anymore. In many cases, however, the Hamiltonian is piecewise constant in time domain, so the evolution can be described using a series of N propagators

$$|\psi\rangle = \left(\prod_{k=N}^{1} U_i\right)|\psi\rangle \tag{2.59}$$

with $U_i = \mathrm{e}^{-\mathrm{i}\frac{\mathcal{H}}{\hbar}t_i}$. While the sequence of Hamiltonians (not reported in Equation 2.59) is written with time running from left to right, the sequence of propagators is written from right to left, *i.e.* the rightmost propagator is the first to be applied. When the Hamiltonian varies continuously with time, the piecewise constant model does not describe the quantum system accurately; in that case, it is possible to write down a formal solution but it is not generally a useful approach in terms of computation and simulation.
The fact that any propagator describing the evolution of a quantum system is unitary has significant consequences:

- every propagator has an inverse, so quantum evolution, with the exception of measurement, is reversible, *i.e.* the cascade of two reciprocally inverse transformations restores the initial state;
- unitary transformations are length preserving, thus ensuring their description as rotations of the vector describing the quantum state.

2.2.4 Measurement

Measurement is the act of carrying out an observation on a given physical system. In particular, it consists in a projection of its state vector onto one of the eigenvectors of its corresponding basis. An important feature, which differs measurement from unitary evolutions, is that it is an **irreversible** operation which suppress superposition.
From an algebraic point of view, it is possible to define a **measurement operator** M_m related to the probability of obtaining the outcome m in the state

$$p(m) = \left\langle \psi \middle| M_m^\dagger M_m \middle| \psi \right\rangle . \tag{2.60}$$

The state of the system after the measurement is

$$|m\rangle = \frac{M_m |\psi\rangle}{\sqrt{P(m)}} . \tag{2.61}$$

According to the completeness equation

$$\sum_m M_m^\dagger M_m = I , \tag{2.62}$$

thus implying that

$$\sum_m P(m) = \sum_m \left\langle \psi \middle| M_m^\dagger M_m \middle| \psi \right\rangle = \sum_m \langle \psi | \psi \rangle = 1 . \tag{2.63}$$

This means that the measurement results can only belong to the basis. This clearly also implies that non-orthogonal states cannot be reliably distinguished by measurement operators. Considering two non-orthogonal states $|\psi_1\rangle$ and $|\psi_2\rangle$ forming

a basis for a generic quantum state $|\psi\rangle = a\,|\psi_1\rangle + b\,|\psi_2\rangle$, since $|\psi_2\rangle$ can be decomposed into a non-zero component $b_\parallel$ parallel to $|\psi_1\rangle$ and a component $b_\perp$ orthogonal to $|\psi_1\rangle$

$$b\,|\psi_2\rangle = b_\parallel\,|\psi_1\rangle + b_\perp\,|\psi_{1_\perp}\rangle\,, \tag{2.64}$$

the quantum state can be written as

$$|\psi\rangle = \left(a + b_\parallel\right)|\psi_1\rangle + b_\perp\,|\psi_{1_\perp}\rangle\,, \tag{2.65}$$

with $|b|^2 = |b_\parallel|^2 + |b_\perp|^2$. The consequence is that measurement can provide $|\psi_1\rangle$ as result even when $|\psi\rangle = |\psi_2\rangle$ is prepared.
The previous description is quite general and it is not limited to two-level systems, such as qubits. Examples more strictly related to these are reported in the following. Given a qubit, the measurement of its state in the computational basis is defined by projection operators:

$$\begin{aligned} M_0 &= |0\rangle\langle 0| = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \\ M_1 &= |1\rangle\langle 1| = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}. \end{aligned} \tag{2.66}$$

For M_0, it is possible to prove that

$$P(0) = \left\langle \psi \,\middle|\, M_0^\dagger M_0 \,\middle|\, \psi \right\rangle = |c_0|^2, \tag{2.67}$$

$$\frac{M_0\,|\psi\rangle}{\sqrt{P(0)}} = \frac{c_0}{|c_0|}\,|0\rangle \simeq |0\rangle\,. \tag{2.68}$$

It must be noticed that $\frac{c_0}{|c_0|} = \mathrm{e}^{i\phi}$ presents a phase contribution that does not play any role. Analogous expressions can be obtained for $|1\rangle$.
Considering a generic two-qubit state

$$|\psi\rangle = a\,|00\rangle + b\,|01\rangle + c\,|10\rangle + d\,|11\rangle \tag{2.69}$$

the measurement operators acting on the left qubit are

$$\begin{aligned} M_0 &= |0\rangle\langle 0| \otimes \mathbb{I}\,, \\ M_1 &= |1\rangle\langle 1| \otimes \mathbb{I}\,, \end{aligned} \tag{2.70}$$

and the state after measurement can be

$$\frac{M_0}{\sqrt{P(0)}} = |0\rangle \otimes \left(\frac{a}{u}\,|0\rangle + \frac{b}{u}\,|1\rangle\right), \tag{2.71}$$

$$\frac{M_1}{\sqrt{P(1)}} = |1\rangle \otimes \left(\frac{c}{v}\,|0\rangle + \frac{d}{v}\,|1\rangle\right), \tag{2.72}$$

where $u = \sqrt{|a|^2 + |b|^2}$ and $v = \sqrt{|c|^2 + |d|^2}$ are the square root of the probabilities of measuring on the left qubit 0 or 1 respectively. This mechanism can clearly extended to multiple qubits. In particular, the observation of k qubits among the total n of a system yields 2^k possible outcomes m_i $(0 \leq i \leq 2^k - 1)$. When the result of the observation of the k qubits is m_i, the state after the observation is projected to the subspace $\mathcal{H}_{m_i}$ with dimension 2^{n-k}.
When qubits are entangled, as already shown in Figure 1.1b, the measurement of one qubit immediately affects the outcome of the measurement on the others. This can be now proved. Given the entangled state

$$|\psi\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}} ,$$

if the first qubit was measured to be 0(1), then the outcome of the measurement of the second qubit is definitely 0(1).

2.3 Additional concepts

In the following subsections, additional concepts of quantum mechanics, which can be considered extremely important for properly understanding quantum information, are described in the following. A formalism compatible with all concepts introduced until this point of the thesis is ensured to be employed.

2.3.1 Mixed states and density matrix

A pure state of a physical system associated with a complex Hilbert space $\mathbb{H}^N$ equipped with an orthonormal basis $|e_i\rangle_{0 \leq i < N}$ is described by a state vector $|\psi\rangle = \sum_i c_i |e_i\rangle$, whose evolution is ruled by the Schrödinger equation. However, many quantum physical systems cannot be described by single pure states, on the contrary they are described by **mixed states**. This means that a semi-classical probability distribution of pure states $|\psi_i\rangle$, each of them occurring with a probability p_i, is associated with the quantum state [25]. This formalism is employed for describing quantum systems interacting with the external environment, for instance, through decoherence and relaxation phenomena. In these cases, the most suitable approach is to describe the system by means of the **density matrix**

$$\rho = \sum_i p_i |\psi_i\rangle \langle\psi_i| \ . \tag{2.73}$$

The density matrix of a pure state is $\rho = |\psi\rangle \langle\psi|$. The elements on the main diagonal of ρ are, by definition, the probabilities of finding the system in a basis state. Hence its trace — *i.e.* the sum of the terms on the main diagonal — must

be equal to one. The time evolution of a system described by a density matrix ρ is given by the Liouville – von Neuman equation:

$$i\hbar\frac{\mathrm{d}\rho}{\mathrm{d}t} = [\mathcal{H}(t), \rho]\,, \tag{2.74}$$

where $\mathcal{H}(t)$ is the Hamiltonian of the system with explicit time dependence and $[x, y] = xy - yx$ is the commutator [7]. As it is routinely done for pure states, also the time evolution of mixed states is customarily represented by employing the unitary time evolution operator $U(t)$:

$$\rho(t) = U(t)\rho(t=0)U(t)^\dagger\,, \tag{2.75}$$

where the time evolution operator $U(t)$ is a formal solution of the **Liouville – von Neumann equation**

$$U(t) = \exp\left(-\frac{i}{\hbar}\int_0^t \mathcal{H}(\tau)\mathrm{d}\tau\right)\,. \tag{2.76}$$

If the Hamiltonian can be regarded as not time-dependent, then this operator reduces to

$$U(t) = \exp\left(-i\frac{\mathcal{H}t}{\hbar}\right)\,. \tag{2.77}$$

Equation 2.74 is valid when the density matrix is assumed to be not affected by non-Hermitian operators called Lindblad operators. When these are present, a generalized equation, called **Lindblad equation**, must be solved: an extension of Equation Equation 2.74 involving a non-unitary contribution to the qubit's evolution [7].

$$\frac{\mathrm{d}\rho}{\mathrm{d}t} = -\frac{i}{\hbar}[\mathcal{H}(t), \rho] + \sum_j [2L_j\rho L_j^\dagger - \{L_j^\dagger L_j, \rho\}]\,, \tag{2.78}$$

where L_j is a Lindblad operator and $\{x, y\} = xy + yx$ is the anticommutator.

2.3.2 Distance metrics

The evolution of the actual physical device will be in general different from the ideal evolution, because of noise, and, thus, the final outcomes will not be identical. Hence, some metrics are usually introduced to determine how close the actual and ideal outputs are.

Fidelity is a scalar quantity that permits to determine the distance between two quantum states. Here the concept of fidelity as introduced in [7] is adopted. The fidelity between two states represented by density matrices ρ_A and ρ_B is:

$$\mathcal{F}(\rho_A, \rho_B) \triangleq \mathrm{Tr}\left(\sqrt{\rho_A^{1/2}\rho_B\,\rho_A^{1/2}}\right)\,, \tag{2.79}$$

where Tr is the matrix trace. If one of the two states is described by a state vector $|\psi\rangle$ and the other by a density matrix ρ, the fidelity can be rewritten as [7]

$$\mathcal{F}(|\psi\rangle, \rho) = \sqrt{\langle\psi| \rho |\psi\rangle}\,. \tag{2.80}$$

When only pure states, described by state vectors, the fidelity can be written as

$$\mathcal{F}(|\psi\rangle_A, |\psi\rangle_B) = \sqrt{\langle\psi|_A \rho_B |\psi\rangle_A} = \sqrt{\langle\psi|_A |\psi\rangle_B \langle\psi|_B |\psi\rangle_A} = |\langle\psi_A|\psi_B\rangle|\,. \tag{2.81}$$

In all the considered cases, it can be shown that $0 \leq \mathcal{F} \leq 1$ and the two states coincide when $\mathcal{F} = 1$.
The **Kullback-Leibler (KL) divergence** [29] is a measure of the distance between a target discrete probability distribution P and a reference one Q. This can be particularly useful in simulations of quantum computers based on the circuit model. For instance, the former can come from experimental data and the latter from theoretical deductions. In other words, the main diagonals of two density matrices are sufficient for computing the KL divergence. For an n-qubit distribution the KL divergence is defined as

$$D_{\mathrm{KL}}(P \parallel Q) \triangleq \sum_{i=0}^{N-1} P(|e_i\rangle) \log_2 \left(\frac{P(|e_i\rangle)}{Q(|e_i\rangle)} \right), \tag{2.82}$$

where $N = 2^n$ and $|e_i\rangle_{0 \leq i < N}$ are the orthonormal eigenstates of the σ_z operator $(|0\cdots0\rangle, \cdots, |1\cdots1\rangle)$. Differently from fidelity, the KL-divergence does not have an upper bound, while the lower one (*i.e.*, $D_{\mathrm{KL}} = 0$) is achieved if and only if the two distributions coincide; therefore, the smaller $D_{\mathrm{KL}}(P \parallel Q)$ is, the closer P and Q are.

2.3.3 No-cloning theorem

According to the **no-cloning theorem**, it is impossible to clone an exact quantum state without destroying the original. It means that a unitary mimicking the *fanout* operation, does not exist. This can be proved *ad absurdum*: a unitary operator U, making a clone of two orthogonal states $|\phi\rangle$ and $|\varphi\rangle$ on another state initialized to $|0\rangle$, is assumed to be available:

$$\begin{aligned} U|\phi 0\rangle &= |\phi\phi\rangle\,, \\ U|\varphi 0\rangle &= |\varphi\varphi\rangle\,. \end{aligned} \tag{2.83}$$

If a quantum state with superposition of the two orthogonal states is taken into account, *e.g.* $|\psi\rangle = \frac{|\phi\rangle + |\varphi\rangle}{\sqrt{2}}$, the cloning transformation must provide for linearity:

$$U\frac{|\phi\rangle + |\varphi\rangle}{\sqrt{2}0} = \frac{1}{\sqrt{2}}(U|\phi 0\rangle + U|\varphi 0\rangle) = \frac{|\phi\phi\rangle + |\varphi\varphi\rangle}{\sqrt{2}}\,.$$

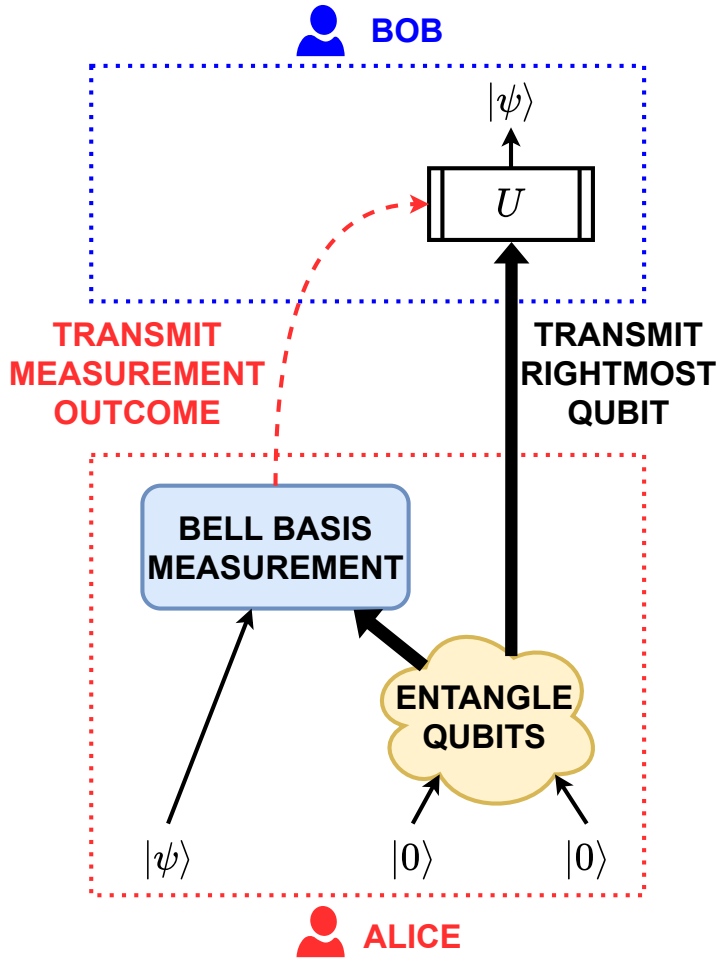


Figure 2.2: Quantum teleportation overview.

However, if U was a cloning transformation, the cloning transformation must also provide

$$U|\psi 0\rangle = |\psi\psi\rangle = \frac{1}{2}(|\phi\phi\rangle + |\phi\varphi\rangle + |\varphi\phi\rangle + |\varphi\varphi\rangle)$$

as final state, but this is different from the previous one. Therefore, the fanout behaviour is not satisfied when the unitary evolution is applied to a superposition state.

Transporting arbitrary quantum states from one system to another is possible, without contradicting the no-cloning theorem. The most famous example of this is **quantum teleportation**, that exploits the entanglement between two remote qubits to transmit an unknown quantum state. Protocol is schematized in Figure 2.2. Alice has initially three qubits $|\psi\rangle = a|0\rangle + b|1\rangle$, $|0\rangle$ and $|0\rangle$; she entangles the two qubits initialized to $|0\rangle$, thus having a state $|\psi\rangle \otimes \frac{|00\rangle+|11\rangle}{\sqrt{2}}$ and she transmits one of these qubits to Bob. Then she measures the two qubits in the so-called Bell

basis, having as eigenstates $\frac{|00\rangle+|11\rangle}{\sqrt{2}}$, $\frac{|00\rangle-|11\rangle}{\sqrt{2}}$, $\frac{|01\rangle+|10\rangle}{\sqrt{2}}$ and $\frac{|01\rangle-|10\rangle}{\sqrt{2}}$. It is possible to prove that the state immediately before measurement in the Bell basis is:

$$|\psi\rangle = \frac{1}{2}[|00\rangle (a|0\rangle + b|1\rangle) + |01\rangle (a|1\rangle + b|0\rangle) + |10\rangle (a|0\rangle - b|1\rangle) + |11\rangle (a|1\rangle - b|0\rangle)] \tag{2.84}$$

The Alice qubits have equal probability $\frac{1}{4}$ of being $|00\rangle, \cdots, |11\rangle$. When the measurement is done, the Bob qubit collapses to $a|0\rangle + b|1\rangle, \cdots, a|1\rangle - b|0\rangle$. All these states can become $|\psi\rangle$ with a unitary evolution U. For this reason, Alice transmits the measurement outcome to Bob, who applies the final U to obtain $|\psi\rangle = a|0\rangle + b|1\rangle$ on his received qubit.
For these reasons, in quantum information terms, the no-cloning theorem permits to do *cut-and-paste* of a state but not to *copy-and-paste* it.

2.3.4 Relaxation and decoherence

Relaxation and decoherence are two dynamic non-ideality phenomena common to all technologies for quantum computing, taking place because of the interactions between qubits, which are not perfectly isolated, and an external environment. The first one is related to the loss of energy of a qubit when it is in its excited state — thus forcing it to reach the ground state after a transient — while the other refers to the dynamic increase of uncertainty about the qubit phase. These phenomena are recognised as two of the main factors currently limiting the performance of real hardware.
The traditional mathematical description of these phenomena is based on solving the Lindblad equation (Equation 2.78). In some relevant quantum information cases (*e.g.* the emission of a two-level atom coupled to the vacuum [7]), **Kraus operators** E_k provide evolutions equivalent to those obtained by solving the Lindblad equation. They affect a density matrix ρ according to the following relation:

$$\mathcal{E}(\rho) = \sum_k E_k \rho E_k^\dagger. \tag{2.85}$$

For an n-qubit system, each operator is a square matrix of size $2^n \times 2^n$ and the total number of involved Kraus operators is at most 2^{2n} [7]. Relaxation of a single qubit can be described in terms of the amplitude damping operators

$$E_0 = \begin{bmatrix} 1 & 0 \\ 0 & \sqrt{1-\gamma} \end{bmatrix} \qquad E_1 = \begin{bmatrix} 0 & \sqrt{\gamma} \\ 0 & 0 \end{bmatrix}, \tag{2.86}$$

where γ is the probability of changing the qubit state from $|1\rangle\langle 1|$ to $|0\rangle\langle 0|$. E_0 and E_1 change the density matrix according to [7]

$$\mathcal{E}_{\text{ampl. damp.}}(\rho) = \begin{bmatrix} \rho_{0,0} + \gamma\rho_{1,1} & \sqrt{1-\gamma}\rho_{0,1} \\ \sqrt{1-\gamma}\rho_{1,0} & (1-\gamma)\rho_{1,1} \end{bmatrix}. \tag{2.87}$$

If amplitude damping operators are applied n consecutive times to the qubit, the probability of measuring $|1\rangle\langle 1|$ reduces as $(1-\gamma)^n\rho_{1,1} = \exp(n\log(1-\gamma))\rho_{1,1}$. If $\gamma = \Gamma\Delta t$ — where Γ is the qubit's decay rate — and $n = \frac{t}{\Delta t}$, a continuous-time expression of amplitude damping can be obtained:

$$\lim_{\Delta t\to 0} \mathrm{e}^{\frac{t}{\Delta t}\log(1-\Gamma\Delta t)}\rho_{1,1} \approx \mathrm{e}^{-\frac{\Gamma\Delta t}{\Delta t}t}\rho_{1,1} = \mathrm{e}^{-\frac{t}{T_1}}\rho_{1,1}, \tag{2.88}$$

where $T_1 = \Gamma^{-1}$ is the relaxation time constant. The same exponential scaling takes place in the enhancement of the probability of measuring $|0\rangle\langle 0|$ while the off-diagonal terms show an exponential scaling with time constant $2T_1$.
Decoherence can be described in terms of phase damping operators

$$E_0 = \begin{bmatrix} 1 & 0 \\ 0 & \sqrt{1-\lambda} \end{bmatrix} \qquad E_1 = \begin{bmatrix} 0 & 0 \\ 0 & \sqrt{\lambda} \end{bmatrix}, \tag{2.89}$$

where λ can be interpreted as the probability that the state of a qubit is randomly perturbed because of distance interactions (*e.g.* electron states with distant electrical charges). The effect of E_0 and E_1 on the density matrix [7] can be written as:

$$\mathcal{E}_{\text{phase damp.}}(\rho) = \begin{bmatrix} \rho_{0,0} & \sqrt{1-\lambda}\rho_{0,1} \\ \sqrt{1-\lambda}\rho_{1,0} & \rho_{1,1} \end{bmatrix}, \tag{2.90}$$

so the probability terms are not affected by this phenomenon. Similarly to amplitude damping, it is possible to define an exponential scaling of the off-diagonal terms in the continuous time limit:

$$\sqrt{1-\lambda}\rho_{0,1} \xrightarrow{\Delta t\to 0} \exp\left(-\frac{t}{T_2}\right)\rho_{0,1}, \tag{2.91}$$

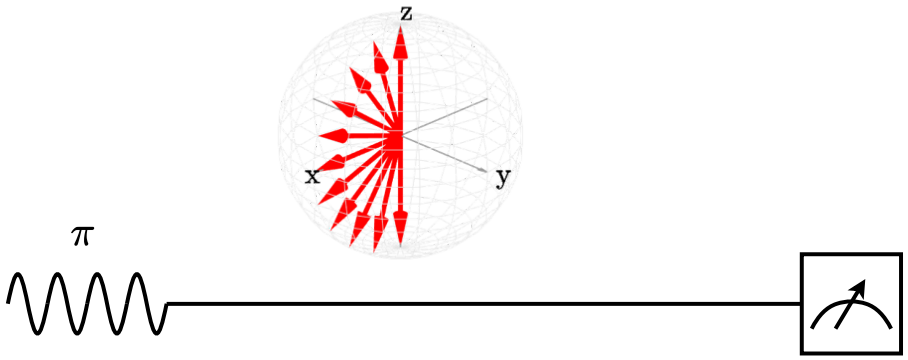
where T_2 is the decoeherence time constant. The simultaneous evaluation of the effects of amplitude damping and phase damping on the off-diagonal elements of the density matrix can be quantified in the definition of an effective time constant [20]:

$$T_{2_{\text{eff.}}} \triangleq \left(\frac{1}{2T_1} + \frac{1}{T_2}\right)^{-1}. \tag{2.92}$$

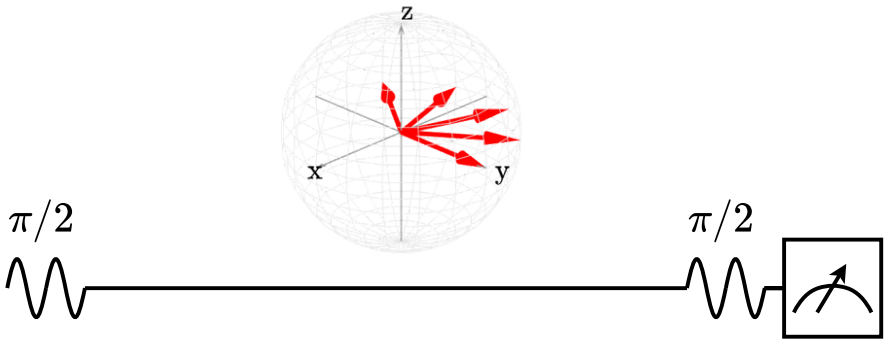
For what concerns the description of Kraus operators involving multiple qubits, some simplifying assumptions are routinely done. In particular, their construction is usually carried out by combining single-qubit operators with tensor products. This approach is supported in many simulators for quantum computing as the `QasmSimulator` in Qiskit [20]. Even though this formalism can reliably describe the dynamics of an arbitrarily large open quantum system, the classical simulation of the dynamics of a multi-qubit system is computationally demanding. In fact, the exponential scaling of both the dimensions and the total number of Kraus operators requires an exponential increase of the memory, for the simulation of noisy qubits. For this reason, there is an urgent need for an alternative description which can provide a good mimic of the evolution of noisy qubits under relaxation and decoherence, resorting to a reduced amount of hardware resources.

Experimental characterization of relaxation and decoherence

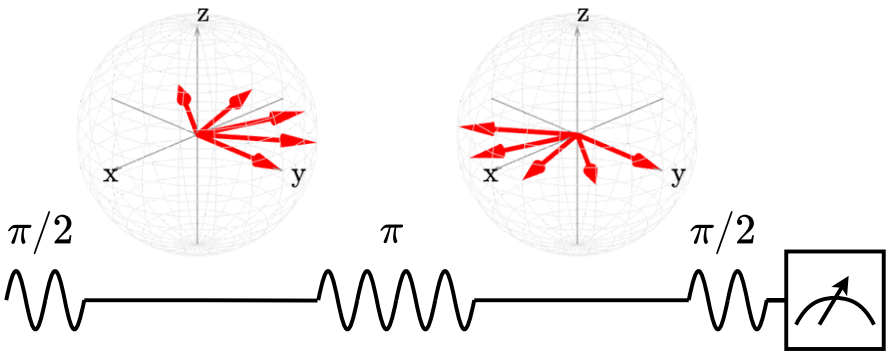
Considering the previous observations concerning the current capabilities of the simulator, **in the following of this thesis T_2 will correspond to T_2^*, *i.e.* the Ramsey fringes time constant**. Techniques typical of spin spectroscopy are employed for estimating the time constants T_1 and T_2. In both cases, repeated experiments involving measurements are involved to obtain a fitting curve associated with the time dependence of the probability of a qubit eigenstate [2], [30].
The experiment for obtaining the relaxation time constant T_1 simply consists in evaluating the time evolution of the probability of measuring $|1\rangle$, assuming that $\mathbb{P}(|1\rangle, t = 0) = 1$. As observable in Figure 2.3a, each iteration of the experiment requires to apply a π-pulse to a qubit initialized to $|0\rangle$, thus ensuring that the probability of measuring $|1\rangle$ is 1, then measurement is done after a delay τ, in which the reduction of $\mathbb{P}(|1\rangle)$ is expected. The measures $\mathbb{P}(|1\rangle, \tau)$ will be well-fitted by an exponential $e^{-\frac{\tau}{T_1}}$, where T_1 is exactly the relaxation time constant.
Calculation of T_2 can be done in two ways: **Ramsey fringes** and **Hahn echo**. In the Ramsey fringes experiment (Figure 2.3b), a qubit initialized to $|0\rangle$ is initially affected by a slightly off-resonance $\frac{\pi}{2}$-pulse on a given axis of the Cartesian space of the Bloch sphere, then a second $\frac{\pi}{2}$ pulse along the same axis, followed by a measurement operation, is applied after a time interval τ. If qubit was not affected by decoherence, the probability of measuring a qubit eigenstate will oscillate at the detuning frequency δ. The effect of decoherence consists in an exponential damping of the probability oscillations of type $\frac{1}{2}(1 + e^{-\frac{t}{T_2}})$, where T_2 is the decoherence time constant. For $t \to \infty$, the classical equal probabilities regime is achieved, where the two eigenstates $|0\rangle$ and $|1\rangle$ have the same probability $\frac{1}{2}$.
In addition to the procedure of Ramsey fringes, Hahn echo (Figure 2.3c) employs an intermediate *refocusing* π-pulse, which reverses low-frequency phase dynamics, thus permitting to enhance the degree of coherence and to obtain longer time constants than those obtained with Ramsey fringes.
In experimental scenarios, the decay time constants of Ramsey fringes and Hahn echo are sometimes called T_2^* and T_2, according to the typical nomenclature of Magnetic Resonance Imaging (MRI). The only decoherence mechanism properly describable by the proposed simulation infrastructure is that of Ramsey fringes. In fact, as it is possible to observe in Section 4.6.2, the same time constant is measured with the two experiments and it corresponds to T_2^*. This can be justified by the fact that the decoherence compact model is not sufficiently sophisticated to describe the partial compensation due to π-refocusing. However, this in some way less critical than expected, since it permits to execute tests and quantum circuits simulations in a worse-case decoherence scenario, considering that $T_2^* < T_2$.



(a) Relaxation.

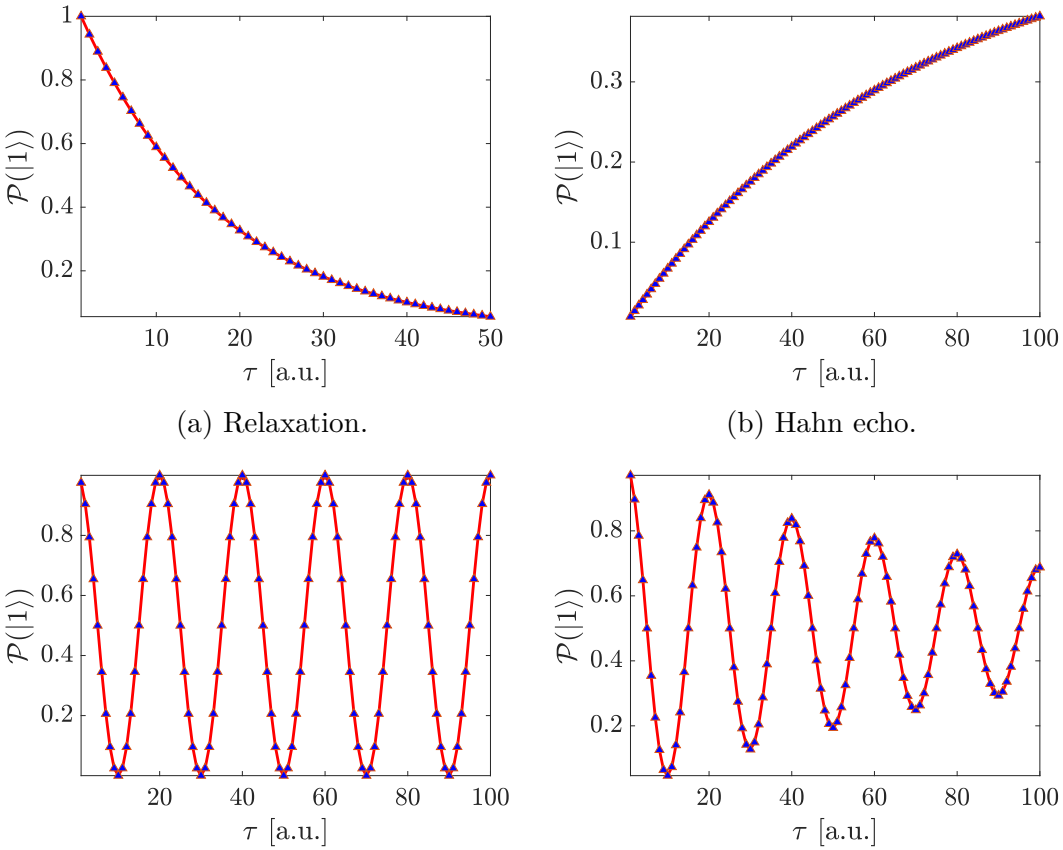


(b) Ramsey fringes.



(c) Hahn echo.

Figure 2.3: Visual demonstration of the experiments for qubit spectroscopy.



(a) Relaxation.

(b) Hahn echo.

(c) Ramsey fringes without decoherence.

(d) Ramsey fringes with decoherence.

Figure 2.4: Characteristic qubit transients, obtained with different experiments.

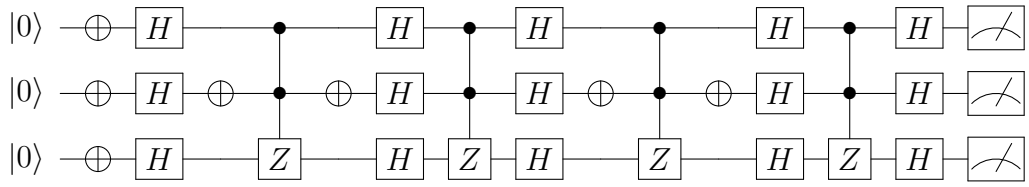
Part II

Quantum circuit model

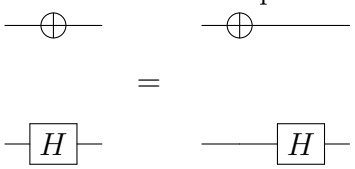
Chapter 3

Introduction to the quantum circuit model

The **quantum circuit**, or **quantum gate array**, **model** is the most consolidated approach for implementing general-purpose quantum computing. According to this paradigm, the evolution of a qubit system is described by a sequence of **quantum gates in the time domain**, each one characterized by an **unitary evolution**. Figure 3.1a shows the three-qubit quantum circuit for maximizing the probability of measuring the eigenstate $|101\rangle$ with Grover's search algorithm. Starting from an initial state — which is usually $|0\cdots0\rangle$, also corresponding to the quantum system ground state — qubits state evolve according to the applied single and multi-qubit gates, which change the probability amplitude, and usually also the measurement probability, of each eigenstate. At the end of the sequence of gates, measurement is done.



(a) Grover's search three-qubit circuit.



(b) Equivalence between parallel and sequential single-qubit gates.

Figure 3.1: Quantum circuits overview.

3.1 Quantum gates

Circuit sections showing a wire correspond to identity gates and single-qubit gates on different qubits are parallel; however, real quantum computers usually execute these operations sequentially, in order to minimize the crosstalk effects which can take place with a simultaneous execution of single-qubit gates. In fact, since each qubit is not sufficiently insulated from the others, it can be affected by the application of single-qubit gates on other qubits, so sequential execution minimizes the effects of this error source. The final state after a sequence of N gates can be written as

$$|\psi\rangle_{\text{final}} = \left(\prod_{k=1}^{N} U_{\text{gate},k} \right) |\psi\rangle_{\text{initial}} \tag{3.1}$$

Similarly to what happens in quantum computing, quantum gates can be:

- **technology-agnostic**, which are those presented in textbooks and are usually more practically useful for building a complex quantum circuit;
- **technology-characteristic**, which are the native unitary evolutions of a given technology, exploitable for building technology-agnostic gates and for optimizing a specific circuit on a given technology.

Some gates, as $R_{\{x,y,z\}}(\theta)$ can belong to both sets. Since this thesis, in the part concerning quantum circuit model, is mainly focused on compact modeling of hardware, wider space is provided to gates belonging to the second set. Consequently, only the most relevant technology-agnostic gates — and the methods for building them with the gates belonging to the other set — are presented. Two notations,



(a) Traditional notation (b) IBM notation

Figure 3.2: Comparison of the two quantum circuit notations.

shown in Figure 3.2 for generating the state $\frac{|10\rangle+|11\rangle}{\sqrt{2}}$ from $|00\rangle$, are nowadays employed in quantum computing community:

1. **traditional** (Figure 3.2a), reported in [7] and in the first publications on quantum circuit design, where the most significant qubit is the highest and the least significant one is the lowest;
2. **IBM** (Figure 3.2b), which is the reverse of the previous one, so the most significant qubit is the lowest.

Since the IBM notation is the most employed nowadays, because of the wide diffusion of the Qiskit software framework, in the following of this thesis this notation is employed for drawing quantum gates and circuits.

3.1.1 Single-qubit gates

In all introductory textbooks of quantum computing, the first described quantum gates for single qubits are the **three Pauli gates** X, Y and Z, the **Hadamard** gate H and the S and T gates.
The unitary evolutions of the first ones correspond to the Pauli matrices $\sigma_{x,y,z}$ descibed in subsection 2.1.8. Their effects on a generic single-qubit state vector are reported in the following:

$$\begin{aligned} X|\psi\rangle &= \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \end{bmatrix} = \begin{bmatrix} c_1 \\ c_0 \end{bmatrix}, \\ Z|\psi\rangle &= \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \end{bmatrix} = \begin{bmatrix} c_0 \\ -c_1 \end{bmatrix}, \\ Y|\psi\rangle &= \begin{bmatrix} 0 & -\mathrm{i} \\ \mathrm{i} & 0 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \end{bmatrix} = \begin{bmatrix} -\mathrm{i}c_1 \\ \mathrm{i}c_0 \end{bmatrix} = -\mathrm{i} \begin{bmatrix} c_1 \\ -c_0 \end{bmatrix}. \end{aligned} \tag{3.2}$$

The X gate swaps the probability amplitudes of the basis states $|0\rangle$ and $|1\rangle$. Since in the case of basis states $X|0\rangle = |1\rangle$ and $X|1\rangle = |0\rangle$, X gate substantially executes a **bit-flip** operation. The Z flips the sign of the probability amplitude of the basis state $|1\rangle$, but it does not change the measurement probabilities of the two basis states. This corresponds to a **phase-flip** operation, which moves the qubit state vector to its diametrically opposed position of the same parallel of the Bloch sphere.
As it is possible to ascertain in the last line of Equation 3.2, Y gate does a **bit-phase-flip** operation, *i.e.* both bit-flip and phase-flip operations.
The Hadamard gate is characterized by an unitary matrix:

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \tag{3.3}$$

When this gate is applied to the basis states $|0\rangle$ and $|1\rangle$:

$$\begin{aligned} H|0\rangle &= \frac{|0\rangle + |1\rangle}{\sqrt{2}}, \\ H|1\rangle &= \frac{|0\rangle - |1\rangle}{\sqrt{2}}, \end{aligned} \tag{3.4}$$

two superposition states, characterized by equal probabilities of measuring the two basis states, are employed. Hadamard gate is often employed at the beginning of quantum algorithms, to obtain an uniform superposition of states, each one encoding a datum to be processed with a quantum algorithm.
The S and T gates (and their Hermitian conjugates $S^\dagger$ and $T^\dagger$) are two diagonal

matrices, corresponding to $\sqrt{Z}$ and $\sqrt[4]{Z}$:

$$\begin{aligned} S\left|\psi\right\rangle &= \begin{bmatrix} 1 & 0 \\ 0 & \mathrm{i} \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \end{bmatrix} = \begin{bmatrix} c_0 \\ \mathrm{i}c_1 \end{bmatrix}, \\ S^\dagger\left|\psi\right\rangle &= \begin{bmatrix} 1 & 0 \\ 0 & -\mathrm{i} \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \end{bmatrix} = \begin{bmatrix} c_0 \\ -\mathrm{i}c_1 \end{bmatrix}, \\ T\left|\psi\right\rangle &= \begin{bmatrix} 1 & 0 \\ 0 & \mathrm{e}^{\mathrm{i}\frac{\pi}{4}} \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \end{bmatrix} = \begin{bmatrix} c_0 \\ \mathrm{e}^{\mathrm{i}\frac{\pi}{4}}c_1 \end{bmatrix}, \\ T^\dagger\left|\psi\right\rangle &= \begin{bmatrix} 1 & 0 \\ 0 & \mathrm{e}^{-\mathrm{i}\frac{\pi}{4}} \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \end{bmatrix} = \begin{bmatrix} c_0 \\ \mathrm{e}^{-\mathrm{i}\frac{\pi}{4}}c_1 \end{bmatrix}. \end{aligned} \tag{3.5}$$

In an experimental scenario, the gates previously described are all implemented in terms of the three **rotations gates** $R_{x,y,z}(\theta)$, introduced in subsection 2.2.2, which describe generic rotation of the state vector along the Cartesian axes:

$$\begin{aligned} R_x(\theta) &= \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) & -\mathrm{i}\sin\left(\frac{\theta}{2}\right) \\ -\mathrm{i}\sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{bmatrix}, \\ R_y(\theta) &= \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) & -\sin\left(\frac{\theta}{2}\right) \\ \sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{bmatrix}, \\ R_z(\theta) &= \begin{bmatrix} \mathrm{e}^{-\mathrm{i}\frac{\theta}{2}} & 0 \\ 0 & \mathrm{e}^{\mathrm{i}\frac{\theta}{2}} \end{bmatrix} = \mathrm{e}^{-\mathrm{i}\frac{\theta}{2}} \begin{bmatrix} 1 & 0 \\ 0 & \mathrm{e}^{\mathrm{i}\theta} \end{bmatrix} = \mathrm{e}^{-\mathrm{i}\frac{\theta}{2}} U_1(\theta), \\ R_{x,y,z}(\theta) &= \mathrm{e}^{-\mathrm{i}\frac{\theta}{2}(\sigma_{x,y,z})^{\frac{\theta}{\pi}}} \end{aligned} \tag{3.6}$$

In the single-qubit case, $R_z(\theta)$ is equivalent to the $U_1(\theta)$ gate, available in the native gates set of IBM quantum hardware [31] and also known as **phase shift** gate, since it adds a phase θ to the probability amplitude of $|1\rangle$. Apart from a negligible global phase contribution, the following equivalences between rotation gates and those previously introduced are valid:

$$\begin{gathered} X = R_x(\pi), \qquad Y = R_y(\pi), \qquad Z = R_z(\pi), \\ S = R_z\left(\frac{\pi}{2}\right), \qquad S^\dagger = R_z\left(-\frac{\pi}{2}\right), \qquad T = R_z\left(\frac{\pi}{4}\right), \qquad T^\dagger = R_z\left(-\frac{\pi}{4}\right). \end{gathered} \tag{3.7}$$

Hadamard gates can be obtained from rotation gates as:

$$R_x(\pi)R_y\left(\frac{\pi}{2}\right) = -\frac{\mathrm{i}}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = -\mathrm{i}H. \tag{3.8}$$

$R_x(\theta)$ and H gates permit to implement rotations $R_z(\theta)$ gates:

$$R_z(\theta) = HR_x(\theta)H = -R_x(\pi)R_y\left(\frac{\pi}{2}\right)R_x(\theta)R_x(\pi)R_y\left(\frac{\pi}{2}\right). \tag{3.9}$$

Previous examples put in evidence that the most important single-qubit gates can be all implemented in terms of $R_{x,y,z}(\theta)$ gates, but they are not the only ones implementable with $R_{x,y,z}(\theta)$. In fact, these gates can implement any single-qubit gate; a proof of the implementability of an universal set of quantum gates, with $R_{x,y,z}(\theta)$ as single-qubit gates, is reported in subsection 3.1.3. Moreover, their implementation at physical level, according to the magnetic resonance theory, is described in detail in section 3.3.3.
Before continuing with multi-qubit gates, an important observation must be done. Considering Equation 3.3 and Equation 3.4, it is possible to ascertain that each column of a unitary matrix corresponds to the evolution of a basis state. This concept is absolutely general for unitary matrices and is not limited to single-qubit gates.

3.1.2 Multi-qubit gates

In standard textbooks of quantum computation theory, quantum gates involving multiple qubits and implementing **controlled** operations are usually introduced. The main feature of these gates is that operations on a set of target qubits is done only if one or more control qubits are in a given state. The simplest controlled gates clearly involve one target and one control which asserts the operation when it is $|1\rangle$. The unitary matrix of a generic single-control-single-target controlled-U gate can be written in terms of an unitary matrix, to be applied on a two-qubit state $|\psi\rangle = |\chi\rangle_{\mathrm{c}} \otimes |\phi\rangle_{\mathrm{t}}$, as:

$$\mathrm{C}(U) = \begin{matrix} \\ |\phi\rangle_{\mathrm{t}} \\ |\phi\rangle_{\mathrm{t}} \end{matrix} \begin{matrix} \begin{matrix} |0\rangle_{\mathrm{c}} & |1\rangle_{\mathrm{c}} \end{matrix} \\ \begin{bmatrix} \mathbb{I} & 0 \\ 0 & U \end{bmatrix} \end{matrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & u_{00} & u_{01} \\ 0 & 0 & u_{10} & u_{11} \end{bmatrix}. \tag{3.10}$$

Equation 3.10 puts in evidence that the unitary matrix of a controlled gate corresponds to a block diagonal matrix, where the evolution of $|\phi\rangle_{\mathrm{t}}$ among $\mathbb{I}$ and U depends on the state of the control. In this section, three *canonical* two-qubit gates are described in the following: the controlled-NOT, the controlled-phase and the controlled-Z. In the most consolidated quantum computing technologies, the first gate is technology-agnostic, while the others can be technology-characteristic.

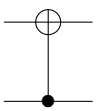


Figure 3.3: Circuit representation of the CNOT gate.

The **controlled-NOT**, well known as either CNOT or CX, whose schematic is available in Figure 3.3. It flips the target qubit when the control qubit is $|1\rangle$ and its unitary matrix is:

$$\text{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \tag{3.11}$$



Figure 3.4: Circuit representations of Cϕ and CZ gates.

From an algebraic point of view, the CNOT unitary matrix swaps the probability amplitudes of eigenstates $|10\rangle$ and $|11\rangle$. The **controlled-phase** (Cϕ) gate is nothing but the controlled version of the phase gate $U_1(\theta)$, providing an additional phase term to $|11\rangle$, while the **controlled-Z** (CZ) gate is a particular case of the previous one with $\phi = \pi$:

$$\begin{aligned} \text{C}\phi = \text{C}U_1(\phi) &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \mathrm{e}^{\mathrm{i}\phi} \end{bmatrix}, \\ \text{C}Z &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}. \end{aligned} \tag{3.12}$$

It is important to precise that the C$R_z(\phi)$ gate does not correspond to the Cϕ gate. In fact, its unitary evolution is:

$$\text{C}R_z(\phi) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \mathrm{e}^{-\mathrm{i}\frac{\phi}{2}} & 0 \\ 0 & 0 & 0 & \mathrm{e}^{\mathrm{i}\frac{\phi}{2}} \end{bmatrix}, \tag{3.13}$$

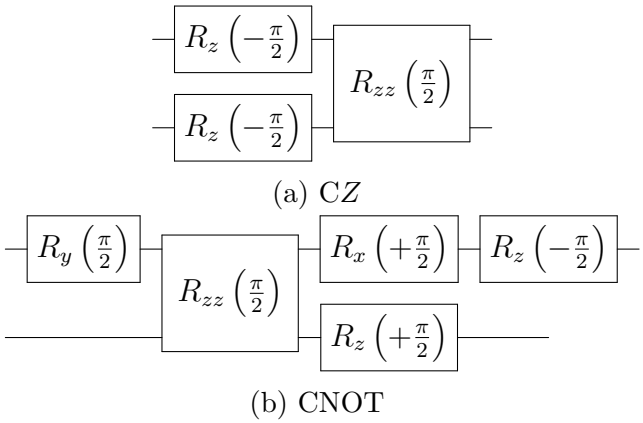
and it is possible to prove that the identity relation between C$R_z(\phi)$ and C$\phi^\dagger$ is not valid:

$$\text{C}R_z(\phi)\text{C}Z^\dagger = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \mathrm{e}^{-\mathrm{i}\frac{\phi}{2}} & 0 \\ 0 & 0 & 0 & \mathrm{e}^{\mathrm{i}\frac{\phi}{2}} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \mathrm{e}^{-\mathrm{i}\frac{\phi}{2}} & 0 \\ 0 & 0 & 0 & \mathrm{e}^{\mathrm{i}\left(\frac{\phi}{2}+\pi\right)} \end{bmatrix}. \tag{3.14}$$

In other words, the phase contribution $e^{-i\frac{\phi}{2}}$ (see Equation 3.6) is not global as in the single-qubit case. The previous proof can be extended to all other pairs involved a controlled-rotation gate and a controlled-Pauli gate.
Cϕ and CZ gates can be directly implemented in some technologies, characterized by the presence of two-level systems, which do not belong to the computational basis and whose resonance frequency depends on other quantum sub-systems belonging to the computational basis. This is the case of the so-called switch-based molecules discussed in section 5.1.
Another two-qubit gate to be mentioned is the $R_{zz}(\theta)$, which is characteristic of some spin-based technologies. Its unitary matrix is:

$$R_{zz}(\theta) = e^{-i\frac{\theta}{2}\sigma_z\otimes\sigma_z} = \begin{bmatrix} e^{-i\frac{\theta}{2}} & 0 & 0 & 0 \\ 0 & e^{i\frac{\theta}{2}} & 0 & 0 \\ 0 & 0 & e^{i\frac{\theta}{2}} & 0 \\ 0 & 0 & 0 & e^{-i\frac{\theta}{2}} \end{bmatrix} = e^{-i\frac{\theta}{2}} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & e^{i\theta} & 0 & 0 \\ 0 & 0 & e^{i\theta} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{3.15}$$

and its combination with single-qubit gates is sufficient for implementing CNOT and CZ gates, as it is possible to ascertain in Figure 3.5. $R_{zz}(\theta)$ is the corresponding



(a) CZ

(b) CNOT

Figure 3.5: Construction of CNOT and CZ gates with $R_{zz}(\theta)$.

unitary evolution of the Hamiltonian, typical of spin-spin longitudinal interactions:

$$\begin{aligned} \mathcal{H} &= \hbar\frac{J}{4}(\sigma_z \otimes \sigma_z)\,, \\ U &= e^{-\frac{i}{\hbar}\mathcal{H}t} = e^{-i\frac{J}{4}(\sigma_z\otimes\sigma_z)t} = R_{zz}\left(\frac{Jt}{2}\right). \end{aligned} \tag{3.16}$$

The parameter J has angular frequency dimensions it can be associated with the interaction strength. It can be negative, thus implying that CZ and CNOT gates can be implemented by changing the sign of all gates reported in Figure 3.5.
When two qubits (labelled as k and $l > k$) are not adjacent, the unitary matrix

associated with a generic two-qubit gate involving them $U_{\text{two-qubit}_{k.l}}$ can be obtained as:

$$\begin{aligned} U_{\text{two-qubit}_{k.l}} &= U_{\text{close}}^{\dagger} U_{\text{two-qubit}_{l-1,l}} U_{\text{close}} \\ U_{\text{close}} &= \prod_{i=k}^{l-2} U_{\text{SWAP}_{k,k+1}} \\ U_{\text{SWAP}} &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \end{aligned} \tag{3.17}$$

where U_{SWAP} is the unitary matrix of a two-qubit SWAP gate (swapping two qubits or equivalently changing the probability amplitudes associated with $|01\rangle$ and $|10\rangle$) and and $U_{\text{close}} = \prod_{i=k}^{l-2} U_{\text{SWAP}_{k,k+1}}$ is a unitary evolution which permits to get the two qubits adjacent through repeated swaps.

3.1.3 Universal set of quantum gates

An universal set of quantum gates is characterized by the fact that any unitary operation can be obtained as a finite sequence of gates belonging to this set. The most consolidated set with this characteristic is the so-called **Clifford+T** [7] made of H, S, T and CNOT gates. In this section, it is proved that the combination of $R_x(\theta)$, $R_y(\theta)$, $R_z(\theta)$ and the Cϕ gates, which are characteristic of all the quantum technologies described in this thesis, is sufficient for implementing the Clifford+T set, except for a negligible phase contribution.
Implementations of Hadamard, S and T gates, based on rotation gates have been already presented in Equation 3.8 and Equation 3.7. If in a given technology $R_z(\theta)$ could be not implemented in a reliable way, it can be built by combining $R_x(\theta)$ and $R_y(\theta)$ gates according to Equation 3.9.

The only missing equivalence is that implementing the CNOT gate involves a

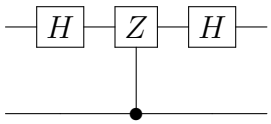


Figure 3.6: CNOT gate constructed with Hadamard and CZ gates.

control qubit and a target qubit. This gate can be implemented on the device of interest by cascading a Hadamard gate on the target qubit, a CZ gate (*i.e.* a Cϕ, with $\phi = \pi$) and another Hadamard gate on the target qubit, as shown in

Figure 3.6. An algebraic proof is reported in the following

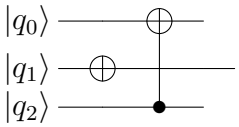
$$(\mathbb{I}\otimes H)CZ(\mathbb{I}\otimes H) = \frac{1}{\sqrt{2}}\begin{bmatrix}1&1&0&0\\1&-1&0&0\\0&0&1&1\\0&0&1&-1\end{bmatrix}\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&-1\end{bmatrix}\frac{1}{\sqrt{2}}\begin{bmatrix}1&1&0&0\\1&-1&0&0\\0&0&1&1\\0&0&1&-1\end{bmatrix}$$
$$= \frac{1}{2}\begin{bmatrix}1&1&0&0\\1&-1&0&0\\0&0&1&1\\0&0&1&-1\end{bmatrix}\begin{bmatrix}1&1&0&0\\1&-1&0&0\\0&0&1&1\\0&0&-1&1\end{bmatrix} = \begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&1&0\end{bmatrix} \tag{3.18}$$

3.2 Formalizing quantum circuits description

Differently from digital circuits, there is not any standard for describing quantum circuit currently. In fact, each quantum computing framework exploits its own commands for describing circuits. However, **OpenQASM 2.0** [32], proposed by IBM in 2017, could be considered a forerunner for a standardization of quantum circuits design. It was born as an interface language to automatize the construction of the first IBM quantum computers runnable *via*-cloud, then it had a such huge spread in quantum computing community that it became a sort of *de-facto* standard for the description of quantum circuits to be run on both real quantum hardware reale or classical simulators.

OpenQASM 2.0 syntax is similar to that of Verilog Hardware Description Language. For example, each code line ends with `;` and comments are introduced with `//`.

OpenQASM 2.0 permits to describe quantum circuits in a structural way, *i.e.* describing single gates and their connections inside a circuit. For example, the following circuit:



(3.19)

Can be written in OpenQASM 2.0 native syntax as:

```
x q1;
cx q2, q0;
```

The first word of each line is the gate to be inserted, while the others are a list of qubits on which gate is applied. In the single-qubit-gate case, list contains only one element, while in those involving two qubits the first qubit is the control, the other is the target.

OpenQASM 2.0 also permits to encapsulate a quantum circuit in a re-utilizable custom gate:

```
gate my_gate q0, q1, q2 {
  x q1;
  cx q2, q0;
}
```

where `my_gate` is the name of the custom gate. The definitions of custom gates must be stored in a file with extension `.inc`, to be included in other circuit description files with the `include` command. In particular, after the `.inc` file inclusion, custom gate can be employed by calling its name and providing after it the list of qubits on which it is applied:

```
include "my_gate.inc";

gate my_second_gate q0, q1, q2, q3 {
  my_gate q0, q3, q1;
  rx(pi/2) q2;
  cu1(1.57) q0, q2;
}
```

Hierarchical circuit design can be obtained in this way, dividing the total complexity of the circuit in smaller building blocks. Top-level circuit must be inserted in a file with extension `.qasm`, which can be run on different backends. This file shows some syntactic peculiarities: apart eventual initial comment lines, the first line must refer to the OpenQASM version to be employed, so `OpenQASM 2.0` in this case. Multiple groups of qubits and classical bits, called quantum and classical registers respectively, can be defined with `qreg` and `creg` respectively. Results must be reported at the end of the algorithm with the command `measure`. A final example is reported in the following.

```
OPENQASM 2.0;

include "my_second_gate.inc";

qreg q[4];
creg c[4];

my_second_gate q[3], q[0], q[1], q[2];
rx(0.54) q[3];
cz q[1], q[0];

measure q[0] -> c[1];
measure q[1] -> c[3];
measure q[2] -> c[0];
measure q[3] -> c[2];
```

For all additional details about OpenQASM 2.0, it is possible to consult [32].
It is important to clarify that in the simulator described in this thesis, the involved OpenQASM 2.0 circuits are very small and not organized according to a modular structure. This is also due to the fact that quantum computers taken into account

involve a too low number of qubits, thus implying that a too sophisticated interpreter of modular quantum circuits is practically useless.
Many benchmark quantum circuits, employable for testing either a simulator or a quantum circuit optimizer, described in OpenQASM 2.0 are available online, especially on GitHub repositories [33, 34]. In this thesis, quantum circuits were mainly employed to evaluate the simulator capabilities in different scenarios, *i.e.* according to the compact model to be tested. All the benchmark circuits reported in this thesis were written *ex-novo* and their names are ruled according to the following notation:

- In quantum circuits implementing boolean functions (such as `fredkin_XXX` and `full_adder_XXXX`) and in `cnot_XX`, each `X` value corresponds to the initial state of the corresponding qubit. Initialization to 1 is done with a $R_x(\pi)$ gate.
- In `grover_XX`, `XX` corresponds to the eigenstate to be searched.
- In `swap_XX`, `XX` are the indices to be swapped.
- In `ghz_N`, creating entangled states of type $\frac{|000\cdots0\rangle+|111\cdots1\rangle}{\sqrt{2}}$, `N` corresponds to the total number of entangled qubits.

3.3 Technology overview

The proves of computational advantages of quantum computing in different applications scenarios forced the necessity of fabricating real quantum hardware. An overview of the technologies proposed for the implementation of quantum computers according to the circuit model are reported in the following.
"Gate Array" is the standard paradigm for general purpose Quantum Computing. It is based on the so-called quantum circuit model [7], according to which all quantum operations are implemented as sequences of quantum gates, each one associated with a particular unitary evolution of the quantum system.

3.3.1 DiVincenzo criteria

David DiVincenzo proposed in 2000 a set of fundamental rules to be satisfied by any potential technology for **general-purpose quantum computation** [35]:

1. "A scalable physical system with well characterized qubits". The qubits of the proposed technology must be well-defined (*e.g.* spins in a molecule) and their scaling in time must always be ensured in order to regularly fabricate more complex and powerful quantum computers.

2. "The ability to initialize the state of the qubits to a simple fiducial state, such as $|000..0\rangle$". According to the quantum circuit model, quantum algorithms always start from an initial basis state; the application of quantum gates permit to change the quantum state by creating superposition of different basis states. The initial state is typically $|000..0\rangle$, which usually corresponds to the lowest energy of the quantum system.

3. "Long relevant decoherence times, much longer than the gate operation time". Decoherence is the most critical non-ideality phenomenon for quantum computing systems. It is related to the interaction between the quantum processor and the external environment and its main effect is the unreliability of the results of a quantum computer. Assuming that an initialization procedure making superposition of quantum states takes place at a certain time instant $t = 0$, the effect of decoherence becomes gradually more significant for $t > 0$, so it is important to ensure that all the operations required by a quantum algorithm are completed in the lowest time interval.

4. "A universal set of quantum gates". According to the quantum circuit model, all quantum operations are implemented as sequences of quantum gates, each one associated with a particular unitary evolution of the quantum system. The combination of quantum gates belonging to a universal set as the Clifford + T [7] (CNOT, Hadamard, S and T) is sufficient for implementing all the possible unitary evolutions, thus quantum gates.

5. "A qubit-specific measurement capability", related to the necessity of having a measurement system ensuring negligible errors.

These criteria have become the *de-facto* rules for proving that a technology is exploitable for the fabrication of a quantum computer. Special-purpose architectures as those for Quantum Annealing do not meet all the rules at the same time.

3.3.2 Overview of technologies for quantum circuit model

Different technologies have been proposed for building a quantum computer. They can be compared in terms of:

- qubit encoding mechanism, which is usually naturally or artificially traced back to a two-level quantum system, such as the quantum-optical two-level atom [36] or the spin-$\frac{1}{2}$ [37];

- connectivity, *i.e.* the capability to directly entangle each possible pair of qubits belonging to the quantum computer;

- timescales for executing reliably a quantum algorithm.

In the following, some of these are introduced. Nowadays, the most consolidated approach for fabricating a qubit is exploiting superconducting devices [38]. This technology is employed by many manufacturers of quantum hardware, among which IBM, Google and Rigetti can be mentioned. In order to understand the mechanism behind superconducting qubits, it is important to precise that the typical quantities of electrical circuits (such as currents, voltages and magnetic fluxes) are subjected to quantization when circuits operate at very-low temperatures (in the order of tens of milli-Kelvin) [39], *i.e.* these physical quantities show discrete values, similarly to position and momentum in *textbook* quantum mechanics. A superconducting qubit is obtained through a non-linear LC circuit. Equivalently to the quantum mechanical harmonic oscillator, in an electrical harmonic oscillator (*i.e.* an LC circuit) the eigenstates of its characteristic quantities (*e.g.* the magnetic flux of the inductor, or equivalently the photon number state $|n\rangle$, typical of quantum optics) are separated by the same energy amount $\frac{\hbar}{\sqrt{LC}}$. The constant transition frequency is not useful for the fabrication of artificial atoms where two particular states are going to be addressed. Superconductivity helps to achieve this target. In fact, a Josephson junction — made of two superconductors separated by an insulating layer — behaves as a non-linear inductor, so if it replaces the inductor, it introduces anharmonicity in the LC circuit. Figure 3.7 shows the differences between the linear and non-linear harmonic oscillators. Anharmonicity provided by Josephson junctions is engineered to have a $|\Delta\omega_{|n\rangle,|n+1\rangle} - \Delta\omega_{|n+1\rangle,|n+2\rangle}|$ sufficiently high to ensure good addressability of the lowest-energy states $|0\rangle$ and $|1\rangle$. The most consolidated superconducting device for quantum computing is called transmon [40] and is characterized by employing a large shunt capacitance in parallel to the Josephson junction, thus ensuring that the energy associated with the non-linear inductance is much higher than the corresponding one of capacitive contributions. Transmon qubits are typically designed in the radio-frequency spectrum ($\Delta\omega_{|0\rangle,|1\rangle} \sim 5\,\text{GHz}$, with anharmonicity in the order of 300 MHz), cooled down in dilution refrigerators below 100 mK and addressed with conventional electronic instruments, *e.g.* frequency synthesizers and spectrum analyzers. The main *pro* of superconducting qubits is the fact that their scalability, in terms of both number of qubits and coherence time, is excellent and can be engineered, improving fabrication and manipulation techniques. The current main limitations of these devices are the partial connectivity of qubits and extremely low temperatures ($\lesssim 100\,\text{mK}$).

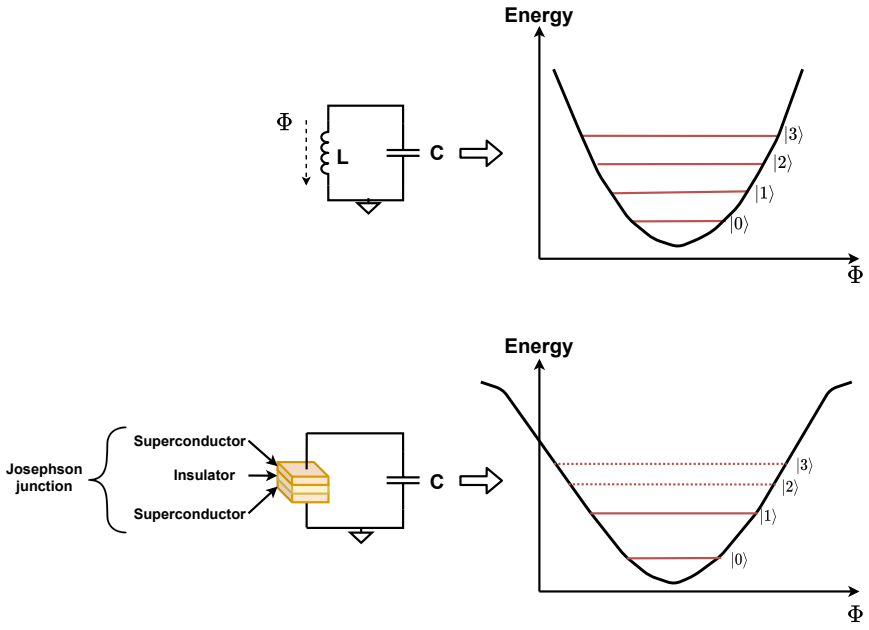


Figure 3.7: Quantized LC circuits. Anharmonicity given by Josephson Junction (lower image) ensures that $|\Delta\omega_{|n\rangle,|n+1\rangle} - \Delta\omega_{|n+1\rangle,|n+2\rangle}|$ is not constant anymore.

Trapped ions [41] is another consolidated technology for quantum computing. In this case, information is encoded onto ions blocked in an ion trap. As shown in Figure 3.8, this is formed by a set of electrodes which force the oscillation of ions around an equilibrium point, thus facilitating their localization. Moreover, these are put at a sufficiently high distance to ensure their single addressability with external radiation. Two types of qubit encodings are available in these systems (also shown in Figure 3.9):

- **Hyperfine**, where two states associated with the nuclear-electron hyperfine interaction are associated with $|0\rangle$ and $|1\rangle$. The two-level system is substantially the same of atomic clocks. The qubit resonance frequency is $\gtrsim 1\,\text{GHz}$, so a microwave control infrastructure is sufficient for manipulating them. Implementations based on $^{43}Ca^{+}$ [42], $^{171}Yb^{+}$ [43] and $^{133}Ba^{+}$ [44] ions are available in the state of art. The American manufacturer IonQ makes a hyperfine-qubit-based computer *via*-cloud.

- **Optical**, where qubits are encoded on two different electron orbitals. The qubit resonance frequency is $\gtrsim 100\,\text{THz}$, so lasers are required for mainpulating these qubit. A famous optical-qubit-based computer is that with $^{40}Ca^{+}$ ions [45], a version of which was built from Austrian manufacturer Alpine Quantum Technologies.

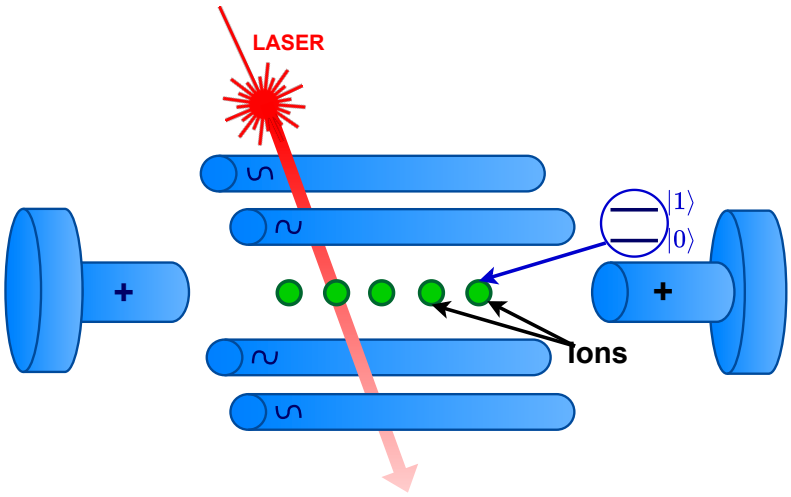
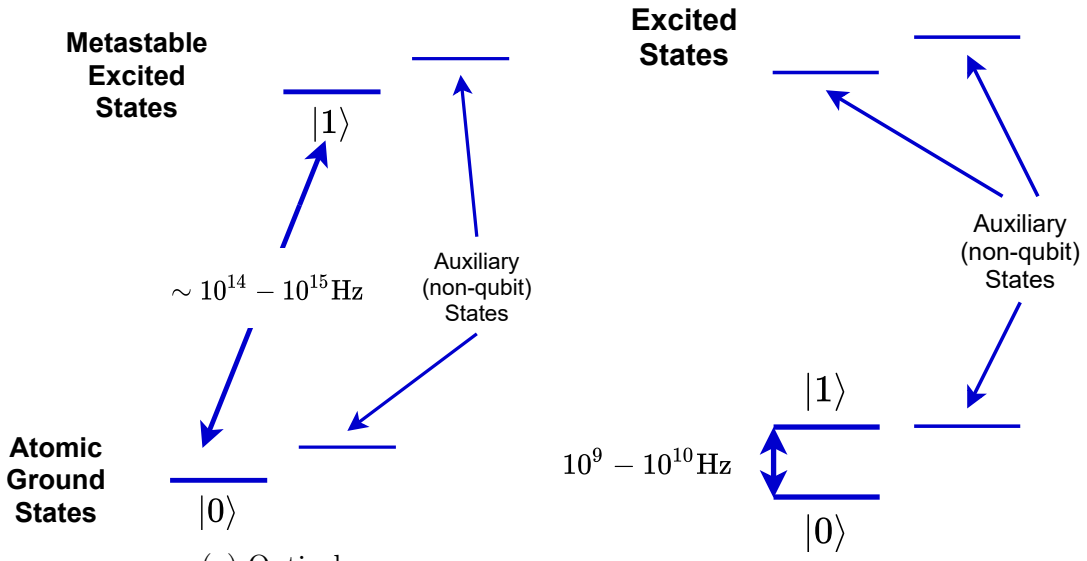


Figure 3.8: Overview of quantum computing with trapped ions.



(a) Optical.

(b) Hyperfine.

Figure 3.9: Qubit encodings in trapped ions.

The main pros of this technology are the full connectivity of qubits and long coherence times, which can overcome 1 s. The main limitations are the sophisticated control infrastructure for trapping and manipulating qubits. These quantum computers can operate at room temperature

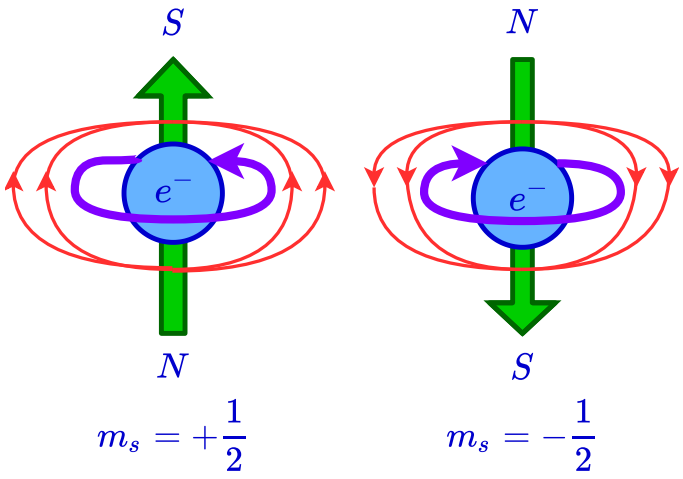


Figure 3.10: Electron spin-$\frac{1}{2}$ overview.

Finally, different solid-state devices, among which silicon quantum-dots [46] can be reminded, and molecules [47] can encode qubits onto **spins**, which are the quantum systems investigated in this thesis. Spin is a quantum mechanical form of angular momentum carried by atomic nuclei and electrons. In particular, it refers to the rotation of a particle around an axis. From a theoretical point of view it is described by a spin angular momentum vector and a **spin quantum number**. This practically means that spin is a discrete quantity, so only a finite number of angular momenta are admitted for a spin system. In the following of this thesis, only spin systems with quantum number equal to $\frac{1}{2}$ (*i.e.* with only two admitted angular momentum vectors) will be taken into account.
Since nuclei and electrons are electrically charged, the spin angular momentum implies a **magnetic moment**, in analogous way with respect to planar currents in classical electrodynamics. In the spin-$\frac{1}{2}$ case, spin behaves as a magnetic dipole with magnetic poles of equal magnitude but opposite polarity, so it behaves like a magnet. Figure 3.10 shows the two admitted spin states (with associated magnetic moments) of an electron. Moreover, the presence of an external magnetic field $B\mathbf{z}$ exerts a torque on spin, depending on its orientation with respect to the field, while the spin vector is subjected to a precession around the $\hat{z}$-axis. In this spin-$\frac{1}{2}$ case, the two spin states associated with the two admitted spin vectors will also have two different energies, so they can be exploited for encoding quantum information (see subsection 3.3.3 for further details). Describing the pros and cons of spin qubits is too generic, in sense that they depend on specific devices. While the qubits readout mechanism can be usually considered quite complex and erroneous, some common pros are the possibility to operate at room temperature, resonance frequencies compatible with microwave instrumentation and reasonably long coherence times. Table 3.1 reports a comparison between different quantum computing technologies.

Table 3.1: Comparison of some quantum computing technologies.

	Superconducting	Trapped ions	Spins (silicon)	Spins (molecules)
Connectivity	Partial	Full	Linear	Full (NMR) Linear (switch-based)
Resonance frequencies	Microwave	Optical (optical) Microwave (hyperfine)	Microwave	Microwave
Coherence timescales	$\gtrsim 100\,\mu s$	$\lesssim 1\,s$	$\gtrsim 6\,\mu s$	$\gtrsim 1\,s$ (NMR) $\gtrsim 1\,\mu s$ (switch-based)
Temperature	$\lesssim 100\,mK$	1 K to 100 K (typical)	$\lesssim 4\,K$	Room (NMR) $\gtrsim 10\,K$ (switch-based)

The terms *NMR* and *switch-based* will be clarified in chapter 5.

3.3.3 Magnetic resonance

Magnetic resonance is the technique employed for changing the state of a nuclear or electron spin. Its formalism is substantially at the basis of the description of coherent transitions in a two-level quantum system and consequently the implementation of single-qubit gates in most of the consolidated quantum computing technologies. The following discussion is based on a generalized formalism, which does not take into account the specific features of a given technology. It is important to precise that **all angular frequencies are assumed to be positive**.
Given a quantum system with two different levels, encoding qubit eigenstates $|0\rangle$ and $|1\rangle$ and characterized by energy difference $\Delta E = E_1 - E_0 = \hbar\omega_0$, its characteristic Hamiltonian can be written as:

$$\mathcal{H}_{\text{stationary}} = -\hbar\frac{\omega_0}{2}\sigma_z\,. \tag{3.20}$$

The eigenvalues of $|0\rangle$ and $|1\rangle$ respectively are respectively equal to

$$\begin{aligned} E_{|0\rangle} &= -\frac{\hbar\omega_0}{2}\,, \\ E_{|1\rangle} &= +\frac{\hbar\omega_0}{2}\,. \end{aligned} \tag{3.21}$$

The transition angular frequency[1] ω_0 is called, in the context of magnetic resonance, **Larmor frequency** and it is related to the angular velocity of the the magnetic moment exhibiting precession about the $\hat{z}$ axis, equivalent to the motion of the qubit on a parallel of the Bloch sphere. Precession does not change the eigenstates measurement probabilities $|c_0|^2$ and $|c_1|^2$ but it changes qubit phase in time.
The spin state can be modified by applying a radio frequency or microwave electromagnetic field along an axis in the xy-plane, forcing a spin precession around a vector in that plane. Assuming that the electro-magnetic field is applied on the x axis, the transverse field Hamiltonian contribution can be written as

$$\mathcal{H}_{\text{transverse}} = \hbar\omega_1 \cos(\omega_r t - \phi)\sigma_x, \tag{3.22}$$

where $\omega_1 = \gamma B_r$ is the precession frequency around the vector in the transverse plane, and that the total Hamiltonian is given by the sum of the two contributions:

$$\mathcal{H} = -\hbar\frac{\omega_0}{2}\sigma_z + \hbar\omega_1 \cos(\omega_r t - \phi)\sigma_x = \frac{\hbar}{2}\begin{bmatrix} -\omega_0 & 2\omega_1 \cos(\omega_r t - \phi) \\ 2\omega_1 \cos(\omega_r t - \phi) & \omega_0 \end{bmatrix}. \tag{3.23}$$

The explicit time-dependence is inconvenient for obtaining the evolution of the quantum system, since integration is computationally hard. The frame of reference is changed from the fixed laboratory frame to a frame rotating with angular frequency ω around the z-axis. In order to change the frame of reference, the unitary transformation [25]

$$U_R = \mathrm{e}^{-\mathrm{i}\frac{\omega_R t}{2}\sigma_z} = \begin{bmatrix} \mathrm{e}^{-\mathrm{i}\frac{\omega_R t}{2}} & 0 \\ 0 & \mathrm{e}^{+\mathrm{i}\frac{\omega_R t}{2}} \end{bmatrix} \tag{3.24}$$

is exploited, which permits given a state $|\psi\rangle$ in the laboratory frame, to write a rotating-frame state with the same eigenvalue probabilities as

$$|\tilde{\psi}\rangle = U_R\,|\psi\rangle\ . \tag{3.25}$$

The Schrödinger equation for the analyzed quantum system can be written in the rotating frame. Reminding that in the laboratory frame it can be written as:

$$\mathrm{i}\hbar\frac{\partial|\psi\rangle}{\partial t} = \mathcal{H}\,|\psi\rangle\ . \tag{3.26}$$

[1] In the following of this thesis, all Hamiltonian operators will involve angular frequencies (quantified in $\mathrm{rad\,s^{-1}}$). However, considering the well-known relation $\omega = 2\pi f$, all numerical values will be reported in frequency units, *i.e.* Hz or multiples, since they facilitate the understanding of the oscillations timescales. For example, the following expressions can be found: $\frac{\omega_0}{2\pi} = 1\,\mathrm{MHz}$ and $\omega_0 = 2\pi \cdot 1\,\mathrm{MHz}$

and observing that $|\psi\rangle = U_R^\dagger|\tilde{\psi}\rangle$, the equivalent formulation can be found:

$$\begin{aligned}
&\mathrm{i}\hbar\frac{\partial(U_R^\dagger|\tilde{\psi}\rangle)}{\partial t} = \mathcal{H}U_R^\dagger|\tilde{\psi}\rangle\,,\\
&\mathrm{i}\hbar\left(\frac{\partial U_R^\dagger}{\partial t}|\tilde{\psi}\rangle + U_R^\dagger\frac{\partial|\tilde{\psi}\rangle}{\partial t}\right) = \mathcal{H}U_R^\dagger|\tilde{\psi}\rangle\,,\\
&\mathrm{i}\hbar U_R^\dagger\frac{\partial|\tilde{\psi}\rangle}{\partial t} = \mathcal{H}U_R^\dagger|\tilde{\psi}\rangle - \mathrm{i}\hbar\frac{\partial U_R^\dagger}{\partial t}|\tilde{\psi}\rangle = \left(\mathcal{H}U_R^\dagger - \mathrm{i}\hbar\frac{\partial U_R^\dagger}{\partial t}\right)|\tilde{\psi}\rangle\,.
\end{aligned} \tag{3.27}$$

Multiplying each side of the equation by U_R on the left, Schrödinger equation in the rotating-frame can be obtained:

$$\begin{aligned}
\mathrm{i}\hbar\frac{\partial|\tilde{\psi}\rangle}{\partial t} &= \tilde{\mathcal{H}}\,|\psi\rangle\ ,\\
\tilde{\mathcal{H}} &= U_R\mathcal{H}U_R^\dagger - \mathrm{i}\hbar U_R\frac{\partial U_R^\dagger}{\partial t}\,,
\end{aligned} \tag{3.28}$$

where the first term of the Hamiltonian is associated with the transformation of $\mathcal{H}$ into a new vector basis, while the second one has the same role of the fictitious forces arising in classical mechanics when working in accelerating frames. The calculations of terms constituting $\tilde{\mathcal{H}}$ are reported in the following.

$$\begin{aligned}
U_R^\dagger &= \begin{bmatrix} \mathrm{e}^{\mathrm{i}\frac{\omega_R t}{2}} & 0\\ 0 & \mathrm{e}^{-\mathrm{i}\frac{\omega_R t}{2}}\end{bmatrix}\,,\\
\frac{\partial U_R^\dagger}{\partial t} &= \mathrm{i}\frac{\omega_R}{2}\begin{bmatrix} \mathrm{e}^{\mathrm{i}\frac{\omega_R t}{2}} & 0\\ 0 & -\mathrm{e}^{-\mathrm{i}\frac{\omega_R t}{2}}\end{bmatrix} = \mathrm{i}\frac{\omega_R}{2}U_R^\dagger\sigma_z\,,\\
-\mathrm{i}\hbar U_R\frac{\partial U_R^\dagger}{\partial t} &= (-\mathrm{i}\hbar)\cdot\left(\mathrm{i}\frac{\omega_R}{2}\right)(U_R U_R^\dagger)\sigma_z = \hbar\frac{\omega_R}{2}\sigma_z = \frac{\hbar}{2}\begin{bmatrix}\omega_R & 0\\ 0 & -\omega_R\end{bmatrix}\,,\\
U_R\mathcal{H}U_R^\dagger &= \frac{\hbar}{2}\begin{bmatrix} \mathrm{e}^{-\mathrm{i}\frac{\omega_R t}{2}} & 0\\ 0 & \mathrm{e}^{\mathrm{i}\frac{\omega_R t}{2}}\end{bmatrix}\begin{bmatrix} -\omega_0 & 2\omega_1\cos(\omega_r t-\phi)\\ 2\omega_1\cos(\omega_r t-\phi) & \omega_0\end{bmatrix}\begin{bmatrix} \mathrm{e}^{\mathrm{i}\frac{\omega_R t}{2}} & 0\\ 0 & \mathrm{e}^{-\mathrm{i}\frac{\omega_R t}{2}}\end{bmatrix} =\\
&= \frac{\hbar}{2}\begin{bmatrix} -\omega_0\mathrm{e}^{-\mathrm{i}\frac{\omega_R t}{2}} & 2\omega_1\cos(\omega_r t-\phi)\mathrm{e}^{-\mathrm{i}\frac{\omega_R t}{2}}\\ 2\omega_1\cos(\omega_r t-\phi)\mathrm{e}^{\mathrm{i}\frac{\omega_R t}{2}} & \omega_0\mathrm{e}^{\mathrm{i}\frac{\omega_R t}{2}}\end{bmatrix}\begin{bmatrix} \mathrm{e}^{\mathrm{i}\frac{\omega_R t}{2}} & 0\\ 0 & \mathrm{e}^{-\mathrm{i}\frac{\omega_R t}{2}}\end{bmatrix}\\
&= \frac{\hbar}{2}\begin{bmatrix} -\omega_0 & 2\omega_1\cos(\omega_r t-\phi)\mathrm{e}^{-\mathrm{i}\omega_R t}\\ 2\omega_1\cos(\omega_r t-\phi)\mathrm{e}^{\mathrm{i}\omega_R t} & \omega_0\end{bmatrix}\,.
\end{aligned} \tag{3.29}$$

Finally, the rotating-frame Hamiltonian can be computed, also exploiting the identity based on Euler's formula $2\omega_1 \cos(\omega_r t - \phi) = \omega_1 \left[e^{i\cdot(\omega_r t - \phi)} + e^{-i\cdot(\omega_r t - \phi)} \right]$:

$$\begin{aligned}
\tilde{\mathcal{H}} &= U_R \mathcal{H} U_R^\dagger - i\hbar U_R \frac{\partial U_R^\dagger}{\partial t} \\
&= \frac{\hbar}{2} \begin{bmatrix} \omega_R - \omega_0 & 2\omega_1 \cos(\omega_r t - \phi) e^{-i\omega_R t} \\ 2\omega_1 \cos(\omega_r t - \phi) e^{i\omega_R t} & \omega_0 - \omega_R \end{bmatrix} \\
&= \frac{\hbar}{2} \begin{bmatrix} \omega_R - \omega_0 & \omega_1 \left\{ e^{i\cdot[(\omega_r - \omega_R)t - \phi]} + e^{-i\cdot[(\omega_r + \omega_R)t - \phi]} \right\} \\ \omega_1 \left\{ e^{-i\cdot[(\omega_r - \omega_R)t - \phi]} + e^{i\cdot[(\omega_r + \omega_R)t - \phi]} \right\} & \omega_0 - \omega_R \end{bmatrix}.
\end{aligned} \tag{3.30}$$

The contributions oscillating at angular frequency $\omega_r + \omega_R$ can be averaged out on a time duration τ which ensures

$$(\omega_r + \omega_R)\tau \gg 2\pi \,. \tag{3.31}$$

This simplification is known as **Rotating Wave Approximation** (RWA) and it is often employed in quantum mechanics for simplifying time-dependent terms. An attempt to understand the reasons behind the negligibility of fast oscillating terms is reported in Appendix A. Thanks to the RWA and assuming that $\omega_R = \omega_r$, the Hamiltonian can be written as

$$\tilde{\mathcal{H}} = \frac{\hbar}{2} \begin{bmatrix} \delta & \omega_1 e^{-i\phi} \\ \omega_1 e^{i\phi} & -\delta \end{bmatrix} = \frac{\hbar}{2} \{ \delta \sigma_z + \omega_1 [\cos(\phi)\sigma_x + \sin(\phi)\sigma_y] \} = \frac{\hbar}{2} \mathbf{w} \cdot \vec{\sigma} \,, \tag{3.32}$$

where $\delta = \omega_r - \omega_0$ is the **detuning**, which is assumed to be not negligible through the RWA, $\mathbf{w}$ is the three-element row vector containing the coefficients of pauli matrices and $\vec{\sigma}$ is the column vector containing the Pauli operators. In this simple case of Hamiltonian (3.32), it is possible to write analytically the unitary evolution associated on a time duration τ, assuming to operate in the neighborhood of $\delta = 0$, to ensure the RWA previously done. The goal is achievable reminding the property of exponentiation of Pauli operators vector reported in Equation 2.50. First of all, $\mathbf{w}$ is normalized:

$$\tilde{\mathcal{H}} = \frac{\hbar}{2} = \frac{\hbar}{2} \frac{\mathbf{w}}{\|\mathbf{w}\|} \cdot \vec{\sigma} = \hbar \frac{\Omega}{2} \left[\frac{\delta}{\Omega} \sigma_z + \frac{\omega_1}{\Omega} \cos(\phi)\sigma_x + \frac{\omega_1}{\Omega} \sin(\phi)\sigma_y \right] , \tag{3.33}$$

in terms of a normalization coefficient $\Omega = \sqrt{\omega_1^2 \cos^2(\phi) + \omega_1^2 \sin^2(\phi) + \delta^2} = \sqrt{\omega_1^2 + \delta^2}$ called **Rabi frequency**. With this form of the Hamiltonian, Equation 2.50 can be

exploited:

$$\begin{aligned} U(t) = e^{-\mathrm{i}\frac{\Omega t}{2}\hat{\mathbf{n}}\cdot\vec{\sigma}} &= \cos\left(\frac{\Omega t}{2}\right)\mathbb{I} - \mathrm{i}\sin\left(\frac{\Omega t}{2}\right)\hat{\mathbf{n}}\cdot\vec{\sigma} \\ &= \cos\left(\frac{\Omega t}{2}\right)\mathbb{I} - \mathrm{i}\sin\left(\frac{\Omega t}{2}\right)\left[\frac{\delta}{\Omega}\sigma_z + \frac{\omega_1}{\Omega}\cos(\phi)\sigma_x + \frac{\omega_1}{\Omega}\sin(\phi)\sigma_y\right] \\ &= \begin{bmatrix} \cos\left(\frac{\Omega\tau}{2}\right) - \mathrm{i}\frac{\delta}{\Omega}\sin\left(\frac{\Omega\tau}{2}\right) & -\mathrm{i}\frac{\omega_1}{\Omega}\sin\left(\frac{\Omega\tau}{2}\right)[\cos(\phi) - \mathrm{i}\sin(\phi)] \\ -\mathrm{i}\frac{\omega_1}{\Omega}\sin\left(\frac{\Omega\tau}{2}\right)[\cos(\phi) + \mathrm{i}\sin(\phi)] & \cos\left(\frac{\Omega\tau}{2}\right) + \mathrm{i}\frac{\delta}{\Omega}\sin\left(\frac{\Omega\tau}{2}\right) \end{bmatrix} \\ &= \begin{bmatrix} \cos\left(\frac{\Omega\tau}{2}\right) - \mathrm{i}\frac{\delta}{\Omega}\sin\left(\frac{\Omega\tau}{2}\right) & -\mathrm{i}\frac{\omega_1}{\Omega}\sin\left(\frac{\Omega\tau}{2}\right)\mathrm{e}^{-\mathrm{i}\phi} \\ -\mathrm{i}\frac{\omega_1}{\Omega}\sin\left(\frac{\Omega\tau}{2}\right)\mathrm{e}^{\mathrm{i}\phi} & \cos\left(\frac{\Omega\tau}{2}\right) + \mathrm{i}\frac{\delta}{\Omega}\sin\left(\frac{\Omega\tau}{2}\right) \end{bmatrix} \end{aligned} \tag{3.34}$$

Equation 3.34 puts clearly in evidence that the Rabi frequency Ω corresponds to the oscillation frequency of the probabilities of measuring the qubit in one of the two eigenstates, or equivalently of the transition probability of one eigenstate to another, in terms of the terms on the anti-diagonal of U:

$$P_{|0\rangle\leftrightarrow|1\rangle}(t) = \left[\frac{\omega_1}{\Omega}\sin\left(\frac{\Omega t}{2}\right)\right]^2 . \tag{3.35}$$

Moreover, it influences the amplitude of this oscillation, which is higher for lower Ω. When the carrier signal is resonant with the transition — *i.e.* $\delta = 0$ and $\Omega = \omega_1$ — probabilities oscillation amplitude is equal to its maximum achievable value 1 and the unitary evolution matrix can be simplified

$$\begin{aligned} \tilde{U} &= \begin{bmatrix} \cos\left(\frac{\omega_1\tau}{2}\right) & -\mathrm{i}\sin\left(\frac{\omega_1\tau}{2}\right)\mathrm{e}^{-\mathrm{i}\phi} \\ -\mathrm{i}\sin\left(\frac{\omega_1\tau}{2}\right)\mathrm{e}^{\mathrm{i}\phi} & \cos\left(\frac{\omega_1\tau}{2}\right) \end{bmatrix} = \\ &= \cos\left(\frac{\omega_1\tau}{2}\right)\mathbb{I} - \mathrm{i}\sin\left(\frac{\omega_1\tau}{2}\right)[\cos(\phi)\sigma_x + \sin(\phi)\sigma_y]. \end{aligned} \tag{3.36}$$

The angle θ related to the amount of the qubit rotation due to the quantum gate is equal to

$$\theta = \omega_1\tau \tag{3.37}$$

so it is possible to vary the rotation angle by varying only ω_1 (*i.e.* the magnetic field intensity), the pulse duration τ or both. In experimental quantum computing τ is usually varied for making different θ values, since it is simpler to have a higher time resolution than an amplitude one.

Single-qubit gates

Rotation gates around the x and y axes $R_{\{x,y\}}(\theta)$ are implemented by pulses with initial phases 0 and $\frac{\pi}{2}$ respectively. Hadamard-like gates can be implemented by

exploiting either a single pulse with anti-resonant carrier frequency ($\delta = \omega_1$ and $\Omega\tau = \pi$) or a couple of resonant pulses

$$R_x(\pi) \cdot R_y\left(\frac{\pi}{2}\right) = -\frac{\mathrm{i}}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix} = -\mathrm{i}H. \tag{3.38}$$

$R_x(\theta)$ and H gates permit to implement rotations around the z axis $R_z(\theta)$ according to the relation

$$R_z(\theta) = HR_x(\theta)H, \tag{3.39}$$

so it would require five consecutive gates, a sequence which would be preferably avoided in order to minimize gate errors and effects of decoherence. Another implementation of $R_z(\theta)$ gates is the following:

$$R_z(\theta) = R_x\left(\frac{\pi}{2}\right)R_y(\theta)R_x\left(-\frac{\pi}{2}\right) \tag{3.40}$$

which slightly reduces the total number of required gates. Something more efficient can be achieved, observing that the effect on the state vector of $R_z(\theta)$ is the same of rotating the x and y axes by $-\theta$ (see Figure 3.11). This permits to keep the effect of $R_z(\theta)$ without applying any physical quantum gate [31], since changing the phases associated with $R_{x,y}(\theta)$ according to the following procedure will be sufficient:

$$x: \quad \phi_x \to \phi_x - \theta \qquad\qquad y: \quad \phi_y \to \frac{\pi}{2} + \phi_x - \theta \tag{3.41}$$

where ϕ_x and ϕ_y are the phases of the external oscillations for implementing $R_x(\theta)$ and $R_y(\theta)$ respectively. These implementation of $R_z(\theta)$ gate is called **virtual-Z** and it does not require any loss of time, thus implying that $R_z(\theta)$ gates are atomic. An important advantage provided by virtual-Z strategy is the implementation of Hadamard gate with only one pulse, since its unitary evolution can be also obtained as

$$H_{\text{virtual}} = R_z\left(\frac{\pi}{2}\right)R_x(\frac{\pi}{2})R_z\left(\frac{\pi}{2}\right). \tag{3.42}$$

If a control hardware, capable of executing generic $R_x(\theta)$ and $R_z(\theta)$ gates, is available, an arbitrary single-qubit rotation U_3 can be implemented in terms of three characteristic angles as:

$$U_3(\theta,\phi,\lambda) = R_z(\phi)R_x\left(-\frac{\pi}{2}\right)R_z(\theta)R_x\left(\frac{\pi}{2}\right)R_z(\lambda)\,. \tag{3.43}$$

According to Equation 3.43, two $\frac{\pi}{2}$-pulses are sufficient for implementing a generic single-qubit gate. It is clear that maximizing R_z gates is a precious resource for reducing the number of pulses to be applied on qubits and then the effects of dynamical non-idealities.

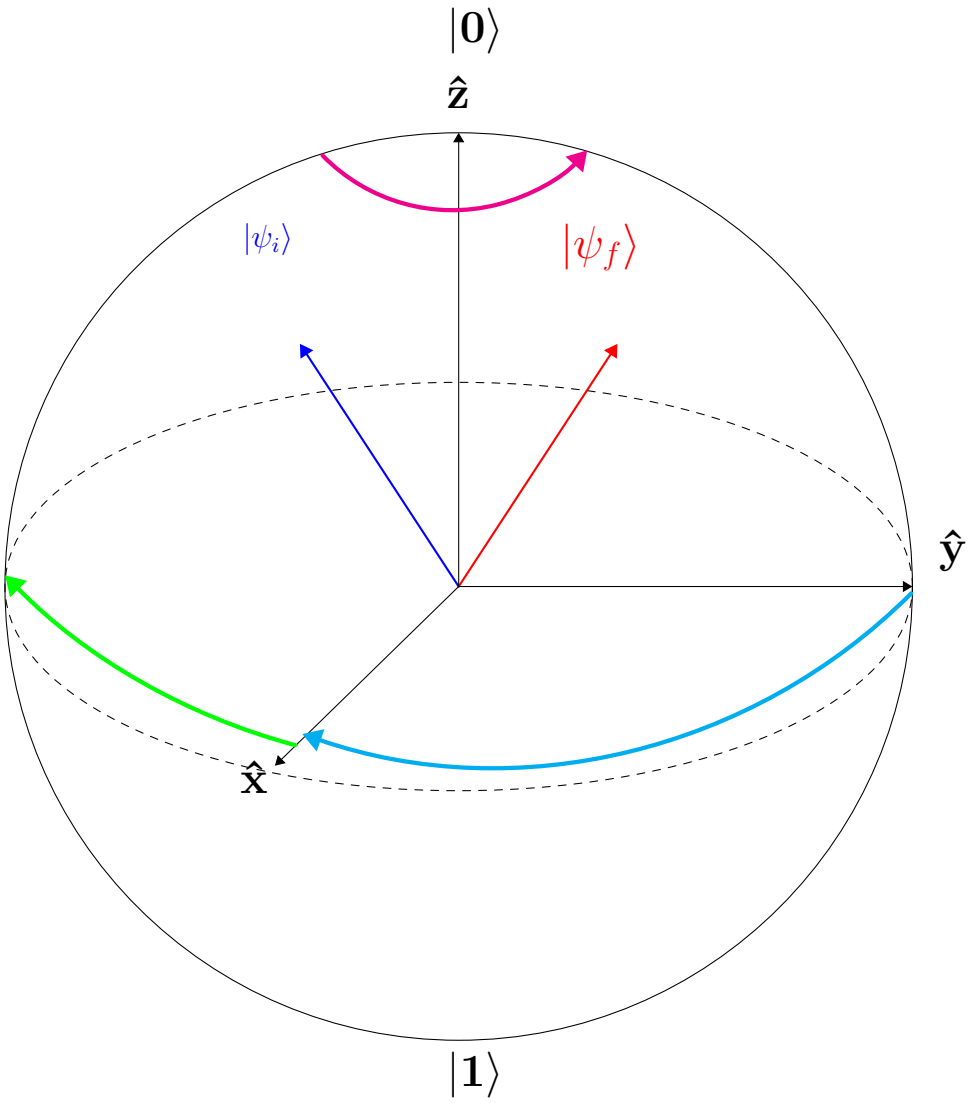


Figure 3.11: Representation of Virtual-Z principle on the Bloch sphere. The final state $|\phi_f\rangle = R_z\left(\frac{\pi}{2}\right)|\psi_i\rangle$ can be achieved by rotating the z-axis by $\frac{\pi}{2}$ or rotating the x and y axes by $-\frac{\pi}{2}$.

When the carrier signal is strongly anti-resonant with the respect to the Larmor frequency - *i.e.* $\delta \gg \omega_1$ - Hamiltonian 3.32 can be approximated as

$$\tilde{\mathcal{H}} \approx \frac{\hbar}{2}\delta\sigma_z \tag{3.44}$$

it means that in the rotating frame the qubit has approximately a precession motion around the z axis with angular frequency δ. This technique is not usually employed for implementing a $R_z(\theta)$ gate, but it is in any case relevant in experimental quantum computing. In fact, this mechanism is fundamental in **Ramsey spectroscopy** and it permits to study the qubits decoherence timescales, in particular to compute the time constant T_2.

The case of nuclear and electron spins

The fomalism of subsection 3.3.3 is usually employed for the description of the evolution of nuclear or electron spin-$\frac{1}{2}$ systems, where spin can assume two states, usually called spin-up $|\uparrow\rangle$ and spin-down $|\downarrow\rangle$ and associated with parallel and anti-parallel orientations with respect to an external polarization static field. The interaction of a spin-$\frac{1}{2}$ with a static field $B_0\mathbf{z}$ along the z-axis provides an energy difference between the two states and it is described by the Hamiltonian:

$$\mathcal{H} = -\hbar\frac{1}{2}\gamma B_0\sigma_z\,, \tag{3.45}$$

where σ_z is a Pauli matrix and γ is the gyromagnetic ratio, which is characteristic of each particle and it is defined as the ratio between its magnetic moment and its angular momentum, so its dimensions are given by the ratio between angular frequency and magnetic field. The experimental value of electron, proton and neutron gyromagnetic ratios, provided by the National Institute of Standards and Technology (NIST), are equal to [48, 49, 50]:

$$\begin{aligned}
\frac{\gamma_e}{2\pi} &= -28\,024.951\,424\,2(85)\,\mathrm{MHz\,T^{-1}}\,,\\
\frac{\gamma_p}{2\pi} &= +42.577\,478\,518(18)\,\mathrm{MHz\,T^{-1}}\,,\\
\frac{\gamma_n}{2\pi} &= +29.164\,693\,1(69)\,\mathrm{MHz\,T^{-1}}\,.
\end{aligned} \tag{3.46}$$

In this context, it is also important to remind another equation for electron gyromagnetic ratio, often employed in the electron spin qubit Hamiltonians:

$$\gamma_e = g_e\frac{\mu_B}{\hbar} \tag{3.47}$$

where g_e is the dimensionless g-factor of the electron and μ_B is Bohr's magneton. Their values, taken from [51, 52], are reported in the following:

$$\begin{aligned}
g_e &= -2.00231930436256(35)\,,\\
\frac{\mu_B}{h} &= 2\pi\frac{\mu_B}{\hbar} = 1.399\,624\,493\,61(42)\times 10^{10}\,\mathrm{Hz\,T^{-1}}\,.
\end{aligned} \tag{3.48}$$

Nuclear spin values of different elements are available at [53]. Some relevant examples are reported in the following:

$$\begin{aligned}\frac{\gamma(^1\mathrm{H})}{2\pi} &= 42.58\,\mathrm{MHz\,T^{-1}}\,,\\ \frac{\gamma(^{13}\mathrm{C})}{2\pi} &= 10.71\,\mathrm{MHz\,T^{-1}}\,,\\ \frac{\gamma(^{19}\mathrm{F})}{2\pi} &= 40.09\,\mathrm{MHz\,T^{-1}}\,,\end{aligned} \tag{3.49}$$

According to the previous observations, it is clear that $\gamma B_0 = \omega_0$ has angular frequency dimensions and ω_0 is nothing but the Larmor frequency describing the spin precession around the z-axis.
In the case of nuclear spins, where $\omega_0 > 0$, Hamiltonians in Equation 3.45 and Equation 3.20 are absolutely equivalent. $|0\rangle$ and $|1\rangle$ are encoded onto the lower-energy and higher-energy eigenstates spin-up $|\uparrow\rangle$ and spin-down $|\downarrow\rangle$ respectively. Nuclear spin manipulation can be done in terms of an oscillating magnetic field with amplitude $2B_r$, angular frequency ω_r and phase ϕ:

$$\begin{aligned}B(t) &= 2B_r\cos(\omega_r t-\phi)\,,\\ \mathcal{H}_{\text{transverse}} &= \frac{\hbar}{2}\gamma B(t)\sigma_x = \hbar 2\omega_1\cos(\omega_r t-\phi)\sigma_x\,,\\ \mathcal{H}_{\text{nuclear}} &= -\hbar\frac{1}{2}\gamma B_0\sigma_z + \frac{\hbar}{2}\gamma B(t)\sigma_x\\ &= \hbar\left[-\frac{\omega_0}{2}\sigma_z + 2\omega_1\cos(\omega_r t-\phi)\sigma_x\right]\,.\end{aligned} \tag{3.50}$$

The previous calculations put in evidence that the evaluation of the nuclear spin-$\frac{1}{2}$ evolution under an oscillating magnetic field can be done exactly with the magnetic resonance model of subsection 3.3.3, without any type of arrangement.
In the case of electron spin, some additional operations must be done. First of all, it must be observed that the electron spin Hamiltonian, exploiting the formulas in Equation 3.45 and Equation 3.50 are:

$$\begin{aligned}\mathcal{H}_{\text{electron}} &= -\hbar\frac{1}{2}\gamma B_0\sigma_z + \frac{\hbar}{2}\gamma B(t)\sigma_x\\ &= \hbar\left[+\frac{\omega_0}{2}\sigma_z - 2\omega_1\cos(\omega_r t-\phi)\sigma_x\right]\,.\end{aligned} \tag{3.51}$$

where the sign flips of the terms associated with ω_0 and ω_1 are due to $\gamma < 0$. In order to exploit the magnetic resonance model of subsection 3.3.3, it is possible to:

1. apply negative magnetic fields $B_{\{0,1\}} < 0$, compensating $\gamma_e < 0$;
2. do a preliminary frame change, keeping the magnetic fields positive.

While a negative-amplitude oscillating field can be easily obtained in an experimental scenario by generating, with a waveform generator, a cosine wave with a systematic π to the oscillation phase, it is not ensured that a negative magnetic static field can be applied to the electron spin. For this reason the second option is chosen, so that $B_0 > 0$. First of all, it is important to precise that the multiple application (even with different U_R) of Equation 3.28 to a given Hamiltonian is not forbidden. In this context, an unitary evolution U_R mapping $|\downarrow\rangle$ and $|\uparrow\rangle$ onto $|0\rangle$ and $|1\rangle$ respectively is required; this is $U_R = \sigma_x$, which is also stationary, thus implying that the second term of Equation 3.28 — characterized by a time derivative — is null. Electron spin Hamiltonian in the new frame is:

$$\begin{aligned}\tilde{\mathcal{H}}_{\text{electron}} &= U_R \mathcal{H}_{\text{electron}} U_R^\dagger = \sigma_x \mathcal{H}_{\text{electron}} \sigma_x = \\ &= \hbar \left[+\frac{\omega_0}{2} \sigma_x \sigma_z \sigma_x - 2\omega_1 \cos(\omega_r t - \phi) \sigma_x \sigma_x \sigma_x \right] \\ &= \hbar \left[-\frac{\omega_0}{2} \sigma_z - 2\omega_1 \cos(\omega_r t - \phi) \sigma_x \right] .\end{aligned} \tag{3.52}$$

$\tilde{\mathcal{H}}_{\text{electron}}$ has the same σ_z term of Hamiltonian (3.20), while the σ_x term has still the opposite sign. In order to do a sign flip, as already said, it is sufficient to change the phase of the oscillation, by adding a systematic π to the oscillation phase:

$$\begin{aligned}\tilde{\mathcal{H}}_{\text{electron}} &= \hbar \left[-\frac{\omega_0}{2} \sigma_z - 2\omega_1 \cos(\omega_r t - \phi + \pi) \sigma_x \right] \\ &= \hbar \left[-\frac{\omega_0}{2} \sigma_z + 2\omega_1 \cos(\omega_r t - \phi) \sigma_x \right] .\end{aligned} \tag{3.53}$$

The final Hamiltonian is equivalent to that in Equation 3.20, thus implying that the magnetic resonance model of subsection 3.3.3 can be applied straightforwardly.

Chapter 4

Compact-modeling-based simulation of quantum computing technologies

Considering the limitations of accessing to real quantum hardware introduced in section 1.3, the development of quantum computing simulators on classical computers could help to estimate and maximize the reliability of the execution a quantum algorithm on a real device, even on a number of qubits higher than those of free-access hardware, taking into account its physical degrees of freedom.
Currently available classical simulators of quantum computers mainly belong to two families:

1. Low-level physical simulators, which numerically solve Schrödinger and Lindblad equations [54]. Even though the accuracy of the description of the quantum physical system is undoubtedly high, this approach could be extremely memory and CPU intensive.

2. Fast high-level simulators [20, 55, 57, 56], which are developed to handle many qubits, thus permitting a classically evaluation on a quantum algorithm, even on a larger data-set than that manageable by a current real quantum computer. Non-ideality phenomena are usually neglected or described by consolidated mathematical models, as Kraus operators formalism for decoherence [7], thus minimizing the simulation latency and to maximise the number of qubits to be simulated.

The first part of this thesis concerns the activities done to find a compromise between the two families. In particular, it is oriented towards the development of a **scalable classical simulation infrastructure of quantum computers**, involving different technologies and intended to fulfil two objectives. The first one is the comparison of the results provided by different technologies for the same

application-driven quantum circuit. The second one is the analysis, for every device, of the performance dependence on both physical parameters and control degrees of freedom, thus enabling their optimization according to some fiducial parameters. Scalability implies that the simulator must execute a set of operations which must not be memory and CPU-intensive. In order to achieve this goal, a simulation approach based on *compact models* has been chosen. The main goal of this approach is describing the expected qubit dynamics without involving numerical resolution of Lindblad equation, thus permitting faster and more scalable simulations. It consists in defining a mathematical model of qubit dynamics — through fitting operations or *ad-hoc* mimic algorithms — which should be a reasonable compromise between low computational complexity and high physical accuracy. This means that the compact model, even if simpler than the standard one, must never neglect the physical characteristics of the quantum system to be modelled.

This thesis and its related articles [60, 59, 58] are the first documents where com-

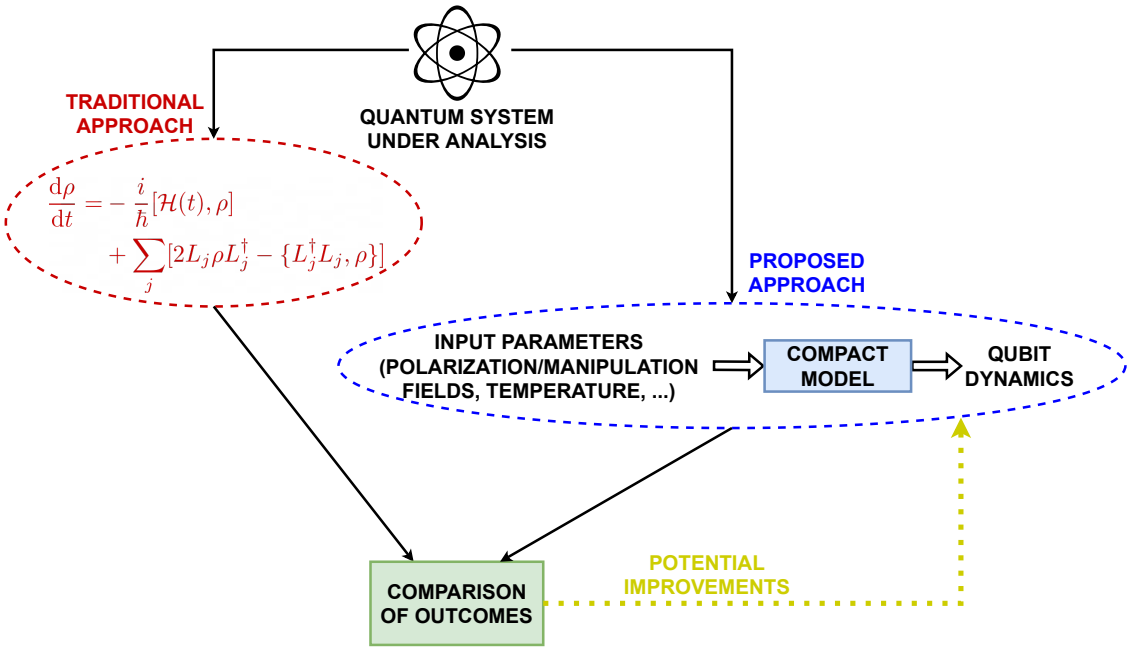


Figure 4.1: Methodology for analysis and validation of compact models.

pact modelling, already consolidated for the simulation of electron devices in Microelectronics Industry [61], is employed for executing scalable technology-dependent simulations of quantum circuits. In a classical scenario, compact models are usually declined into Simulation Program with Integrated Circuit Emphasis (SPICE) netlists or Verilog-A hardware descriptions containing the mathematical model associated with the target device. In the quantum computing context, these are formalized in terms of **MATLAB** scripts, integrated in a **compact-model-based**

simulator of quantum computing technologies belonging to the circuit model domain, whose development and validation constituted an important part of the doctorate activities described in this thesis. Details about the simulator and its organization are available in Appendix B.
Figure 4.1 shows the typical approach for defining, validating and improving a compact model. Starting from an initial Hamiltonian or phenomenological analysis of the quantum system, compact model is defined as a *black box* which provides the qubit response depending on a set of external parameters, which can be related to both polarization (*e.g.* magnetic static fields) and manipulation (*e.g.* electromagnetic waves for implementing quantum gates). The traditional description approaches are taken as reference for validating and improving the proposed one, through a comparison of the outcomes provided by quantum circuits belonging to a benchmark.
In this chapter, an overview of the approach for building compact models is shown and the overall description of the simulation methodology is provided. Moreover, the models of some features common to different quantum technologies, *i.e.* relaxation and decoherence dynamics and off-resonance mechanism, are described and the corresponding validation results are reported.

4.1 Overview of simulation infrastructure

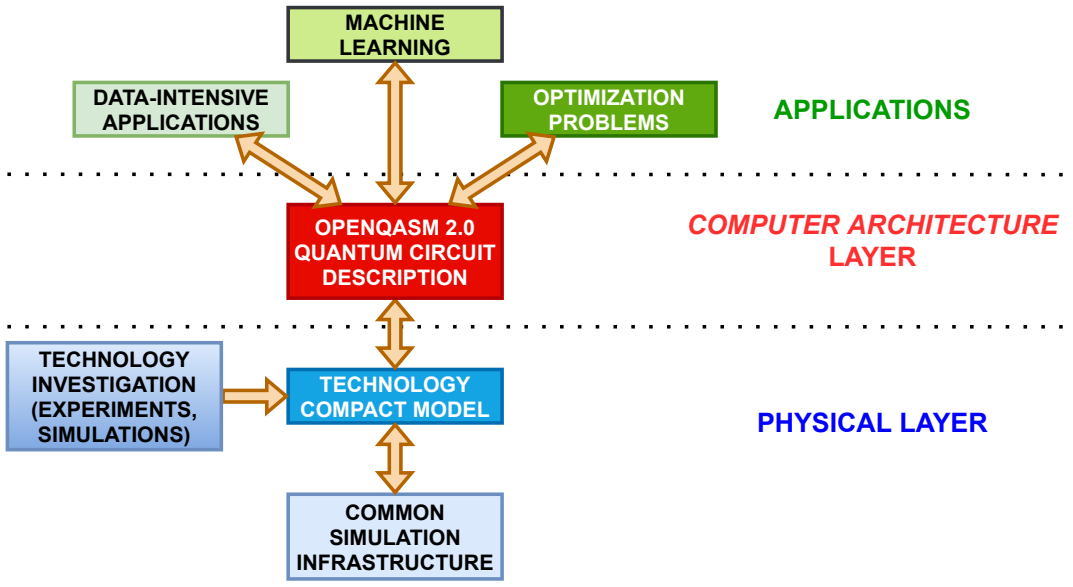


Figure 4.2: Block scheme of the complete simulation infrastructure.

Quantum computing simulators are usually interfaced with consolidated frameworks for the development of application-specific quantum algorithms, such as

Qiskit from IBM [20] and Cirq by Google. The same has been done for the developed simulator, which has been organized according to the block scheme reported in Figure 4.2. Three layers can be clearly observed and these are described from bottom to top, since the definition of the physical layer of the simulator was preliminary to the others.
A simulation procedure common to all quantum technologies has been defined, over which compact models have been defined and developed in such a way that the main dynamics of each device can be uniformed to the routines of the simulator itself. The construction of a compact model and its adaption to the common simulation infrastructure depend on an investigation of the technology for which the model is built. Some phenomena are substantially the same for all technologies, *e.g.* the evolutions associated with technology-native single and two-qubit gates, others are exclusive of a given device, *e.g.* the residual spin-spin coupling in Nuclear Magnetic Resonance (NMR) devices.
When all the routines for describing the standard quantum gates of a given technology are available, circuit design can be done in an agnostic way with respect to simulator routines. For this reason, interfaces between the simulator and the quantum ciruits description language OpenQASM 2.0 have been developed, thus allowing the simulation of a generic quantum circuit on different modelled technologies. Quantum circuits compilation strategies could also belong to this *computer architecture* layer; in fact, the availability of an OpenQASM 2.0 interface in a compiler is sufficient to evaluate circuits generated by it with the developed simulator. Finally, since OpenQASM 2.0 is supported by multiple frameworks, the developed simulator could be employed, in the future, as an additional backend for Qiskit and Cirq, thus permitting to estimate the performance of available technologies in different applicative scenarios, such as optimization problem solving, machine learning and data-intensive applications. However, it is important to precise that this activity is absolutely germinal and not reported in the following of this thesis. It is reminded here that the simulator, such as the OpenQASM 2.0 interpreter, was entirely developed in MATLAB, according to the Object Oriented Programming (OOP), where a class is associated with each module described in Section 4.2. It is reminded that this chapter does not provide the whole description of the simulator organization, which is object of Appendix B, but it is focused on the methodology for the description and simulation of quantum technologies with compact models.

4.2 Organization of a compact model

A compact model of a quantum technology is substantially a *black box* providing the qubit response depending on **external stimuli** and **degrees of freedom associated with environment**. This would permit to estimate the performance of a device when a certain quantum circuit is applied on it. The definition of a

compact model clearly depends on the chosen methodology for the simulation of noisy qubits. In this section, the idea is to provide a general idea of the main actors of a compact model and of their contributions in the simulation of a quantum computer.

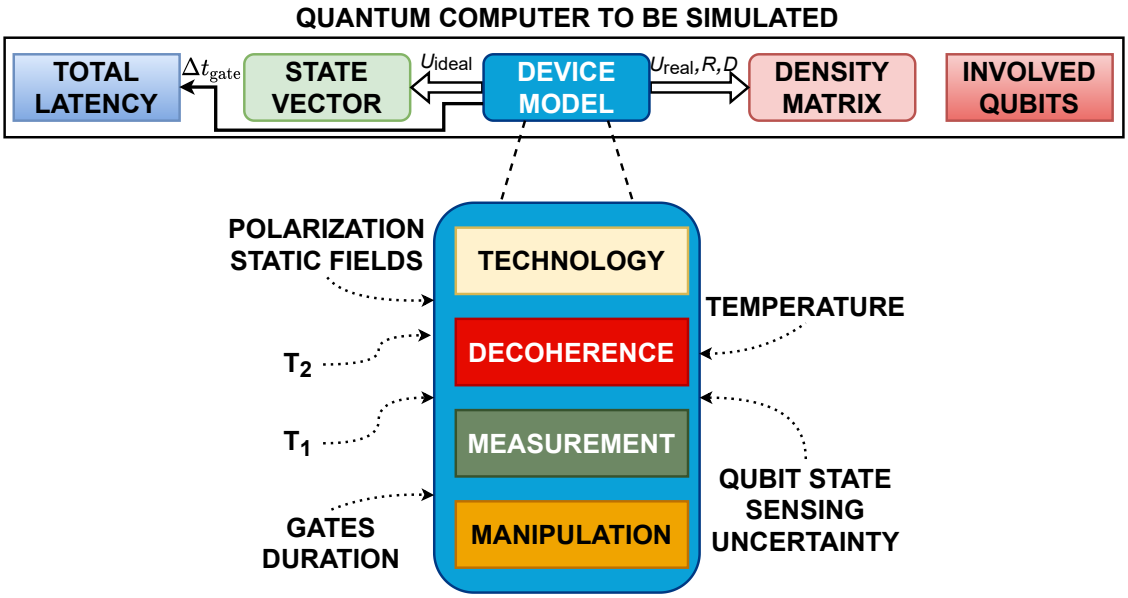


Figure 4.3: Overview of a quantum computing system, where the device model has the role of properly manipulating the state vector and the density matrix.

It is important to remind that classical simulation of qubits can be done in terms of evolution of **density matrix** ρ. Given a quantum circuit, each gate is applied sequentially and it forces an evolution of the quantum system, according to Equation 2.75. The comparison between the final state of noisy qubits, provided by ρ, and of the corresponding ideal qubits, given by the **state vector** $|\psi\rangle_{\text{ideal}}$, permits to estimate the reliability of the execution of the circuit on a real device. This operation can be done in terms of the state **fidelity**, defined in Equation 2.80.
Figure 4.3 introduces the main actors of a compact-model-based simulator. The box on top of Figure contains all the main modules for the simulation:

- The density matrix ρ, which takes into account non-ideality phenomena of quantum gates and of target hardware.
- The **ideal state vector** $|\psi\rangle_{\text{ideal}}$, which evolves exclusively according to the ideal unitary evolutions of gates of the quantum circuit to be simulated. It is practically employed the reference state, to be compared with the density matrix in terms of fidelity.

- The **device compact model**, which provides the evolution matrices for $|\psi\rangle_{\text{ideal}}$ and ρ, according to external degrees of freedom, such as polarization static fields, durations or amplitudes of electromagnetic pulses for the implementation of quantum gates, temperature and sensitivity of the devices employed for measuring the qubit state[1].

- A boolean array of **involved qubits**, with length equal to the total number of qubits in a device. Each location of the array is associated with a qubit and it is asserted when the corresponding qubit has been subjected to at least one quantum gate. This is employed for establishing the probabilities repartition in relaxation (more details are available in section 4.6).

- An accumulator variable, exploited for estimating the **total latency** of a quantum circuit, given by the sum of the durations of gates belonging to it, executed in a strictly sequential way. It is inserted for debugging reasons.

These modules are exploited in all simulations. Moreover, independently on the target technology and its exclusive features, the same *top-level* operations for updating the qubits state can be executed:

- **Single-qubit gate**, described according to the magnetic resonance model in subsection 3.3.3. The unitary evolution to be applied to the $|\psi\rangle_{\text{ideal}}$ is always obtained from Equation 3.34 (with all the required tensor products with identity matrices for untargeted qubits), while that updating ρ depends on the technology to be simulated (additional details will be provided in the following of this thesis).

- **Two-qubit gate**, which is the multi-qubit gate reliably executable on all available quantum computing technologies. Each technology is characterized by its native two-qubit gate — *e.g.* Cϕ in in Cr_7Ni, $R_{zz}(\theta)$ in NMR, $R_{xx}(\theta)$ in trapped ions [41], *etc.* — and in all cases the parameters for their definition are the two involved qubits and the evolution angle θ.

- **No-operation**, for evaluating the effects of some dynamic non-ideal phenomenon, such as relaxation, similarly to what happens in an experimental setup (details in section 2.3.4).

- **Measurement**, which consists in eventual permutations or partial traces calculations to get the probability distributions of qubits to be measured, properly sorted.

[1] Temperature and sensing uncertainty are future perspectives.

The compact model is responsible to provide, for each quantum gate to be simulated, the gate duration Δt_{gate} (to increment the accumulator associated with the total circuit latency) and the following **matrices**:

- a unitary matrix U_{ideal}, to be applied onto $|\psi\rangle_{\text{ideal}}$ and related to the **ideal evolution of qubits with a certain quantum gate**;
- a unitary matrix U_{real} to be applied onto ρ, describing the **non-ideal qubits evolution** with a certain quantum gate, involving not only the expected dynamics, but also unitary non-idealities of the target technology (*e.g.* undesired inter-qubit couplings or off-resonance mechanisms);
- non-unitary matrices R and D affecting ρ, according to the evolution of two **non-ideal phenomena**: relaxation and decoherence (details in section 4.6).

In the definition of a compact model, four main sub-models have been identified, which can be seen in the bottom part of Figure 4.3. The first one is related to the characteristic features of the **technology** to which the device belong. From a practical point of view, this sub-model permits to compute the unitary evolutions associated with single and two-qubit gates and of unitary non-idealities, *e.g.* residual qubit coupling with open switch in Cr_7Ni supramolecular complexes or off-resonance effects in homonuclear NMR qubits.
The second sub-module is the **decoherence model** of the technology, strictly related to the relaxation and decoherence timescales $T_{1,2}$. This can be arranged to the characteristic decoherence dynamics of a given device. In the current infrastructure, all the devices employ the general model described in section 4.6.
The **measurement mechanism** could slightly modify the probability distribution of the qubits eigenstates. Routines of this sub-model are called when a measurement operation is called and they modify the density matrix according to the main features of the qubits measurement system, *e.g.* uncertainty or systematic offsets to be compensated. Current model executes ideal measurements and already permits to execute a measurement on a number of qubits lower than the total of a given device through partial trace calculations.
Finally, a model of the **system for manipulating qubits** is available. From a practical point of view, it contains all the characteristic qubit control parameters, such as waveform, duration and amplitude of each electromagnetic pulse employed for implementing a quantum gate on a certain qubit. In the current implementation of the simulator the manipulation of all qubits is assumed, for sake of simplicity, to be done in the same way, *i.e.* **fixing one among duration and amplitude of rectangular pulses**.
The organization is sufficiently modular to ensure that all possible quantum computing technologies can be simulated in the same way. In fact, the data structures associated with the qubits state are always the density matrix ρ and the state vector $|\psi\rangle_{\text{ideal}}$. Moreover, for each quantum gate, the compact model returns the gate

delay and U_{ideal}, U_{real}, R and D matrices, affecting ρ and $|\psi\rangle_{\text{ideal}}$. Details about simulation methodology and scheduling are reported in section 4.3.

4.3 Compact-model-based simulation methodology

The general procedure for simulating a complete quantum circuit based on compact models was proposed in [59]. Taking into account the scheme in Figure 4.3, the main steps are reported in the following:

1. A device model, with all its characteristic features (native unitary evolutions, manipulation, relaxation and decoherence timescales $T_{1,2}$, *etc.*) is instantiated and included in the top module, containing the ideal state vector $|\psi\rangle_{\text{ideal}}$, the density matrix ρ, the list of involved qubits and the latency accumulator.

2. The system is initialised in its pure ground state, corresponding to $|\psi\rangle_{\text{ideal}} = |0\rangle$ and $\rho_{\text{init}} = (|0\rangle\langle 0|)^{\otimes n}$.

3. Each quantum gate is applied sequentially and updates ρ and $|\psi\rangle_{\text{ideal}}$, according to the matrices U_{ideal}, U_{real}, R and D obtained with the compact model.

4. After a measurement operation call, the infrastructure provides in output the ideal outcomes, those obtained resorting to the proposed model and the corresponding fidelities.

This section is focused on the description of the third and fourth steps. For what concerns the third step, simulation of a single quantum gate with compact models, *i.e.* the procedures for computing U_{ideal}, U_{real}, R and D, is introduced. It is precised that the same workflow is also employed for the generation of matrices associated with two-qubit gates. Figure 4.4 shows the single-qubit-gate simulation workflow and the garnet-red circles permit to understand how the operations are ordered, according to six steps.
First of all, the gate to be simulated must be properly interpreted, *i.e.* the target qubit, the oscillation phase ϕ associated with the rotation axis and the angle θ of the single-qubit gate must be properly recognized (step 0). In the simulation of a two-qubit gate, step 0 must recognize the two involved qubits and the angle associated with the characteristic two-qubit gate of the technology to be simulated (*e.g.* the ϕ or the Cϕ or the θ of the $R_{zz}(\theta)$).

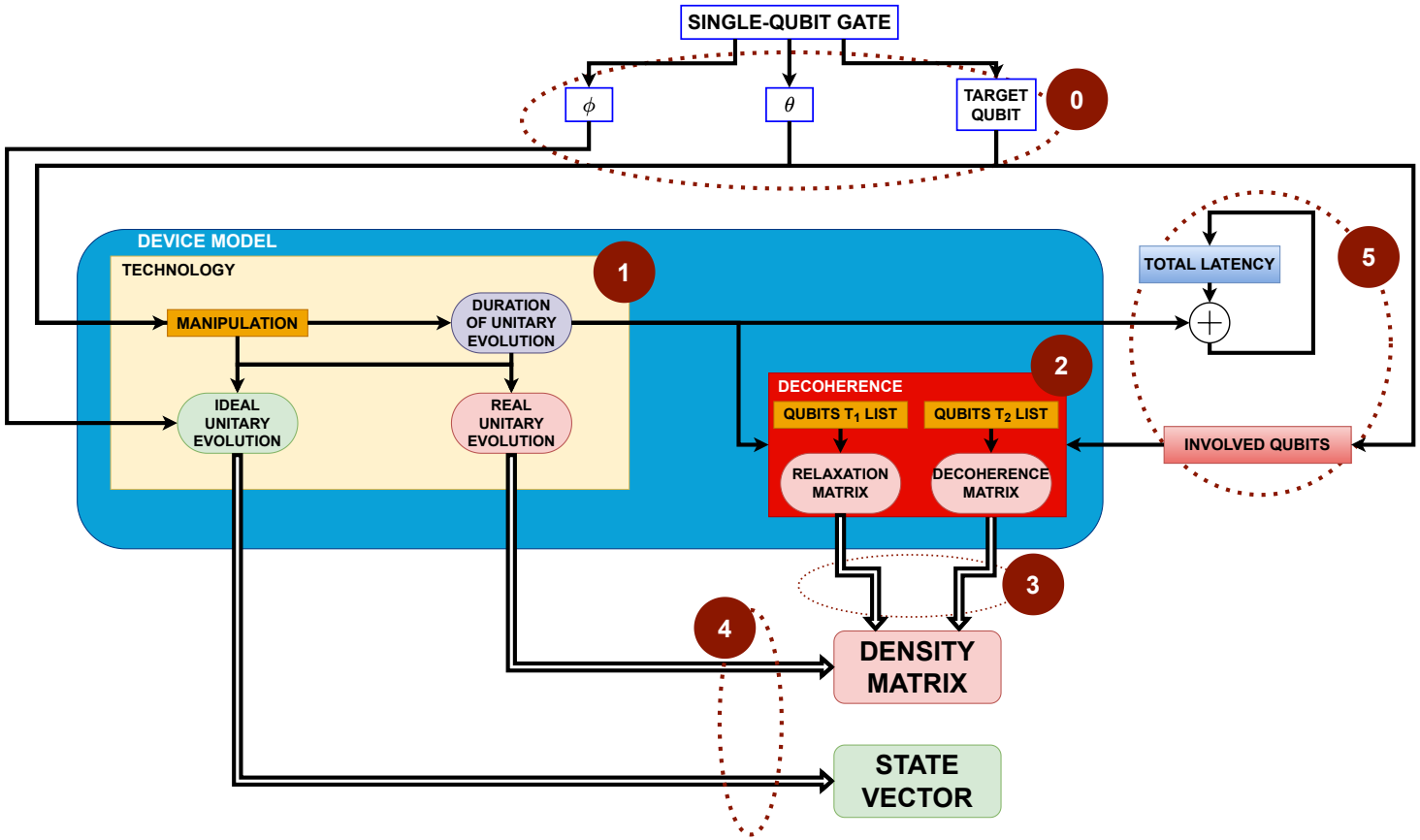
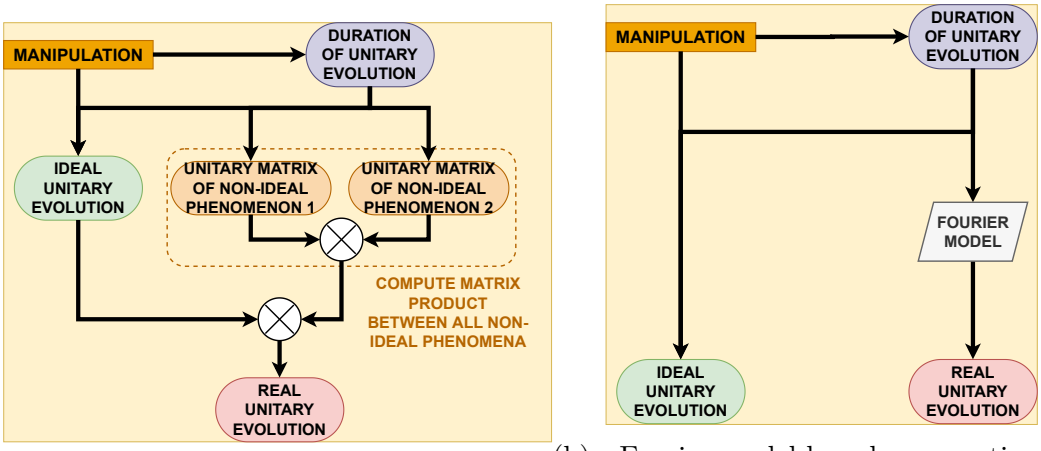


Figure 4.4: Overview of a single-qubit gate simulation. The garnet-red circles indicate the operations ordering.

At the completion of this preliminary operation, the manipulation parameters of the target qubit are involved to derive the timescales for the execution of the quantum gate, according to the magnetic resonance model described in subsection 3.3.3 (step 1). With this information, U_{ideal}, U_{real}, R and D can be computed, through the models inside the technological and decoherence modules (steps 1 and 2). In the current implementation of the simulator, single-qubit-gate U_{ideal}, R and D are computed in the same way for all technologies, while the calculation of U_{real} depends on the specific model of each technology. Figure 4.5 shows two available technology-dependent methods for the generation of U_{real}. In the left case (Figure 4.5a), unitary matrices associated with unwanted phenomena and U_{ideal} are initially treated separately, then they are multiplied with each other to obtain U_{real}. This method is similar to a Trotter – Suzuki expansion (Equation 2.41) and could be employed where there is a significant Hamiltonian contribution, *e.g.* that of magnetic resonance, and other perturbation-like terms, *e.g.* the *open-switch* qubits interaction in switch-based molecules (further details are available in subsection 5.1.3). Figure 4.5b shows a mechanism employed in models of Nuclear Magnetic Resonance quantum computers, where U_{real} calculation, instead of exploiting U_{ideal}, is based on an autonomous model based on the Fourier spectral analysis of electro-magnetic pulses (further details are available in Sections 4.5 and 5.2.1).

Non-unitary matrices (step 3) are applied before the unitary ones (step 4). The



(a) Unitary matrices product.

(b) Fourier-model-based generation of U_{real}.

Figure 4.5: Possible generation mechanisms of U_{real} in the available compact models.

reason of this choice can be traced back to phenomenology; in particular, considering the description of qubit spectroscopy experiments in subsection 2.3.4, relaxation and decoherence emerge after the end of the application of the π or $\frac{\pi}{2}$-pulse. This can be interpreted — at least in a first approximation — as the fact that the evolution due to these non-ideality phenomena depends on the *past* of the quantum

system state and takes place during the current quantum gate. In order to properly emulate this mechanism, the application to ρ of R and D before U_{real} ensures that R and D modify, on the same time duration of the quantum gate, the qubits state taking into account its *past*. In the case of the first gate, assuming that the initial state is $\rho = |0\rangle\langle 0|$, only the unitary matrix of the quantum gate affects the density matrix, since R and D are ineffective with respect to $\rho = |0\rangle\langle 0|$, as expected experimentally.
Finally, the list of qubits subjected to at least one quantum gate and the total circuit latency accumulator can be updated (step 5).
The simulation of two-qubit gates is done in the same way, while measurement operations are treated in a quite different way, so they require additional comments. First of all, it is important to precise that, in the current implementation of the simulator, measurements are assumed to be ideal, *i.e.* without any offset error to be corrected or measurement uncertainty. Figure 4.6 shows the flow chart of measurement operation. In the middle, the main flow chart is shown, while on the two sides two sub-charts are reported, which are related to the main required operations, involving the manipulation of the density matrix, according to the qubits to be measured (left blue sub-chart) and their order in the classical register (right green sub-chart). In order to understand the flow chart, it is convenient to report a pratical example, involving a system of five qubits, labelled with a number from 1 to 5. The following measurement operation must be done:

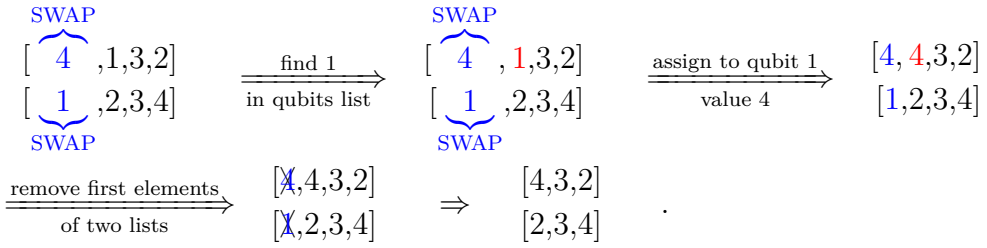
$$\text{measure}([5,1,4,3],[1,2,3,4])\,,$$

where the first list reports the indexes of qubits to be measured and the second one the indexes of classical bits onto which the qubit to be measured in the same list position must be mapped. After checking that the two lists have equal length, which must be lower or equal than the total number of qubits of the target device, partial trace and index updates operations, reported in the blue sub-chart, are executed. Partial trace operations on non-involved qubits (only that labelled as 2, in this case) are executed to obtain a density matrix associated with qubits 1, 3, 4 and 5 and an updated qubits lists equal to [4,1,3,2]. At this point, qubits reordering, described in the green sub-chart, must be done. Attention must be focused on the dark green sub-blocks, which describe the permutation algorithm. In order to facilitate the comprehension, it is convenient to report the two lists of qubits and classical bits to be measured:

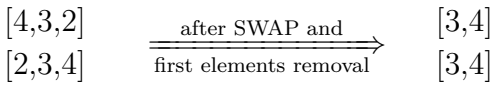
$$\begin{matrix}[4,1,3,2] \\ [1,2,3,4]\end{matrix}\,.$$

Since the first two elements, 4 and 1, are not equal, the unitary matrix associated with an ideal SWAP gate between qubits 4 and 1 — built according to the scheme in Equation 3.17 — is applied to the density matrix thus permitting that the qubit

to be associated with the classical bit 1 is in the right position. At this point, the qubit previously labelled as 1 must be the new 4, so the updated list becomes, after all the reported steps:



The first two elements of the updated list (4 and 2) are not equal, so the SWAP operation must be done again according to the previous scheme:



Finally, in the last two iterations, SWAP operations are not required and the final density matrix has qubits ordered in the desired way. At this point, considering that measurements are ideal, the whole density matrix or its main diagonal, corresponding to the eigenstates probability distribution, can be returned to complete the measurement simulation instruction.
Before concluding, it is important to precise that, in the current implementation of the simulator, measurements are always the last operations of the tested quantum circuits, since a mechanism allowing the update of the qubits density matrix, according to the outcome of those measured, has not been already developed. However, a little *coding effort* at software level is expected to be sufficient to achieve this goal.

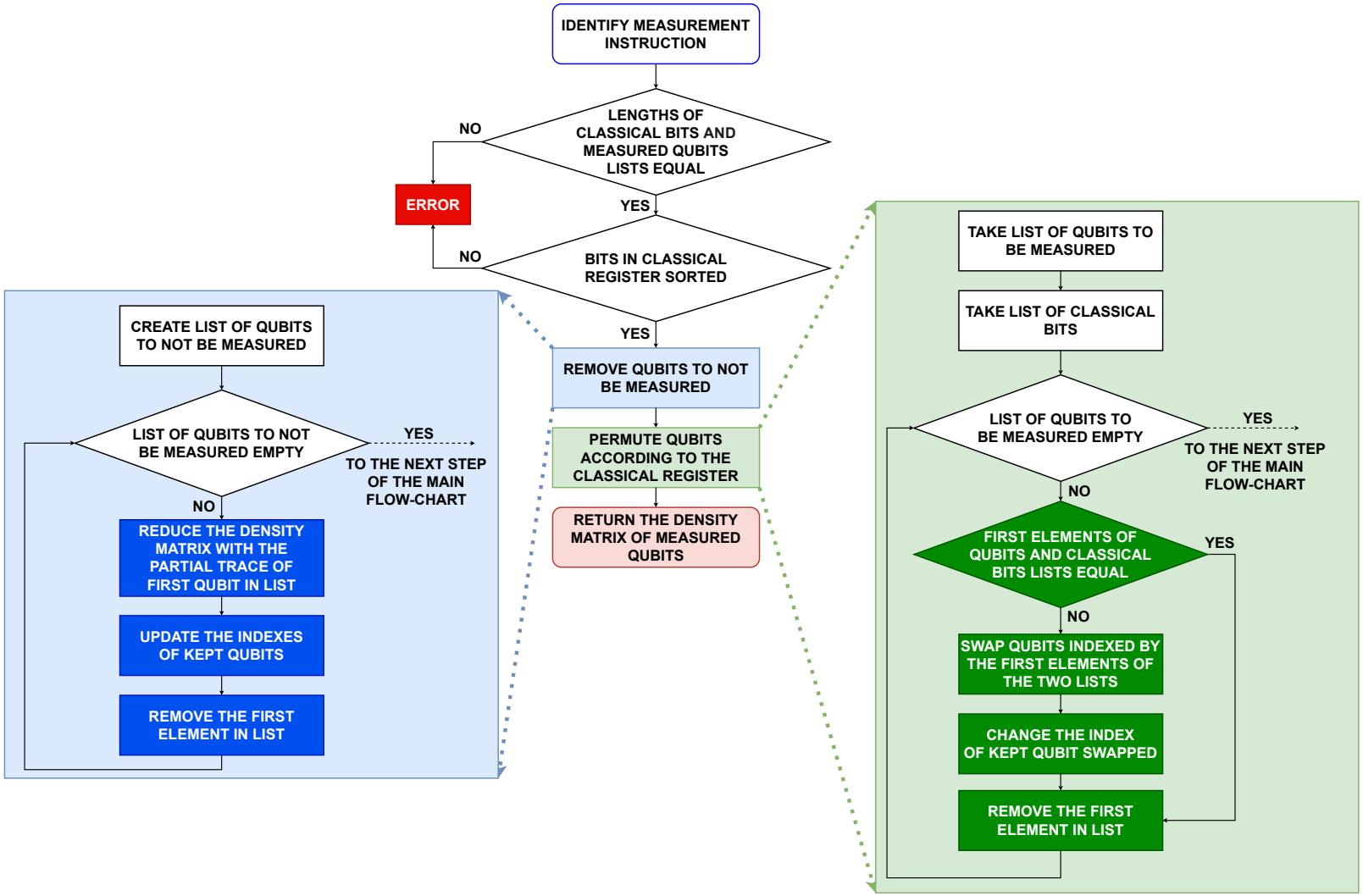


Figure 4.6: Flow chart of measurement operation.

4.4 Transversal sub-models

Compact models can be classified in two main categories:

- **transversal**, common to different quantum computing technologies;
- **exclusive** of a given technology.

Transversal compact models are usually employed as sub-modules of more complex ones, encapsulating the description of a complete device. In the following sections, an overview of two sub-models — which can be arranged for their usage with all the quantum technologies available in the simulator — is reported. These are the models describing relaxation, decoherence and the evolution of qubits under an electromagnetic pulse not resonant with them (off-resonance). While relaxation and decoherence are already available in all the defined technology models, off-resonance model is currently employed only in the model of Nuclear Magnetic Resonance quantum computers. However, it is important to precise that it can be theoretically extended to all quantum technologies affected by crosstalk.
For each compact model, after an initial description of the physical phenomena at its basis, a description is reported, such as the testing methodologies to prove their effectiveness in a simulative scenario.

4.5 Fourier model for off-resonance description

In most of quantum computing technologies, it is not possible to ensure that an oscillating pulse for targeting a specific qubit would not affect the others of the system. This is due to either the impossibility of creating an electromagnetic oscillation sufficiently narrow to point a single qubit, as in homonuclear molecular quantum computers, or crosstalk phenomena which could take place in superconducting devices [62]. Figure 4.7 shows how an excitation signal resonant with the qubit Q2 also slightly affects the other qubit Q1.

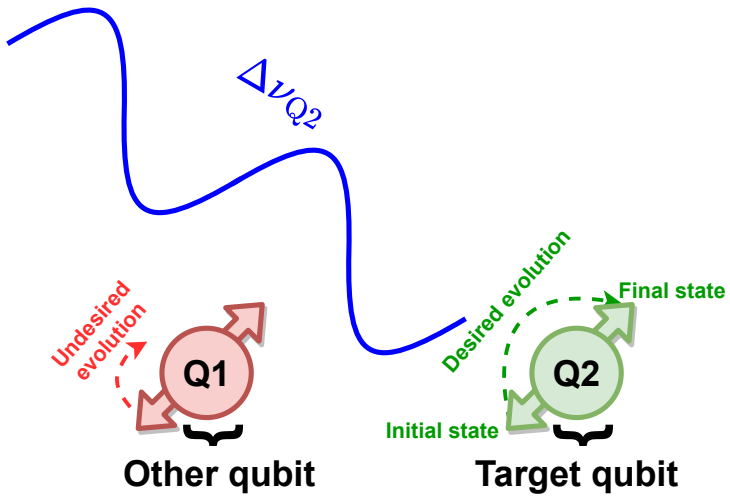


Figure 4.7: Visual description of off-resonance mechanism.

4.5.1 Analytical descrption of the phenomenon

The single-qubit magnetic resonance model, introduced in subsection 3.3.3, could be not sufficient for the description of the evolution of a quantum system involving multiple qubits with close resonance frequencies, since the operating conditions for Rotating Wave Approximation (RWA) are not always guaranteed. This implies that calculation of the exact unitary evolution requires to carry out a time integration, which is highly expensive from a computational point of view. An example, involving two uncoupled qubits `q0` and `q1`, with difference between resonance frequencies $\Delta\omega_0$, is reported in the following.
Assuming that two excitation signals, one for each qubit, can be applied to the system:

$$B_k(t) = 2B_{r,k}\cos(\omega_{r,k}t - \phi)\,, \tag{4.1}$$

that the rotating frame angular frequency of each qubit ω_k is equal to the external excitation angular frequency $\omega_{r,k}$ (on its hand equal to the resonance angular frequency $\omega_{0,k}$) and that an external oscillation resonant with `q0` is applied to the system ($B_{r,0} \neq 0$, $B_{r,1} = 0$), the Hamiltonian can be written — arranging the

demonstration in [60] — as:

$$\begin{aligned}\frac{\widetilde{\mathcal{H}}_{r,0}}{\hbar} = \frac{\omega_{1,0}}{2}\Big\{ &\Big[\cos(2\omega_{r,0}t-\phi_0)+\cos(\phi_0)\Big](\mathbb{I}\otimes\sigma_x)+\\ &+\Big[\sin(2\omega_{r,0}t-\phi_0)+\sin(\phi_0)\Big](\mathbb{I}\otimes\sigma_y)+\\ &+\frac{\gamma_1}{\gamma_0}\Big[\cos(\Omega_r t-\phi_0)+\cos(\Delta\omega_r t+\phi_0)\Big](\sigma_x\otimes\mathbb{I})+\\ &+\frac{\gamma_1}{\gamma_0}\Big[\sin(\Omega_r t-\phi_0)+\sin(\Delta\omega_r t+\phi_0)\Big](\sigma_y\otimes\mathbb{I})\Big\},\end{aligned} \tag{4.2}$$

where ω_{1_0} is the precession frequency of `q0`, ϕ_0 is the phase of the oscillating excitation signal, $\frac{\gamma_1}{\gamma_0}$ is, inheriting the spin formalism, the ratio of the qubits precession factors (equal to the ratio of their resonance frequencies), $\Delta\omega_r = \omega_{r,1} - \omega_{r,0}$ and $\Omega_r = \omega_{r,1} + \omega_{r,0}$. Assuming to evaluate the evolution on timescales which permit to employ the RWA on additive frequency terms ($2\omega_{r,0}, \Omega_r \gg \frac{2\pi}{\tau}$), the Hamiltonian can be written as:

$$\begin{aligned}\frac{\widetilde{\mathcal{H}}_{r,0}}{\hbar} = \frac{\omega_{1,0}}{2}\Big\{ &\cos(\phi_0)\cdot(\mathbb{I}\otimes\sigma_x)+\sin(\phi_0)\cdot(\mathbb{I}\otimes\sigma_y)+\\ &+\frac{\gamma_1}{\gamma_0}\left[\cos(\Delta\omega_r t+\phi_0)\cdot(\sigma_x\otimes\mathbb{I})+\sin(\Delta\omega_r t+\phi_0)\cdot(\sigma_y\otimes\mathbb{I})\right]\Big\},\end{aligned} \tag{4.3}$$

Analogously to the single-qubit and additive frequency cases (see Equation 3.31), the RWA can be reasonably employed on the time-dependent terms involving $\Delta\omega_r$ when

$$|\Delta\omega_r\tau| \gg 2\pi\,. \tag{4.4}$$

When resonance angular frequencies are significantly different from each other ($|\Delta\omega_r\tau| \to \infty$), such as in heteronuclear molecules, time-dependent terms can be easily neglected in all practical cases. On the other hand, when they are close to each other ($\Delta\omega_r \to 0$), as in homonuclear molecules, where the difference in resonance frequencies is in the order of some parts per million, inequality 4.4 can be ensured by only enhancing the pulse duration τ. A pulse that satisfies this requirement is known as **soft pulse**, in contrast with hard pulses, routinely employed when $|\Delta\omega_r\tau| \to \infty$.

The approximated soft-pulse Hamiltonian, where each qubit k is targeted by a soft-pulse which negligibly affects the others and $\omega_k = \omega_{r,k} = \omega_{0,k}$, can be written as:

$$\frac{\widetilde{\mathcal{H}}}{\hbar} = \frac{1}{2}\sum_{k=0}^{n-1}\omega_{1,k}\left[\cos(\phi_k)\sigma_{x_k}+\sin(\phi_k)\sigma_{y_k}\right]. \tag{4.5}$$

Even though this Hamiltonian is simple and not time-dependent, it does not properly describe the off-resonance mechanism, *i.e.* when $\Delta\omega_r \to 0$.

4.5.2 Fourier model

In order to find a compromise between accuracy and computational complexity, a model based on the Fourier transform — which permits to treat the target qubit according to Equation 4.5 and taking at the same time into account the off-resonance evolution of non-addressed qubits— has been defined.
According to a practical rule of Nuclear Magnetic Resonance spectroscopy, higher addressability of spins with resonance frequencies close to each other can be achieved with longer pulses in the time domain, since their spectra are narrower, so they are more selective in the frequency domain. From a mathematical point of view, soft pulses must make the off-resonance term of the Hamiltonian in Equation 4.3 negligible in the corresponding unitary evolution. This experimental argument can help in the definition of a compact description of single-qubit gates based on the Fourier analysis. Before going into the detail, it could be convenient to understand the expected outcome of the analysis and how this affects the simulation of qubits with close resonance frequencies. The goal is to compute an effective single-qubit-gate unitary evolution when the l^{th} qubit among n is rotated by an amount θ. An equivalent resonant oscillation is assumed to be applied on each qubit k and each oscillation, in general, differs from the others in terms of amplitude, so that each qubit is subjected to a different effective rotation angle θ_{eff_k}, which is computed by taking into account the Fourier spectrum of the radiation pulse. The total unitary evolution is

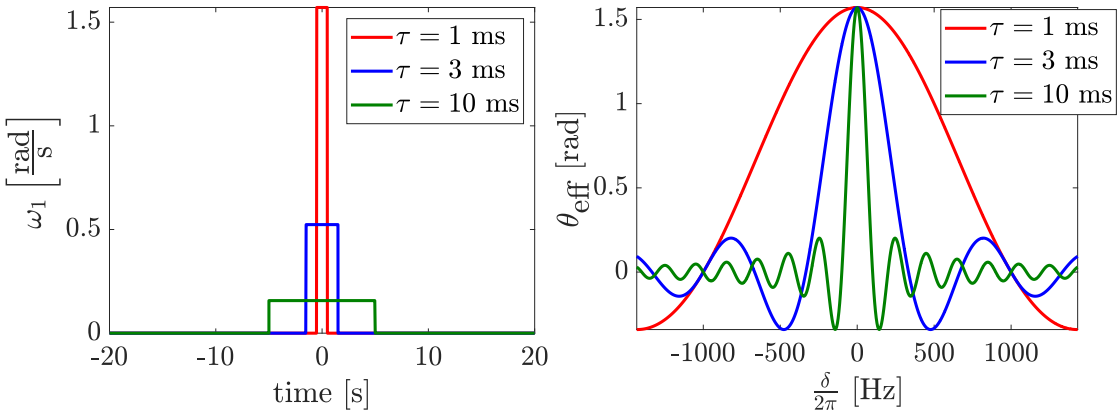
$$U(\tau) = \bigotimes_{k=0}^{n-1} \exp\left\{-i\frac{\theta_{\text{eff}_k}}{2}[\cos(\phi_l)\sigma_x + \sin(\phi_l)\sigma_y]\right\}, \tag{4.6}$$

where ϕ_l is the phase of the oscillating signal chosen for rotating the target qubit about a specific axis in the xy-plane and θ_{eff_k} is the effective rotation angle of the k^{th} qubit. In an ideal case, $\theta_{\text{eff}_l} = \theta$ and $\theta_{\text{eff}_k} = 0$ for each qubit k different from the target one.
The Fourier analysis starts from observing that the effect of an external electric or magnetic field — with amplitude $2A_r$, oscillating on the $\hat{x}$ axis with angular frequency ω_r — on a qubit of gyromagnetic ratio γ_n can be substantially described in terms of an equivalent precession signal:

$$\omega_{\text{prec}}(t) = 2m(t)\cos(\omega_r t - \phi), \tag{4.7}$$

where $m(t) = A_r\gamma_n \cdot s(t) = \omega_1 s(t)$ is a modulating signal whose dimension is an angular frequency. If it is assumed that $m(t)$ corresponds to the equivalent precession signal in a frame rotating at angular frequency ω_r about the $\hat{z}$ axis under the RWA, the rotation angle of the target qubit is given by:

$$\theta = \int_{-\infty}^{+\infty} m(t)\mathrm{d}t. \tag{4.8}$$



(a) Time domain. (b) Frequency domain.

Figure 4.8: Effective rotation angles with three different port precession signals.

If $m(t)$ is assumed to be equal to $\omega_1 \mathbf{P}_\tau$, *i.e.* a port signal of duration τ and amplitude ω_1, the angle is exactly equal to

$$\theta = \int_{-\infty}^{+\infty} \omega_1 \mathbf{P}_\tau \mathrm{d}t = \omega_1 \int_0^\tau \mathrm{d}t = \omega_1 \tau, \tag{4.9}$$

coherently with the formalism reported in subsection 3.3.3. The Fourier transform of $m(t)$ is equal to

$$\mathcal{F}\{m(t)\} = M(\delta) = \int_{-\infty}^{+\infty} m(t) \mathrm{e}^{-\mathrm{i}\delta t} \mathrm{d}t, \tag{4.10}$$

where $\delta = \omega - \omega_r$ is the the independent variable related to the effective angular frequency in the rotating frame, so the frequency domain of interest is centred at frequency ω_r. The defined Fourier spectrum has angular dimensions; this implies that the effective rotation angle of a qubit with resonance frequency ω_k can be computed as the value of the spectrum at frequency $\delta_k = \omega_{0,k} - \omega_r$. For the port signal $m(t) = \omega_1 \mathbf{P}_\tau$ the spectrum is

$$\mathcal{F}\{\omega_1 \mathbf{P}_\tau\} = \omega_1 \tau \,\mathrm{sinc}\left(\frac{\delta\tau}{2\pi}\right) = \omega_1 \tau \frac{\sin\left(\frac{\delta\tau}{2}\right)}{\frac{\delta\tau}{2}} = \omega_1 \tau \frac{\sin\left(\pi\frac{\delta\tau}{2\pi}\right)}{\pi\frac{\delta\tau}{2\pi}}, \tag{4.11}$$

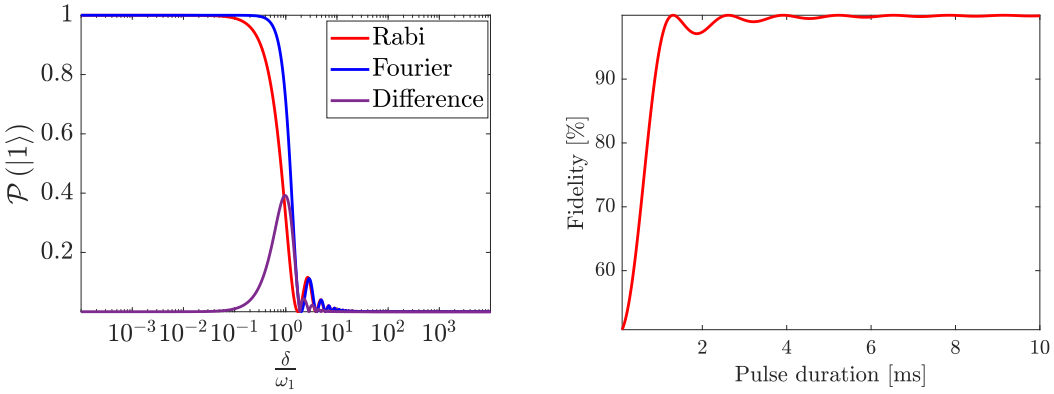
so that the effective angle of the k^{th} qubit with detuning δ_k is computed as

$$\theta_{\text{eff}_k} = \omega_1 \tau \,\mathrm{sinc}\left(\frac{\delta_k \tau}{2\pi}\right). \tag{4.12}$$

The effective angle of the target qubit, for which δ tends to 0, is equal to $\omega_1\tau$, as expected by Equation 4.9. According to Equation 4.11, a longer pulse shows

a narrower main lobe, thus fast achieving negligible rotation angles for undesired qubits. Figure 4.8a shows three port precession signals, associated with $\frac{\pi}{2}$-pulses with different duration τ. The corresponding spectra are all equal to $\frac{\pi}{2}$ when $\delta \to 0$, but the one corresponding to the pulse with the shortest time duration and highest amplitude (red in Figure 4.8b) shows the largest lobe. This implies that, given a detuning frequency δ_k, the effective rotation angles are in general further from $\omega_1\tau$ for larger pulses, since spectrum tends to fast achieve values close to 0.
This model is not employable only with port modulating signals. In fact, it can be employed with any $m(t)$ as Gaussian or numerically-optimised pulses, which are usually adopted in experimental quantum computing.

4.5.3 Validation



(a) Probabilities obtained with standard Rabi and approximated Fourier models, when a single qubit is manipulated.

(b) Dependency of fidelity on pulse duration for a two-qubit system, where each qubit is affected by a $\frac{\pi}{2}$-pulse.

Figure 4.9: Functional tests of the Fourier model.

In order to have an idea of how model works in a practical scenario, two different tests were done. The first one can be seen as an analysis of the capabilities of the Fourier model in emulating the canonical one described in subsection 3.3.3 (also called Rabi model from Rabi oscillations), at least for what concerns transition probabilities. In particular, a π-pulse is assumed to be applied on a single qubit in state $|0\rangle$, whose precession frequency in resonance regime is ω_1, and the difference of probabilities obtained with the Rabi model and with the approximated Fourier

model is computed:

$$\begin{aligned}\epsilon_{\text{Fourier}} &\triangleq \left|\mathbb{P}_{\text{Rabi}} - \mathbb{P}_{\text{Fourier}}\right| \\ &= \left| \frac{1}{1+\left(\frac{\delta}{\omega_1}\right)^2} \sin^2\left[\frac{\sqrt{1+\left(\frac{\delta}{\omega_1}\right)^2}\pi}{2}\right] - \sin^2\left\{\frac{1}{2}\left[\frac{\sin\left(\frac{\pi\delta}{2\omega_1}\right)}{\frac{\pi\delta}{2\omega_1}}\cdot\pi\right]\right\}\right|\end{aligned} \tag{4.13}$$

Figure 4.9a shows the dependence of the probability of measuring $|1\rangle$ after the π-pulse on the ratio between detuning δ and precession frequency ω_1. It is possible to ascertain that the probability difference is generally negligible, apart from a range $10^{-1} \leq \frac{\delta}{\omega_1} \leq 10^{1}$, where the probability difference does not overcome 0.4 in any case. However, that range could be not very critical, because in a practical scenario qubits fabrication (affecting δ or equivalently $\Delta\omega_0$) and control (affecting ω_1) are expected to be engineered in such a way that $\frac{\delta}{\omega_1}$ does not belong to the critical interval. In conclusion, in a reasonable manipulation scenario, where $\frac{\delta}{\omega_1} \to \infty$ for not addressed qubits and $\frac{\delta}{\omega_1} \to 0$ for that to be manipulated, Fourier models describes the single-qubit manipulation in a reasonable approximation.
In the second test, simulations for estimating the fidelity of single-qubit gates in a two-qubit system was done. In particular, the dependence of fidelity on single-qubit-gate duration was evaluated. In all the simulations, qubits were assumed to have infinite coherence timescale and to not interact, thus ensuring that the only involved non-ideality phenomenon is off-resonance. Moreover, qubits were sequentially affected by a $R_x\left(\frac{\pi}{2}\right)$.
Figure 4.9b shows the fidelity dependency on $\frac{\pi}{2}$-pulse duration, with difference between resonance frequencies $\frac{\Delta\omega_0}{2\pi} \simeq 765\,\text{Hz}$. Apart from ringing with local maxima when the pulse duration is an integer multiple of $T = 2\pi/\left|\Delta\omega_0\right| \sim 1.31\,\text{ms}$, it is possible to observe that fidelity asymptotically gets close to 1 for longer duration, thus proving that off-resonance mechanism is minimized by enhancing the gate duration, as expected from soft pulse theory.

4.6 Relaxation and Decoherence

As already introduced in subsection 2.3.4, relaxation and decoherence are one of the most critical non-ideality phenomena in quantum computing technologies and they are canonically described in terms of Kraus operators, which could be extremely heavy from a computational point of view, especially for what concerns the required memory amount. In this thesis, the methodology for compact modelling these phenomena in a quantum circuits simulation already described in [60] is reported, such as the additional validation results.

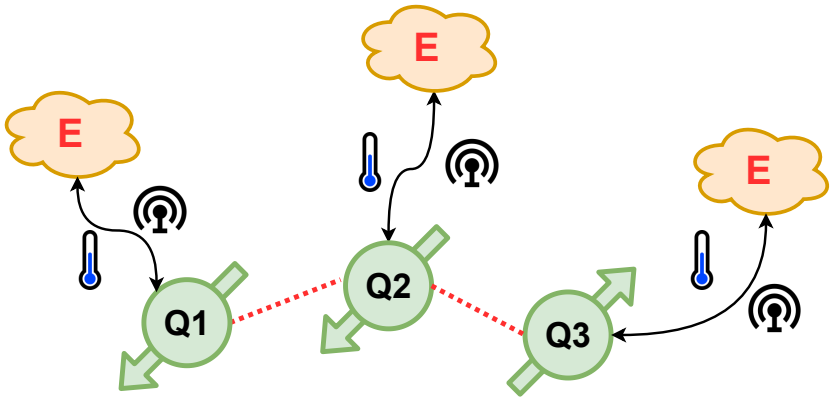


Figure 4.10: Equal local baths model, where each qubit interacts thermodynamically or electro-magnetically with a specfic environment.

4.6.1 Proposed model

The proposed approach is based on the assumption that the reference physical system employed for describing the dynamics of noisy qubits is based on **equal local baths** [63], shown in Figure 4.10, where each qubit interacts with a separated environment and all the environments are equal, so the decoherence dynamics of each qubit does not depend on the others. In [59] the first attempts for describing the effects of relaxation and decoherence were proposed. The effects of relaxation and decoherence on a single-qubit density matrix in the time domain t can be written as [7]:

$$\begin{bmatrix} (a-a_0)\mathrm{e}^{-\frac{t}{T_1}}+a_0 & b\mathrm{e}^{-\frac{t}{T_2}} \\ b^*\mathrm{e}^{-\frac{t}{T_2}} & (a_0-a)\mathrm{e}^{-\frac{t}{T_1}}+1-a_0 \end{bmatrix}, \tag{4.14}$$

where a and a_0 are the probabilities of measuring the qubit in $|0\rangle\langle 0|$ for $t=0$ and $t\to\infty$ respectively and T_1 and T_2 are the relaxation and effective decoherence time constants. It can be observed that the effects of relaxation and decoherence are applied locally in density matrices, thus permitting to define a simplified strategy for describing these effects, without storing all the Kraus operators. Decoherence of an n-qubit system can be modelled with a decoherence matrix D of size $2^n\times 2^n$:

$$\begin{aligned} D &= \bigotimes_{i=n-1}^{0} D_i = D_{n-1}\otimes\cdots\otimes D_0, \\ D_i &\triangleq \begin{bmatrix} 1 & \mathrm{e}^{-\frac{t}{T_{2,i}}} \\ \mathrm{e}^{-\frac{t}{T_{2,i}}} & 1 \end{bmatrix}, \end{aligned} \tag{4.15}$$

where $\otimes$ refers to the Kronecker product, D_i is the decoherence matrix of the i^{th} qubit among n, with decoherence time constant $T_{2,i}$. The element-by-element product of matrix D with the density matrix is sufficient for evaluating the effect

of decoherence. It is possible to ascertain that this is the expected analytical decoherence dynamics of the density matrix in a system with equal local baths [63]. Relaxation was described in [59] by a matrix R, constructed analogously to the n-qubit decoherence matrix:

$$\begin{aligned} R &= \bigotimes_{i=n-1}^{0} R_i = R_{n-1} \otimes \cdots \otimes R_0, \\ R_i &\triangleq \begin{bmatrix} 1 & 1 \\ 1 & \mathrm{e}^{-\frac{t}{T_{1,i}}} \end{bmatrix}, \end{aligned} \tag{4.16}$$

where R_i is the relaxation matrix of the i^{th} qubit among n, with relaxation time constant $T_{1,i}$. The terms on the main diagonal of R, with the exception of that on the top left associated with the ground basis state $|0\cdots0\rangle\langle0\cdots0|$, are always lower than 1, thus permitting to describe their loss of probability. In order to computationally simplify the description of relaxation, the probability loss was always provided to the ground state, according to the following procedure:

1. compute the element-by-element product between relaxation and density matrices, so that the generic (i, j) element of the resulting matrix is $\rho_{\text{el-by-el}_{i,j}} = R_{i,j}\rho_{i,j}$;
2. compute the excited levels loss term as $l = \sum_{i=1}^{2^n-1}(1 - R_{i,i})\rho_{i,i}$ and add it to $\rho_{\text{el-by-el}_{0,0}}$;
3. assign $\rho_{\text{el-by-el}}$ to ρ.

Even though this procedure describes a dynamics involving a loss of probability of higher-energy states and preserves $\sum_{i=0}^{2^n-1}\rho_{i,i} = 1$, the probability re-distribution between eigenstates is not very accurate, since the ground state and the higher-energy states acquire too much and too little probabilities respectively, if compared to those expected in a real relaxation transient. Moreover, the use of the relaxation matrix R does not equally change the off-diagonal coherence terms of the density matrix, since it tends to change more significantly the terms towards the bottom right corner of ρ. For this reason, an approach for describing the probability update can be proposed.
The basic principle is ensuring that a higher probability is provided to states with a larger number of 0 (*i.e.* more qubits in the ground state) and a lower time constant T_1, since they will be subjected to faster transients. Moreover, probability re-distribution is not fixed *a-priori* but is dynamically changed during the simulation, depending on qubits affected by at least one quantum gate.
The first operation of the new approach is evaluating the total lost probability due

to relaxation, computing a **relaxation vector**, instead of a relaxation matrix:

$$\underline{\mathbf{r}} = \bigotimes_{i=n-1}^{0} \left[\exp\left(-\frac{t}{T_{1_i}} \right) \right] = \left[1, \exp\left(-\frac{t}{T_{1_0}} \right), \cdots, \exp\left(-\sum_{i=0}^{n-1} \frac{t}{T_{1_i}} \right) \right]^{\mathsf{T}} = [r_{0,0}\,,\, r_{1,1}\,, \cdots\,,\, r_{2^n-1,2^n-1}]^{\mathsf{T}}, \tag{4.17}$$

so that it is possible to compute the probability lost by each eigenstate as $(1 - r_{k,k})\rho_{k,k}$ and the total lost probability $\mathbb{P}_{\text{lost tot.}} = \sum_{k=0}^{2^n-1}(1-r_{k,k})\rho_{k,k}$. The relaxation vector affects only the main diagonal of the density matrix and it does not provide any imbalance in the exponential decay of the off-diagonal elements. At this point the new **probability distribution procedure** is applied, which assigns to each eigenstate $|k\rangle$ a probability amount

$$\mathbb{P}_{\text{acquired by } |k\rangle} = \frac{w_{|k\rangle}}{\sum_k w_{|k\rangle}} \mathbb{P}_{\text{lost tot.}}, \tag{4.18}$$

where $w_{|k\rangle}$ is the weight of each eigenstate and it depends on the employed qubits, at the moment of the relaxation evaluation and since the beginning of the quantum circuit simulation. Given the set $\mathbb{Q}$ of qubits excited at least once with dimension $\dim(\mathbb{Q})$, $2^{\dim(\mathbb{Q})}$ eigenstates $|k\rangle$ are involved in the relaxation process and each state has a weight equal to:

$$w_{|k\rangle} = \sum_{l \in \mathbb{Q}} \frac{1}{T_{1_l}} (1 - b_l), \tag{4.19}$$

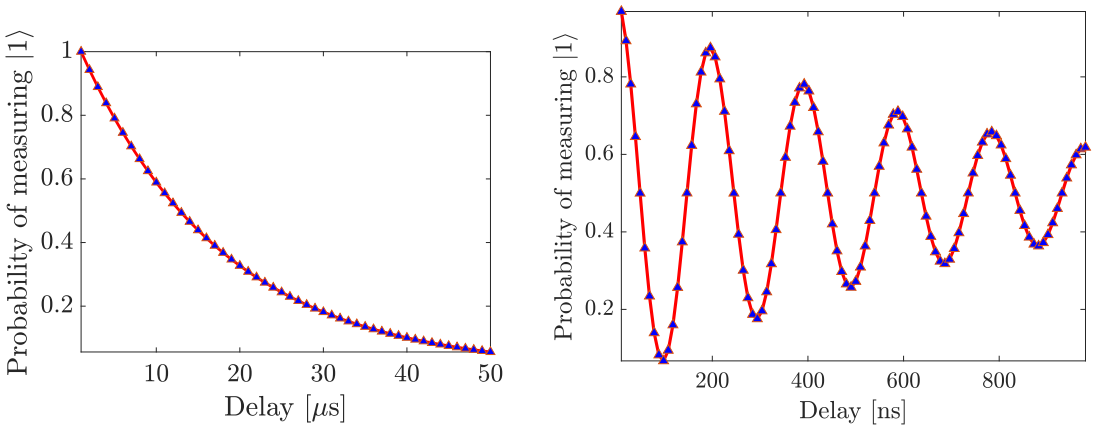
where $b_l = \{0, 1\}$ is the binary value of the qubit l belonging to $\mathbb{Q}$. Therefore, the states with a higher number of $b_l = 0$ and lower relaxation time constants have a higher weight. For example, if some quantum gates have been applied only to two qubits i and j among n, the relaxation process will involve 2^2 eigenstates $|0 \cdots 0\, b_i\, 0 \cdots 0\, b_j\, 0 \cdots 0\rangle$. Each eigenstate $|k\rangle$ among the 2^2 involved has a relaxation weight equal to $w_{|k\rangle} = \frac{1}{T_{1_i}}(1 - b_i) + \frac{1}{T_{1_j}}(1 - b_j)$. Independently from the employed qubits, $|1 \cdots 1\rangle$ has always a weight equal to 0, as it is clear from Equation (4.19). It is important to further clarify how the weight of a given eigenstate is updated in a simulation, depending on the qubits on which quantum gates have been applied at least once, by considering the case of $|0 \cdots 0\rangle \langle 0 \cdots 0|$. If, at the beginning of a quantum circuit, a quantum gate is applied to the qubit 0, $|0 \cdots 0\rangle$ has a weight equal to $\frac{1}{T_{1_0}}$. Finally, if all qubits have been employed since the beginning of the circuit, its weight is $\sum_{i=0}^{N-1} \frac{1}{T_{1_i}}$.
The evaluation of relaxation can be summarised as follows:

1. compute the probability lost for each eigenstate $(1 - r_{k,k})\rho_{k,k}$;
2. compute the total lost probability $\mathbb{P}_{\text{lost tot.}} = \sum_k (1 - r_{k,k})\rho_{k,k}$;
3. evaluate the eigenstates $|k\rangle$ affected by relaxation;

4. compute $w_{|k\rangle}$ and consequently $\mathbb{P}_{\text{acquired by } |k\rangle}$;
5. update the main diagonal of the density matrix according to $\rho_{k,k} = (1 - r_{k,k})\rho_{k,k} + \mathbb{P}_{\text{acquired by } |k\rangle}$.

This simulation strategy permits to significantly reduce the amount of required memory, since a square matrix of size $2^n \times 2^n$ and a vector of length 2^n are sufficient for estimating the dynamics.

4.6.2 Results



(a) Estimation of T_1.

(b) Estimation of T_2 through Ramsey fringes.

Figure 4.11: Simulation of spectroscopic experiments for estimating the relaxation and decoherence time constants $T_1 = 17\,\mu\text{s}$ and $T_2 = 680\,\text{ns}$.

The proposed methodology for the description of the dynamics of non-ideality relaxation and decoherence phenomena discussed in subsection 4.6.1 was validated in two ways.
First of all, the simulations of the experiments for the quantification of T_1 and T_2 were done. Figure 4.11 shows the results obtained for a qubit with expected time constants $T_1 = 17\,\mu\text{s}$ and $T_2 = 680\,\text{ns}$ and resonance frequency $\frac{\omega_0}{2\pi} = 169.9833\,\text{GHz}$. For what concerns relaxation, Figure 4.11a shows the decay of probability of $|1\rangle$, after the application of a $R_x(\pi)$ pulse; in order to properly simulate the experiment, the probability was computed after incremental delays, each one 1 µs higher than the previous one. An exponential decay of type $\text{e}^{-\frac{t}{17\,\mu\text{s}}}$ was expected; an `exp1` fitting on MATLAB, of type $f(t) = a{\cdot}\text{e}^{b\cdot t}$ provided $a = 1.061$ and $b = -0.058\,82\,\mu\text{s}^{-1} \simeq -\frac{1}{17\,\mu\text{s}}$, with 95% confidence bounds. T_2 was obtained with the simulation of a Ramsey experiment. Before detailing the mechanism, reader is invited to take into

account magnetic resonance equations in subsection 3.3.3, in particular those involving the Rabi frequency, and the description of qubit spectroscopy experiments in section 2.3.4. The whole simulation was done in a rotating frame with angular frequency $\omega_0 - \delta$, equal to that of the assumed external oscillation. Detuning δ was set equal to $2\pi \cdot 0.0051\,\text{GHz}$. An initial $R_x\left(\frac{\pi}{2}\right)$ gate allowed the qubit state to reasonably achieve the equator of the Bloch sphere, where qubit can have its free evolution, that was in turn described by a $R_x(0)$. This gate with null amplitude allowed, according to Equation 3.34, the simulation of the desired evolution in terms of an effective $R_z(\theta)$, with an angular frequency equal to the Rabi frequency $\sqrt{0^2 + [\omega_0 - (\omega_0 - \delta)]^2} = \delta$. MATLAB fitting provided an value for T_2 equal to 679.98 ns, absolutely compatible with the expected one. Both results proved that the simulated transients are coherent with the expected ones.

Simulation of complex quantum circuits

Table 4.1: Relaxation and decoherence time constants of two the NMR quantum computers employed in the simulations for validating the proposed model.

Dummy quantum computer

	`q[0]`	`q[1]`	`q[2]`	`q[3]`
T_1	2.60 s	1.40 s	2.40 s	1.70 s
T_2	0.15 s	0.21 s	0.05 s	0.12 s

This section aims at validating the proposed decoherence and relaxation model in a more applicative approach, *i.e.* through the comparison of the eigenstates probability distributions of different quantum circuits, obtained with the compact model and with a consolidated decoherence model: the thermal relaxation error employable by Qiskit's `QasmSimulator`. This error estimate is based on Kraus operators, computed — according to Qiskit [20] documentation — with the time constants T_1 and T_2 of each qubit, quantum gate duration τ, and excited-state thermal population p_1. The time constants must clearly be the same in both simulations and p_1 is set to 0, to mimic the relaxation transient described in subsection 2.3.4. Thermal relaxation error is built with quantum gates:

$$\begin{aligned} U_1(\phi) &= R_z(\phi)\,, \\ U_2(\phi,\lambda) &= R_z(\phi) R_y\left(\frac{\pi}{2}\right) R_z(\lambda)\,, \\ U_3(\theta,\phi,\lambda) &= R_z(\phi) R_x\left(-\frac{\pi}{2}\right) R_z(\theta) R_x\left(\frac{\pi}{2}\right) R_z(\lambda)\,, \end{aligned} \tag{4.20}$$

and all R_z gates are virtual, to guarantee that the latency of the quantum circuit is the same in all considered cases. Hence, the duration of a U_2 gate is set equal to

that of a $R_{\{x,y\}}\left(\frac{\pi}{2}\right)$, while that of a U_3 gate is the double of the previous one. In this simulation, a *dummy* four-qubit quantum computer is employed. This device involves only relaxation and decoherence as non-idealities and their corresponding $T_{1,2}$ timescales are different reported in Table 4.1. On the other hand the same duration for $R_{\{x,y\}}\left(\frac{\pi}{2}\right)$ gates (185 µs) is employed for all qubits
Table 4.2 reports some comparison results, in terms of highest probability and KL-divergence. It is important to precise that, apart from ghz_4 (whose expected final state is $\frac{|0000\rangle+|1111\rangle}{\sqrt{2}}$), all the simulated circuits are expected to have a unique eigenstate with non-null probability in an ideal case. This choice is due to the fact that these circuits permit to simplify the comparison of the probability distributions obtained with the two simulations.
The most probable eigenstates are the same in all cases. Moreover, the probability distributions corresponding to two-qubit circuits, even if simulated on a four-qubit device, are very close to each other, while the KL-divergence is slightly higher in those involving three or four qubits, as the reversible full-adder and the Fredkin gate. However, it is possible to observe that in these cases the maximum probability with the proposed model is always lower than that obtained with the Qiskit's thermal relaxation model, thus implying that the proposed model permits to reasonably execute simulations in a worse-case scenario.
In conclusion, it is possible to say that the proposed compact model does not allow the compact-model-based simulator users to underestimate the effects of relaxation and decoherence, when it is compared to a standard representation.

Table 4.2: Comparison of the most probable outcomes on a cytosine quantum computer, according to the thermal relaxation model of Qiskit and the simulation strategy based on the proposed relaxation model.

Quantum Circuit	Thermal relaxation state, prob.	Proposed model, prob. KL-Divergence
cnot_0010	$\|0011\rangle$, $\mathbb{P} = 0.9946$	$\|0011\rangle$, $\mathbb{P} = 0.9952$ KLD $= 0.00086$
grover_0010	$\|0010\rangle$, $\mathbb{P} = 0.9846$	$\|0010\rangle$, $\mathbb{P} = 0.9793$ KLD $= 0.00158$
swap_03	$\|1000\rangle$, $\mathbb{P} = 0.9878$	$\|0010\rangle$, $\mathbb{P} = 0.9846$ KLD $= 0.00181$
toffoli_0110	$\|1110\rangle$, $\mathbb{P} = 0.8893$	$\|1110\rangle$, $\mathbb{P} = 0.8641$ KLD $= 0.02404$
full_adder_0011	$\|1100\rangle$, $\mathbb{P} = 0.8431$	$\|1110\rangle$, $\mathbb{P} = 0.7247$ KLD $= 0.08194$
fredkin_0110	$\|1010\rangle$, $\mathbb{P} = 0.9122$	$\|1110\rangle$, $\mathbb{P} = 0.8506$ KLD $= 0.05188$
ghz_4	$\|0000\rangle$, $\mathbb{P} = 0.4889$ $\|1111\rangle$, $\mathbb{P} = 0.4906$	$\|0000\rangle$, $\mathbb{P} = 0.4879$ $\|1111\rangle$, $\mathbb{P} = 0.4858$ KLD $= 0.01500$

Chapter 5

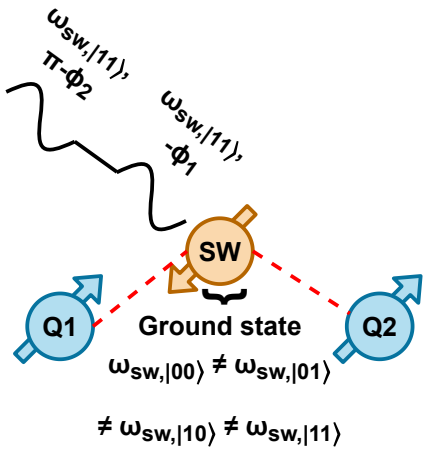
Molecular technologies for quantum computing

Molecules have been proposed as candidates for the fabrication of a quantum computer since the beginning of experimental quantum computing. Information is encoded onto spins of electrons or nuclei and the implementation of gates depends on the capabilities to manipulate single spins and to force interactions between couples of these.
Molecular quantum computers are undoubtedly milestones of quantum information processing; suffice to say that in 1998 one of these has been employed for factoring 15 with the Shor's algorithm [64].
Molecular quantum technologies analyzed in this thesis can be roughly divided into two main families: **switch-based** and **Nuclear Magnetic Resonance** (NMR). Even though more details about the physics of these devices and the development of their compact models are reported in the following, it is already possible to do a functional comparison between them.
Both families exploit a magnetic static field to ensure energy splitting between spin states, so to encode qubit, as already described in section 3.3.3. The main difference between switch-based and NMR families of molecular quantum computers concerns two-qubit gate implementation. In switch-based molecules, interaction between a pair of qubits is *indirect*, *i.e.* it depends on the interaction of each qubit with an intermediate spin called switch, while in NMR spins are directly coupled with the so-called J-coupling. Another common feature of these devices is that they do not involve many qubits (at most four in those simulated, at most twelve in experimental NMR).
In this context it is important to clarify the reasons behind the choice of these devices. As already introduced in subsection 1.5.1, these systems are not extremely complex, in terms of mechanisms for implementing quantum gates and of non-ideality phenomena, so this *simplicity* was exploited to build the first compact models and to define the general methodology for building and simulating compact

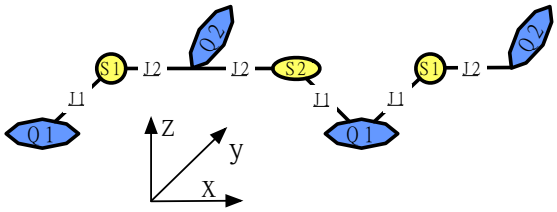
models described in section 4.2 and section 4.3. Moreover, these are expected to be a starting point for the development of more complex models, also associated with technologies different from the currently examined. For example, all the solid-state technologies (semiconductors, diamond, *etc.*) — encoding quantum information onto spins and potentially integrable on systems with a classical processing part — can be mentioned again in this context. Moreover, there are some interesting physical features. For example, NMR operates at room temperature, differently to most of other quantum technologies (such as superconducting qubits, working at $\sim 15\,\mathrm{mK}$), while in switch-based devices the number of qubits can be theoretically scaled with supramolecular chemistry techniques, thus overcoming a limitation of NMR, whose qubits control is in general quite sophisticated.
The chapter is organized in two main sections and another one with concluding remarks. The first one describes switch-based quantum computers, while the second one is related to NMR. In both cases, after an initial description of the characteristic Hamiltonian and of functioning of the technology, the corresponding compact model is presented and the obtained results, in terms of comparison of the outcomes given by this model and the canonical description of the quantum computer, are reported.

5.1 Switch-based molecules

Switch-based molecules encode quantum information onto both electron or nuclear spins. Their main characteristic is the presence of spins called **switches**, which are interspersed between a pair of spin qubits. Supramolecular chemistry techniques would permit with some molecules to create linear chains where qubits and switches are alternated, according to the scheme in Figure 5.1b.



(a) Conceptual mechanism for implementing the two-qubit gate.



(b) Qubit-switch chain, potentially obtainable with supramolecular chemistry techniques.

Figure 5.1: Overview of switch-based molecules

The resonance frequency of a switch depends on the state of its adjacent qubits and this property can be exploited for the implementation of the Cϕ gate. For the analysis of these devices, it must be assumed that **qubits and switches are never entangled**, so a quantum system made of N qubits and $N-1$ switches can be described by two independent state vectors with 2^N and 2^{N-1} probability amplitudes respectively:

$$|\psi\rangle_{\text{system}} = |\psi\rangle_{\text{qubits}} \otimes \left(|S_{z_{\text{sw}}}\rangle = \left|-\frac{1}{2}\right\rangle\right)^{\otimes N-1}, \tag{5.1}$$

where switches are assumed to be associated with electron spins. The **computational basis** is defined for states where each switch is *open*, *i.e.* in its ground state ($|S_{z_{\text{sw}}}\rangle = \left|-\frac{1}{2}\right\rangle$). Basis $|0\rangle$ and $|1\rangle$ is encoded onto the spin qubit eigenstates with lower and higher energy respectively. The *open switch* states are exploited for the implementation of single-qubit gates: magnetic anisotropy is responsible of the different qubits resonance frequencies, thus allowing their independent rotation.
For what concerns two-qubit molecules, the switch transition angular frequency is different in all possible cases:

$$\begin{aligned}
&\Delta\omega_{|00\rangle\otimes(|S_{z_{\text{sw}}}\rangle=\left|-\frac{1}{2}\right\rangle)\rightarrow|00\rangle\otimes(|S_{z_{\text{sw}}}\rangle=\left|+\frac{1}{2}\right\rangle)} \neq \\
&\Delta\omega_{|01\rangle\otimes(|S_{z_{\text{sw}}}\rangle=\left|-\frac{1}{2}\right\rangle)\rightarrow|01\rangle\otimes(|S_{z_{\text{sw}}}\rangle=\left|+\frac{1}{2}\right\rangle)} \neq \\
&\Delta\omega_{|10\rangle\otimes(|S_{z_{\text{sw}}}\rangle=\left|-\frac{1}{2}\right\rangle)\rightarrow|10\rangle\otimes(|S_{z_{\text{sw}}}\rangle=\left|+\frac{1}{2}\right\rangle)} \neq \\
&\Delta\omega_{|11\rangle\otimes(|S_{z_{\text{sw}}}\rangle=\left|-\frac{1}{2}\right\rangle)\rightarrow|11\rangle\otimes(|S_{z_{\text{sw}}}\rangle=\left|+\frac{1}{2}\right\rangle)}.
\end{aligned} \tag{5.2}$$

For a molecule with more than two qubits, it has been proved that the transition frequency of each spin is only affected by its adjacent spin qubits and not by the

others. This phenomenon is exploited for the implementation of the two-qubit Cϕ gate (Equation 3.12), whose unitary matrix is reported again in the following:

$$\mathrm{C}\phi = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \mathrm{e}^{\mathrm{i}\phi} \end{bmatrix} .$$

In order to easily derive the unitary evolution matrix of the Cϕ gate in these molecules, it is convenient to analyze the evolution of each basis state ($|00\rangle, \cdots, |11\rangle$) separately, reminding that each column of the unitary matrix corresponds to the evolution of one basis vector (end of subsection 3.1.1). In the case of the Cϕ unitary matrix, the leftmost column refers to the evolution of $|00\rangle = [1,0,0,0]^T$, while the rightmost refers to $|11\rangle = [0,0,0,1]^T$. In these devices, the Cϕ gate is realized by a couple of oscillating π-pulses resonant with the transition $|11\rangle \otimes (|S_{z_{\mathrm{sw}}}\rangle = \left|-\frac{1}{2}\right\rangle) \Leftrightarrow |11\rangle \otimes (|S_{z_{\mathrm{sw}}}\rangle = \left|+\frac{1}{2}\right\rangle)$. When the spin qubits are both $|1\rangle$, the spin switch is rotated and it accumulates two phases contributions, one for each rotation, thus obtaining a total phase shift $\mathrm{e}^{\mathrm{i}\phi}$ depending on the difference of the phases of the two consecutive oscillations (for the further proof, giving a look to Equation 3.34 and Equation 3.36 is suggested):

$$\begin{aligned} & |11\rangle \otimes \left(|S_{z_{\mathrm{sw}}}\rangle = \left|-\frac{1}{2}\right\rangle\right) \\ & \xrightarrow[\omega=\Delta\omega_{|11\rangle},\phi=-\phi_1]{\text{First } \pi \text{ pulse}} |11\rangle \otimes \left[|S_{z_{\mathrm{sw}}}\rangle = \mathrm{e}^{-\mathrm{i}\left(\frac{\pi}{2}+\phi_1\right)} \left|+\frac{1}{2}\right\rangle\right] \\ & \xrightarrow[\omega=\Delta\omega_{|11\rangle},\phi=\pi-\phi_2]{\text{Second } \pi \text{ pulse}} |11\rangle \otimes \left[|S_{z_{\mathrm{sw}}}\rangle = \mathrm{e}^{\mathrm{i}(\phi_2-\phi_1)} \left|-\frac{1}{2}\right\rangle\right] . \end{aligned} \tag{5.3}$$

Then, reminding that $A \otimes kB = kA \otimes B$, where A and B are two vectors and k a scalar term, it is possible to *pass* the accumulated phase contribution $\mathrm{e}^{\mathrm{i}(\phi_2-\phi_1)}$ to the qubits state vector:

$$\begin{aligned} & |11\rangle \otimes \left(|S_{z_{\mathrm{sw}}}\rangle = \mathrm{e}^{\mathrm{i}(\phi_2-\phi_1)} \left|-\frac{1}{2}\right\rangle\right) = \\ & \mathrm{e}^{\mathrm{i}(\phi_2-\phi_1)} |11\rangle \otimes \left(|S_{z_{\mathrm{sw}}}\rangle = \left|-\frac{1}{2}\right\rangle\right) . \end{aligned} \tag{5.4}$$

If qubits are not both equal to $|1\rangle$, the off-resonance is sufficiently high (see inequalities in Equation 5.1) to not rotate the spin switch, so the evolution in Equation 5.3 does not take place and no significant phase amount is accumulated. In conclusion, the global unitary evolution on the qubit state vector is the same of a Cϕ gate (with $\phi = \phi_2 - \phi_1$). According to [65], a couple of pulses — with carrier frequency resonant with the transition $|11\rangle \otimes (|S_{z_{\mathrm{sw}}}\rangle = \left|-\frac{1}{2}\right\rangle) \Leftrightarrow |11\rangle \otimes (|S_{z_{\mathrm{sw}}}\rangle = \left|+\frac{1}{2}\right\rangle)$ and modulated by a Gaussian function with peak value of 50 G and 99% of area in

about 6 ns — permits to obtain the evolution of a CZ gate, *i.e.* a Cϕ with $\phi = \pi$. Two switch-based molecules have been taken into account: **Cr_7Ni supramolecular complexes** [65, 66] and **vanadyl moieties** [67]. A comparison between the two technologies is reported in Table 5.1. The main differences between them, in terms of quantum gates frequencies and coherence timescales, are related to the type of spins exploited for encoding quantum information. In fact, vanadyl moieties show both nuclear and electron spins, while in Cr_7Ni rings only electron spins are employed.

Table 5.1: Comparison of the switch-based molecules examined in this thesis.

	Vanadyl	**Cr_7Ni rings**
Spin types	Nuclear (qubit) Electron (switch)	Electron
Quantum gates frequencies	RF ($R_k(\theta)$) MW (CZ)	MW-IR
Coherence timescales	$\gg 1\,\mu s$ (qubit) $1\,\mu s$ (switch)	$\gtrsim 1\,\mu s$ (qubit) $2\,\mu s$ (switch)

RF = Radio Frequency, MW = Microwave, IR = Infrared

5.1.1 Vanadyl moieties

The vanadyl architecture is based on two vanadium ions with both electronic and nuclear spins (spin-$\frac{7}{2}$). Ions are not directly connected by a bridging ligand, thus implying that the dipolar through-space coupling is expected to be the most relevant magnetic interaction [67].

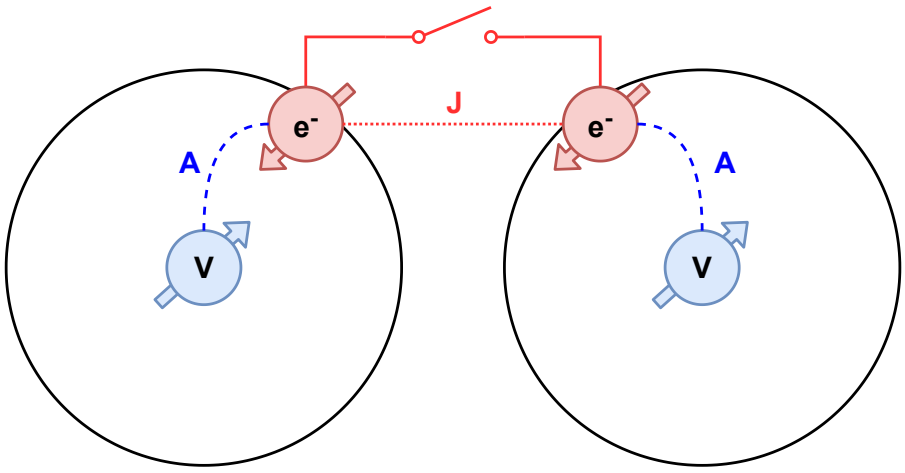


Figure 5.2: Conceptual scheme of the quantum computer based on vanadyl moieties.

The system, whose conceptual scheme is reported in Figure 5.2, is described by the following Hamiltonian:

$$\frac{\mathcal{H}_{\text{vanadyl}}}{\hbar} = \sum_{i=1}^{2} \sigma_{z_{\frac{7}{2},i}} A_{z_i} \sigma_{z_i} + \mu_B \frac{1}{2} \sum_{i=1}^{2} g_{z_i} B_0 \sigma_{z_i} + \frac{J}{2} \sigma_{z_1} \sigma_{z_2} \,, \tag{5.5}$$

where $\sigma_{z_{\frac{7}{2},i}} = \text{diag}\left(\left[+\frac{7}{2}, +\frac{5}{2}, \cdots, -\frac{5}{2}, -\frac{7}{2}\right]\right)$ can be seen as the Pauli matrix σ_z associated with the i^{th} nuclear spin-$\frac{7}{2}$ encoding qubit, σ_{z_i} is the *canonical* Pauli matrix associated with the i^{th} electron spin, J describes the interaction between electron spins, while A is the hyperfine interaction between nuclear and electronic spins. All values (in frequency units) are available in [67]. For $B > 0.3\,\text{T}$ spin states are practically factorized because of the large difference in their Zeeman energies (so the rotation of each nuclear spin does not depend on the state of the other), while the rotation of coupled electron spins depends on the state of both nuclei, because of the hyperfine coupling [67]. The eigenstates $\left|+\frac{7}{2}\right\rangle$ and $\left|+\frac{5}{2}\right\rangle$ of nuclear spins encode qubits, while the electron ones are exploited to implement the switch for the CZ gate.
The Zeeman diagram, showing the dependence of spin energy on the magnetic static field, is plotted in Figure 5.3a. Eigenstates with excited switch have energy constant with the magnetic static field. Moreover, it is possible to ascertain from Figure 5.3b that the switch resonance frequency when both qubits are equal to $|0\rangle$ is definitively different from the others, thus making it the most suitable eigenstate for the implementation of the two-qubit gate. If the frequency of the excitation is set to $\Delta\omega_{|00\rangle\otimes(|S_{z_{\text{sw}}}\rangle=\left|-\frac{1}{2}\right\rangle)\rightarrow|00\rangle\otimes(|S_{z_{\text{sw}}}\rangle=\left|+\frac{1}{2}\right\rangle)}$, the global unitary evolution will be $\text{diag}([\text{e}^{\text{i}\phi}, 1, 1, 1])$ and the effective quantum gate is that in Figure 5.4.

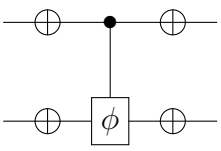
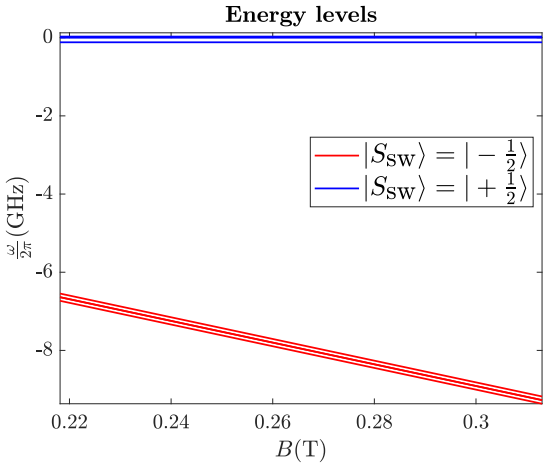
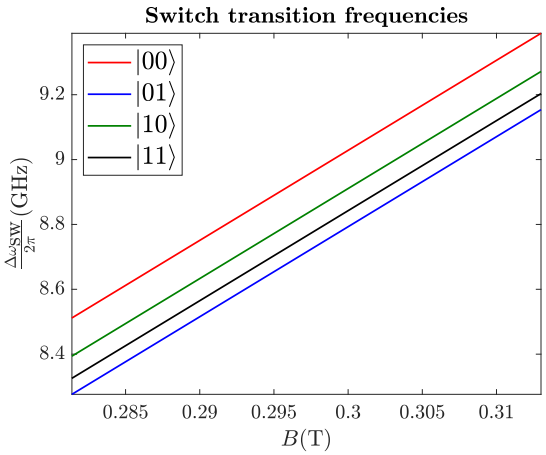


Figure 5.4: Circuit representation of the effective Cϕ of vanadyl moieties.



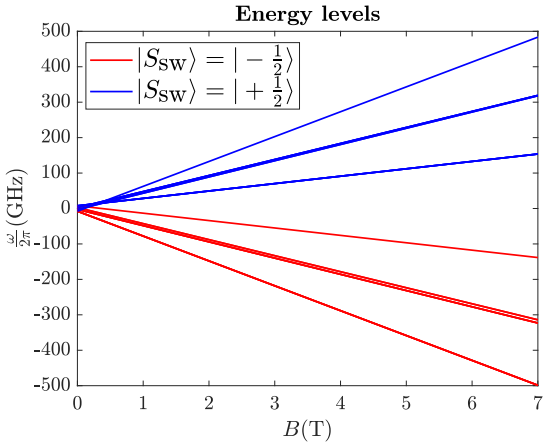
(a) Computational basis is encoded onto the eigenstates related to red lines (*open switch*).



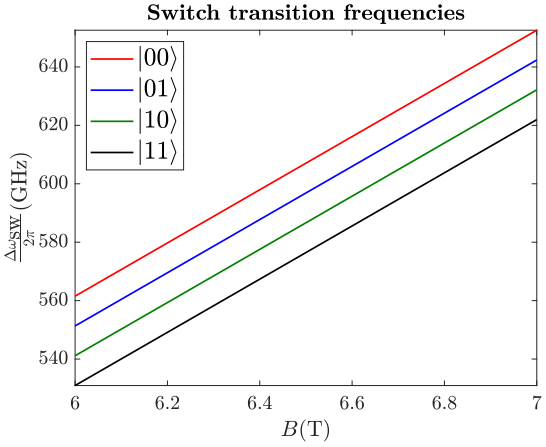
(b) For a given magnetic static field, spin switch resonance frequencies are not equally spaced.

Figure 5.3: Zeeman diagram for a vanadyl system involving two qubits and one switch.

5.1.2 Cr_7Ni supramolecular complexes



(a) Computational basis is encoded onto the eigenstates related to red lines (*open switch*).



(b) For a given magnetic static field, different resonance frequencies are equally spaced.

Figure 5.5: Zeeman diagram for a Cr_7Ni system involving two qubits and one switch.

These molecules, proposed in [65, 66], are based on metal-organic Cr_7Ni rings, each characterized by a spin-$\frac{1}{2}$ qubit. Assemblies of low-spin coupling rings have been

made through switchable links, each of which is based on a Co^{2+} ion and is also characterized by a spin-$\frac{1}{2}$ at low temperature ($T = 5\,\mathrm{K}$). As already written in the introduction of this chapter, supramolecular chemistry could be exploited to link individual components with different spatial configurations, thus influencing the properties of the resulting supra-molecules and potentially creating qubit-switch chains, as that reported in Figure 5.1b. Magnetic anisotropy is responsible of the switches transition frequency shift, which depends — for a given switch — exclusively on the states of its adjacent spin qubits.
The generic Hamiltonian of Cr_7Ni supramolecular complexes, involving N qubits and $N-1$ switches, can be obtained from the analysis of two and three-qubit molecules [65, 66]. This is written according to the schematic representation - to be read from left to right - of the supramolecular complex in Figure 5.1b:

$$\begin{aligned}\frac{\mathcal{H}}{\hbar} =& \mu_\mathrm{B} \sum_{i=1}^{N} \left(\mathbf{Q}_i \cdot \underline{g}_{i_\mathrm{qubit}} \cdot \mathbf{B}\right) + \mu_\mathrm{B} \sum_{i=1}^{N-1} \left(\mathbf{S}_i \cdot \underline{g}_{i_\mathrm{sw}} \cdot \mathbf{B}\right) \\ &+ \sum_{i=1}^{N-1} \left(\mathbf{Q}_i \cdot \underline{J}_{i_\mathrm{qubit},i_\mathrm{sw}} \cdot \mathbf{S}_i\right) + \sum_{i=1}^{N-1} \left(\mathbf{S}_i \cdot \underline{J}_{i_\mathrm{sw},i+1_\mathrm{qubit}} \cdot \mathbf{Q}_{i+1}\right) ,\end{aligned} \tag{5.6}$$

where μ_B is the Bohr's magneton (Equation 3.48), $\mathbf{S}_i$ and $\mathbf{Q}_i$ are — apart from the reduced Planck constant $\hbar$ — the spin-$\frac{1}{2}$ operators vectors (*e.g.* $\mathbf{S} = \mathbf{S}_x\hat{x} + \mathbf{S}_y\hat{y} + \mathbf{S}_z\hat{z} = \frac{1}{2}\sigma_x\hat{x} + \frac{1}{2}\sigma_y\hat{y} + \frac{1}{2}\sigma_z\hat{z} = \frac{1}{2}\overrightarrow{\sigma}$) of the i^th Co^{2+} and the i^th Cr_7Ni spins respectively, $\underline{g} = \mathrm{diag}\left([g_x, g_y, g_z]\right)$ is the spectroscopic tensor describing the interaction of the magnetic moments with an external magnetic field $\mathbf{B}$, and $\underline{J} = \mathrm{diag}\left([J_x, J_y, J_z]\right)$ is the anisotropic exchange tensor describing the interaction between adjacent spins. Different $\underline{J}$ and $\underline{g}$ interactions are responsible of the different easy axes of spin qubits sharing the same switch. It is possible to ascertain from Figure 5.1b that the molecule has a periodical structure, in which all the operators assume regularly only two values: $\mathbf{Q}_i = \mathbf{Q}_{1(2)}$, $\mathbf{S}_i = \mathbf{S}_{1(2)}$, $\underline{J}_{i_\mathrm{qubit},i_\mathrm{sw}} = \underline{J}_{1(2)}$ and $\underline{J}_{i_\mathrm{sw},i+1_\mathrm{qubit}} = \underline{J}_{2(1)}$ when i is odd(even).
The parameters of the molecule employed in simulations are available in [65, 66] and, with the same, it is possible to obtain the Zeeman diagram in Figure 5.5a, with magnetic static field applied along the $\hat{z}$ axis only ($\mathbf{B} = B\hat{z}$). It is possible to ascertain that, for a standard operating regime of Cr_7Ni molecules, $B \gtrsim 3\,\mathrm{T}$, frequencies are in the order of magnitude of 100 GHz, definitively higher than those of vanadyl moieties.
Apart from decoherence, the main source of non-ideality is the residual interaction between pairs of qubits when the intermediate switch is in the off state ($|S_{z_\mathrm{sw}}\rangle = \left|-\frac{1}{2}\right\rangle$). In fact, a residual interaction depending on $\frac{J_{1,2}}{2\pi} \sim 1\,\mathrm{GHz}$ [65] is always present and it does not allow a perfect qubits insulation. Assuming that a static field B is applied on the $\hat{z}$ axis and $g\mu_B B \gg \underline{J}_{1,2}$, the qubit-qubit interaction can be described by the following Hamiltonian, which is obtained by assuming that spin-spin interactions $\underline{J}_{1,2}$ are second-order perturbations of the Hamiltonian (5.6)

for two qubits, when $|S_{z_{\text{sw}}}\rangle = \left|-\frac{1}{2}\right\rangle$ [66]:

$$\begin{aligned}\frac{\mathcal{H}_{\text{eff}}}{\hbar} =& \Gamma_{xx}\mathbf{Q}_{1x}\mathbf{Q}_{2x} + \Gamma_{yy}\mathbf{Q}_{1y}\mathbf{Q}_{2y} \\ &+ \lambda_1\mathbf{Q}_{1z} + \lambda_2\mathbf{Q}_{2z} + C\,,\end{aligned} \tag{5.7}$$

where $\mathbf{Q}_{1k=\{x,y,z\}} = \mathbf{Q}_{k=\{x,y,z\}} \otimes I$ and $\mathbf{Q}_{2k=\{x,y,z\}} = I \otimes \mathbf{Q}_{k=\{x,y,z\}}$ are the spin operators of the two qubits and the scalar terms multiplying the operators are:

$$\begin{aligned}\Gamma_{xx} &= -\frac{g_{1z}J_{1y}J_{2x}\left(g_{\text{Coz}}^2 - g_{2z}^2\right) + J_{1x}\left[J_{2x}g_{\text{Coz}}\left(g_{\text{Coz}}^2 - g_{1z}^2 - g_{2z}^2\right) + g_{2z}J_{2y}\left(g_{\text{Coz}}^2 - g_{1z}^2\right)\right]}{4\mu_B B\left(g_{\text{Coz}}^2 - g_{1z}^2\right)\left(g_{\text{Coz}}^2 - g_{2z}^2\right)}\,, \\ \Gamma_{yy} &= -\frac{g_{2z}J_{1y}J_{2x}\left(g_{\text{Coz}}^2 - g_{1z}^2\right) + J_{2y}\left[J_{1y}g_{\text{Coz}}\left(g_{\text{Coz}}^2 - g_{1z}^2 - g_{2z}^2\right) + g_{1z}J_{1x}\left(g_{\text{Coz}}^2 - g_{2z}^2\right)\right]}{4\mu_B B\left(g_{\text{Coz}}^2 - g_{1z}^2\right)\left(g_{\text{Coz}}^2 - g_{2z}^2\right)}\,, \\ \lambda_1 &= -\frac{J_{1z}}{2} - \frac{2g_{\text{Coz}}J_{1x}J_{1y} + g_{1z}\left(J_{1x}^2 + J_{1y}^2\right)}{8\mu_B B\left(g_{\text{Coz}}^2 - g_{1z}^2\right)}\,, \\ \lambda_2 &= -\frac{J_{2z}}{2} - \frac{2g_{\text{Coz}}J_{2x}J_{2y} + g_{2z}\left(J_{2x}^2 + J_{2y}^2\right)}{8\mu_B B\left(g_{\text{Coz}}^2 - g_{2z}^2\right)}\,.\end{aligned} \tag{5.8}$$

Apart from a constant term, Γ_{xx} and Γ_{yy} interactions induce an unwanted evolution when the switch is turned off. The unwanted evolution can be controlled by the size of the applied field or by the size of the Co-ring exchange, to obtain high-fidelity single qubit gates. Also in the case of non-ideality phenomena, the parameters are available in [65, 66].

5.1.3 Compact model

The development of the compact model for switch-based molecules was gradual and started from the analysis of Cr_7Ni devices in [58]. In this article, a first model for the native quantum gates of this technological family was presented, which allowed a preliminary noiseless simulation of quantum circuits. Non-ideality phenomena were only considered in parallel to the simulation. In particular, the time evolution of the qubits fidelity depending on each phenomenon, which was considered independent from the others, was estimated, to understand the timescales on which non-idealities could be neglected. Latencies of different circuits were compared with these timescales, to evaluate which of them could be reasonably executed on a Cr_7Ni quantum computer. The first simulations taking into account relaxation and decoherence were presented in [59]. Reminding the contents of section 4.6, the decoherence model was that consolidated in the simulation infrastructure, while the relaxation one was still preliminary. In particular, this was based on the relaxation matrix (Equation 4.16) and did not employ the probability redistribution algorithm. The current models for relaxation and decoherence for all switch-based

molecules are those consolidated in the current version of the simulation infrastructure; and values for relaxation and decoherence time constants are available in [65, 67].
The mechanism for implementing quantum gates is the same for all switch-based molecules, independently on their nature, so all compact models have the same gates descriptions. The pre-computation of qubits resonance frequencies in the two technologies permitted to avoid the continuous management of the whole Hamiltonians in Equation 5.5 and Equation 5.6 and to exploit simpler operations, such as Equation 3.34. Values for nominal magnetic static fields are reported in Table 5.2. Angular frequencies — for $B \gtrsim 3\,\mathrm{T}$ and $B \gtrsim 0.3\,\mathrm{T}$ in Cr_7Ni and vanadyl cases respectively — can be obtained, with good approximation, through a simple proportion $\Delta\omega_{B\neq B_{\mathrm{nominal}}} = \frac{B}{B_{\mathrm{nominal}}}\Delta\omega_{B_{\mathrm{nominal}}}$.

Table 5.2: Resonance angular frequencies with the two technologies.

Resonance angular frequency	Cr_7Ni ($B = 7\,\mathrm{T}$)	Vanadyl ($B = 0.315\,\mathrm{T}$)
$\Delta\omega_{0_1}$	$2\pi \cdot 151.32\,\mathrm{GHz}$	$2\pi \cdot 93\,\mathrm{MHz}$
$\Delta\omega_{0_2}$	$2\pi \cdot 169.98\,\mathrm{GHz}$	$2\pi \cdot 93\,\mathrm{MHz}$
$\Delta\omega_{\lvert 00\rangle\otimes(\lvert S_{z_{\mathrm{sw}}}\rangle=\left\lvert -\frac{1}{2}\right\rangle)\rightarrow\lvert 00\rangle\otimes(\lvert S_{z_{\mathrm{sw}}}\rangle=\left\lvert +\frac{1}{2}\right\rangle)}$	$2\pi \cdot 561.53\,\mathrm{GHz}$	$2\pi \cdot 9.389\,\mathrm{GHz}$
$\Delta\omega_{\lvert 01\rangle\otimes(\lvert S_{z_{\mathrm{sw}}}\rangle=\left\lvert -\frac{1}{2}\right\rangle)\rightarrow\lvert 01\rangle\otimes(\lvert S_{z_{\mathrm{sw}}}\rangle=\left\lvert +\frac{1}{2}\right\rangle)}$	$2\pi \cdot 541.13\,\mathrm{GHz}$	$2\pi \cdot 9.154\,\mathrm{GHz}$
$\Delta\omega_{\lvert 10\rangle\otimes(\lvert S_{z_{\mathrm{sw}}}\rangle=\left\lvert -\frac{1}{2}\right\rangle)\rightarrow\lvert 10\rangle\otimes(\lvert S_{z_{\mathrm{sw}}}\rangle=\left\lvert +\frac{1}{2}\right\rangle)}$	$2\pi \cdot 551.33\,\mathrm{GHz}$	$2\pi \cdot 9.272\,\mathrm{GHz}$
$\Delta\omega_{\lvert 11\rangle\otimes(\lvert S_{z_{\mathrm{sw}}}\rangle=\left\lvert -\frac{1}{2}\right\rangle)\rightarrow\lvert 11\rangle\otimes(\lvert S_{z_{\mathrm{sw}}}\rangle=\left\lvert +\frac{1}{2}\right\rangle)}$	$2\pi \cdot 530.93\,\mathrm{GHz}$	$2\pi \cdot 9.203\,\mathrm{GHz}$

For what concerns single-qubit gates, these are described according to the Trotter–Suzuki-like scheme in Figure 4.5a. In particular, it is given by the product of two unitary matrices, independent on each other: one related to the pulse effect on qubits (Equation 3.34), the other associated with the residual interaction of qubits when spin switches are *open*, whose construction is detailed at the end of this chapter subsection. It is important to remind that the unitary evolution of untargeted qubits is the identity. This is due to the fact that compact models for these technologies were defined before that for off-resonance and integration was not done at the time of writing this thesis. In any case, a small effort, in terms of software development, is expected to be sufficient for involving the Fourier model in those of switch-based molecules.
The rotating frame describing single-qubit gates does not properly describe the unitary evolution of a spin in a strong anti-resonance regime, where $|\delta| \to \infty$. This could imply a bad estimation of the spin switch evolution under an external excitation not resonant with its transition frequency, thus making difficult the description of the native Cϕ gate. In [58] a first approximated method, looking for a compromise between the effective Cϕ unitary evolution of these devices and the

simple equations of the spin-$\frac{1}{2}$ dynamics in the rotating frame, was proposed. It was based on the description of the effect of the intermediate spin switch evolution on a couple of qubits as:

$$M = \mathrm{diag}\left([\mathrm{e}^{\mathrm{i}P_{00}\phi}, \mathrm{e}^{\mathrm{i}P_{01}\phi}, \mathrm{e}^{\mathrm{i}P_{10}\phi}, \mathrm{e}^{\mathrm{i}P_{11}\phi}]\right), \tag{5.9}$$

where P_{ij} is the transition probability $|ij\rangle\,(|S_{z_{\mathrm{sw}}}\rangle = \left|-\frac{1}{2}\right\rangle) \Leftrightarrow |ij\rangle\,(|S_{z_{\mathrm{sw}}}\rangle = \left|+\frac{1}{2}\right\rangle)$. The total C$\phi$ gate would be obtained by applying two pulses, with phases $-\phi_1$ and ϕ_2 respectively. Assuming that the two consecutive pulses are equal (with the exception of phase), P_{ij} does not change in sequence, so the gate can be described in terms of matrix product as

$$U_{\mathrm{C}\phi} = M_2 M_1 = \begin{bmatrix} \mathrm{e}^{\mathrm{i}P_{00}\Delta\phi} & 0 & 0 & 0 \\ 0 & \mathrm{e}^{\mathrm{i}P_{01}\Delta\phi} & 0 & 0 \\ 0 & 0 & \mathrm{e}^{\mathrm{i}P_{10}\Delta\phi} & 0 \\ 0 & 0 & 0 & \mathrm{e}^{\mathrm{i}P_{11}\Delta\phi} \end{bmatrix}, \tag{5.10}$$

with $\Delta\phi = \phi_2 - \phi_1$. The transition frequencies $|ij\rangle\,(|S_{\mathrm{sw}}\rangle = \left|-\frac{1}{2}\right\rangle) \Leftrightarrow |ij\rangle\,(|S_{\mathrm{sw}}\rangle = \left|+\frac{1}{2}\right\rangle)$ are sufficiently different for ensuring, on time scales of some nanosecond and for $\omega_1\tau = \pi$, switch transition probabilities $P_{11} \sim 1$ and $P_{ij}\phi \ll 1$ for $|ij\rangle \neq |11\rangle$, so each diagonal matrix can be approximated as:

$$\begin{aligned} M_k &= \mathrm{diag}\left([\mathrm{e}^{\mathrm{i}P_{00}\phi_k}, \mathrm{e}^{\mathrm{i}P_{01}\phi_k}, \mathrm{e}^{\mathrm{i}P_{10}\phi_k}, \mathrm{e}^{\mathrm{i}\phi_k}]\right) \\ &\approx \mathrm{diag}\left([1, 1, 1, \mathrm{e}^{\mathrm{i}\phi_k}]\right). \end{aligned} \tag{5.11}$$

Matrix $U_{\mathrm{C}\phi}$ would be approximately a Cϕ, where the target phase is properly the phase difference of the two pulses. The approximated model is assumed to be valid in a regime where the phase of the complex exponential can be neglected, thus $\mathrm{e}^{\mathrm{i}P\phi} \approx 1$. In particular, $P\phi \leq \frac{1}{10}$ has been considered a good approximation; since $\phi \leq \pi$ in all the cases of interest, the transition probability must be $P \leq \frac{1}{10\pi} \simeq 0.032$, which is negligible even in the canonical model. According to Equation 3.35, it is possible to compute a lower threshold for the approximation:

$$\max\{P\} = \left|\frac{\omega_1}{\Omega}\right|^2 = \frac{1}{1 + \left(\frac{\delta}{\omega_1}\right)^2} \leq \frac{1}{10\pi}, \tag{5.12}$$

and consequently $\left(\frac{\delta}{\omega_1}\right)^2 \geq 10\pi - 1 \simeq 10\pi$. Since π-pulses are employed for rotating the spin switch, the precession angular frequency is $\omega_1 = \frac{\pi}{\tau}$; finally, the following condition ensuring a good approximation can be obtained:

$$\frac{\delta\tau}{\pi} \geq \sqrt{10\pi} \Rightarrow \delta_f\tau \geq \sqrt{\frac{5\pi}{2}} \simeq 2.8\,, \tag{5.13}$$

where $\delta_f = \frac{\delta}{2\pi}$ is the frequency detuning.
The same model was also employed in [59], then it has been improved in order to get the description closer to the real dynamics, *i.e.* to consider the whole round-trip evolution of the spin switch (Equation 5.3 and Equation 5.4). For each eigenstate, an auxiliary spin vector was initialized to $|\psi\rangle_{\text{sw, aux}} = [1,0]^\dagger$, assuming that the spin switch state is always pure (thus not requiring a density matrix) and that evolution always starts with the switch in its ground state. Then, two unitary matrices U_1 and U_2 are computed according to Equation 3.34, to mimic the sequence of the two π-pulses, with the excitation frequency and phase of each external pulse set according to Equation 5.3. If $\mathbb{P}_{U_1}$ — *i.e.* the transition probability associated with U_1 and given by the magnitude square of the off-diagonal terms of that matrix — is higher than a given threshold (currently 0.98), the total evolution $U_2U_1 |\psi\rangle_{\text{sw, aux}}$ is computed and the accumulated phase will be the global one of the auxiliary spin vector at the end of the round trip to be *provided* to the basis state:

$$\begin{aligned} U_2U_1 \begin{bmatrix} 1 \\ 0 \end{bmatrix} &\simeq \mathrm{e}^{\mathrm{i}\phi} \begin{bmatrix} 1 \\ 0 \end{bmatrix} , \\ \mathrm{C}\phi\, |k\rangle &= \mathrm{e}^{\mathrm{i}\phi}\, |k\rangle . \end{aligned} \tag{5.14}$$

When the condition on $\mathbb{P}_{U_1}$ is not satisfied, the scalar value on the main diagonal associated with the eigenstate $|k\rangle$ will be equal to:

$$\mathrm{C}\phi\, |k\rangle = \mathrm{e}^{\mathrm{i}\mathbb{P}_{U_2}(\pi-\phi)} \mathrm{e}^{\mathrm{i}\mathbb{P}_{U_1}\pi}\, |k\rangle = \mathrm{e}^{\mathrm{i}[(\mathbb{P}_{U_1}+\mathbb{P}_{U_2})\pi - \mathbb{P}_{U_2}\phi]}\, |k\rangle , \tag{5.15}$$

with $\mathbb{P}_{U_2}$ transition probability of U_2. It is evident that lower are transition probabilities, closer to 1 is the complex exponential in Equation 5.15. The choice of involving the $\mathbb{P}_{U_2}$ is to *dirt* the accumulated phase of the target eigenstates, which is assumed to be slightly different from 0 with an off-resonant excitation, even if quite far from resonance condition. In any case, it would be possible, with a minimum coding effort, to simply use $\mathrm{e}^{\mathrm{i}\cdot 0} = 1$ with untargeted qubits.
With the available resonance frequencies in Table 5.2, the evolution of Equation 5.14 takes place only with the target eigenstate, which is usually $|k\rangle = |11\rangle$, while in the other three cases Equation 5.15 can be employed.
The main difference between Cr_7Ni and vanadyl models concerns the simulation of the qubit-qubit Unwanted Evolution (UE), which is typical of the first technology. This can be achieved by taking into account the unitary evolution of the effective interaction Hamiltonian $\mathcal{H}_{\text{eff}}$, removing the contributions due to fast rotations of single spins around $\hat{z}$ (*i.e.* those associated with $\lambda_{1,2}$), since they introduce a negligible phase shift. The unitary evolution of residual qubit-qubit interaction can be written as:

$$U_{\text{UE}}(t) = \mathrm{e}^{+\frac{\mathrm{i}}{\hbar}\mathcal{H}_{\text{eff}_{\text{diag}}}\tau} \mathrm{e}^{-\frac{\mathrm{i}}{\hbar}\mathcal{H}_{\text{eff}}\tau} , \tag{5.16}$$

where $\mathcal{H}_{\text{eff}_{\text{diag}}}$ contains the terms on the main diagonal of the effective Hamiltonian. Evolution in Equation 5.16, valid for a two-qubit system, is approximately of type:

$$U_{\text{UE}}(t) \approx \text{diag}([1, \mathrm{e}^{\theta_{\text{eff}_{\text{UE}}}}, \mathrm{e}^{-\theta_{\text{eff}_{\text{UE}}}}, 1]) , \tag{5.17}$$

where $\theta_{\text{eff}_{\text{UE}}}$ is a sort of effective angle of the unitary evolution. A generalized description for N qubits is not available in the state of the art. However, it can be observed in the Supplementary Figure 29 of [65] that the timescale Δt_{UE} for which fidelity is greater or equal 90% scales as $\frac{1}{N-1}$ with the number of qubits $N > 2$, *i.e.*

$$\Delta t_{\text{UE}}(N) = \Delta t_{\text{UE}}(N-1) \cdot \frac{N-1}{N} = \frac{\Delta t_{\text{UE}}(2)}{N-1}\,. \tag{5.18}$$

Moreover, it is declared that entangled states $\frac{|0101\cdots\rangle+|1010\cdots\rangle}{\sqrt{2}}$ are the most error-prone. For this reason, an empirical mathematical model for obtaining that fidelity reduction evolution for a generic N-qubit system has been investigated. First of all, $U_{\text{UE}}(t)$ is computed for a two-qubit molecule and $\theta_{\text{eff}_{\text{UE}}}$ is estimated; then this angle is multiplied by $N-1$, to ensure that the N-qubit system evolves $N-1$ times faster than the two-qubit one, thus implying the same reduction trend for fidelity. Finally, it has been found that the expected unitary matrix should be:

$$U_{\text{UE}_N}(t) = \mathrm{e}^{-\mathrm{i}(N-1)\theta_{\text{eff}_{\text{UE}}}\left\{\bigoplus_{k=1}^{2^N-2}\left[(-1)^{k-1}\sigma_z\sigma_z\right]\right\}}\,, \tag{5.19}$$

where $\bigoplus$ refers to the construction, through the direct sum, of a block diagonal matrix with elements $(-1)^{k-1}\sigma_z\sigma_z = \text{diag}([(-1)^{k-1},(-1)^k,(-1)^k,(-1)^{k-1}])$. This non-ideality is evaluated in the simulation at each iteration, in particular after the evolution of the qubits density matrix ρ due to a single-qubit gate with duration τ. In particular, $\theta_{\text{eff}_{\text{UE}}}$ is first computed according to Equation 5.17 (which is reminded to be valid for a two-qubit device), then the effective angle is exploited to compute, according to Equation 5.19, $U_{\text{UE}}(\tau)$, which is finally applied to ρ.

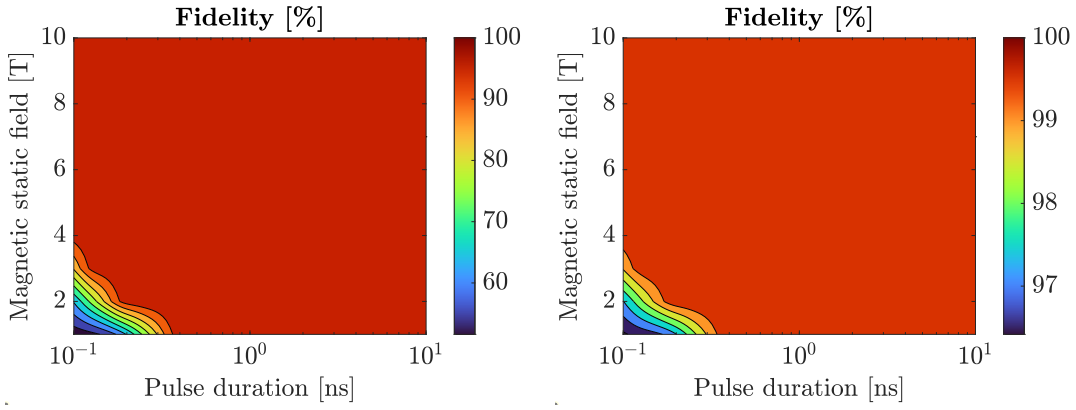
5.1.4 Modelling results

It is possible to say that the evaluation of the results for switch-based molecules permitted to build the complete simulation infrastructure described in this thesis. In fact, since these were the first analyzed devices, they did not only allow the gradual construction of a general compact modelling methodology, but also that of benchmarking models themselves.

The first results for switch-based technologies were reported in [58]. As already explained in subsection 5.1.3, simulations at the publication time of that article did not directly involve non-ideality phenomena, *i.e* they permitted only to understand how it was possible to implement a quantum circuit with the native gates of the technology. This inclusion was virtually done by estimating the maximum time duration, for each non-ideality phenomenon, ensuring that all the corresponding fidelities were higher than 0.9 and then choosing the lowest one as critical time duration for a given molecule. The critical timing thresholds resulted $\tau_M = 144\,\text{ns}$ and $\tau_M = 94\,\text{ns}$ and $\tau_M = 70\,\text{ns}$ for the two and three-qubit molecules respectively,

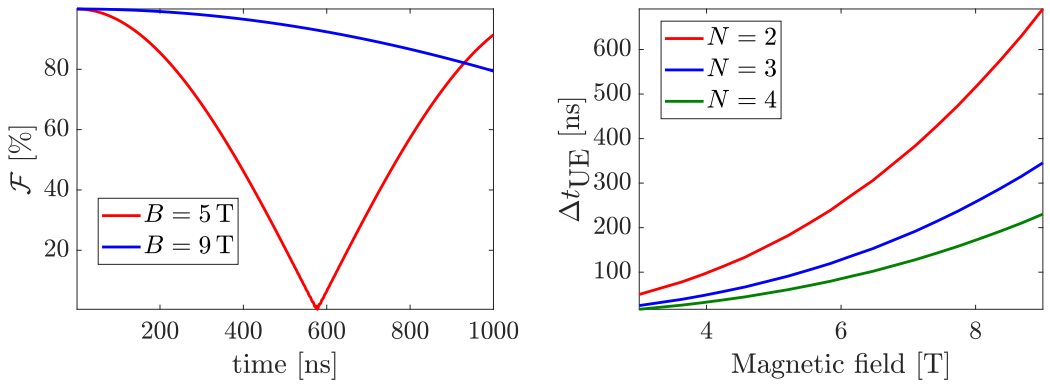
assuming that the magnetic static field $B = 7\,\mathrm{T}$. The comparison of the latency of each tested quantum circuit and the obtained time durations critical thresholds permitted to understand if the circuit could be reliably executed on the molecular quantum computer. The time interval for each gate/algorithm was computed assuming that all native gates were executed sequentially. Moreover, the pulse duration τ was fixed (while pulse amplitude changed proportionally to the gate rotation amount θ) and set to the highest value sufficient for ensuring that at least the 80% (so the majority) of the employed benchmark circuits satisfied the condition $\Delta t < \tau_M$ and that the Controlled-phase gate could be reasonably approximated by Equation 5.15. In particular, τ was equal to 5 ns, 1.7 ns and 0.47 ns for two, three and four-qubit nanomagnets respectively, while the duration of two-qubit gates was the double, since they require two consecutive pulses.
Results of [58] were undoubtedly germinal and presented some features to be necessarily improved, *e.g.* the fact that simulation was noiseless. However, an additional validation with the new version of the compact model, in the same simulation setup, can be done and is reported in the following. It is important to precise that the provided modelling results are those for Cr_7Ni, since they are also affected, different from vanadyl moieties, by interaction with open switch.
Considering the information previously reported, it is possible to ascertain that characterization of switch-based molecules involves multiple degrees of freedom, associated with different non-ideality phenomena. For this reason, the results are reported gradually, in sense that those associated with a specific phenomenon, making all the other effects negligible (*e.g.* $T_{\{1,2\}} \to \infty$), are initially reported, then the complete simulation results, including all the effects at the same time, are shown for some benchmark circuits. First of all, it is important to precise that in all tests relaxation could be totally neglected, since, according to [65], $T_1 = 17.73(33)\,\mu\mathrm{s}$, a value much greater than those associated with decoherence and qubit-qubit interaction.

The approximated description of the Cϕ gate, described in subsection 5.1.3, was validated and the results for two Cϕ gates are reported in Figure 5.6. In particular, the **fidelity for pure states** (Equation 2.81) — which compares the final state vectors of a two-qubit system initialized to $|\psi\rangle = \frac{|00\rangle+|01\rangle+|10\rangle+|11\rangle}{2}$ and subjected to the ideal and the approximated Cϕ gates — is shown. It is possible to ascertain that in a standard operating regime ($B \gtrsim 5\,\mathrm{T}$ and $\tau \gtrsim 1\,\mathrm{ns}$) the two states are susbtantially equal, thus proving that the approximated model can reasonably approximate the Cϕ-gate evolution. This is due to the fact that the inequality in Equation 5.13 is well satisfied. In particular, $\min(\delta_f) \simeq 14\,\mathrm{GHz}$ for $B = 5\,\mathrm{T}$, so for $\tau \geq 1\,\mathrm{ns}$ the condition $\delta_f \tau > 14 > 2.8$ is always satisfied. For what concerns the effect of the unwanted evolution U_{UE} due to qubits interaction with switch off, the time evolution of the fidelity for pure states was exploited for the evaluation of this phenomenon. In particular, the ideal and non-ideal qubits states $|\psi_I\rangle$ and $|\psi\rangle$, not affected and affected by U_{UE} respectively, were compared. First



(a) $\phi = \pi$ (CZ). (b) $\phi = \frac{\pi}{4}$.

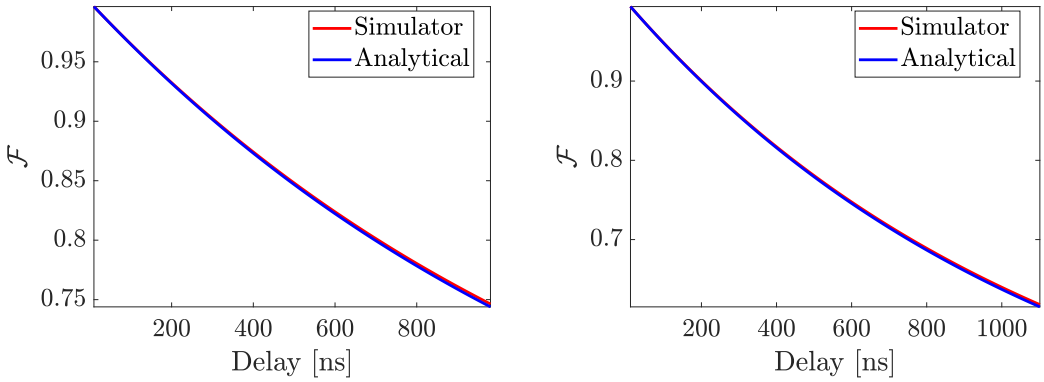
Figure 5.6: Fidelity of Cϕ gate for a two-qubit Cr_7Ni molecule with respect to different degrees of freedom.



(a) Fidelity time evolution of two two-qubit molecules with different magnetic static fields.

(b) Dependency of Δt_{UE} on of the applied magnetic field for two and three-qubit molecules.

Figure 5.7: Analysis of the residual qubit-qubit interaction on the fidelity of the system. The values of the J tensor are those in [65].



(a) Two qubits.

(b) Three qubits.

Figure 5.8: Fidelity reduction due to simultaneous decoherence of multiple qubits subjected to Markovian dynamics.

tests were done on the two-qubit state $|\psi(0)\rangle = \frac{|01\rangle+|10\rangle}{\sqrt{2}}$, which is one of the states mostly affected by this non-ideality phenomenon. The fidelity time evolutions for the two-qubit molecule, under two different magnetic static fields and assuming $T_{\{1,2\}} \to \infty$, is reported in Figure 5.7a. It is possible to ascertain that longer high-fidelity timescales can be achieved with a higher magnetic static field. The same observation can be done for Figure 5.7b, for what concerns the dependence of Δt_{UE}, *i.e.* the time interval after which $\mathcal{F} < 90\%$, on the applied magnetic field, for two, three and four-qubit devices.
Similarly to the residual qubit-qubit interaction, a fidelity due to decoherence was defined. In the supplementary information of [65] the analysis of the effects of decoherence for qubits, assumed to be uncorrelated for what concerns this non-ideality phenomenon, is done. In a worst case scenario, where qubit is on the equator of the Bloch sphere, fidelity evolves according to:

$$\mathcal{F} = \sqrt{\frac{1 + \mathrm{e}^{-\frac{t}{T_2}}}{2}}\,, \tag{5.20}$$

while in a system involving N qubits with equal T_2, fidelity evolves according to:

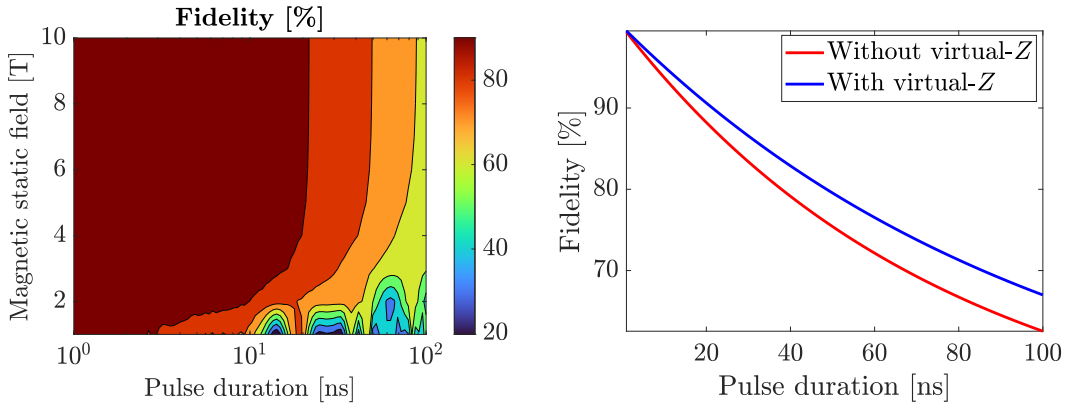
$$\mathcal{F} = \left(\frac{1 + \mathrm{e}^{-\frac{Nt}{T_2}}}{2}\right)^{\frac{N}{2}}\,, \tag{5.21}$$

where the exponent $-\frac{Nt}{T_2}$ is obtained by the fact that the decoherence time constant is divided by N. The dynamics reported in [65] is the same regime of the decoherence model described in section 4.6, for this reason a simulation of fidelity evolution under decoherence was done with the available simulator, to prove that the obtained fidelity corresponds to that obtained with Equation 5.21. In the validation of this error, the decoherence of each qubit in molecule was considered independent from the applied magnetic field and equal to $T_2 = 2 \cdot 683\,\text{ns}$ for $T = 2.8\,\text{K}$, according to the available information in [65]. Plots in Figure 5.8 show that the curves associated with Equation 5.21 and simulator are substantially overlapped for both two and three-qubit molecules, thus proving the correctness of simulation.
Finally, the simulation of quantum circuits, involving simultaneously all the phenomena, was done. First of all, the results — for some circuits simulated in [58] and others employed as benchmark in different parts of this thesis, with the same simulation setup of [58] — are provided in Table 5.3. It is possible to observe that in all cases fidelity of quantum circuits with latency comparable with the obtained τ_M in [58] is high (close to 99%), so it is possible to conclude that the critical timescales τ_M are excessively stringent. In addition to the simulations of Table 5.3, simulations involving multiple degrees of freedom — at both control/manipulation and circuit levels — were done with the available compact model. Figure 5.9 shows the fidelities of Grover's search of eigenstate $|10\rangle$ and of a reversible half-adder on

two-qubit and three-qubit molecules respectively. For each circuit, the left figure shows fidelity dependence on pulse duration and magnetic static field, while the fidelities of the same circuit — fixing $B = 7\,\mathrm{T}$ and tuning the pulse duration and the implementation strategy of $R_z(\theta)$ gates — are reported the right figure. It is possible to ascertain, as expected from the previous single-non-ideality simulations, that lower execution timescales — achievable with shorter pulses or with less gates, *i.e.* exploiting virtual-Z gates — and higher magnetic static fields — implying higher spin switch energies splitting and weaker open-switch interactions — permit to enhance the fidelity.
These simulations put clearly in evidence the capabilities of the compact-model-based simulator in estimating the reliability execution of a quantum circuit on a target device, according to different degrees of freedom.

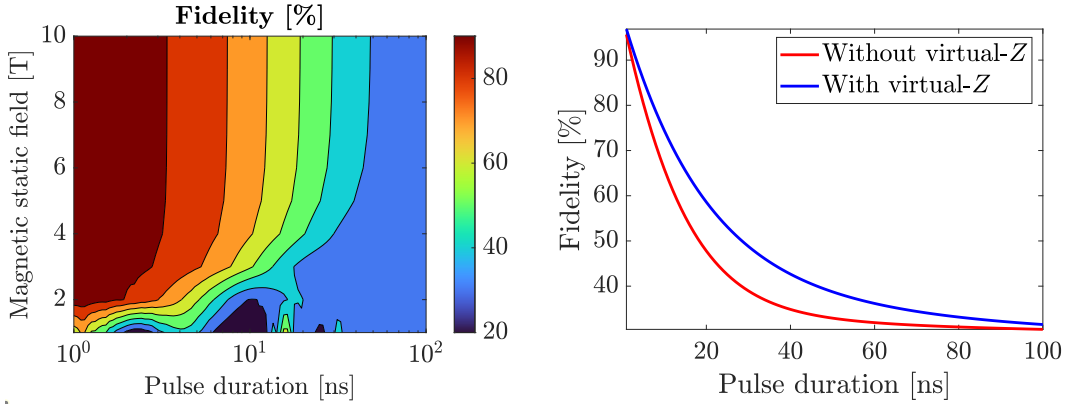
Table 5.3: Fidelities for some benchmark circuits, employing the same simulation setup of [58] (fixed pulse duration).

Circuit	Number of qubits, τ	Latency with parameters of simulations in [58]	Fidelity obtained with last version of simulator
bell	2, 5.00 ns	40.00 ns	0.9890
swap_10	2, 5.00 ns	95.00 ns	0.9875
qft2	2, 5.00 ns	120.00 ns	0.9833
grover_10	2, 5.00 ns	100.00 ns	0.9741
ghz	3, 1.70 ns	23.80 ns	0.9910
toffoli_011	3, 1.70 ns	173.40 ns	0.9609
half_adder	3, 1.70 ns	630.70 ns	0.8907
fredkin_0110	4, 0.47 ns	95.88 ns	0.9735
entanglement swapping	4, 0.47 ns	49.35 ns	0.9727



(a) Pulse duration and magnetostatic field (two-qubit grover_10).

(b) Virtual-Z gates exploitation (two-qubit grover_10).



(c) Pulse duration and magnetostatic field (three-qubit reversible half adder).

(d) Virtual-Z gates exploitation (three-qubit reversible half adder).

Figure 5.9: Fidelity of two quantum circuits on two and three-qubit Cr_7Ni molecules, with respect to different degrees of freedom.

5.2 Nuclear Magnetic Resonance

Nuclear Magnetic Resonance (NMR) quantum computing encodes qubit onto spin-$\frac{1}{2}$ of nuclei of atoms belonging to molecules, which are diamagnetic and usually liquid at room temperature.
Assuming a magnetic static field applied along the $\hat{z}$ axis, the Hamiltonian for a system of coupled nuclear spin-$\frac{1}{2}$ is:

$$\frac{\mathcal{H}_{\text{NMR}}}{\hbar} = -\frac{1}{2}\sum_i \omega_{0_i}\sigma_{z_i} + \frac{1}{4}\sum_{i<j} J_{ij}\sigma_{z_i}\sigma_{z_j}\,, \tag{5.22}$$

where ω_{0_i} is the transition frequency of the i^{th} spin-$\frac{1}{2}$ and J_{ij} is the electron-assisted through-bond coupling strength of i^{th} and j^{th} nuclear spins. The complete Hamiltonian should also involve interactions in the transverse plane of type $\sigma_{\{x,y\}}\sigma_{\{x,y\}}$, with strength equal to J_{ij}. However, they have been not reported, because in experimental scenarios the time duration of quantum gates unitary evolutions are sufficiently long for ensuring a reasonable application of the RWA, according to the analytical expression derived in [60].
From a practical point of view, the two terms of the Hamiltonian are associated with single-qubit and two-qubit gates respectively. In fact, the first one involves each spin independently on the others, while the other involves pairs of spins and is exploited for implementing the fundamental two-qubit gate of NMR quantum computing:

$$U_J(\tau) = \mathrm{e}^{-\mathrm{i}\frac{J}{4}\tau\sigma_z\sigma_z} = \mathrm{e}^{-\mathrm{i}\frac{\frac{J}{2}\tau}{2}\sigma_z\sigma_z} = R_{zz}\left(\frac{J\tau}{2}\right)\,. \tag{5.23}$$

Figure 5.10 shows the implementation of CZ gate with NMR native gates. The

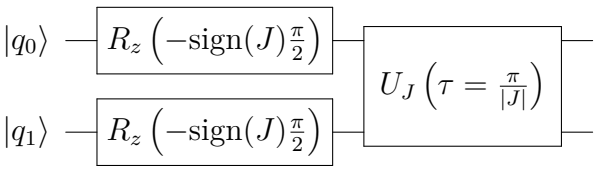


Figure 5.10: CZ gate with NMR native gates.

sign of J affects the $R_z\left(\frac{\pi}{2}\right)$ gates to be implemented on single qubits. The duration of the two-qubit evolution is reported assuming that J has angular frequency dimensions; if J-coupling was reported in frequency dimensions, $\tau = \frac{1}{2|J|}$. Moreover, if R_z gates can be implemented virtually, the gate duration of CZ corresponds to that of U_J.
A typical NMR Hamiltonian in weak coupling regime usually shows $\max\{|J_{ij}|\} \ll \min\{|\Delta\omega_0|\}$, where $\min\{\Delta\omega_0\}$ is the minimum difference between spin transition frequencies, so single-qubit gates can be applied on each spin with negligible effects of J-coupling interactions (at least in a first approximation) for gate timescales

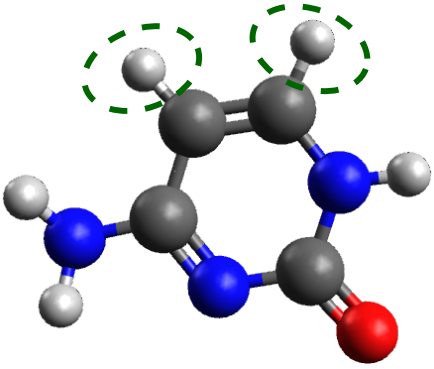
$\tau \ll \frac{2\pi}{\max(J_{i,j})}$.
NMR quantum computers can be divided in two main families, depending on the nuclei encoding quantum information. If at least two atoms of the same isotope are exploited for computation, device is named **homonuclear**, otherwise it is **heteronuclear**. Cytosine (Figure 5.11a) and crotonic acid (Figure 5.11b) — involving two ^{1}H and four ^{13}C nuclear qubits respectively — belong to the first family, while ^{13}C-labeled diethyl fluoromalonate — exploiting three nuclear spins ^{1}H, ^{13}C and ^{19}F for encoding information — belongs to the other. For what concerns homonuclear qubits, considering the contents of section 3.3.3, it may seem that identical isotopes are characterised by the same resonance frequency, thus not permitting to ensure their single addressability. However, in liquid-state NMR nuclei are affected by slightly different magnetic static local fields due to the fact that circulating currents in molecular electron clouds are induced by the external field. These currents in turn generate an additional magnetic field, thus forcing different effective Larmor frequencies. The isotropic chemical shielding σ permits to quantify the dependence of the local field on the applied field as $B_{\text{loc}} = (1-\sigma)B_0$ [68]. B_{loc} clearly affects the Larmor frequency $\omega_0 \triangleq |\gamma_n|(1-\sigma)B_0$ and this resonance frequency shift is known as **chemical shift**. From an experimental point of view, chemical shifts are measured with respect to a reference; in particular, denoting with subscripts *mol* and *ref* the molecule under experiment and the reference compound respectively, the chemical shift δ is defined as [68, 69]:

$$\delta \triangleq \frac{\omega_{\text{mol}} - \omega_{\text{ref}}}{\omega_{\text{ref}}} = \frac{|\gamma_n|(1-\sigma_{\text{mol}})B_0 - |\gamma_n|(1-\sigma_{\text{ref}})B_0}{|\gamma_n|(1-\sigma_{\text{ref}})B_0} = \frac{\sigma_{\text{ref}} - \sigma_{\text{mol}}}{1-\sigma_{\text{ref}}}\,. \tag{5.24}$$

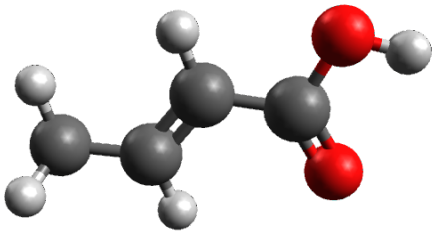
The reference frequencies are measured on particular compounds and are reported in the Spectral Database for Organic Compound SDBS [70]. Table 5.4, Table 5.5 and Table 5.6 report the characteristic parameters of the three molecules. The isotropic chemical shifts (in parts per million or in frequency measurement unit) are reported on the main diagonal, while the J-coupling constants (in frequency measurement unit) are the off-diagonal terms and the time-constants T_1 and T_2 are located in the last two rows. In the case of crotonic acid, multiple references provided compatible measured values of chemical shift and qubits interactions.
Before going in the detail of the compact model, some premises are required. First of all, a large ensemble of spins is involved in conventional NMR experiments, so the qubits density matrix at thermal equilibrium (high-temperature limit) is not that of a pure state, but it is always that of a mixed one [69]:

$$\rho_0 = \frac{\exp\left(-\frac{\mathcal{H}_{\text{NMR}}}{k_{\text{B}}T}\right)}{\text{tr}\left\{\exp\left(-\frac{\mathcal{H}_{\text{NMR}}}{k_{\text{B}}T}\right)\right\}} \sim \left(\frac{\mathbb{I}}{2}\right)^{\otimes n} + \Delta\rho\,, \tag{5.25}$$

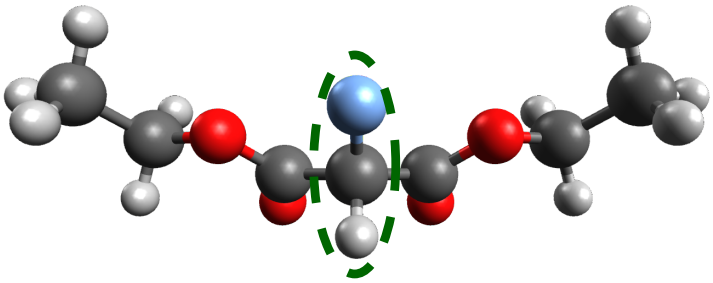
where k_{B} is the Boltzmann constant and T is the temperature. The first term is a uniformly mixed ensemble of all possible states, while the second term represents



(a) Cytosine.



(b) Crotonic acid, where all ^{13}C encode quantum information.



(c) ^{13}C-labeled diethyl fluoromalonate.

Figure 5.11: Examples of NMR quantum computers. In cytosine and diethyl fluoromanolate cases, nuclei encoding qubits are marked with green ellipses.

a small deviation. This state is inadequate for computation and several techniques [75] are experimentally exploited to obtain **pseudopure fiducial initial states** as

$$\rho_0 = \left(\frac{\mathbb{I}}{2}\right)^{\otimes n} + \Delta\rho \longrightarrow \rho_0 = \left(\frac{\mathbb{I}}{2}\right)^{\otimes n} + \alpha \left(|0\rangle \langle 0|\right)^{\otimes n}, \tag{5.26}$$

where α is a small constant.
Secondly, in a molecule with several coupled qubits, the two-qubit gate between a specific pair of qubits requires only their corresponding J-coupling. This implies that the effect of other J-couplings must be properly removed during the execution time of the desired gate. This can be done with the **refocusing** technique. The basic principle to remove an unwanted coupling between qubits $|q_k\rangle$ and $|q_m\rangle$ for a time τ is to leave $|q_k\rangle$ to evolve freely between 0 and $\frac{\tau}{2}$ and then reverse its evolution from $\frac{\tau}{2}$ to τ, applying a π-pulse on qubit $|q_m\rangle$ at time $t = \frac{\tau}{2}$. A final π-pulse on qubit $|q_m\rangle$ ensures that both qubits return to their initial state. In terms of unitary

Table 5.4: Physical parameters of the crotonic acid: the isotropic chemical shifts are reported in [70, 71], the J-coupling constants (properly $\frac{J}{2\pi}$) are taken from [71] and the time constants $T_{1,2}$ are available in [71, 72].

Crotonic Acid				
	q[0]	q[1]	q[2]	q[3]
q[0]	172.35×10^{-6}			
q[1]	72.36 Hz	122.38×10^{-6}		
q[2]	1.18 Hz	69.72 Hz	147.55×10^{-6}	
q[3]	7.04 Hz	1.46 Hz	41.64 Hz	18.08×10^{-6}
T_1	10.20 s	5.60 s	5.30 s	5.10 s
T_2	0.79 s	0.66 s	0.92 s	0.84 s

Table 5.5: Physical parameters of the crotonic acid: the isotropic chemical shifts are reported in [70], the J-coupling constants (properly $\frac{J}{2\pi}$) are taken from [73], such as $T_{1,2}$.

Cytosine		
	q[0]	q[1]
q[0]	5.970×10^{-6}	
q[1]	7.10 Hz	7.498×10^{-6}
T_1	7.00 s	7.00 s
T_2	1.00 s	1.00 s

Table 5.6: Physical parameters of the ^{13}C-labeled diethyl fluoromalonate. All the parameters are taken from [74] and the magnetic static field is expected to be $B = 14.09\,\mathrm{T}$.

Diethyl fluoromalonate			
	q[0]	q[1]	q[2]
q[0]	3350.60 Hz		
q[1]	47.50 Hz	−1 110 998.38 Hz	
q[2]	161.60 Hz	−191.50 Hz	12 899.91 Hz
T_1	3.20(38) s	3.30(15) s	3.00(44) s
T_2	1.20(18) s	1.30(22) s	1.40(29) s

evolution, this corresponds to:

$$\begin{aligned}
&[R_x(\pi)\otimes\mathbb{I}]\,U_J\left(\frac{\tau}{2}\right)[R_x(\pi)\otimes\mathbb{I}] = U_J\left(-\frac{\tau}{2}\right) = [\mathbb{I}\otimes R_x(\pi)]\,U_J\left(\frac{\tau}{2}\right)[\mathbb{I}\otimes R_x(\pi)]\,,\\
&[R_x(\pi)\otimes\mathbb{I}]\,U_J\left(\frac{\tau}{2}\right)[R_x(\pi)\otimes\mathbb{I}]\,U_J\left(\frac{\tau}{2}\right) = U_J\left(-\frac{\tau}{2}\right)U_J\left(\frac{\tau}{2}\right) = \mathbb{I}\,.
\end{aligned} \tag{5.27}$$

Refocusing schemes for multi-qubit systems can be derived from these principles and are discussed in [76].

5.2.1 Compact model

The definition of a compact model for NMR was incremental. In [59], a first model for heteronuclear quantum computers, which inherited some features of the Cr_7Ni described in [58], was presented. In particular, the same mechanism for describing single-qubit gates was employed, *i.e.* the global unitary evolution is that reported in Figure 4.5a and is given by the product of two contributions: one associated with magnetic resonance mechanism, the other corresponding to the J-coupling-only evolution of qubits (last term of Equation 5.22).
The main limitation of this model is that it completely neglects the off-resonance phenomena which reasonably take place with homonuclear qubits. For this reason, in [60] the so-called Fourier model (section 4.5) was proposed. In the evaluation of the unitary evolution of single-qubit gates under the Fourier model, two approaches were initially taken into account:

1. treating the Fourier-model (Equation 4.5) and J-coupling Hamiltonians separately, so the global unitary evolution was obtained with the Trotter–Suzuki-like product;

2. complex exponentiation of the sum of the two Hamiltonians.

The two-evolution-product approximation is theoretically applicable since the time duration of single-qubit gates is significantly shorter than $\frac{1}{J_{\max}}$, according to section 5.2.
Relaxation and decoherence are described by the usual model of section 4.6.
It is important to precise that the compact model of NMR devices is based on three main assumptions:

1. The quantum computer initial state is fiducially assumed pure and equal to $|0\rangle\langle 0|$.

2. Refocusing is assumed to be intrinsically applied when a two-qubit gate is evaluated on a pair of qubits, thus avoiding to complicate excessively the simulation, which would depend on the refocusing scheme.

3. In the evaluation of the chemical shifts, the reference resonance frequencies of each element are assumed to be, for sake of simplicity, $\omega_{\text{ref}} = \gamma_n B_0$, with γ_n taken from [53]. In the evaluation of off-resonance effects with the Fourier model, this choice is not effective, since the *exact* chemical shift values are employed, so the differences between the qubits resonance frequencies are always the same, independently on the choice of ω_{ref}.

5.2.2 Results

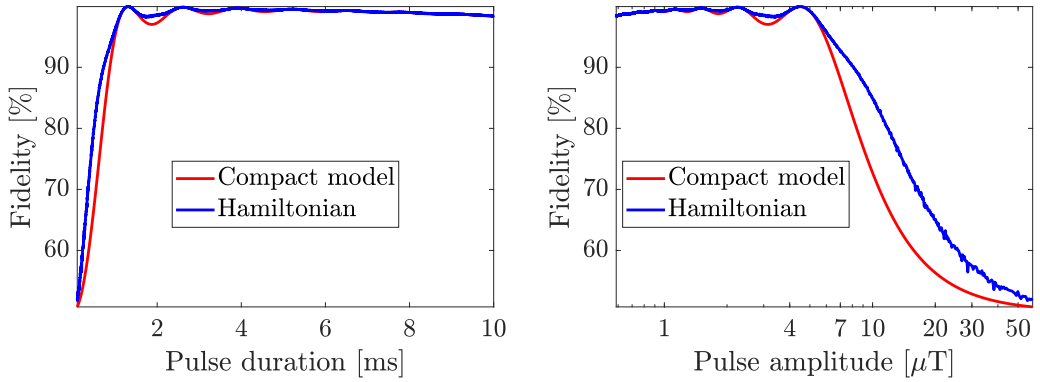
Two different simulative tests are presented for the NMR compact model. First of all, it is important to precise that the reported tests involve homonuclear quantum computers (cytosine and crotonic acid), since their simulation is expected to be harder. In fact, they are affected by more non-ideality phenomena than heteronuclear devices. Suffice it to remind that off-resonance cannot be easily neglected.
The first test consisted in the pulse amplitude optimization for each molecule. The goal was maximizing the fidelity when a $R_x\left(\frac{\pi}{2}\right)$ gate is sequentially applied to all qubits, for different values of the pulse width and amplitude. The pulse parameters were assumed equal for all qubits. This benchmark was chosen because the $\frac{\pi}{2}$-rotation forces a superposition state on the equator of the Bloch sphere, sensitive to the unwanted J-coupling evolution, mainly causing a rotation about $\hat{z}$ axis. The simulation results obtained with the compact model were compared with those with direct Hamiltonian integration, employed as reference model, and using the same decoherence model.
The obtained results were preliminary to the second test, consisting in the simulation of quantum circuits on the two molecules. Pulse amplitudes were assumed fixed and their values was equal to the optimal ones given by the first test. The results of the proposed model were compared with those obtained via direct integration and those obtained with Qiskit employing thermal relaxation model, according to the approach outlined in section 4.6.2.

5.2.3 Optimization of single-qubit-gate duration

The fidelity dependence on pulse amplitudes and widths is reported in Figures 5.12 and 5.13. The $\gtrsim 1\,\text{ms}$ range is due to the fact that simulations were focused in the soft pulses regime, thus facilitating qubits addressing. The fidelity plots (Figure 5.12) for cytosine obtained with the two approaches are quite well overlapped, especially for pulse duration/amplitudes compatible with soft pulses. It is also possible to observe in both cases a remarkable ringing with local maxima when τ is an integer multiple of $\frac{2\pi}{|\Delta\omega_0|} \sim 1.31\,\text{ms}$, *i.e.* the reciprocal of the difference of the qubits resonance frequencies; this is a remarkable trace of the presence of off-resonance mechanism. The pulse width shall be chosen such that:

$$\frac{2\pi}{|\Delta\omega_0|} \ll \tau \ll \frac{2\pi}{|J|} \longrightarrow 1.31\,\text{ms} \ll \tau \ll 142.86\,\text{ms}\,. \tag{5.28}$$

Considering Figure 5.12, the practical upper limit is lower than 142.86 ms because of decoherence and relaxation phenomena. Moreover, the sensitivity of the fidelity $\mathcal{F}$ to the fluctuations of the field ($\frac{\partial\mathcal{F}}{\partial B_r}\frac{B_r}{\mathcal{F}}$) and pulse width ($\frac{\partial\mathcal{F}}{\partial\tau}\frac{\tau}{\mathcal{F}}$) should be taken into account. The leftmost peak of the fidelity versus time plot ($\tau = 1.31\,\text{ms}$, $B_r = 4.50\,\mu\text{T}$) is quite narrow and sensible to time variations, while the second



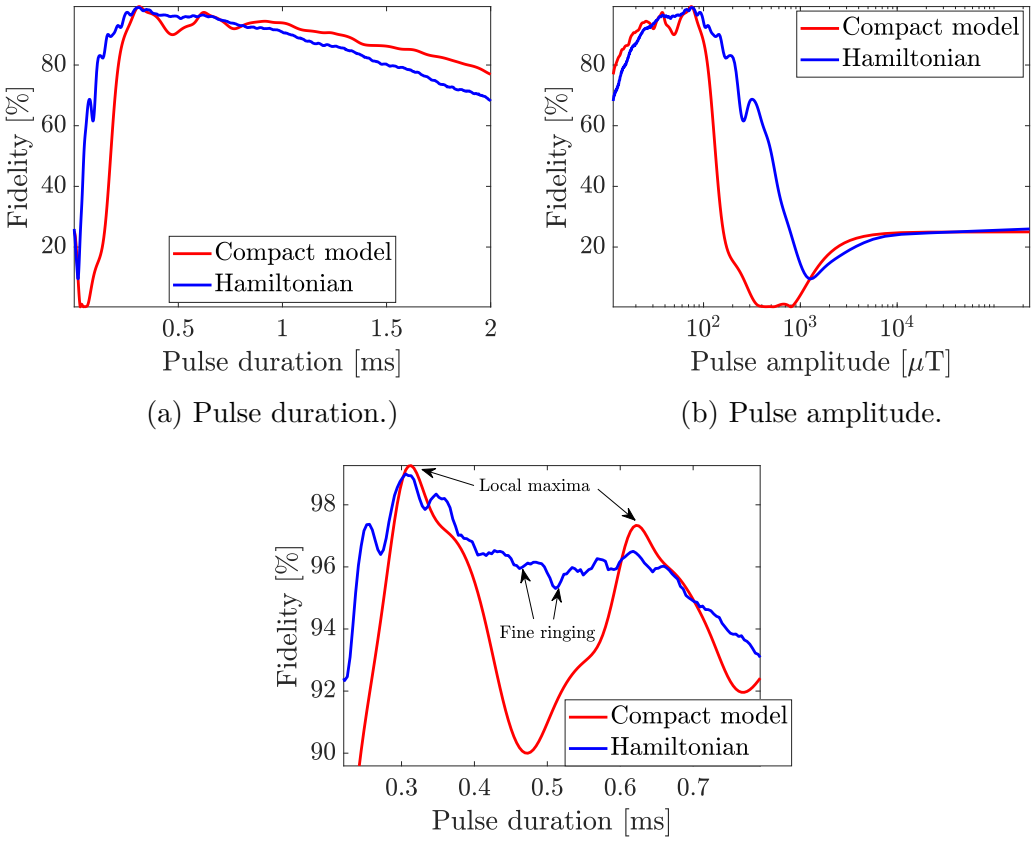
(a) Pulse duration. (b) Pulse amplitude.

Figure 5.12: Dependency of fidelity on pulse amplitude and duration for a two-qubit cytosine molecule.

one ($\tau = 2.62\,\text{ms}$, $B_r = 2.25\,\mu\text{T}$) is a reasonable trade-off between fidelity, pulse parameters sensitivity and decoherence. For this reason, the pulse amplitude was fixed to $B_r = 2.25\,\mu\text{T}$ for the quantum circuits of the second benchmark.
For what concerns crotonic acid, the curves (Figure 5.13) are less overlapped than cytosine case. Compact model fidelity is lower than the Hamiltonian integration one for short pulse durations, while it is higher for long pulse durations, which are not going to be practically taken into account because relaxation and decoherence are significant in that range. For this reason, it is possible to say that the approximation provided by the compact model is reasonable for the pulse durations/amplitudes with highest fidelity. Moreover, the fidelity plot ringing is less evident with Hamiltonian integration than with compact model. However, as it is possible to see in Figure 5.13c, in both cases local maxima occur in the neighbourhood of integer multiples of $2\pi/\left|\Delta\omega_{0,\text{min}}\right| = 0.32\,\text{ms}$, while the fine ringing has a periodicity close to $2\pi/\left|\Delta\omega_{0,\text{max}}\right| = 0.05\,\text{ms}$. However, this *good* range of pulse width — where fidelity never achieves the values 0.9925 and 0.9894 with compact model and Hamiltonian integration respectively — is limited and significantly small. This is due to the fact that the ratio between the largest J-coupling constant and the minimum difference in chemically shifted resonance frequencies is smaller than cytosine case:

$$\frac{2\pi}{\left|\Delta\omega_{0,\text{min}}\right|} \ll \tau \ll \frac{2\pi}{\left|J_{\text{max}}\right|} \longrightarrow 0.32\,\text{ms} \ll \tau \ll 13.82\,\text{ms}\,. \tag{5.29}$$

Being the fidelity relatively low, the operating point was set to the maximum value $B_r = 75.45\,\mu\text{T}$ ($\tau = 0.32\,\text{ms}$).



(a) Pulse duration.)

(b) Pulse amplitude.

(c) Pulse duration zoom.

Figure 5.13: Dependency of fidelity on pulse amplitude and duration for a four-qubit crotonic acid.

Simulation of complex quantum circuits

After setting the best control parameters for cytosine and crotonic acid, quantum circuits were tested, according to the methodology described in section 2.3.4, involving thermal relaxation error for relaxation and decoherence and forcing the same gate duration for the $U_{\{1,2,3\}}$ and CZ gates employed by the Qiskit reference simulation. In addition to Qiskit, quantum circuit simulations based on Hamiltonian direct integration were also executed, to evaluate the errors of the Fourier model in the simulation of complex circuits. Figure 5.14 and Figure 5.15 show the probability distributions for Grover's search and Bernstein-Vazirani's quantum circuits on cytosine and crotonic acid, respectively, which are characterized, in the ideal qubits simulation regime, by the presence of one eigenstate with measurement probability equal to 1 ($|10\rangle$ and $|1101\rangle$ respectively). Figures show that the expected outcome is always the most probable: the probability distributions are very close to each

other in the cytosine case, while in the crotonic acid one the expected eigenstate has a slightly higher probability in Qiskit simulation than in the others. This is due to the fact that the relaxation mechanism of Qiskit, associated with Kraus operators, involves fewer eigenstates in the relaxation dynamics than those of the model described in subsection 4.6.1, so the expected outcome has higher probability. This effect tends to be more evident in circuits involving more qubits. Moreover, Fourier model does not significantly change simulations with respect to Hamiltonian integration.
Considering the previous comments, it is possible to properly interpret the results in Tables 5.7 and 5.8, where — for each circuit of the employed benchmark — the **most probable eigenstates**, obtained with each simulation approach, and the **KL-divergence** between the probability distributions obtained with Qiskit and the other simulation methods, are reported. The most probable eigenstates are the same in all cases. Moreover, the probability distributions corresponding to two-qubit circuits, even if simulated on a crotonic acid molecule and regardless of the circuit depth, are very close to each other, while the KL-divergence is slightly higher in those involving three or four qubits, *e.g.* the reversible full-adder. However, the highest probability differences and KL-divergences between the reference and the proposed models, which occur for circuits like the reversible full-adder, are not actually critical, since in these cases the probability value of the expected solution is, with all simulation strategies, sufficiently low (0.5575, 0.3810 and 0.3932 for the full-adder with thermal relaxation, Fourier model and Hamiltonian integration, respectively) for considering unreliable, and thus useless, the execution of those circuits on the target molecules. Moreover, the maximum probabilities obtained with the compact model are always lower than those based on thermal relaxation, thus ensuring that the proposed methodology does not underestimate the qubits' non-idealities.
In conclusion, the proposed compact model does **not underestimate** non-ideality phenomena, with respect to standard representations, such as Hamiltonian integration and thermal relaxation error, so it is possible to reasonably consider it a reliable method for simulating quantum circuits considering NMR quantum computers as backends.

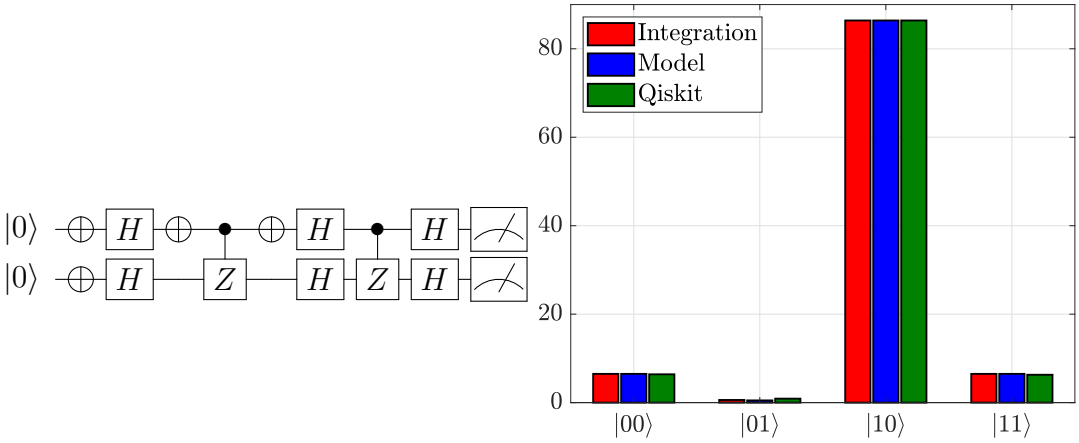


Figure 5.14: Probability distribution resulting from the simulation of the execution of Grover's quantum algorithm (binary solution equal to 10) on a cytosine molecule, adopting the direct integration, the proposed model and Qiskit `QasmSimulator`.

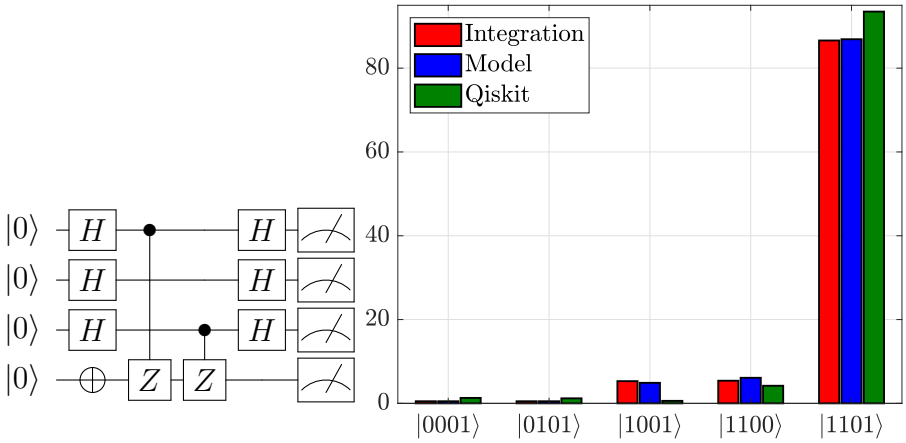


Figure 5.15: Probability distribution resulting from the simulation of the quantum circuit of Bernstein-Vazirani's algorithm (binary string to be found equal to 101) on a crotonic acid molecule, adopting the direct integration, the proposed model and Qiskit's `QasmSimulator`. Only the eigenstates with non-null probabilities are reported.

Table 5.7: Comparison of the most probable outcomes on a cytosine quantum computer, according to the thermal relaxation model of Qiskit and two simulation strategies based on the proposed relaxation model. Virtual-Z gates are assumed to be employed.

Quantum Circuit	Thermal relaxation state, prob.	Fourier state, prob. KL-Divergence	Hamiltonian state, prob. KL-Divergence
cnot_01	$\|11\rangle$, $\mathbb{P} = 0.9541$	$\|11\rangle$, $\mathbb{P} = 0.9550$ KLD $= 0.010727$	$\|11\rangle$, $\mathbb{P} = 0.9556$ KLD $= 0.008759$
dcx_01	$\|10\rangle$, $\mathbb{P} = 0.9117$	$\|10\rangle$, $\mathbb{P} = 0.9120$ KLD $= 0.001612$	$\|10\rangle$, $\mathbb{P} = 0.9100$ KLD $= 0.002051$
iswap_01	$\|10\rangle$, $\mathbb{P} = 0.8641$	$\|10\rangle$, $\mathbb{P} = 0.8557$ KLD $= 0.000514$	$\|10\rangle$, $\mathbb{P} = 0.8558$ KLD $= 0.000508$
dj_balanced	$\|01\rangle$, $\mathbb{P} = 0.8641$	$\|01\rangle$, $\mathbb{P} = 0.8621$ KLD $= 0.001697$	$\|01\rangle$, $\mathbb{P} = 0.8616$ KLD $= 0.001729$
grov_10_IBMQ	$\|10\rangle$, $\mathbb{P} = 0.8640$	$\|10\rangle$, $\mathbb{P} = 0.8556$ KLD $= 0.000498$	$\|10\rangle$, $\mathbb{P} = 0.8567$ KLD $= 0.000332$
grov_10	$\|10\rangle$, $\mathbb{P} = 0.8639$	$\|10\rangle$, $\mathbb{P} = 0.8639$ KLD $= 0.001531$	$\|10\rangle$, $\mathbb{P} = 0.8640$ KLD $= 0.001656$

Table 5.8: Comparison of the most probable outcomes on a crotonic acid quantum computer, according to the thermal relaxation model of Qiskit and two simulation strategies based on the proposed relaxation model. Virtual-Z gates are assumed to be employed.

Quantum Circuit	Thermal relaxation state, prob.	Fourier state, prob. KL-Divergence	Hamiltonian state, prob. KL-Divergence
bern_vaz_101	$\|1101\rangle$, $\mathbb{P} = 0.9349$	$\|1101\rangle$, $\mathbb{P} = 0.8688$ KLD $= 0.071052$	$\|1101\rangle$, $\mathbb{P} = 0.8662$ KLD $= 0.082697$
toffoli_0110	$\|1110\rangle$, $\mathbb{P} = 0.7176$	$\|1110\rangle$, $\mathbb{P} = 0.6282$ KLD $= 0.277338$	$\|1110\rangle$, $\mathbb{P} = 0.6384$ KLD $= 0.253923$
fulladder_0011	$\|1001\rangle$, $\mathbb{P} = 0.5575$	$\|1001\rangle$, $\mathbb{P} = 0.3810$ KLD $= 0.584241$	$\|1001\rangle$, $\mathbb{P} = 0.3932$ KLD $= 0.567896$
swap_03	$\|1000\rangle$, $\mathbb{P} = 0.8645$	$\|1000\rangle$, $\mathbb{P} = 0.8582$ KLD $= 0.005852$	$\|1000\rangle$, $\mathbb{P} = 0.8591$ KLD $= 0.005729$
grov_10_IBMQ	$\|0010\rangle$, $\mathbb{P} = 0.9797$	$\|0010\rangle$, $\mathbb{P} = 0.9337$ KLD $= 0.040235$	$\|0010\rangle$, $\mathbb{P} = 0.9296$ KLD $= 0.049765$
grov_10	$\|0010\rangle$, $\mathbb{P} = 0.9796$	$\|0010\rangle$, $\mathbb{P} = 0.9614$ KLD $= 0.010340$	$\|0010\rangle$, $\mathbb{P} = 0.9464$ KLD $= 0.031118$

5.3 Concluding remarks

This thesis chapter reports the description of the compact models for two molecular quantum technologies. Their corresponding validation was done sequentially: from the analysis of specific physical features and non-ideality phenomena of each device taken on their own (*e.g.* off-resonance in NMR and residual interaction between qubits with *open-switch* configuration in the other case) to the simulation of quantum circuits employing a compact model with all its physical characteristics, also inheriting the models for relaxation and decoherence described in the previous chapter.
The overall compact modelling results can be considered encouraging, since they are on the one hand coherent with theory, on the other the computational complexity is lower than a canonical physical simulation. Moreover, for what concerns the two analyzed technologies, it was possible to define operating regions where the effects of non-ideality phenomena do not significantly affect the functioning of the devices. For example, the magnetic static field in Cr_7Ni-based quantum computers shall be greater or equal than 7 T, while the amplitude of the oscillating magnetic field for crotonic acid shall be $B_r = 75.45\,\mu T$, to look for a compromise between the minimizations of the effects of off-resonance and decoherence.
The detection of these regions is also useful for a physical-conscious design of the real device, its contour control infrastructure and of the applications to be run efficiently on it. Moreover, it is important to say that the *good* operating regions for each device have been also chosen as default configurations in the simulator (`params.m` files in Appendix B), thus facilitating the end user.
Even though the complexity of the analyzed devices is not significantly high, they are hoped to be exploited, in the future, as reference for the construction of the compact models of more sophisticated quantum devices and technologies, not strictly molecular.

Part III

QUBO modelling

Chapter 6

Introduction to binary optimization problems

In this chapter, the main features of the equivalent **Ising** and **Quadratic Unconstrained Binary Optimization** (**QUBO**) models are introduced. These are employed for describing **combinatorial optimization problems** involving binary variables and they van be employed in different applicative scenarios: from pharmaceutics [77] to routing [78] and placemenet [79] optimization.
An introduction to classical and quantum techniques employed for solving Ising/QUBO problems is also provided, with a couple of examples of combinatorial optimization problems based on this formuation.

6.1 QUBO model

Before starting with the description of the main features of QUBO problems, it could be convenient to understand the meaning of the acronym:

- **Quadratic** refers to the fact that second power is the highest one applied to variables;
- **Unconstrained** refers to the absence of constraints applied to variables;
- **Binary** refers to the fact that **unipolar** binary variables $x \in [0,1]$ are involved;
- **Optimization** refers to the fact that this model is employed for minimizing or maximizing the characteristic objective function of the problem.

A QUBO cost function can be written as:

$$f([x_{N-1}; \cdots ; x_0]) = \sum_{i=0}^{N-1} \sum_{j=0}^{i} q_{ij} x_i x_j \,, \tag{6.1}$$

where $[x_{N-1};\cdots;x_0] = \mathbf{x}$ is the vector containing each binary variable x_i and q_{ij} is the quadratic coefficient of $x_i x_j$. Problem's solution is the binary string $[x_{N-1};\cdots;x_0]_{\text{opt}}$ associated with the lowest value of the cost function. In the particular case of binary and unipolar variables, the square value is equal to the value itself ($x_i x_i = x_i^2 = x_i$), thus implying that the coefficients q_{ii} correspond to those on single variables. For this reason, the cost function can be re-written as:

$$f([x_{N-1};\cdots;x_0]) = \sum_{i=0}^{N-1} a_i x_i + \sum_{i=0}^{N-2}\sum_{j>i} b_{ij} x_i x_j \,, \tag{6.2}$$

where a_i e b_{ij} are single-variable and pair coefficients respectively. An example of QUBO function involving three variables is the following:

$$f(\mathbf{x}) = 6x_2 - 9x_1 - 8x_0 + 7x_2x_1 - 7x_2x_0 - 2x_1x_0 \,. \tag{6.3}$$

Figure 6.1 shows the profile of the cost function. The optimal solution is the

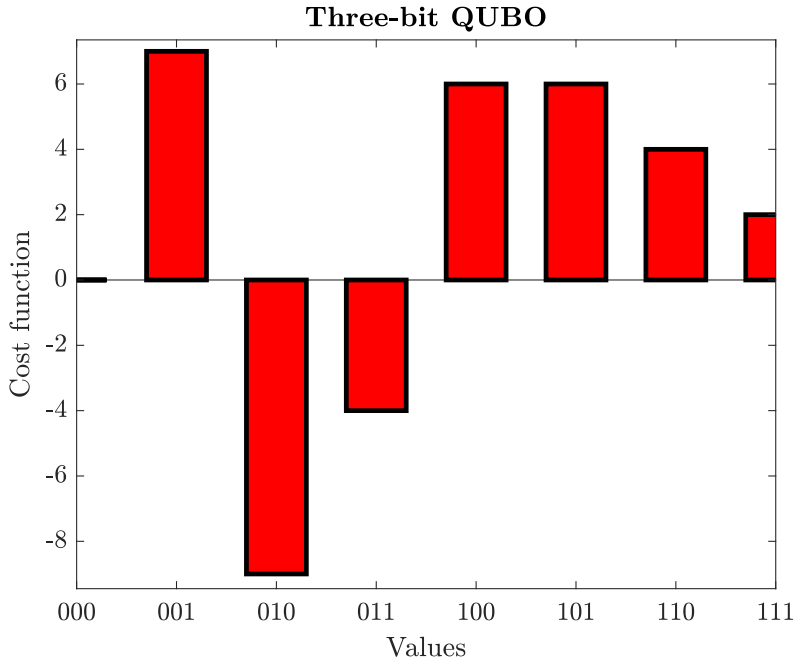


Figure 6.1: Profile of the cost function of Equation 6.3.

binary string $[0;1;0]$, whose cost value is equal to -9. The proof of $f([0;1;0]) = -9$ is reported in the following, obtained by replacing $[x_2;x_1;x_0]$ values in the cost function:

$$f([0,1,0]) = (+6\cdot 0) + (-9\cdot 1) + (-8\cdot 0) + (+7\cdot 0\cdot 1) + (-7\cdot 0\cdot 0) + (-2\cdot 1\cdot 0) = -9$$

An equivalent QUBO formulation is given by matrix notation:

$$f(\mathbf{x}) = \mathbf{x}^T \cdot Q \cdot \mathbf{x} \,, \tag{6.4}$$

where Q is a square matrix of order N with the coefficients of the problem and $\mathbf{x}^T$ is the transpose column vector with binary values $\mathbf{x}$. It can be written in two equivalent forms: upper triangular and symmetric, which are reported in the following for the cost function in Equation 6.3:

$$\begin{aligned} Q_{\text{triangular}} &= \begin{bmatrix} 6 & 7 & -7 \\ 0 & -9 & -2 \\ 0 & 0 & 7 \end{bmatrix}, \\ Q_{\text{symmetric}} &= \begin{bmatrix} 6 & \frac{7}{2} & -\frac{7}{2} \\ \frac{7}{2} & -9 & -1 \\ -\frac{7}{2} & -1 & 7 \end{bmatrix}. \end{aligned} \tag{6.5}$$

The upper triangular form is obtained by replacing the q_{ij} coefficients of the symmetric matrix with $q_{ij} + q_{ji}$, for all i and $j > i$, and the q_{ij}, for $j < i$, with zero. The proof of the equality of all expressions is reported in the following:

$$\begin{bmatrix} x_0 & x_1 & x_2 \end{bmatrix} \begin{bmatrix} 6 & 7 & -7 \\ 0 & -9 & -2 \\ 0 & 0 & 7 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x_0 & x_1 & x_2 \end{bmatrix} \begin{bmatrix} 6x_0 + 7x_1 - 7x_2 \\ -9x_1 - 2x_2 \\ 7x_2 \end{bmatrix}$$
$$= x_0(6x_0 + 7x_1 - 7x_2) + x_1(-9x_1 - 2x_2) - 7x_2^2$$
$$= 6x_0 + 7x_0x_1 - 7x_0x_2 - 9x_1 - 2x_1x_2 - 7x_2$$
$$\begin{bmatrix} x_0 & x_1 & x_2 \end{bmatrix} \begin{bmatrix} 6 & \frac{7}{2} & -\frac{7}{2} \\ \frac{7}{2} & -9 & -1 \\ -\frac{7}{2} & -1 & 7 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x_0 & x_1 & x_2 \end{bmatrix} \begin{bmatrix} 6x_0 + \frac{7}{2}x_1 - \frac{7}{2}x_2 \\ \frac{7}{2}x_0 - 9x_1 - x_2 \\ -\frac{7}{2}x_0 - x_1 + 7x_2 \end{bmatrix}$$
$$= x_0(6x_0 + \frac{7}{2}x_1 - \frac{7}{2}x_2) + x_1(\frac{7}{2}x_0 - 9x_1 - x_2) + x_2(-\frac{7}{2}x_0 - x_1 + 7x_2)$$
$$= 6x_0 + \frac{7}{2}x_0x_1 - \frac{7}{2}x_0x_2 + \frac{7}{2}x_0x_1 - 9x_1 - x_1x_2 - \frac{7}{2}x_0x_2 - x_1x_2 + 7x_2$$
$$= 6x_0 + 7x_0x_1 - 7x_0x_2 - 9x_1 - 2x_1x_2 - 7x_2$$

Two important properties of QUBO functions are reported in the following, exploitable to facilitate the solution of some problems:

- If the sign of all the coefficients q_{ij} of a function $f(\mathbf{x})$ is changed, the new optimum solution corresponds to the combination maximizing $f(\mathbf{x})$.
- Multiplying all the coefficients q_{ij} of a function $f(\mathbf{x})$ with the same positive factor $\alpha > 0$, f is scaled accordingly, but the optimum solution does not change.

Some combinatorial optimization problems are quite easy to translate in QUBO model, as shown in [11], while others require additional steps. These can be classified into three categories [80]:

- **without constraints**, for example the partitioning problem;
- **with equality constraints** like in the traveling salesman problem;
- **with inequality constraints** like in the knapsack problem.

Despite what the name would suggest, a large part of these can be formulated with the QUBO model, adding **quadratic penalties** to the objective function, instead of imposing constraints in the classical sense. These **penalty functions** are equal to zero for feasible solutions and they show positive penalty amounts for infeasible solutions.
Equality constraint can be written very easily: if the sum of a set of binary variables $[x_{N-1}, \cdots, x_0]$ must be equal to a given number $K \in \mathbb{Z}^+$, the following penalty can be written:

$$P_{\text{equality}} = \left(\sum_{i=0}^{N-1} x_i - K \right)^2 = 0 \,. \tag{6.6}$$

It can be ascertained that $P_{\text{equality}} = 0$ when K variables are equal to 1, otherwise $P_{\text{equality}} > 0$. Inequality constraints can be treated as equality ones by adding M auxiliary bits $[a_{M-1} \cdots a_0]$, to be exploited for having an equality constraint:

$$\begin{aligned} P(\sum_i x_i \geq K) &= \left(\sum_{i=0}^{N-1} x_i - K - \sum_{l=0}^{M-1} 2^l a_l \right)^2 , \\ P(\sum_i x_i \leq K) &= \left(\sum_{i=0}^{N-1} x_i - K + \sum_{l=0}^{M-1} 2^l a_l \right)^2 . \end{aligned} \tag{6.7}$$

The total number of auxiliary variables must be $\log_2(N-K)$ and $\log_2(K)$ for majority and minority constraints respectively. It is important to clarify that if binary variables have weights $w_i \in \mathbb{Z}$ (thus $\sum_{i=0}^{N-1} w_i x_i$), the equality and inequality constraints can be written in the same way, only considering the a_i values. In this case, the total number of auxiliary variables of majority constraint must be $\log_2(\sum_{i=0}^{N-1} w_i x_i - K)$.
The **penalty functions** of some common two and three-variable constraints P to be added are reported in Table 6.1.

In a generic optimization problem involving $[x_1, \cdots, x_N] = \mathbf{x}$ binary variables, a QUBO function, like that in Equation 6.2, can be obtained with the simplification of the function given by the sum of a reference cost function $f_{\text{cost}}(\mathbf{x})$ and constraints on variables $c_k(\mathbf{x})$:

$$f(\mathbf{x}) = f_{\text{cost}}(\mathbf{x}) + \sum_k \alpha_k c_k(\mathbf{x}) \,, \tag{6.8}$$

with α_k weight of the constraint c_k. In a practical scenario, the choice of α_k is fundamental for ensuring the achievement of the problem's optimal solution and its dimensioning is an important issue. If a constraint must be absolutely satisfied

Classical constrain	Equivalent penalty
$x + y = 1$	$P(1 - x - y + 2xy)$
$x + y \leq 1$	$P(xy)$
$x + y \geq 1$	$P(1 - x - y + xy)$
$x = y$	$P(x + y - 2xy)$
$x \leq y$	$P(x - xy)$
$x_1 + x_2 + x_3 \leq 1$	$P(x_1x_2 + x_1x_3 + x_2x_3)$

Table 6.1: Known QUBO penalty functions for constraints evaluation, arranged from [11].

(**hard constraint**), α_k must be sufficiently large, in order to avoid violations. When a constraint is less critical (**soft constraint**), in sense that respecting it would be preferable, but a violation is accepted, a small value for α_k is permitted. Avoiding extremely large or soft penalty parameters for hard and soft constraint is required for not affecting the quality of the solution. In fact, if α_k is too large, the cost function could become too flat, making difficult to evaluate the optimal solution; on the other hand, with a too small penalty parameter, a constraint could ignored.
In practical scenarios, software routines — as the `Python` library **qubovert** [81] — permit to build a QUBO model for an optimization problem, starting from an initial function and a set of constraints. Moreover, some routines are available for solving the problem, exploiting **brute-force**, *i.e.* evaluating all possible combinations, or **simulated** and **quantum annealing**.

6.2 Ising model

The **Ising** model is a physical-mathematical model used in statistical mechanics for describing **ferromagnetism** . It consists of a system of **dipolar atomic spins**, which can be in one among two discrete states ± 1, depending on their orientation (spin-up or spin-down), and are arranged in a lattice which allows interaction between neighbour spins. This model can be described by the following N-spin Hamiltonian:

$$H(\mathbf{s}) = \sum_{i=1}^{N} h_i \cdot s_i + \sum_i \sum_{j \neq i} J_{ij} \cdot s_i \cdot s_j \,. \tag{6.9}$$

This includes two types of interaction:

- **External-field** h, whose sign is related to the preferred orientation of a spin with the respect to that of an applied external field. The size of h quantifies the dependency of the spin orientation on the external field and it

also represents the weight of the energy contribution of a single spin with the respect to the others;

- **Inter-spin** J between neighbour spin pairs, whose sign is related to the preferred reciprocal alignment of the spins pair (ferromagnetic or anti-ferromagnetic). Each pair contribution has to be summed to obtain the overall interaction energy.

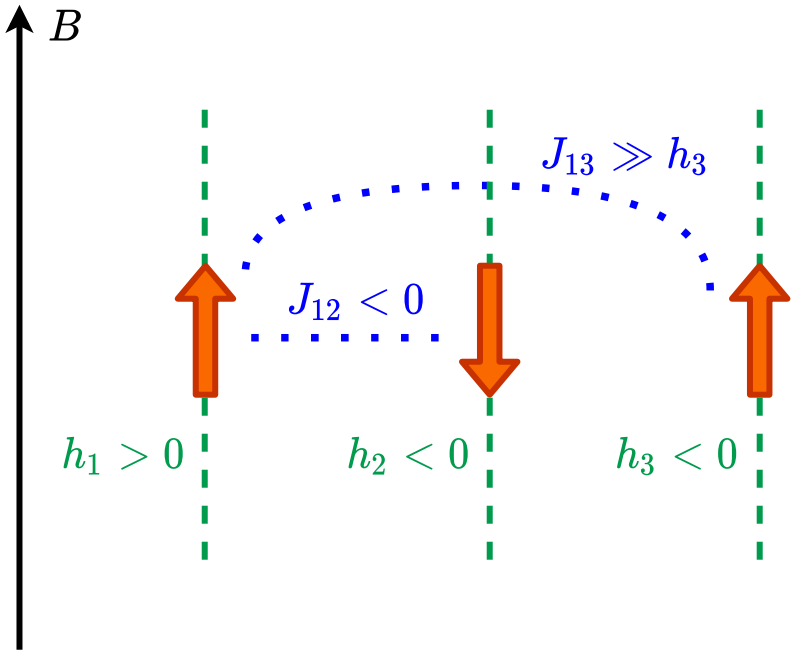


Figure 6.2: Example of Ising system.

Figure 6.2 shows an example of an Ising system with three spins. It is interesting to observe that weight J_{13} is much higher than h_3, thus forcing an orientation of the rightmost spin parallel to the magnetic field B (and to the leftmost spin), in opposition to h_3, which tends to force an anti-parallel orientation of that spin with respect to B.
It is possible to prove that **Ising** and **QUBO** models are perfectly equivalent. The fundamental relations between Ising and QUBO variables s_i and q_i are the following:

$$\begin{aligned} q_i &= \frac{1+s_i}{2} \\ s_i &= 2q_i - 1 \end{aligned} \tag{6.10}$$

Given a QUBO formulation, the corresponding Ising one can be obtained by replacing q_i with the corresponding expression involving s_i and *vice-versa*.
A matrix-notation Ising formulation can be also written as:

$$f(\mathbf{s}) = \mathbf{s}^T J \mathbf{s} + \mathbf{h}^T \mathbf{s} + o\,, \tag{6.11}$$

where $\mathbf{s} = [s_{N-1}; \cdots ; s_0]$ is a column vector of bipolar binary variables $s_i \in [-1; +1]$, J is the matrix containing all the inter-spin elements of the Hamiltonian, $\mathbf{h}$ is a vector with all the external-field coefficients and o is a negligible offset term. The relations for obtaining $\mathbf{h}$ and J from Q (apart from an offset) and *vice-versa* are reported in the Equation 6.12:

QUBO to Ising	Ising to QUBO
$J_{ij} = \dfrac{q_{ij}}{4}, \quad \text{with } i \neq j$	$Q_{ij} = 4J_{ij}, \quad \text{with } i \neq j$
$h_i = \dfrac{1}{4}\left(\overbrace{\sum_{j\neq i} q_{ij}}^{\text{row fixed}} + \overbrace{\sum_{i\neq j} q_{ij}}^{\text{column fixed}}\right)$	$Q_{ii} = 2\left(\sum_i h_i - \overbrace{\sum_{j\neq i} J_{ij}}^{\text{row fixed}} - \overbrace{\sum_{i\neq j} J_{ij}}^{\text{column fixed}}\right)$

(6.12)

Many libraries for managing combinatorial optimization problems can do these conversions automatically. For example, `qubovert` can translate a combinatorial optimization problem in both models using **QUBO** and **QUSO** classes [81], where the second acronym stays for Quadratic Unconstrained Spin Optimization, which is nothing but an optimization model described by an Ising model.

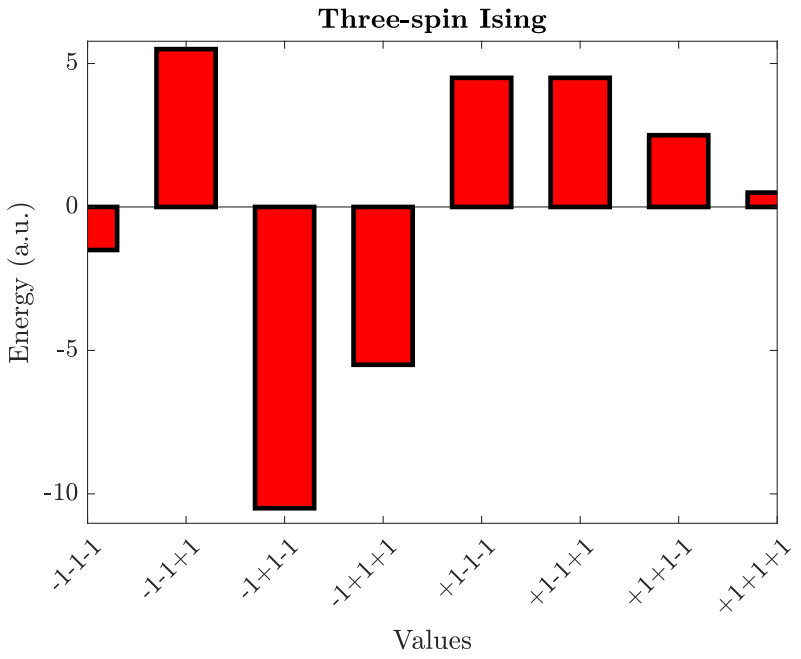


Figure 6.3: Energy profile of the Ising model equivalent to QUBO one in Equation 6.3.

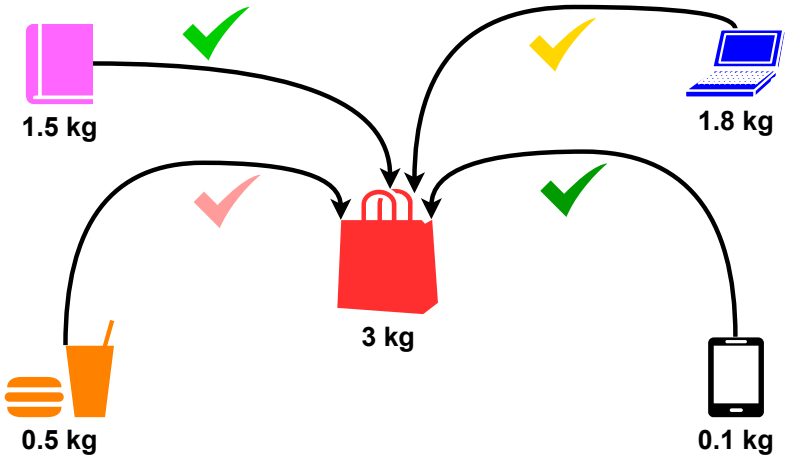
Figure 6.3 shows the equivalent Ising model for QUBO of Figure 6.1. Apart from an offset, it is possible to ascertain that the profiles are the same.

6.3 Problems overview

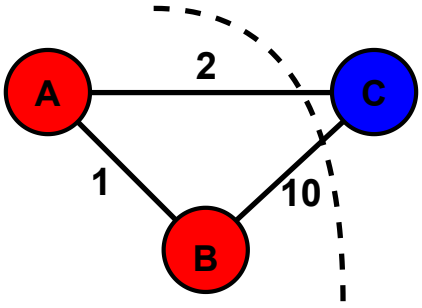
As already said in the introduction of this chapter, QUBO/Ising formalism permits to describe optimization problems employable in many application fields. An overview of two optimization problems is reported in the following.
An important family of QUBO problems is the **knapsack** one, whose conceptual scheme is reported in Figure 6.4a. Given a set of objects X, each of which is labelled as x_i and is characterized by a weight $w_i \in \mathbb{N}$, it is required to define the best sub-set to be put into a bag, ensuring that the total weight does not exceed a threshold $W \in \mathbb{N}$:

$$0 < \sum_{i=1}^{\dim(X)} w_i x_i \leq W \,. \tag{6.13}$$

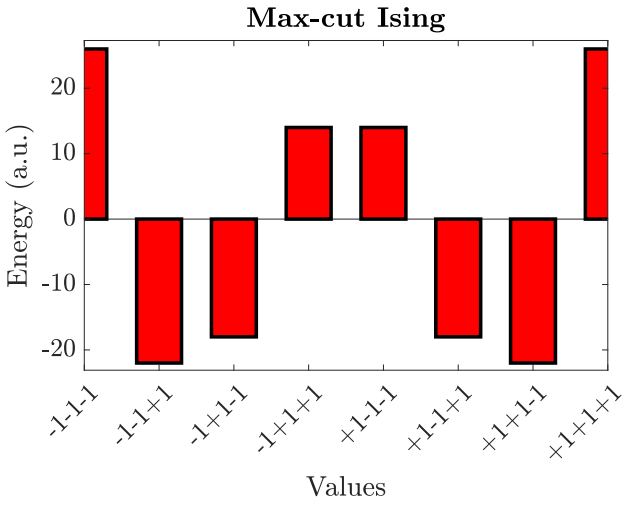
The upper bound constraint can be obtained straightforwardly according to Equation 6.7, while for the lower bound strict inequality case it is convenient to force a greater-or-equal constraint, forcing a lower threshold $\delta > 0$. This formulation will be employed for solving the Joint Transmission problem, described in Section 7.2.2.



(a) Knapsack problem, with different objects preferences.



(b) Three-vertices max-cut problem to be solved.



(c) Max-cut energy profile.

Figure 6.4: Overview of problems which can be formulated with QUBO model.

In addition to the inequality constraints, it is possible to force the preference of some objects. In this case, the cost function is written as:

$$f_{\text{knapsack}}(\mathbf{x}) = f_{\text{inequality}}(\mathbf{x}) - \sum_i p_i x_i \,, \tag{6.14}$$

where $p_i x_i$ are penalty contributions, to force the choice of specific objects. In particular, higher is the p_i associated with the i^{th} object, higher is the preference for its insertion. When $p_i = 1\,\forall i$, the penalty contributions tend to maximize the total number of objects to be inserted in the knapsack.
Another binary combinatorial problem employed in many practical scenarios is the **max-cut** for the partitioning of a graph into two complementary sub-sets S and $\overline{S}$, ensuring the maximization of the sum of weights over all the edges across the two vertex subsets. The simplest approach for mapping a max-cut problem is usually described by an **Ising** formulation of type:

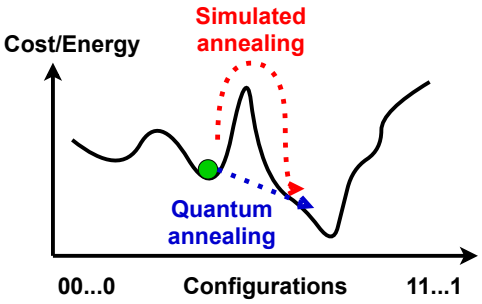
$$H(\mathbf{s}) = \sum_i \sum_{j \neq i} J_{ij} s_i s_j \,, \tag{6.15}$$

where J_{ij} is the weight of the edge between vertices i and j. Each vertex assumes value 1 if it belongs to the first subset of the graph S or -1 if it belongs to the complementary subgraph, $\overline{S}$. It is possible to prove that the minimum-energy state corresponds to the max-cut optimal solution. Figure 6.4 shows a three-vertices max-cut problem. It is possible to ascertain that the corresponding Ising energy profile is symmetric; this is due to the fact that complementary combinations, as $[-1;-1;+1]$ and $[+1;+1;-1]$ are conceptually equivalent, since they split vertices in the two halves in the same way.

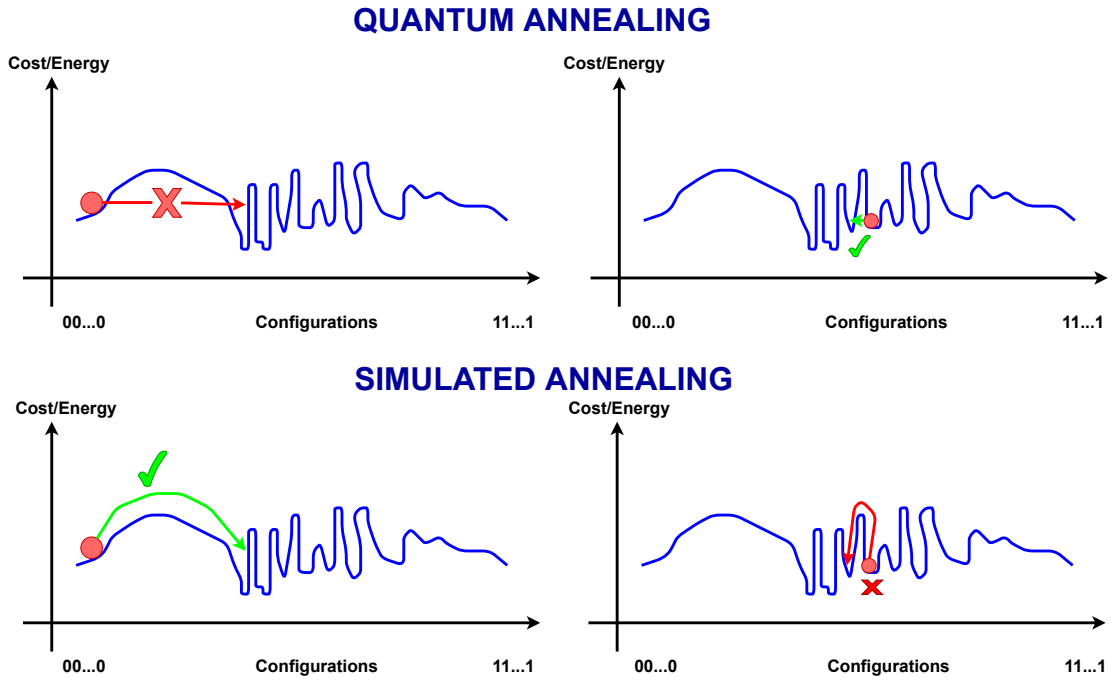
6.4 Approaches to QUBO problem solving

QUBO/Ising problems can be solved with different algorithms: annealing-based, genetic [82], population-based [83], *etc.* This section is focused on **simulated and quantum annealing techniques**. The second one can be seen as a *quantum version* of the first one, which simulates the **physical annealing** process for solids, which is able to modify a material's property through thermalization. In particular, the crystalline solid is heated over the so-called re-crystallization temperature. In this way, atoms are moved in the lattice; then the solid is cooled very slowly, to force an optimal settling of atoms in the crystal lattice, with minimum mechanical stress.
In other words, annealing algorithms try to achieve the minimum-energy configuration of the profile associated with the problem. More details about simulated and quantum annealing are reported in the following. In this context, taking into account Figure 6.5, it is sufficient to say that simulated annealing works well with

low and large barriers of the problem profile through *thermal barrier jumps*, while — on the contrary — quantum annealing is more efficient with high and narrow barriers thanks to the **tunneling** effect.



(a) Overcoming barriers.



(b) Exploring heterogeneous profiles.

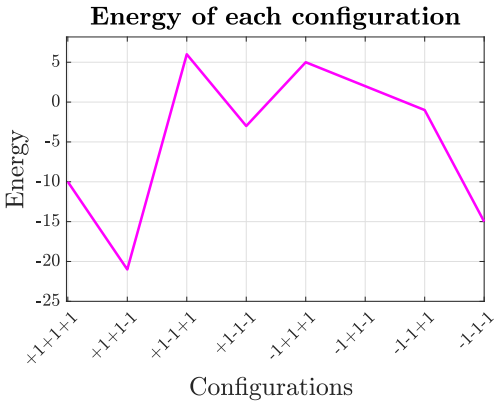
Figure 6.5: Comparison of simulated and quantum annealing techniques.

6.4.1 Simulated annealing

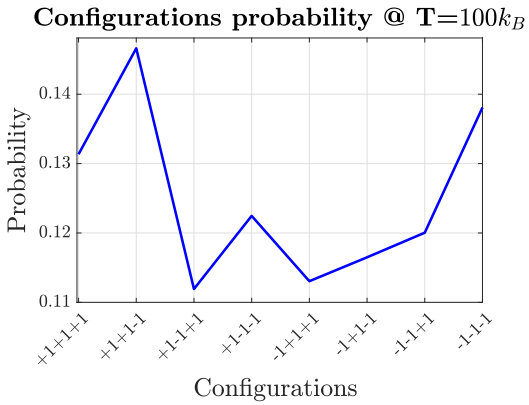
From a historical point of view, simulated annealing was proposed in 1983 [84]. As already written, it simulates physical annealing through thermalization; in order to do that, an **equivalent temperature**, defining the search radius in the solution

space, is introduced in the algorithm. At the beginning of the algorithm execution, temperature is high to ensure a high level of freedom in the solutions space exploration, *i.e.* that all configurations of the optimization problem have the same occurrence probability. Then temperature is lowered to limit the exploration. If process is properly done, it leads to a direct achievement of a low-energy configuration.

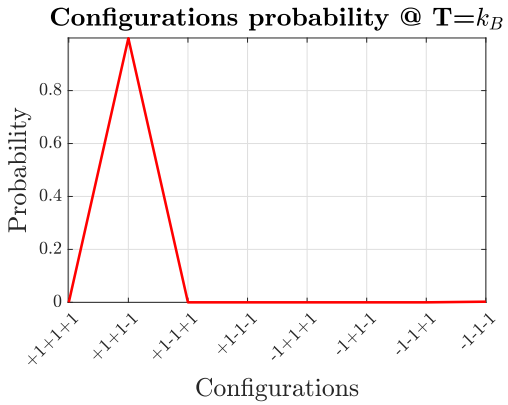
The main steps of simulated annealing are [85]:



(a) Reference energy profile.



(b) Probability distribution at the beginning of simulated annealing.



(c) Probability distribution at the end of simulated annealing.

Figure 6.6: Solving an Ising problem with simulated annealing: effects of temperature.

1. Consider an initial arbitrary solution S as the optimal one and set the temperature parameter T to a high value;

2. Consider a random neighbor S' of the solution S, obtained with an incremental local change.

3. Evaluate the energy difference between the two solutions $\Delta = H(S') - H(S)$.
4. Replace S with S', if $\Delta \leq 0$, otherwise do the substitution randomly with a probability $e^{\frac{-\Delta}{T}}$.
5. Reduce the temperature T.
6. Repeat from step 2, until $T = 0$.

Figure 6.6 shows a conceptual example of Ising problem solved with simulated annealing. For high temperatures, all solutions have a non-negligible probability of being obtained, while for low ones the optimal solution is achieved, if the annealing process was properly executed.
The solution update is done according to the **Metropolis-Hastings** algorithm [86]. In practice, the evaluation of a new solution depends on an **acceptance rate**, on its hand related to the Δ and T, which permits to avoid to be trapped in a local minimum of the profile. The acceptance rate is described by an exponential **Boltzmann distribution**:

$$P(S) = \frac{1}{\sum_{S \in \mathbb{S}} e^{-\frac{E(S)}{T}}} \cdot e^{-\frac{E(S)}{T}}, \tag{6.16}$$

where $\mathbb{S}$ is the set with all possible solutions of the optimization problem S. The update policy emulates **thermal fluctuations** of a physical system. Figure 6.7 shows

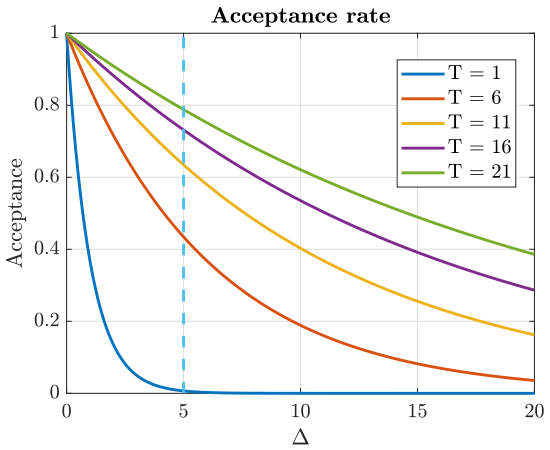


Figure 6.7: Acceptance rate, for different temperatures.

the acceptance rate dependence on $\Delta > 0$ (*i.e.* in a potentially worse situation), for different T values, It is possible to ascertain that acceptance rate decreases with increasing Δ and that, for a fixed Δ value, it is higher for higher T.
Defining the best evolution for the temperature parameter is an important issue for increasing the success probability. A thermal decrease of $T(t) = \frac{c}{\log(1+t)}$, with c

depending on the size and the characteristics of the problem, was proposed in [87] to achieve the optimal solution of a generic combinatorial optimization process.
Even though simulated annealing is a well-consolidated optimization problem solver, it is not completely satisfactory. In fact, it is not able to give good results for cost functions with high absolute derivative in some points, corresponding to high and narrow peaks in the energy profile, since thermal fluctuations are not generally efficient, unless very high initial temperature and long iteration time are not taken into account.
Simulated annealing can be implemented in software, thus permitting its execution on classical computing hardware, such as CPU.

6.4.2 Quantum annealing

Quantum annealing was proposed as an alternative of simulated annealing, where thermal fluctuations are replaced with **quantum fluctuations** [10]. From a practical point of view, quantum annealing is suitable for Ising problems, which can be embedded into a quantum annealer, whose system is characterized by a Hamiltonian:

$$\begin{aligned} H(t) &= -\sum_{ij} J_{ij}\sigma_{z_i}\sigma_{z_j} - h\sum_i \sigma_{z_i} - \Gamma(t)\sum_i \sigma_{x_i} = \\ &= H_0 - \Gamma(t)\sum_i \sigma_{x_i}, \end{aligned} \tag{6.17}$$

where H_0 is the standard Ising model on which the problem is mapped, $\Gamma(t)$ is the **time-dependent applied transverse field**, σ_{x_i} and σ_{z_i} are the Pauli matrices associated, respectively. The transverse field $\Gamma(t)$ has the same role of temperature parameter in simulated annealing. In particular, it stimulates tunneling between system eigenstates, encoding the possible solutions of the combinatorial optimization problem. This quantum phenomenon generates an equivalent **global search**, instead of a local one like that typical of simulated annealing and facilitates the overcoming of high and narrow barriers of the profile (as previously said). The achievement of the lowest-energy configuration requires that the amplitude of the transverse field must decrease slowly.
Figure 6.8 shows the schematic evolution of a quantum annealer for solving a two-variable Ising problem considering two equivalent reference systems: a spin system (the standard one) and a bathtub filled with water, which will have a final shape compatible with the problem to be solved. At the beginning, the high transverse field forces a quantum superposition state, where the eigenstates show equal probability amplitude. It can be intuitively seen as a mechanism where all spins are aligned in $\hat{x}$ direction and the bathtub is flat. If the transverse field is reduced sufficiently slowly (corresponding in the bathtub case to force a deformation), the instantaneous ground state of the Hamiltonian can be followed, until reaching the

ground state of H_0, corresponding to the optimal solution. The **adiabaticity** of the quantum evolution is undoubtedly the main issue of this approach.

Quantum annealing clearly requires quantum hardware. The most consolidated

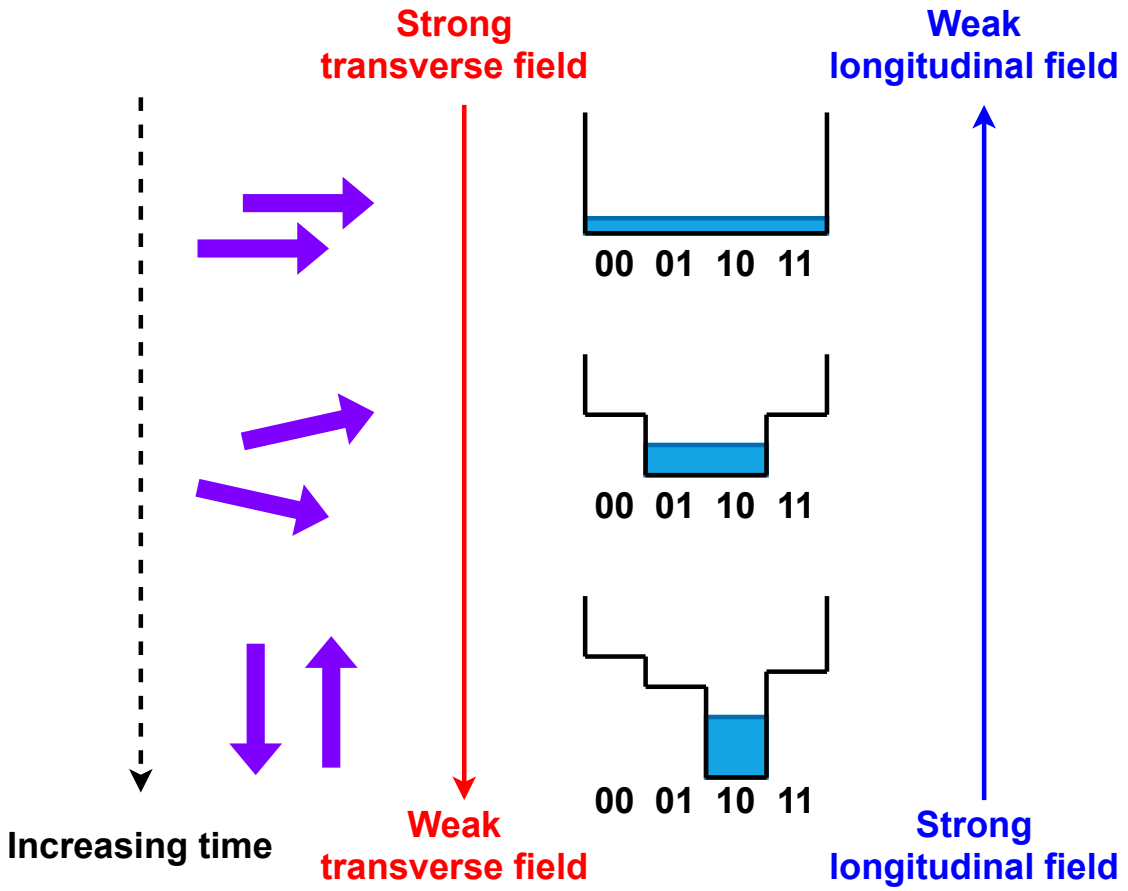


Figure 6.8: Conceptual scheme of quantum annealing evolution with spins and the bathtub filled with water.

technology for building these devices is the superconducting one and is affected by the issues introduced in section 1.3 concerning decoherence and connectivity. An alternative approach emerged in the last years consists in designing classical emulators of quantum annealers, to be implemented in hardware on FPGAs or in software on GPUs. In particular, they can be organized according to two main families:

- **Path integral quantum Monte Carlo**, emulating tunneling, superposition and entanglement through a set of replicas of the spin system. It was implemented on both GPU [88] and FPGA [89];
- **Simulated bifurcation**, proposed by Toshiba [90] and [91] and emulating

quantum annealers with a network of non-linear Kerr oscillators. It was implemented on FPGA and GPU.

Another possibility is to use quantum-inspired solvers, such as the **Digital annealer**, proposed by Fujitsu. This is able to verify in parallel and in real time all the possible adjacent solutions of the of the optimization problem (differing for the value of only one variable), in a way trying to simulate the superposition of the qubits states. This algorithm has been implemented in hardware, initially on FPGA [92], then on an Application-Specific Integrated Circuit (ASIC) [93].

Chapter 7

Quantum-inspired techniques for Joint Transmission allocation problem

Joint Transmission (JT) more commonly named as Coordinated MultiPoint (CoMP) in mobile access network, refers to a wide range of different techniques for transmission and/or reception. The main idea of CoMP is the dynamic coordination of transmission and/or reception at multiple geographically separated sites with the aim to enhance system performance and end-user service quality. If the user equipment (UE) receives signals from multiple cell sites and their transmitted signals are coordinated, the downstream performance experienced by the UE can be increased significantly. Also for the upstream flow, in case it is implemented, the performance can improve if the multiple cell sites receiving the signal by the same UE are coordinated. An optimization problem, looking for the best users who can be served in JT by a MIMO system, can be defined to improve the effectiveness of the mechanism.
A pure *brute-force* approach is not feasible, because the time required for evaluating all possible combinations grows very rapidly with the total number of involved users. As already said in subsection 1.5.2, execution time is critical, since the optimization problem shall be solved continuously at *run-time*, according to the continuos time variation of users linked to the mobile cellular network.
In order to reduce the time required for a global exploration of the solutions space, another preliminary selection problem, solvable heuristically, can be defined. For these reasons, a JT-served users scheduling algorithm, based on a QUBO model, was defined and is proposed in this chapter. As detailed in the following sections, this model is going to reduce the total number of users to be examined with the *brute-force* approach. QUBO formalism has been chosen to allow a potential future compatibility with both quantum computers, which are expected to provide, in the future, a computational advantage — in terms of both solution quality and

execution time — with respect to their classical counterparts.
After an introduction to JT formalism and to the optimization problem itself, subsection 7.2.1 and subsection 7.2.2 describe the current limitations in solving this problem with quantum computers and the proposed QUBO model respectively. After some details on software development in subsection 7.2.3, subsection 7.2.4 reports all the obtained results, employing both synthetic and simulator-provided data. Some final comments are reported in section 7.3.
This activity was done in a research contract between the Department of Electronics and Telecommunications of Politecnico di Torino and TIM S.p.A., commissioning the problem and providing the fundamental data for model construction and testing.

7.1 Theoretical model

Before starting with the protocol description, some basic concepts and the typical signal theory formalism of Telecommunications Engineering are presented. First of all, it is important to clarify that phasor-domain representation for sinusoidal signals is assumed to be always employed. The fundamental physical quantity is the electric power normalized to an implicit resistance $P_{\text{normalized}}$. Moreover, the relationship between $P_{\text{normalized}}$ and voltage is the following

$$P_{\text{normalized}} = |V|^2 , \tag{7.1}$$

thus implying that the measurement unit V^2 can be used to quantify normalized power (equivalently to W/Ω) and that the normalized power of a voltage signal v_i is given by its magnitude square $|v_i|^2$.
In a system involving N_{sig} signals, these are assumed to be not simultaneously transmitted, *i.e.* they are transmitted in a time interval Δt sequentially, each for a time duration $\frac{\Delta t}{N_{\text{sig}}}$. With this assumption, the total power will be equal to the sum of powers of each signal, which is different from the power of the sum of all signals:

$$P = \sum_{i=1}^{N_{\text{sig}}} |v_i|^2 \neq \left| \sum_{i=1}^{N_{\text{sig}}} v_i \right|^2 . \tag{7.2}$$

In the following, the formalism for MIMO wireless communication systems and JT, arranged from [94], is reported. Here, it is fundamental to make a premise: **the term *power*, whenever mentioned, is going to refer to the normalized power**.

7.1.1 MIMO model

Joint Transmission is going to be employed in MIMO systems, whose conceptual scheme is reported in Figure 7.1. It involves N_{BS} transmitting Base Stations (BSs)

and S receiving users; for sake of simplicity, each communication terminal has only one antenna. The mathematical model is based on **voltage signals**, so power calculation must follow the rules in Equation 7.1 and Equation 7.2.

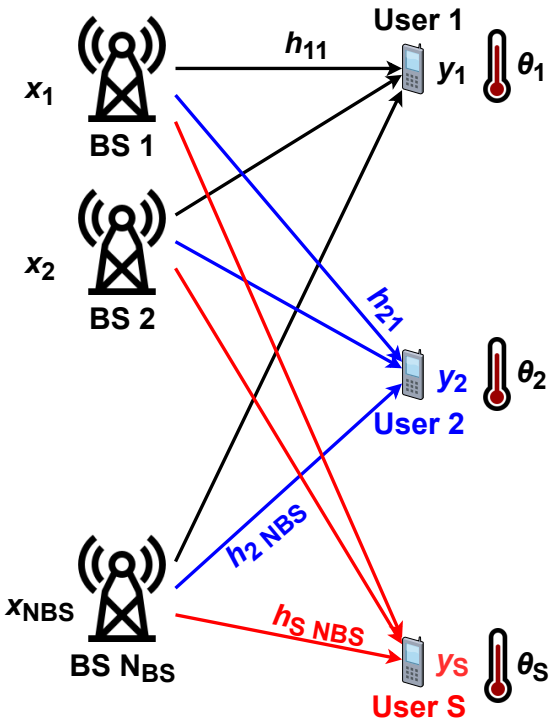


Figure 7.1: Conceptual scheme of a wireless MIMO system.

Each BS transmits a symbol x_n, which is received by a user s as:

$$y_s = h_{sn} x_n, \tag{7.3}$$

where h_{sn} is a channel coefficient related to path-loss attenuation and random phase shift of the transmitted signal. In a linear MIMO system with N_{BS} base stations and S users, it is possible to write the transfer function of the whole system as a matrix-vector product:

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_S \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & \cdots h_{1N_{\mathrm{BS}}} \\ h_{21} & h_{22} & \cdots h_{2N_{\mathrm{BS}}} \\ & \ddots & \ddots \\ h_{S1} & h_{S2} & \cdots h_{SN_{\mathrm{BS}}} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{N_{\mathrm{BS}}} \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \\ \vdots \\ n_S \end{bmatrix} = H\mathbf{x} + \boldsymbol{\theta}\,, \tag{7.4}$$

where $H \in \mathbb{C}^{S \times N_{\mathrm{BS}}}$ is the complex **channel matrix** and $\mathbf{x}$, $\mathbf{y}$ and $\boldsymbol{\theta}$ are **signal vectors** associated, respectively, with transmitted, received and noise signals. The last one is always present, provides an additive contribution to the received signal

and is usually considered thermal in this model. According to Equation 7.4, the total signal received by user s is given, apart from the thermal noise signal θ_s, by the product between the s^{th} row of H and $\mathbf{x}$.
In a standard MIMO system, signal transmitted by each base station corresponds to a symbol b_s to be received by user s. For sake of simplicity, it is assumed in the following analysis that each user receives only one symbol; this implies that $N_{\text{BS}} = S$ and the signal $x_s = b_s$, transmitted by the base station s, is going to be sent to user s. The signal received by the same can be written as:

$$y_s = \sum_{n=1}^{N_{\text{BS}}} h_{sk} x_n + \theta_s = \overbrace{h_{ss} x_s}^{\text{useful}} + \overbrace{\sum_{n \neq s} h_{sn} x_n}^{\text{interference}} + \overbrace{\theta_s}^{\text{noise}} . \tag{7.5}$$

Equation 7.5 puts in evidence that each users does not only receive the useful channel-modulated signal $h_{ss}x_s$ and thermal noise, but also the signals transmitted by other base stations, which constitute an interference contribution. In order to quantify the effects of noise and interference, it is possible to compute for each received signal the Signal-to-Interference-plus-Noise Ratio(SINR) as:

$$\text{SINR} \triangleq \frac{P_{\text{useful}}}{P_{\text{interference}} + P_{\text{noise}}} = \frac{|h_{ss} x_s|^2}{\sum_{n \neq s} |h_{sn} x_n|^2 + \sigma_{n_s}^2} , \tag{7.6}$$

where $\sigma_{n_s}^2$ is thermal noise power. Interference and noise significantly affect the maximum transmission rate of information over a noisy communication channel. In fact, according to the Shannon-Hartley formula, the transmission rate r over a channel of bandwidth B can be computed as:

$$r \triangleq B \log_2(1 + \text{SINR}) , \tag{7.7}$$

thus implying that transmission rate can be increased by increasing the bandwidth or the SINR. In a practical wireless telecommunication scenario, bandwidth cannot be changed, so the only way for increasing r is increasing SINR, reducing noise and interference or amplifying the useful signal.

7.1.2 Joint transmission mechanism

Joint Transmission (JT) goal is removing interference.

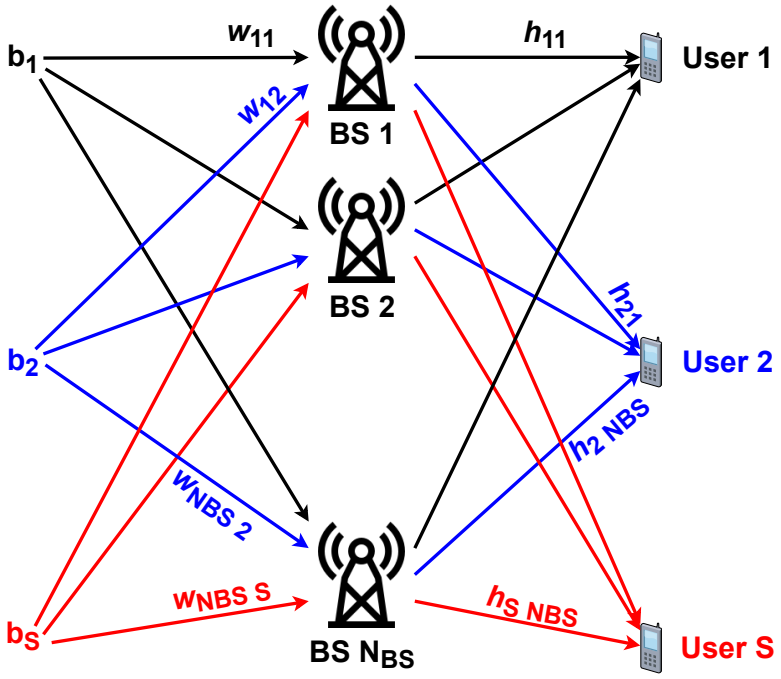


Figure 7.2: Conceptual scheme of a MIMO system with Joint Transmission.

According to Figure 7.2, symbols b_s are distributed over all the base stations through **beamforming** technique, so each base station transmits a signal which is the linear combination of all the involved symbols:

$$\begin{aligned} x_n &= \sum_{s=1}^{S} w_{ns} b_s, \\ \mathbf{x} &= \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{N_{\mathrm{BS}}} \end{bmatrix} = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1S} \\ w_{21} & w_{22} & \cdots & w_{2N_{\mathrm{BS}}} \\ & \ddots & \ddots & \\ w_{N_{\mathrm{BS}}1} & w_{N_{\mathrm{BS}}2} & \cdots & w_{N_{\mathrm{BS}}S} \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_S \end{bmatrix} = W\mathbf{b}\,. \end{aligned} \tag{7.8}$$

W is the $N_{\mathrm{BS}} \times S$ **beamforming matrix** and it describes how symbols in vector $\mathbf{b}$ are distributed over all the N_{BS} base stations, which on their hand transmit signals belonging to vector $\mathbf{x}$. Substituting Equation 7.8 in Equation 7.4, it is possible to prove that received signals are equal to

$$\mathbf{y} = HW\mathbf{b} + \boldsymbol{\theta}\,. \tag{7.9}$$

The main difference between matrices H and W is that each element h_{sk} of the first depends on the environment between two terminal antennas, while the whole W matrix can be engineered. Since it is desired to maximize the similarity between $\mathbf{y}$ and $\mathbf{b}$, the beamforming matrix should be chosen to have:

$$HW = \mathbb{I}_{\dim\{S\}}\,, \tag{7.10}$$

where $\mathbb{I}_{\dim\{S\}}$ is the identity matrix with size $\dim\{S\}$. If this condition is satisfied, any symbols interference does not take place and the signal received by each user is only affected by thermal noise:

$$y_s = x_s + \theta_s \,. \tag{7.11}$$

Finally, the transmission rate with JT mechanism is:

$$r_s = B \log_2 \left(1 + \frac{|b_s|^2}{\sigma_{n_S}^2} \right) . \tag{7.12}$$

The condition in Equation 7.10 can be achieved by forcing the beamforming matrix equal to

$$W = H^{\dagger} \,, \tag{7.13}$$

where $H^{\dagger}$ is the **Moore-Penrose inverse** matrix of W. When square matrices are involved ($N_{\text{BS}} = S$), the Moore-Penrose inverse matrix corresponds to the standard inverse.
Beamforming substantially pre-compensates all the channel effects, to ensure that the received signal has the same form of Equation 7.11. However, since channel provides power attenuation, the satisfaction of Equation 7.13 also requires signal amplification, but it would possible that the required base station transmission power $\sum_{s=1}^{S} |w_{ns} b_s|^2$ (where w_{ns} the element of the beamforming matrix on the n^{th} row and the s^{th} column) is higher than its maximum supported $P_{\max_{\text{BS},n}}$. In order to ensure that $\sum_{s=1}^{S} |w_{ns} b_s|^2 \leq P_{\max_{\text{BS},n}}$, power attenuation factor p_s must be defined for each symbol; from a mathematical point of view, the effect of each power attenuation factor is taken into account with a scalar $\sqrt{p_s}$ to be multiplied with each symbol b_s. The received signals vector and transmission rate of user s can be written as:

$$\begin{aligned} \mathbf{y} &= HW \begin{bmatrix} \sqrt{p_1} b_1 \\ \sqrt{p_2} b_2 \\ \vdots \\ \sqrt{p_S} b_S \end{bmatrix} + \boldsymbol{\theta} = (\sqrt{\mathbf{p}} \circ \mathbf{b}) + \boldsymbol{\theta}, \\ r_s &= B \log_2 \left(1 + \frac{p_s |b_s|^2}{\sigma_{n_S}^2} \right) , \end{aligned} \tag{7.14}$$

where $\sqrt{\mathbf{p}}$ is the vector with the square root of each power attenuation and $\circ$ is the element-wise product. For sake of simplicity, $b_s = 1\,\text{V}$, thus $|b_s|^2 = 1\,\text{V}^2$.
All data required for the definition of the scheduling problem are available. Given the channel matrix H of a MIMO system involving N_{BS} base stations and N_{UE} potential users ($N_{\text{BS}} = 3$ and $N_{\text{UE}} = 30$ are usually considered in current research works concerning JT users allocation), the subset of users S_{served} served by Joint

Transmission with *maximum total transmission rate* is going to be found. In particular, considering all possible S_{served}, the following optimization problem is going to be solved [94]:

$$\begin{aligned} &\textbf{find} \ \max\{r_{S_{\text{served}}}\} = \max\left\{\sum_{s\in S_{\text{served}}} B\log_2\left(1+\frac{p_s}{\sigma_n^2}\right)\right\}, \\ &\textbf{with} \ 0 < P_n = \sum_{s\in S_{\text{served}}} |w_{ns}|^2 p_s \leq P_{\max_{\text{BS},n}} \quad \forall n \in [1; N_{\text{BS}}]. \end{aligned} \tag{7.15}$$

The optimization problem substantially requires two operations: **iterating among subsets** and **finding the optimal power attenuation factors for each subset**.
The generic theoretical model and the definition of the optimization problem have been presented. In the following, different approaches for achieving the optimal subset, implementing these tasks, are going to be presented. These will be fundamental for the definition of a global algorithm for allocating users.

7.1.3 Choice of power attenuation factors

Equation 7.15 puts in evidence that power attenuation factors p_s are critical for solving the allocation problem, since their values constitute a compromise between the transmission rate maximization and the satisfaction of power constraint. In the definition of an allocation algorithm, the choice of an approach for finding these factors is clearly necessary.
A very simple — on the edge of *naïf* — approach could be obtained by assuming that all users belonging to S_{served} have the same power attenuation factor and that only one base station transmits to maximum power. In particular, given a potential subset S_{served}, the factor permitting a maximum transmission power is found for each *base station* independently on others. At this point N_{BS} potential p_s are available and the lowest one is chosen, thus ensuring that each base station does not overcome the power threshold. An example with $N_{\text{BS}} = 3$ and $\dim\{S_{\text{served}}\}$ is reported in the following:

$$\begin{aligned} \begin{bmatrix} |w_{11}|^2 & |w_{12}|^2 \\ |w_{21}|^2 & |w_{22}|^2 \\ |w_{31}|^2 & |w_{32}|^2 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} &= \begin{bmatrix} |w_{11}|^2 p_1 + |w_{12}|^2 p_2 \\ |w_{21}|^2 p_1 + |w_{22}|^2 p_2 \\ |w_{31}|^2 p_1 + |w_{32}|^2 p_2 \end{bmatrix} = \\ &\overset{\text{if } p_1=p_2}{=} \begin{bmatrix} (|w_{11}|^2 + |w_{12}|^2)p \\ (|w_{21}|^2 + |w_{22}|^2)p \\ (|w_{31}|^2 + |w_{32}|^2)p \end{bmatrix} \leq \begin{bmatrix} P_{\max_{\text{BS}}} \\ P_{\max_{\text{BS}}} \\ P_{\max_{\text{BS}}} \end{bmatrix}. \end{aligned} \tag{7.16}$$

$$p = \min\left\{\begin{bmatrix} P_{\max_{\text{BS}}}/(|w_{11}|^2 + |w_{12}|^2) \\ P_{\max_{\text{BS}}}/(|w_{21}|^2 + |w_{22}|^2) \\ P_{\max_{\text{BS}}}/(|w_{31}|^2 + |w_{32}|^2) \end{bmatrix}\right\}$$

The main limitation of this simple approach is that two antennas could transmit information with a power much lower than the maximum one, so JT cannot be executed at its best.
In order to have a better tuning of the parameters, it could be convenient to exploit more sophisticated approaches. Convex optimization is a serious candidate, since the problem to be solved is convex. Many optimizers are available in terms of software libraries to be employed in environments for numerical calculus. This is the case of CVX [95] for MATLAB and CVXPY [96] for Python.

7.2 Scheduling algorithms

The hardest computational task in the analysis of JT MIMO wireless systems is the choice of the optimal served users subset. The simplest approach for finding this is a ***brute-force*** one, which evaluates all possible subsets. Its description is reported in Algorithm 1. Calculation of power attenuation factors p_s can be done by exploiting one of the two methods proposed in subsection 7.1.3. The main limitation of the *brute-force* approach is the factorial scaling of the number of iterations with the number of potential users N_{UE}. In fact, if S must be selected, the total number of subsets is given by the binomial coefficient:

$$N_{\text{subsets}} = \binom{n}{k} = \frac{n!}{k!(n-k)!}, \tag{7.17}$$

which implies the feasibility of the approach only with few users ($N_{\text{UE}} \lesssim 20$). However, even if computationally hard, *brute-force* must not be completely neglected, since it ensures in any case to find the optimal solution. For this reason, it is preferable to keep it, trying to limit the total number of users on which it is applied. A possible strategy could consist in doing a preliminary selection of users with a faster approximated optimization and to exploit the *brute-force* method only on this selected subset, thus reducing the total number of combinations processed by it.

7.2.1 Applying quantum computing to Joint Transmission

The main features of quantum computers, independently from the computational paradigm (section 1.2), are reported in the following:

- Thanks to superposition, a quantum computer can simultaneously evaluate the same operation on multiple eigenstates, each one associated with a datum.
- Quantum algorithms are probabilistic, so obtaining the optimal solution of a problem is not ensured. Moreover, the probability of getting the optimum is further reduced by errors and non-ideality phenomena.

- Current quantum hardware is very limited, for what concerns both the number of employable qubits and reliability of its native operations. Even though classical simulators of quantum computers can help in evaluating the results provided by larger quantum circuits involving ideal qubits, they can require high memory allocation.

According to these features and what described in section 7.1, the complete replacement of a classical computer with a quantum computer in JT scheduling problem is currently unfeasible, especially for what concerns p_s calculation. Suffice it to say that a systematic quantum advantage in the execution of arithmetic and algebraic operations, fundamental in this optimization problem, is not guaranteed.
Previous observations permit to conclude that, even though quantum computing cannot solve on its own the JT users allocation problem, it can be exploited to accelerate its resolution, assisting classical computing in specific tasks which are hard for it. A quantum-classical co-processing methodology is here proposed and its conceptual block scheme is available in Figure 7.3.

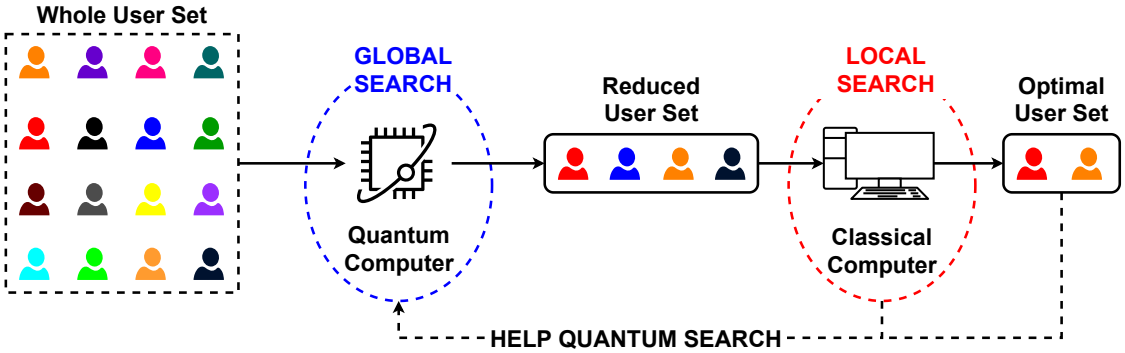


Figure 7.3: Block scheme of a hybrid solver for Joint Transmission allocation problem.

Quantum computer is responsible of a first selection of users among the N_{UE} available and it provides to the classical computer a potential solution or a set of users smaller than the initial one. A *brute-force* search of the optimal solution is then executed on the subset. If quantum computer is properly calibrated, the obtained solution $S_{\mathrm{served}_{\mathrm{algorithm}}}$ will correspond to $S_{\mathrm{served}_{\mathrm{best}}}$, achieved with the *brute-force*-only strategy. In order to achieve $S_{\mathrm{served}_{\mathrm{best}}}$, classical computer can provide a feedback to the quantum unit in order to increase its selectivity, thus implementing an iterative procedure, along the lines of consolidated variational quantum algorithms, such as the Variational Quantum Eigensolver [8] and the Quantum Approximate Optimization Algorithm (QAOA) [1], [9]. In all these algorithms, quantum computer is a *probe* of the whole solutions space, which is iteratively *calibrated* by a classical computing unit according to the results provided by last iteration.

A first hybrid approach based on QUBO model, thus runnable on quantum annealers, is reported in subsection 7.2.2. This modelling has been chosen since both real quantum annealers and classical emulators permit to manage a higher number of variables than their corresponding ones based on quantum circuit model.

Algorithm 1 Find $S_{\text{served}_{\text{best}}}$, with corresponding $r_{\max}$.

Require: channel matrix H, involving all users
Require: thermal noise σ_n^2
Require: maximum transmission power of each base station $P_{\max_{\text{BS},n}}$
  $r_{\max} = 0$
  for all S_{served} **do**
    $H_{\text{current}} \leftarrow H(S_{\text{served}})$
    $W_{\text{current}} \leftarrow H_{\text{current}}^{\dagger}$
    compute $p_s(W_{\text{current}}, P_{\max_{\text{BS},n}})$
    $r_{S_{\text{served}}} \leftarrow \sum_{s \in S_{\text{served}}} B \log_2 \left(1 + \frac{p_s}{\sigma_n^2}\right)$
    if $r_{S_{\text{served}}} > r_{\max}$ **then**
      $r_{\max} \leftarrow r_{S_{\text{served}}}$
      $S_{\text{served}_{\text{best}}} \leftarrow S_{\text{served}}$
    end if
  end for
  return $S_{\text{served}_{\text{best}}}$, $r_{\max}$

7.2.2 QUBO formulation for Joint Transmission

The first proposal of quantum-compliant JT problem solving strategy is substantially a sophisticated QUBO **knapsack problem** (section 6.3), looking for the maximization of the total Signal-to-Noise ratio (thus of the transmission rate), without overcoming the transmission power limits of each BS. The proposed model is described in the following; before starting, it is important to precise that the classical computer must provide to the quantum counterpart all the terms related to beamforming $|w_{ns}|^2$ and p_s, which are the QUBO weights of the knapsack problem..
The total number of variables of the proposed QUBO model is $N_{\text{UE}} \cdot N_{\text{BS}}$, where N_{UE} and N_{BS} are the total number of users and BSs respectively. In order to simplify the understanding of the organization of these variables, it is convenient to consider their arrangement according to a **matrix structure** $N_{\text{UE}} \times N_{\text{BS}}$, *i.e.* a grid like that in Figure 7.4a. **Each line is associated with a user and each column is associated with a BS**. This data structure is thought for ensuring to each BS a subset of exclusive boolean variables for the N_{UE} users, so a specific knapsack problem is going to be solved for each transmitting antenna, moreover in

parallel to the others. The choice of an unique knapsack problem, where the same variables are employed for evaluating the power constraints of each BS, would seem extremely hard to be solved, from a computational point of view. Each grid variable x_{sn} is equal to 1 (red circles in Figure 7.4a) if BS n serves user s. Since QUBO result is a set of potential users, the problem substantially corresponds to finding the best $x_{s,n} = 1$.
The proposed QUBO cost function presents four terms:

1. Maximization of SNR

$$f_{\text{cost}}(\mathbf{x}) = -\sum_{s} \frac{p_s}{\sigma_\theta^2} \sum_{n} x_{s,n}, \tag{7.18}$$

   where the minus sign is employed to convert a maximum value search into a minimum one. This term tends to force all $x_{s,n}$ to 1. As observable in Figure 7.4b, boolean variables belonging to the same row/user have the same p_s. Moreover, different purple intensities refer to the different p_s, and consequently SNR, of users.

2. BS serving exactly the same users. This is equivalent to say that, given a row of the grid, all the variables must be equal to each other (Figure 7.4c), *i.e.* the sum of their values must be 0 or N_{BS}. This row constraint is thought for ensuring that the knapsack problems, which are reminded to be one for each BS, choose the same users, as requested by JT.

$$c_{\text{user on all BS}}(\mathbf{x}) = \left[\left(\sum_{n} x_{s,n}\right) - N_{\text{BS}}\right]^2 \cdot \left(\sum_{n} x_{s,n}\right)^2 \quad \forall s. \tag{7.19}$$

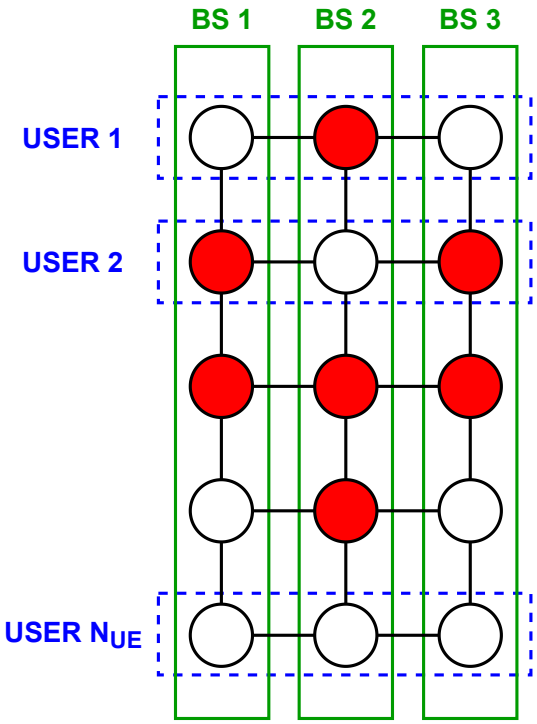
3. Number of users served by each BS equal to $\dim\{S_{\text{served}}\}$. From a practical point of view, each column must have $\dim\{S_{\text{served}}\}$ variabiles equal to 1 (Figure 7.4d). This constraint avoids that a BS does not serve any user.

$$c_{\text{user on single BS}}(\mathbf{x}) = \left[\left(\sum_{s} x_{s,n}\right) - \dim(S_{\text{served}})\right]^2 \quad \forall n. \tag{7.20}$$

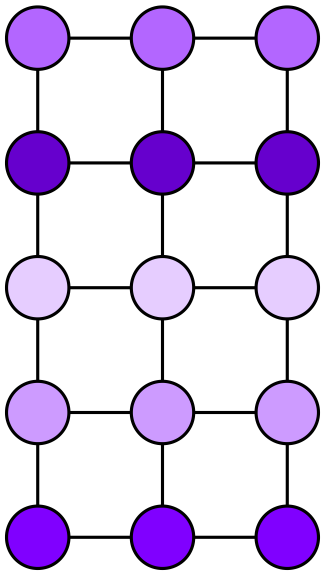
4. Knapsack problem concerning transmitted power of BSs for a given users subset:

$$c_{\text{power BS}}(\mathbf{x}) = c\left(0 < \sum_{s \in \{s_1, s_2, s_3\}} |w_{s,n}|^2 p_s x_{s,n} \leq P_{\max}\right) \quad \forall \{s_1, s_2, s_3\}, \forall n. \tag{7.21}$$

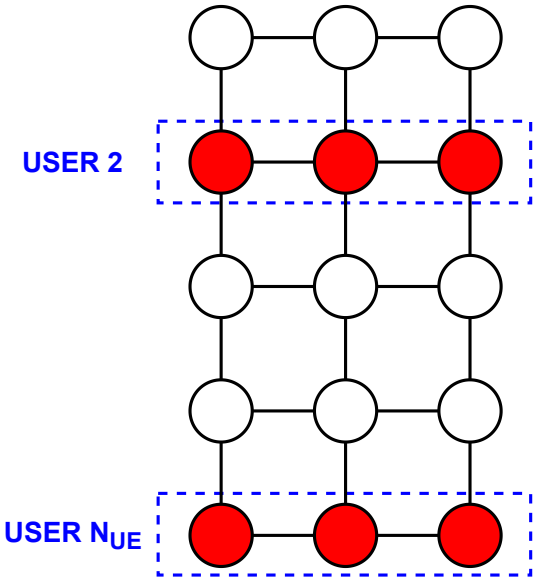
   In order to simplify knapsack evaluation, power values $|w_{s,n}|^2 p_s$ and P_n are quantified in mW or tens of mW and treated as integer with truncation (*e.g.* 39.7 mW is considered as 39 mW).



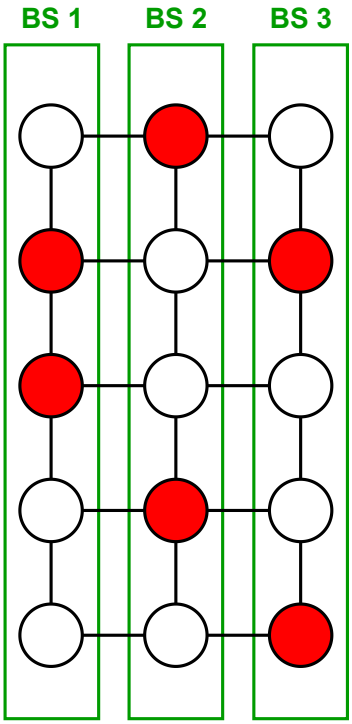
(a) Boolean variables grid.



(b) Contribution associated with the maximization of SNR.



(c) Constraint concerning equality of users served by all base stations.



(d) Constraint on the number of users served by each base station.

Power and row-column constraints tend to compensate each other. In fact, the first one would force all variables to 0, while the others — even before limiting the number of users to be served — try to ensure that $\exists x_{s,n} : x_{s,n} = 1, \forall n$. It is also possible to say that row and column constraints link the different knapsack problems by forcing them to choose $\dim(S_{\text{served}})$ users belonging to the same row without overcoming the transmission power limits of a BS.
The current critical features of this model are reported in the following:

- The whole channel matrix must be known.
- All the arithmetic and algebraic calculations are done by a classical computer and are provided to QUBO in terms of constraint contributions.
- Model is sensible to constraint weights α_k.
- The achievement of the ideal solution, where the optimal $\dim\{S_{\text{served}}\}$ rows are coloured[1], is not guaranteed.

While the first two features are intrinsic of the model, the others can be managed and arranged. Cost function sensitivity with respect to α_k could be characterized and controlled by the software responsible of building the QUBO with or without classical-to-quantum feedback mechanism[2], while a solution not corresponding to the optimal one can be properly arranged:

- If, as usually expected, the number of users chosen by QUBO would be greater than the expected $\dim\{S_{\text{served}}\}$, the QUBO solution will be considered a subset of potential users among which the best subset is going to be chosen.
- If variables belonging to a row were not all equal to each other, considering transmitting base stations are usually odd, **majority-voting** could be employed, so all rows where variables equal to 1 are more than those equal to 0 would be chosen. If the configuration in Figure 7.4a is taken into account, users 2 and 3 would be considered valid solutions.

7.2.3 Software development

The proposed model has been implemented in software; in particular, a Python library — based on to the Object-Oriented Programming (OOP) paradigm — has been developed for solving the Joint Transmission with the approaches previously described. In order to develop this library, an interface between `qubovert` and CVXPy, which are thought for solving optimization problems based on QUBO and

[1] The configuration in Figure 7.4c could be an optimal solution for $\dim\{S_{\text{served}}\} = 2$.

[2] Calibration is currently manual and feedback is not implemented.

convex formalism respectively, has been developed.
Each class is associated with a different solver, so it is possible to continuously update the available solving approaches. The currently available solvers are:

- `JointTransmissionBruteForceSolver`, which employs only the brute-force approach. It supports both convex optimization and *naïf* approaches.
- `JointTransmissionQUBOAssistedSolver`, based on the proposed methodology, where QUBO selects potential users and a brute-force solver, *i.e.* a `JointTransmissionBruteForceSolver` object, finds the optimal subset.

Before building the QUBO model, it is possible to choose the resolution of the trasmission power: $\Delta P = 1\,\text{mW}$ or $\Delta P = 10\,\text{mW}$. The choice of this value affects the total number of involved variables, since different knapsack problems are defined.
In all the executed tests, QUBO was solved with *simulated annealing* solvers, provided by libraries `qubovert` and `neal` in D-Wave Ocean SDK, but interfacing the model with other solvers, *e.g.* a real quantum annealer, is possible. When CVXPy is involved, the Splitting Conic Solver (SCS) [97] is always employed.
The main advantage provided by OOP is that each object/solver can work on its own; this implies that a JT problems with different subsets of BSs, each involving different users, can be solved in parallel, *e.g.* employing multi-thread programming. The parallel evaluation of multiple JT problems is a typical scenario, since — in an environment with tens of BSs and practical JT involving three of them — users are separated in subgroups, one for each BS triplet.
Apart from constraints weights calibration, currently done *by hand*, other issuess observed in the development were:

- **The set of potential users could be lower than dim(S_{served}).** In order to overcome this, weights of the constraints related to transmission rate maximization and power were temporarily doubled and halved respectively, because this variation is expected to permit the achievement of a QUBO solution with a larger subset of potential users. This procedure can be repeated multiple (currently five) times: if QUBO does not achieve a reasonable solution after these repetitions, JT problem is only solved with brute-force-*naïf* approach.
- **Convex optimizer could provide negative power attenuation factors.** In this case, the *naïf* approach is immediately employed, on the same subset of potential users, thus ensuring a valid solution.

7.2.4 Results

Different tests have been done for the validation of the proposed model. It is important to precise that all these tests were executed on the D-Wave Neal **simulated**

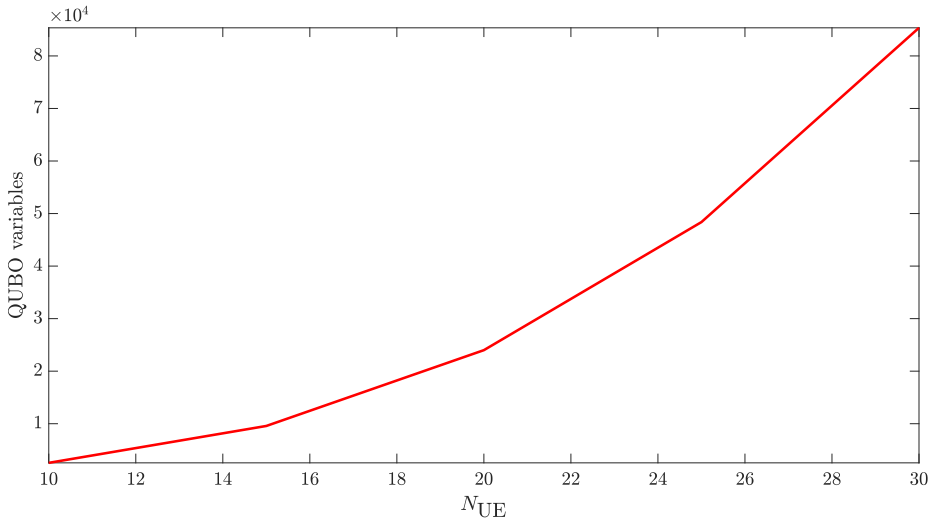


Figure 7.5: Total number QUBO variables, with different total number of users.

annealer (sampling fifty values), instead of a quantum annealer, mainly for two reasons. The first one is that the defined QUBO problems involve a number of variables significantly high for a real quantum annealer, which could be further increased after minor embedding. Figure 7.5 shows the total number of QUBO variables in different problems involving N_{UE} users. It is possible to ascertain that in all cases these are in the order of thousands.

The other reason is that the simulated annealing algorithm can be run on local

Parameter	Random	Simulator-based
Power resolution ΔP	10 mW	
Initial power partition factors	random, with upper limit $3 \cdot 10^{-6}$	random, with upper limit $5 \cdot 10^{-14}$
Weight of number of served users constraint	100	
Weight of transmission power constraint	$[5 \cdot 10^{-6}, 5 \cdot 10^{-6}, 5 \cdot 10^{-6}]$	
Transmission rate weight	2.5	

Table 7.1: QUBO building parameters.

machines, thus making easier, at least initially, the validation of the algorithm with repeated tests. Moreover, at the time of development and testing, a quantum annealing emulator was not available, so it was not possible to exploit this additional

on-premises solution.
Tests can be divided into two main sets, depending on the type of subchannel employed for testing the scheduling algorithm:

- **random**, where a random subchannel matrix was generated for a single triangle of BS, changing in different tests the total number of users involved in JT;
- **simulator-based**, where the subchannel matrices of a MATLAB-based MIMO simulator available in TIM S.p.A. — involving 52 subchannels, 57 BSs and 570 users — were taken into account.

All parameters were computed by considering each BS triplet independently on the others. In all the examined cases, the matrix H of each subchannel was computed as the element-by-element product of two matrices:

$$H = L \circ P \,, \tag{7.22}$$

where L is the path loss matrix in linear units, *i.e.* each element is equal to $10^{\frac{\text{Path Loss}}{20}}$, and P is a complex *dephasing* matrix, associated with the phase shift of each signal during its transmission from a base station to an user. Details about the construction of L and P matrices are available in the specific subsections of each set. Noise power was always considered constant and equal to -109 dBm, on one hand for sake of simplicity, on the other to stay in a worst-case scenario. QUBO building parameters are reported in Table 7.1. Apart from the initial power partition factors, the other constraints were the same in all the executed tests; the power resolution for knapsack problems was always fixed to $\Delta P = 10\,\text{mW}$, to reduce the total number of involved variables in each QUBO problem to be solved.
The metrics employed in the two types of tests — reported in the two following subsections — were slightly different. This is due to the fact that tests were done sequentially, starting from random-matrix-based ones. The experience acquired with the first tests permitted to improve those with matrices provided by the MATLAB MIMO simulator (which were undoubtedly more critical and interesting from an applicative point of view), exploiting metrics which could permit to better evaluate the effective capabilities of the proposed solving methodology.
In conclusion, it is important to precise that all the following results refer to tests executed on the server VLSI22 of the VLSI Lab of the Department of Electronics and Telecommunications of Politecnico di Torino, with Intel Xeon Gold 6134 CPU and a 100 GB RAM.

Random matrices

The first tests were done on a single BS triplet, varying the total number of users involved in the optimization problem N_{UE} and employing H matrices generated

by the element-by-element multiplication of L and P random matrices. In particular, L random values belonged to an uniform distribution between $-110\,\mathrm{dB}$ and $-70\,\mathrm{dB}$, while the random real and imaginary parts of each complex number $a + \mathrm{i}b$ of P where obtained according to a Gaussian distribution, with mean value and standard deviation equal to $\frac{1}{\sqrt{2}}$ and 1 respectively.
In these random-matrices-based tests, convex optimization library CVXPy was employed to optimize the power partition factors, in both the *brute-force*-only solver and in the *brute-force* step of the proposed algorithm. For each N_{UE} value, tests were repeated one hundred times, to estimate statistical figures of merit with different H random matrices. The optimal solution rate was initially employed:

$$o_r \triangleq \frac{n_{\mathrm{opt.}}}{N_{\mathrm{tot.}}} \cdot 100\,[\%]\,, \tag{7.23}$$

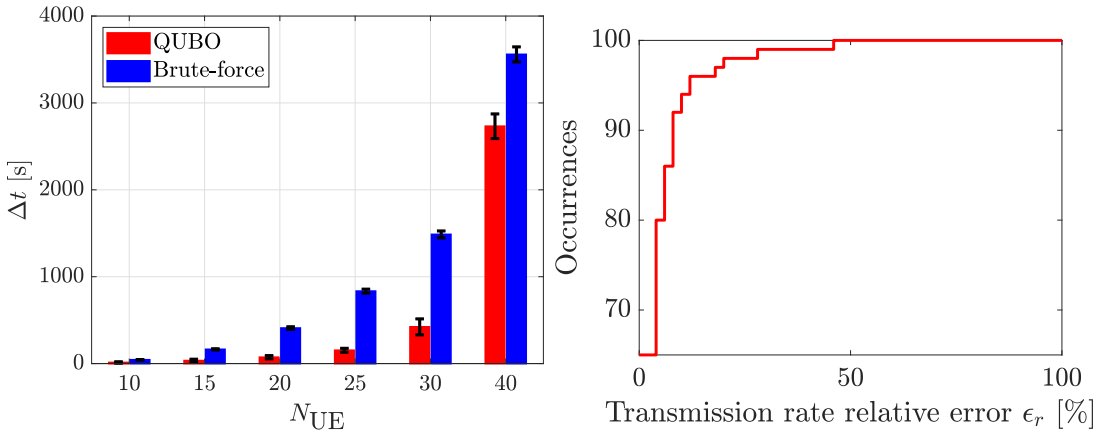
where $n_{\mathrm{opt.}}$ is the total number of times in which the optimal solution (identified with the *brute-force* solver on all possible users) was found and $N_{\mathrm{tot.}} = 100$ is the total number of repetitions of the execution. However, since the achievement of the optimal solution is not always ensured, the model capability of approaching the optimal solution was evaluated. For this goal, **transmission rate relative error** was exploited:

$$\epsilon_r \triangleq \left| 1 - \frac{r_{\mathrm{found}}}{r_{\mathrm{opt.}}} \right| \cdot 100\,[\%]\,, \tag{7.24}$$

where r_{found} is the transmission rate obtained with the proposed method and $r_{\mathrm{opt.}}$ is the optimal transmission rate, given by a *brute-force* approach.
Finally, latencies of the QUBO-based and *brute-force*-only approaches were computed for each test repetition to prove an eventual computational speed-up provided by the proposed approach.

Table 7.2: Optimal solution rate with random H matrices and $n_{\mathrm{repetitions}} = 100$.

N_{UE}	o_r
10	62%
15	71%
20	61%
25	54%
30	42%
40	45%



(a) Comparison of the latencies.

(b) Cumulative distribution of ϵ_r.

Figure 7.6: Response of the proposed algorithm to random channel matrices, with $N_{\mathrm{UE}} = 30$.

Figure 7.6a compares execution latencies for different N_{UE}. It is possible to ascertain that the hybrid approach has lower latencies than the brute-force-only one, with a difference of latencies which becomes lower for higher N_{UE}.
Considering Table 7.2, the optimal solution rate o_r becomes lower for higher N_{UE}, even though it is always greater than 40%, *i.e.* it is achieved in quite less than half executed tests. Moreover, a higher number of repetitions is expected to *damp* the *oscillations* of o_r values and to force a trend substantially asymptotic for $N_{\mathrm{UE}} \geq 30$, with the employed QUBO setup. Figure 7.6b shows the transmission rate relative error with $N_{\mathrm{UE}} = 30$ and $n_{\mathrm{repetitions}} = 100$, which provided the lowest o_r. It is possible to observe that in about 80% of tests $\epsilon_r \leq 4\%$, and $\epsilon_r \leq 10\%$ in more than 90% of cases.
These results permit to declare that, even though the optimal solution is not always found, acceptable solutions are usually achieved by the QUBO-based solver when tested on random subchannel matrices, with a total execution time lower than the *brute-force*-only approach (at least when the convex optimizer is involved).

Matrices provided by the MIMO simulator

In this section the results obtained with data acquired through the TIM S.p.A. MIMO simulator are reported and commented.

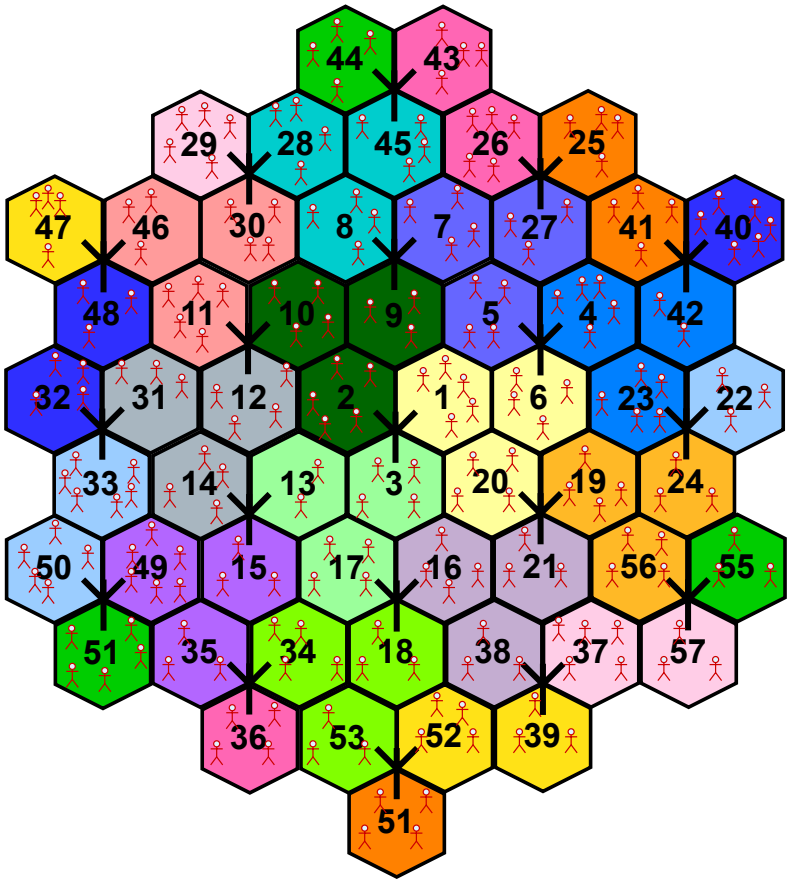


Figure 7.7: Base stations involved in TIM S.p.A. simulation. Cells with the same colour form a BS triplet.

The simulated MIMO system involved 52 independent sub-channels, 57 BSs with two transmitting antennas — highlighted in Figure 7.7 — and 570 users. Fixing the subchannel and assuming that each BS exploited only one antenna for transmission, independent BS triplets (highlighted with different colours in Figure 7.7) cooperating in JT could be defined. If the 570 users were separated in different subgroups, one for each BS triplet, the optimization problems could be solved in parallel, because of absence of data dependencies. The procedure for separating users, thus creating parallel QUBO problems (one for each BS triplet) is detailed in the following.
TIM S.p.A. MIMO simulator returned three matrices:

- `H`, corresponding to the P matrix of random tests.
- `RxPower`, quantifying in decibel the path loss on the whole transmission spectrum (without separating it in sub-bands). Each element `RxPower` could be

`NaN`, corresponding to a missing link between a BS and an user, so it could be interpreted as a 0 in linear units.

- `CQI_ue`, associated with the *connection* quality between each BS and each user on the whole transmission spectrum (*i.e.* without separating it in sub-bands).

From a practical point of view, each 57×570 subchannel matrix H was obtained similarly to that for random tests, *i.e.* with an element-by-element product of L e P matrices (Equation (7.22)). In particular, `CQI_ue` was not taken into account and the received power was assumed to be divided uniformly on the 52 sub-bands, so the path loss on each sub-channel — required for computing the L matrix — was obtained by dividing each `RxPower` value (in linear units) by 52.
With this mechanism, it was possible to obtain a 57×570 matrix for each subchannel, but this was not sufficient, since the goal was obtaining, for a given subchannel, a set of $3 \times N_{\mathrm{UE}_{\mathrm{triplet}}}$ matrices, one for each BS triplet. In order to do that, each user was associated with a triplet if the corresponding `RxPower` values in linear units were greater than 0 for all the BSs of a triplet. For "peripheral" triplets, such as 29-37-57 (pink hexagons in Figure 7.7), users served by only two BSs were taken into account, since the total number of users served by all the BSs of the triplet could be extremely low, thus limiting excessively the optimal solution search.
Before going in the details of the results, it is important to specify the QUBO simulation setup. The constraints weights in the QUBO construction (reported in Table 7.1) were the same for problems looking for two or three Joint Transmission users, to ensure a similar simulation setup in the two cases. For each subchannel and BS triplet, one thousand repeated tests were done. The results reported in the following were only obtained with a *brute-force* solver based on the *naïf* approach, since the available computational resources and time for completing tests were not sufficient to test the algorithm with the convex optimizer with all the available data. Suffice it to say that repeated tests with the *naïf* approach required more than one month.
Different statistical parameters were exploited to estimate the quality of the algorithm. The optimal solution efficiency o_r exploited for random tests was not taken into account, because it was expected too low in this scenario and because identifying the quality of sub-optimal solutions was considered more interesting to establish the capabilities of the proposed methodology with realistic data. The computed fiducial parameters were:

- **Transmission rate computed with the hybrid approach**, to be compared with the optimal one, surely obtained with a *brute-force* approach.
- **Users reduction factor** G, given by the ratio between the total number of users associated with a BS triplet and the number of selected users after the

selection done by QUBO. This parameter permitted to estimate the number of *brute-force* iterations avoided thanks to QUBO, so the latency reduction of the *brute-force* step of the procedure.

Plots in Figure 7.8 are reported to have an idea of the advantages provided by the reduction of the total number of combinations. Plot on the left column show, for four G values, the ratio between the total number of combinations when two (top) or three users (bottom) among n are chosen ($\binom{n}{2}$, $\binom{n}{3}$) and the corresponding ones when the total number of users is $\frac{n}{G}$ ($\binom{\frac{n}{G}}{2}$, $\binom{\frac{n}{G}}{3}$). It is possible to ascertain that, for a given n, a higher G lowers the total number of iterations. The same conclusion can be achieved for the figures in the right column, where the number of users selected by QUBO is assumed fixed and equal to $\frac{n}{G} = 10$; for example, in the three-user served case, the total number of *brute-force* iterations is reduced by about nineteen times with $G = \frac{5}{2}$.

General comments on the obtained results can be done by reporting only those for

Triplet	Number of served users	Users associated with triplet	μ_G	σ_G
1-6-20	3	30	2.82	0.30
8-28-45	3	28	2.78	0.34
11-30-46	3	35	3.18	0.34
12-14-31	3	23	2.54	0.27
25-41-54	2	60	5.74	0.75
32-40-48	2	58	5.73	0.65
39-47-52	2	61	5.94	0.71

Table 7.3: Users selection done by QUBO on subchannel 2.

subchannel 12, since an analogous trend was observed for all the others. Table 7.3 reports the mean value μ_G and the standard deviation σ_G of G for some BS triplets of subchannel 12. In both two and three-user served cases, the QUBO model selected $\frac{n}{G} \simeq 10$ users, where $G \simeq 6$ and $G \simeq 2$ in the first and in the second case respectively. This implies, reminding the plots in Figure 7.8, a reduction of the total number of brute-force iterations by one order of magnitude (usually at least twenty times). Considering the timescales comparison in Figure 7.6a, this reduction of the total number of iterations could be particularly useful with the convex optimizer, since the total time required for solving the problem with the hybrid approach is expected to be lower than the corresponding one based on the *brute-force*-only strategy.

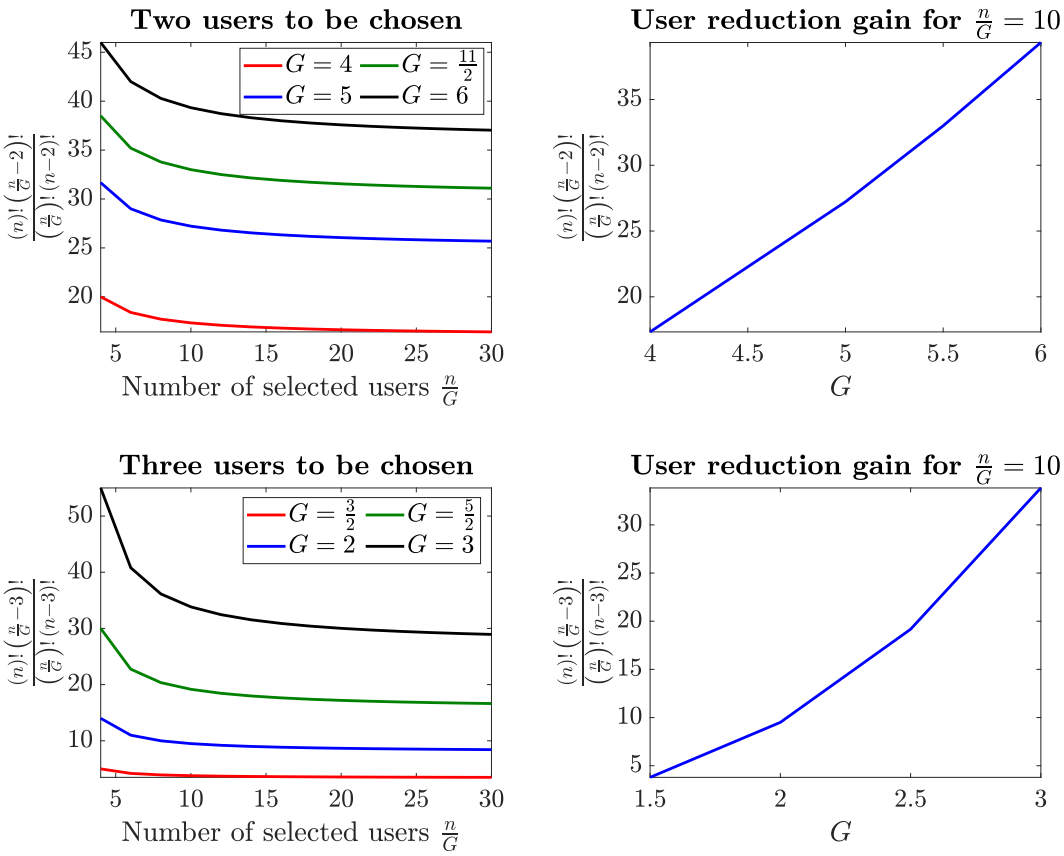


Figure 7.8: **Left column plots**: Reduction of the total number of *brute-force* iterations with the number of selected users. **Right column plots**: Reduction of the total number of *brute-force* iterations with G.

Figure 7.9 shows the cumulative distribution of the transmission rates of four triplets, two serving two users (32-40-48 and 44-51-55) and the others serving three users (1-6-20 and 11-30-46). The legend of each plot also reports mean values and standard deviations of transmission rates of the considered BS triplets. In all the examined cases, the cumulative distributions have the highest derivative in correspondence of the optimal transmission rate, thus proving that this was the mostly found by the algorithm during the executed tests. However, this derivative is not significantly higher than the others, thus implying that the optimal transmission rate is not the most frequent in a significant way. In any case, the results can be considered quite encouraging. In fact, observing that in all the reported cases, the mean values are higher than the average of the highest and lowest obtained transmission rates, it is possible to say that the cumulative distributions are more concentrated to higher transmission rate values.
In conclusion, the tests executed on MIMO simulator data proved that the hybrid

approach permitted to find a solution in definitely less iterations than a pure *brute-force* approach and that the hybrid algorithm selected JT users in a reasonable way, *i.e.* it looked for users combinations with higher transmission rate.

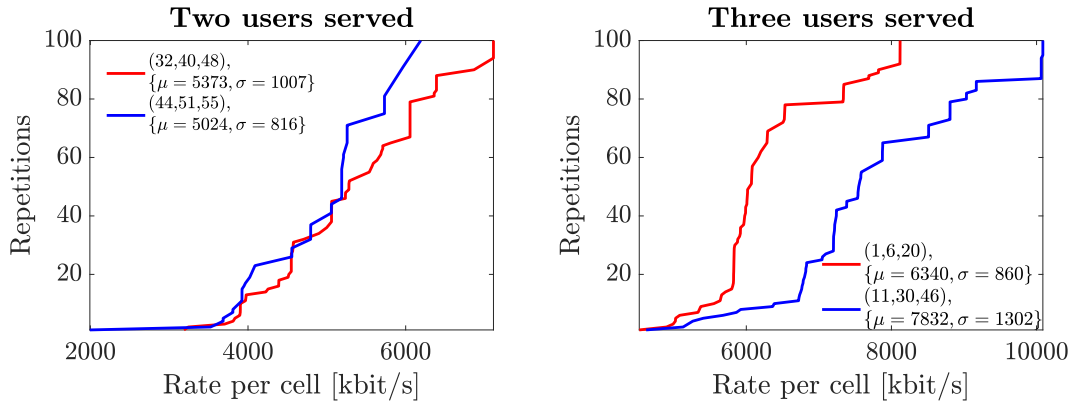


Figure 7.9: Transmission rate cumulative distributions for four base stations triplets. Values μ e σ are the mean value and standard deviation of transmission rate.

7.3 Concluding remarks

The proposed approach, based on the co-operation between a preliminary user selection with a QUBO model and a *brute-force* solver, permitted to achieve an acceptable trade-off between execution time and users selection accuracy. In fact, tests done with random data permitted to achieve the optimal solutions and good sub-optimal ones, with total execution times lower than a *brute-force*-only method (Figure 7.6), while — with data provided by TIM S.p.A. simulator — QUBO reduced the total number of users to be provided to the *brute-force* method by one order of magnitude in all the examined cases (Table 7.3) — thus reducing the total time associated with the second step of the complete algorithm — and the transmission rate associated with obtained solutions are characterized by a cumulative distribution which tends to be distributed towards the optimal value (Figure 7.9). The obtained results permit to declare that the proposed methodology, where a quantum-compliant QUBO model is exploited of a preliminary less-accurate selection of optimal users, can be effective for solving this type of problem in terms of both accuracy and execution time. In any case, this can be undoubtedly improved, as reported in section 8.2.
In conclusion, it is possible to say that the JT use-case permitted to verify the validity of this optimization problem solving hybrid methodology, which can be extended or arranged for other types of scheduling or other similar optimization problems.

Part IV

Conclusions and Future Perspectives

Chapter 8

Future perspectives

In both domains of analysis, a quite good compromise between solution quality and computational complexity was achieved. However, the obtained results are not definitive and works can be undoubtedly improved; for this reasons, some future perspectives are going to be presented in the following.

8.1 Simulation based on compact models

For what concerns simulation of quantum computers based on the circuit model, future perspectives associated with two main domains are proposed in the following.

8.1.1 Improving and extending compact models

In the current version of the simulator, manipulation and measurement mechanisms must be absolutely improved. The first one involves only rectangular pulses, which are not practically employed in experiments, that usually involve Gaussian or numerically-optimized pulses, while the second one is currently ideal and does not involve uncertainty due to the readout mechanisms of real hardware. For what concerns manipulation, the change of modulation will slightly affect the calculation of gates unitary matrices, but it will not require an upsetting of the methodology for computing these. For example, the Fourier model for off-resonance can be employed in these cases in a straightforward way, since changing the spectra from which θ_{eff_k} are obtained will be sufficient. Improving the compact model for measurement is more challenging. A possible solution could be to build, after the analysis of real qubit state readout systems, a measurement errors matrix affecting the eigenstates probabilities, in the wake of calibration matrices employed in Qiskit to partially compensate errors [20]. This matrix should be applied on the whole or the reduced density matrices, thus also making easy the interface with the operations belonging to the flow-chart in Figure 4.6.
On the other hand, the simulation infrastructure cannot be limited to molecules

with spin qubits. The methodology of definition of compact models and the organization of the simulator can be exploited to quite easily create models of non-molecular devices, also inheriting and arranging some features of the molecular ones. For example, as already clarified in chapter 4, relaxation, decoherence and off-resonance models are general and can be inserted in those of new examined devices. Relaxation could be adapted to dynamics whose steady-state is not $|0\cdots0\rangle\langle0\cdots0|$ (modifying the weights of the re-distribution algorithm would be sufficient). Moreover, the Fourier model can be extended to the effects of electromagnetic radiations, *e.g.* due to crosstalk phenomena, on every technology with qubits having resonance frequencies close to each other.
In conclusion, a possible technological domain — from which it could be convenient to start the exploration of new technologies — is suggested. This is that of solid-state devices encoding quantum information onto spins, such as semiconductor quantum-dots [46] or or nitrogen-vacancy (NV) centers in diamond [98]. The reason behind this choice would be that, since spin-qubits are employed also in this case, inheriting molecular compact models is expected to be generally easy. In any case, the analysis can also start from non-spin-based technologies, such as superconducting qubits, thanks to the sufficiently high level of abstraction of compact models.

8.1.2 Methodology for simulator software development

An important requirement for the simulator is the capability of scaling the number of evaluable qubits in the coming years, like real quantum hardware, thus allowing users to estimate the performance of their contemporary devices. Considering that the amount of memory to be allocated for simulation scales exponentially with the number of qubits (more precisely that a density matrix with 2^{2n} complex numbers is associated with n noisy qubits), smart strategies for the storage of the density matrix shall be found. Apart from all the optimization strategies associated with the quantum supremacy diatribe [12] and [13] — which either distribute the data structure on different levels of the memory organization of a computer or slightly change simulation in the paradigm, moving from pure Schrödinger representation of quantum states to a Schrödinger-Heisenberg one — a possible approach for the reduction of memory requirements, compatible with the current quantum state representation employed in the simulator, could be to exploit the repetitions of complex numbers in the density matrix, a typical condition in some practical quantum algorithms. Therefore, alternative data structures, which store the recurring values of the density matrix only once, could be investigated to enable fast and reliable simulation of tens or hundreds of noisy qubits in the future. A simulation approach based on decision diagrams [99] could be a good starting point for achieving this goal.
In addition to the previous considerations on memory allocation, an additional

improvement of the simulator at software level could be to change programming language, considering that MATLAB is interpreted, so — even though is extremely versatile and optimized for numerical calculations and linear algebra — it does not permit to execute all the hard optimizations on execution time and memory that are possible on compiled languages, such as C/C++. An idea of the *performance mismatch* between compiled and interpreted languages can be found in [100], but MATLAB is not mentioned. This operation can undoubtedly provide long-term advantages, but it clearly requires a preliminary translation effort for moving all the infrastructure in a compiled programming language.

8.2 Joint Transmission

For what concerns JT, the modularity of the developed infrastructure ensures to insert, in future, additional methodologies for solving the target optimization problem. In fact, the algorithm is expected to be improved and some suggestions are reported in the following.
First of all, it is reminded that all tests described in chapter 7 were done with a purely classical infrastructure. In fact, QUBO problems were solved only with simulated annealing. Quantum annealing could increase the search quality, but — considering what is written in section 6.4 — the QUBO model weights should be arranged to obtain a global cost function suitable for quantum annealing. In the following, some general comments for improving the current QUBO, not strictly related to a specific solver, are reported. The first two suggestions can be applied to current implementation, while the last two are more general.
A first improvement can be achieved by exploiting the feedback mechanism shown in Figure 7.3. This could permit to employ power partition factors more suitable for each user and more reasonable weights for each constraint. Feedback can be executed during the standard running of the algorithm or in a preliminary calibration step, where some data previously computed or generally available are exploited, similarly to what is done during the training of neural networks.
In order to reduce the execution time associated with QUBO model, a *warm-up* mechanism could be implemented, when the machine executing the JT solver is turned on. In particular, if the subchannel matrix is expected to not dramatically change on the one minute timescale, repeated executions could be executed to give to solver an initial idea of the optimal subset, thus starting from a better problem configuration. At the end of this initial phase, the infrastructure could monitor the subchannel matrix in time, even in real-time, in an eventual optimistic scenario. In any case, this *warm-up* could be done in preliminary tests, to empirically define a strategy for the initial conditions of the optimization problem to be solved in time.
The last two hints could reduce the QUBO complexity. The first one consists in operating on smaller users subsets. For example, multiple QUBO models involving

at most ten-twelve users could be initially associated with a BS triplet on a given subchannel, then their optimal solutions could be compared to find the best solution. The second hint could be the simplification of the QUBO problem itself, *e.g.* considering only the knapsack part for power constraint.
Finally, in addition to QUBO improvements, validation must be enhanced. A comparison with purely classical solvers not based on QUBO modeling, such as [94], must be done to evaluate the capabilities of the proposed strategy with respect to the state of art and to prove that quantum-compliant modelling (*e.g.* QUBO) can provide an effective computational advantage in solving the JT optimization problem.

Chapter 9

Conclusions

The focus of this thesis was presenting new contributions in classical simulation of quantum computers through compact models and in Joint Transmission users scheduling. In both cases, the workflow was analogous:

1. A preliminary investigation and understanding of the main features of classical simulation of quantum computers and Joint Transmission systems was done, to acquire consciousness of their degree of criticality.

2. Some specific algorithmic methodologies were defined, to try to overcome the main issues.

3. Validation was done, comparing the results with canonical references, either available in the state of art or implemented in parallel to the proposed solution.

In the context of quantum technologies simulation, starting from the analysis of the qubits physical systems, in terms of Hamiltonian characterisation and experimental qubit dynamics, some innovative strategies for the compact-model description of their qubits system, involving some critical non-ideality phenomena, such as decoherence and off-resonance, were defined and proposed. The reliability of the proposed approach for the description of noisy qubits has been demonstrated, comparing the results provided by the implemented models with those obtained with numerical solutions of the Schrödinger equation and with existing noise models in Qiskit's `QasmSimulator`. The obtained results are encouraging for a future accurate simulation of noisy quantum circuits with compact models, since solving any time-dependent Lindblad equation is not required (thus significantly reducing the simulation latency) and some simplifications, reducing the total amount of memory for describing the evolution of a qubit system under non-ideality phenomena, can be done. Furthermore, the results highlight that reliable qubit dynamics can be achieved without solving any time integral.
In the Joint Transmission context, an algorithm for finding the best subset of users

served with JT was proposed. It is hybrid, on both solving strategy (QUBO-brute-force) and eventually computational paradigm (quantum-classical) levels. In particular, the QUBO model is responsible of selecting a non-definitive subset of users, which is then processed in a *brute-force* to find the best users. The obtained results proved that the proposed approach can reduce the total execution time with respect to a *brute-force*-only approach, when a convex optimizer is exploited to find the best users power partition factors, and that, even with the *naïf* uniform power partition among users, it can achieve on average results close to the optimal one.
Although there are some improvable features, the results already obtained permit to look to the future with optimism. On one hand, an accurate infrastructure for classical simulations of noisy quantum computers could be successfully employed by circuit designers, with the consciousness that the provided results are very close to those expected in an experimental setup, on the other the JT problem solver could be further improved to be employed in real MIMO wireless system.
In any case, the activities described in this thesis are hoped to be a starting point for a conscious engineering, at both hardware and software level, of quantum computation, capable of effectively solving problems having a positive impact on Society and life.

Part V

Appendices

Appendix A

Understanding the Rotating Wave Approximation with a simple scalar integral example

The Rotating Wave Approximation (RWA) is undoubtedly one of the most important and employed approximations for the description of qubit dynamics. An attempt of providing an intuitive comprehension of this approximation is reported in the following, considering a simple scalar case.
The integral of the sum of two complex exponentials, those in the top-right element of the rotating frame Hamiltonian in Equation 3.30, is taken into account:

$$\int_0^\tau \mathrm{e}^{\mathrm{i}\cdot[(\omega_r-\omega_R)t-\phi]} + \mathrm{e}^{-\mathrm{i}\cdot[(\omega_r+\omega_R)t-\phi]}\mathrm{d}t\,, \tag{A.1}$$

where ω_r and ω_R are the angular frequencies of the external oscillation and of the rotating frame respectively. Assuming, without loosing generality, that $\phi = 0$ and defining $\delta = \omega_r - \omega_R$ and $\Sigma = \omega_r + \omega_R$, the integral can be written as

$$\int_0^\tau \mathrm{e}^{\mathrm{i}\delta t} + \mathrm{e}^{-\mathrm{i}\Sigma t}\mathrm{d}t\,. \tag{A.2}$$

The integral can be computed analytically:

$$\begin{aligned}\int_0^\tau \mathrm{e}^{\mathrm{i}\delta t} + \mathrm{e}^{-\mathrm{i}\Sigma t}\mathrm{d}t &= \left(\frac{1}{\mathrm{i}\delta}\mathrm{e}^{\mathrm{i}\delta t}\Big|_0^\tau\right) + \left(-\frac{1}{\mathrm{i}\Sigma}\mathrm{e}^{-\mathrm{i}\Sigma t}\Big|_0^\tau\right) = -\mathrm{i}\frac{\mathrm{e}^{\mathrm{i}\delta\tau}-1}{\delta} + \mathrm{i}\frac{\mathrm{e}^{-\mathrm{i}\Sigma\tau}-1}{\Sigma} \\ &= -\mathrm{i}\frac{\cos(\delta\tau)+\mathrm{i}\sin(\delta\tau)-1}{\delta} + \mathrm{i}\frac{\cos(\Sigma\tau)-\mathrm{i}\sin(\Sigma\tau)-1}{\Sigma}\,.\end{aligned} \tag{A.3}$$

For all possible τ, δ and Σ values, both numerators have finite magnitude, with the same lower and upper bounds:

$$0 \leq |(\cos(\theta)-1) \pm \mathrm{i}\sin(\theta)| \leq 2\,. \tag{A.4}$$

The RWA condition is reported in Equation 3.31 and it substantially consists in $\Sigma \to \infty$. Moreover, in a typical scenario it is assumed that $\omega_R \to \omega_r$, *i.e.* $\delta \to 0$, thus implying that the integral can be written in terms of two limits:

$$\begin{aligned}\lim_{\Sigma\to\infty,\delta\to 0} -\mathrm{i}\frac{\mathrm{e}^{\mathrm{i}\delta\tau}-1}{\delta}+\mathrm{i}\frac{\mathrm{e}^{-\mathrm{i}\Sigma\tau}-1}{\Sigma} &= \lim_{\Sigma\to\infty}\left(\lim_{\delta\to 0} -\mathrm{i}\frac{\mathrm{e}^{\mathrm{i}\delta\tau}-1}{\delta}+\mathrm{i}\frac{\mathrm{e}^{-\mathrm{i}\Sigma\tau}-1}{\Sigma}\right)\\ &= \lim_{\Sigma\to\infty}\left(\lim_{\delta\to 0} -\mathrm{i}\frac{1+\mathrm{i}\delta\tau-1}{\delta}+\mathrm{i}\frac{\mathrm{e}^{-\mathrm{i}\Sigma\tau}-1}{\Sigma}\right) \qquad \text{(A.5)}\\ &= \lim_{\Sigma\to\infty}\left(\tau+\mathrm{i}\frac{\mathrm{e}^{-\mathrm{i}\Sigma\tau}-1}{\Sigma}\right).\end{aligned}$$

The calculation of the limit for $\delta \to 0$ puts in evidence that in resonance condition the integral substantially depends on the integration time τ. Considering the inequalities in Equation A.4, the limit of the integral for $\Sigma \to \infty$ can be finally obtained:

$$\lim_{\Sigma\to\infty,\delta\to 0}\int_0^\tau \mathrm{e}^{\mathrm{i}\delta t}+\mathrm{e}^{-\mathrm{i}\Sigma t}\mathrm{d}t = \lim_{\Sigma\to\infty}\left(\tau+\mathrm{i}\frac{\mathrm{e}^{-\mathrm{i}\Sigma\tau}-1}{\Sigma}\right)=\tau\,. \qquad \text{(A.6)}$$

In conclusion, the only significant integral contribution is that associated with $\mathrm{e}^{\mathrm{i}\delta t}$. An approximation similar to that here presented is expected to be valid for the *real* RWA of the Hamiltonian integration required by the Schrödinger equation.

Appendix B

Organization of the simulator

In this appendix, an overview of the compact-model-based simulator organization is provided. It does not have a definitive name, anyway it is currently called **QuanTO**. This is a pun, obtained from *quantum* and TO, which is the initial of the Turin Province, to stress the fact that it is a quantum hardware simulator developed in Turin. It is entirely developed in MATLAB, since, in initial prototype development phase, an environment able to easily manage vectors and matrices was convenient. Object Oriented Paradigm (OOP) was employed to ensure a more portable and reusable code, thus facilitating the insertion of new modules *ex-novo* or their creation from others already available (according to inheritance and polymorphism principles, typical of OOP).
The simulator is organized according to a modular structure, which is hoped to permit the integration of additional technologies and new functionalities. In the following, different figures, describing the organization of the simulator, are shown. It is important to precise that transparent directories, pointed by dashed arrows, are not currently available and expected to be integrated in the future. Figure B.4 shows the main directories of QuanTO. The four sub-directories of the *root* `matlab-quanto` are:

- `simulation_config`, which contains the compact models. It is possible to observe that there are four sub-directories, each of which associated with one of the four fundamental modules of the compact model detailed in section 4.2. Particular attention is provided to the contents of `devices`, marked with the light blue circle.
- `simulation_exec`, which contains the testbench environment. It currently supports only the simulation of hardware for quantum computation and the sub-directory `quantum_computing`, marked with the orange circle, is detailed in the following of this Appendix.
- `quantum_information_utils`, containing functions for the generation of useful matrices to be employed in compact models.

- `interfaces`, which will contain, in the future, scripts for interfacing the simulator with consolidated frameworks for quantum computation and information, such as Qiskit [20].

An overview of the main sub-directories and classes already developed is reported in the following. Each of these is described in a dedicated section.

B.1 simulation_exec/quantum_computing

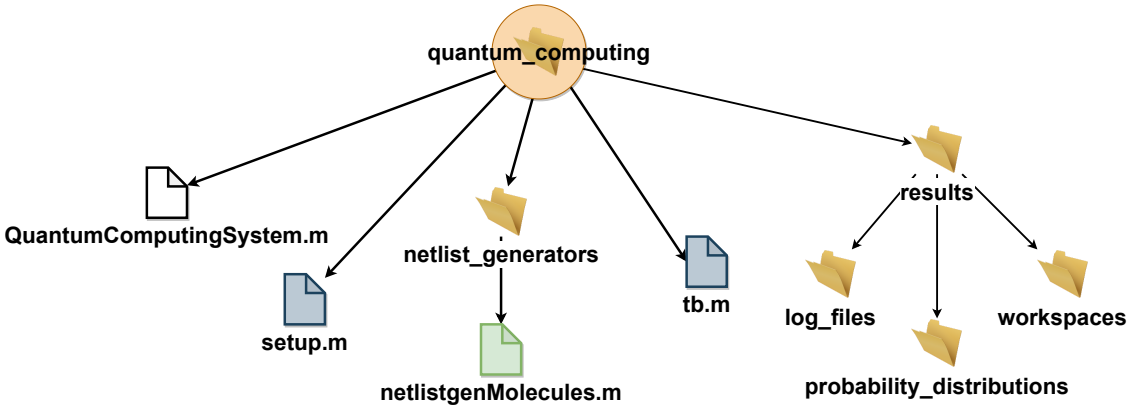


Figure B.1: `simulation_exec/quantum_computing` content.

`matlab-quanto/simulation_exec/quantum_computing` contains all the files required for launching a simulation:

- `QuantumComputingSystem.m`, a file containing the homonymous class, associated with the quantum computer to be simulated and schematized in Figure 4.3.
- `setup.m`, a file containing some boolean simulation settings:
  - `amplitudeFixed`, for choosing if pulses have fixed amplitude (true) or fixed duration (false).
  - `virtualZ_flag`, for employing virtual-Z gates;
  - `save_workspace`, for saving the workspace at the end of the simulation;
  - `debug_mode`, for printing the eigenstates probability distribution at each step (quantum gate) of the simulation;
  - `writeMeasurementResultsOnFile`, for writing the final eigenstates probability distribution on a text file.

- `netlist_generators/`, a directory containing scripts called netlist generators. In the context of traditional electronics, a netlist is a file containing all components and connections of a circuit. SPICE netlists are a relevant example. For what concerns this simulator, a netlist is a script containing the sequence of simulator operations (substantially methods of a `QuantumComputingSystem` object) to be executed. Netlist can be written manually or automatically, through scripts which translate OpenQASM 2.0 instructions into simulator commands. Figure B.2 shows a schematic representation of this translation. Apart from comments which are ignored, native commands of OpenQASM 2.0 (*i.e.* the gates belonging to the library `qelib1.inc`) are properly interpreted are translated into simulator commands. OpenQASM 2.0 `measure` instructions are always postponed at the end of the quantum circuit, *i.e.* after all gates. In this parsing procedure, apart from the syntax evaluation, the connectivity between qubits is verified for a correct simulation of two-qubit gates. In the current implementation of this interpreter, only the native two-qubit gates of a target technology can be employed in the input OpenQASM 2.0 file. In the NMR case, assuming that $R_z(\theta)$ can be virtually implemented, $R_{zz}(\theta)$ and C-Z are both supported. Virtual-Z gates are properly evaluated with these scripts; an array with the ϕ_x of each qubit is stored and updated according to Equation 3.41, whenever a $R_z(\theta)$ gate is implemented. Moreover, if `debugMode` is asserted, the netlist also contains commands for printing the eigenstates probability distribution (associated with the density matrix) after each quantum gate.

- `results/`, a directory containing three sub-directories:

  - `workspaces/`, storing the MATLAB workspace at the end of a simulation;
  - `probability_distributions`, which stores the probability distributions associated with the classical register, after a measurement call;
  - `log_files`, which stores `.log` files containing the probability distribution associated with the whole density matrix of the simulated quantum computer, the fidelity at the end of the quantum circuit (before measurement) and the total time latency associated with the quantum circuit.

  The files of these directories are thought to facilitate post-processing tasks and have a formatted name `<initialLabel>_<device>_<date>_<time>.<suffix>`, where `<initialLabel>` can be `probability_distribution` or `workspace` only, `<time>` has format `HH-MM-SS` (the separator `-` is employed to avoid directory errors under Windows operating systems) and `suffix` can be `log`, `txt` or `mat` for log files, probability distributions and workspaces respectively.

- `tb.m`, the testbench script, containing all the main operations for succesfully completing a simulation, also schematized in Figure B.3. It is possible to define six main steps:

  1. Executing `setup.m` script, to set boolean flags associated with simulation settings to be employed in the following of the simulation.
  2. Executing a **parameters function**, automatically instantiating an object of the device to be simulated. The parameters files, one for each technology, are characterzied by the suffix `_params.m` and shown in Appendix B.3. They also support some input arguments, such as the magnetic static field, the `amplitudeFixed` flag and the total number of involved qubits (in the Cr_7Ni case).
  3. Instantiation of a `QuantumComputingSystem` object, with the device to be simulated.
  4. Creation of the netlist script, also taking into account the two-qubit gates connectivity of the quantum computer and the eventual assertion of `virtualZ_flag`.
  5. Launching the netlist script, with each gate conceptually simulated according to the graph in Figure 4.4.
  6. Save additional files, such as the workspace and the log files, according to their associated boolean flags.

For sake of completeness, `QuantumComputingSystem` properties and methods are reported in the following:

`property` `stateVector`, which stores the ideal qubits state.

`property` `densityMatrix`, which stores the noisy qubits state.

`property` `involvedQubits`, which is a boolean array employed to identify qubits subjected by at least one quantum gate, to be involved in relaxation process evaluation, according to the model in subsection 4.6.1.

`property` `totalExecutionDelay`, which is the latency time variable, incremented according to the execution time of each quantum gate. It is practically an accumulator variable.

`method` `QuantumComputingSystem`, which is the object constructor, receiving as argument the object of the device to be simulated.

`method` `GetStateVector`, returning the current ideal qubits state vector.

`method` `GetDensityMatrix`, returning the current noisy qubits density matrix.

method `GetFidelity`, returning the current fidelity (Equation 2.80).

method `GetProbabilityDistributionDensityMatrix`, returning the probability distribution associated with the noisy qubits density matrix. It can also return the distribution of a single qubit of the simulated system, exploiting the partial trace with respect to all the other qubits.

method `GetTotalExecutionDelay`, returning the total delay at a given point of the execution

method `singleQubitGate`, which is the simulator *instruction* for single-qubit gates. Given the target qubit and the parameters θ and ϕ — associated with the rotation amount and axis — as input arguments, this function updates the state vector and the density matrix, according to the scheme in Figure 4.4.

method `singleQubitGateExternalFreq`, which substantially does the same thing of the previous function, with the only exception that resonance condition is not assumed, so the frequency of the external oscillation must be provided as input argument.

method `twoQubitGate`, which is the simulator *instruction* for two-qubit gates. Given the target qubit and the angle ϕ of the characteristic unitary evolution of the device to be simulated as input arguments, this function updates the state vector and the density matrix.

method `nopDelayForSpectroscopyExperiments`, which evaluates the effect of a No-OPeration (NOP) on the simulated quantum computer on a finite time interval provided as input argument.

method `measure`, which receives the lists of qubits to be measured and of their corresponding classical bits and implements the measurement mechanism detailed in Figure 4.6. If the binary flag `writeOnFile` is asserted, it saves the probability distribution associated with the measured qubits in the directory `results/probability_distribution/`.

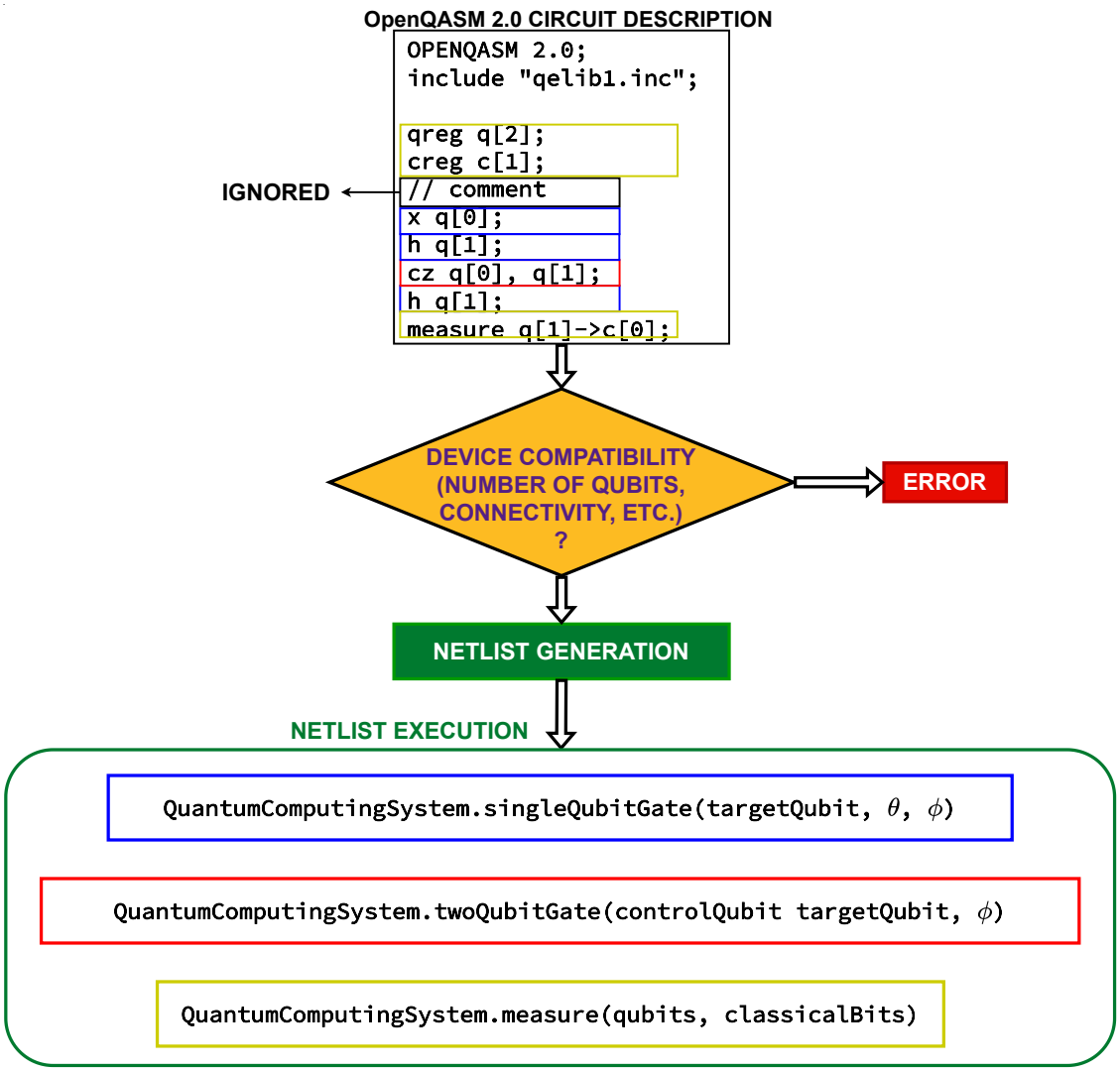


Figure B.2: Mechanism for creating the simulation netlist from an OpenQASM 2.0 file.

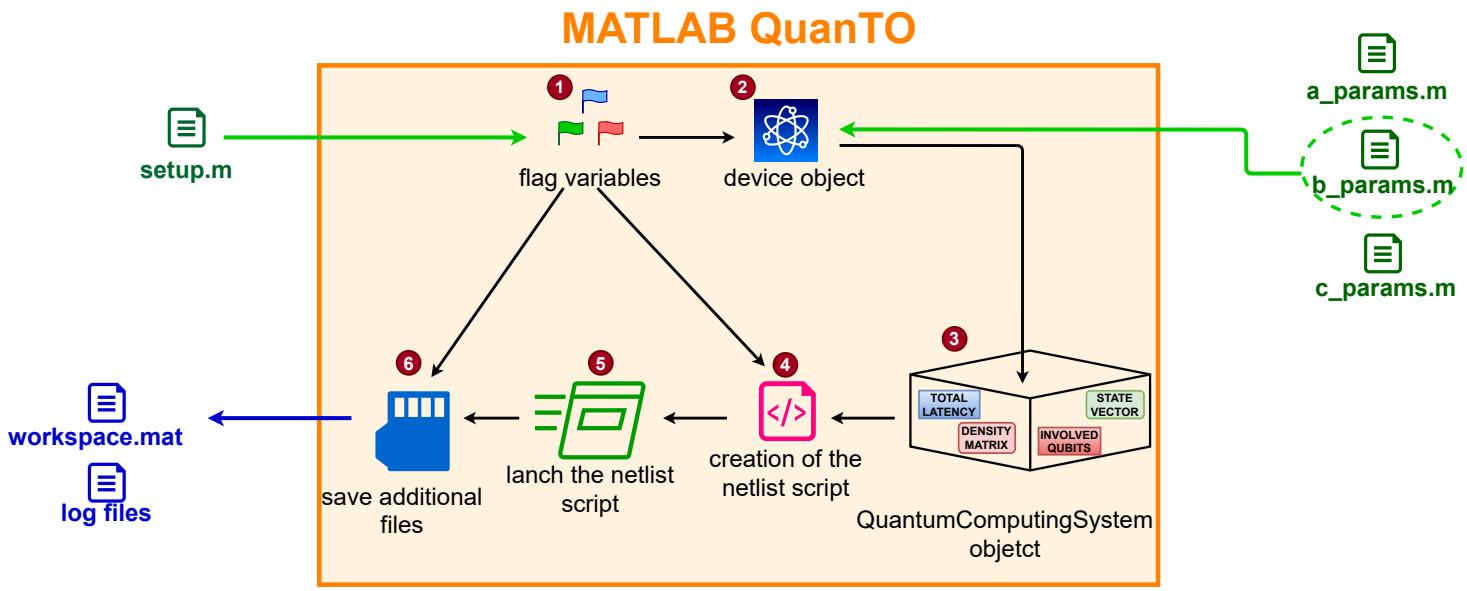


Figure B.3: Testbench workflow.

B.2 quantum_information_utils/

The directory `matlab-quanto/quantum_information_utils/` contains functions which can be transversally employed in simulation:

- `computePartialTrace.m`, which computes the partial trace on a certain qubit of the density matrix, thus returning a new density matrix not considering that qubit.
- `portunitarymatrixeval.m`, which determines the unitary evolution of a qubit according to Equation 3.34.
- `computeCloseQubitsMatrix.m`, which determines the unitary evolution $U_{\text{close}} = \prod_{i=k}^{l-2} U_{\text{SWAP}_{k,k+1}}$ for getting two qubits k and $l > k$ adjacent through repeated swaps (Equation 3.17). An example of application of this function is the computation of the unitary evolution associated with the J-coupling between two non-adjacent qubits $R_{zz_{k,l}}\left(\frac{J_{k,l}\tau}{2}\right) = U_{\text{close}}^{\dagger} R_{zz_{l-1,l}}\left(\frac{J_{k,l}\tau}{2}\right) U_{\text{close}}$.
- `computeSWAPMatrix.m`, which determines unitary evolution of the SWAP gate between qubits k and $l > k$ as $U_{\text{close}}^{\dagger} U_{\text{SWAP}_{l-1,l}} U_{\text{close}}$.
- `computeJCouplingHamiltonian.m`, which determines the J-coupling Hamiltonian $\sum_{\{i,j\}} J_{i,j}\sigma_{z_i}\sigma_{z_j}$ or $\sum_{\{i,j\}} J_{i,j}(\sigma_{x_i}\sigma_{x_j} + \sigma_{y_i}\sigma_{y_j} + \sigma_{z_i}\sigma_{z_j})$ (depending on a boolean flag), given in input a matrix with the J-couplings between each pair of qubits.

B.3 simulation_config/quantum_computing

The directory `matlab-quanto/simulation_config/quantum_computing` contains all the elements necessary for building a compact model. It is characterized by four sub-directories, each one associated with a sub-module of a compact model (section 4.2):

- `decoherence`, containing classes associated with decoherence model. Only the generic `DecoherenceModel` class, implementing the algorithms for computing the relaxation and decoherence matrices in subsection 4.6.1, is currently available.
- `manipulation`, containing classes associated with manipulation mechanism. The generic `ExternalManipulation` class is currently available. It contains all parameters for manipulating each qubit of a device according to the magnetic resonance model (pulse duration or amplitude). Moreover, it is also responsible of evaluating the effective off-resonance angles according to the Fourier model described in section 4.5.

- `measurement`, containing classes associated with measurement mechanism. The generic `Measurement` class is currently available, implementing the ideal measurement described in section 4.3.

- `devices`, containing all classes associated with the technology sub-module of the compact model and their corresponding parameters function files.

The organization of `devices` is more complex than the others and it is shown in Figure B.5. All models are inherited from the class `QuantumComputingTechnology`, characterized by the following properties and methods:

property `name`, which is a string containing the name of the device, practically employed only in saved results files.

property `numQubits`, which is the total number of qubits of the system under analysis.

property `singleQubitFrequencies`, which is a vector containing the resonance frequencies (in Hz unit or its multiples) of each qubit.

property `twoQubitGateFrequencies`, which is a generic data structure containing frequencies (in Hz unit or its multiples) associated with two-qubit gates, in the case of switch-based molecules, it is declined in a matrix (`numQubits-1`) $\times$ 4, containing the resonance frequency of each spin switch in all possible configuration of its adjacent qubits, while in the NMR case it is a `numQubits` $\times$ `numQubits` matrix with J-couplings.

property `decoherenceModel`, which is an object of the class associated with the decoherence model. The generic `DecoherenceModel` class is currently employed in all compact models.

property `manipulation`, which is an object of the `ExternalManipulation` class.

property `durationFixedN_amplitudeFixed`, a boolean flag which indicates if the simulation of quantum gates described by the magnetic resonance model works with fixed pulse amplitude (true) or fixed pulse duration (false).

property `measurementMechanism`, which is an object of the `Measurement` class.

method `QuantumComputingTechnology`, which is the object constructor receiving in input all the property values (apart those which can be initialized to default values).

method `GetSingleQubitGateUnitaryMatrices`, which is called in `singleQubitGate` method of class `QuantumComputingSystem` (section B.1). After receiving in input the target qubit, the rotation angle θ and the initial phase ϕ associated with the rotation axis, it returns the unitary matrices for updating state

vector and density matrix (called U_{ideal} and U_{real} in section 4.3) and the gate time duration.

method `GetExternalFrequencySingleQubitGateUnitaryMatrices`, which is analogous to the previous method, with the only difference that the frequency of the external excitation is also provided as input argument.

method `GetTwoQubitGateUnitaryMatrices`, which is employed in the `twoQubitGate` method of the `QuantumComputingSystem` class (section B.1) and returns the same matrices of the previous methods for two-qubit gates, having as input parameters the two involved qubits and the rotation amount ϕ of the characteristic two-qubit unitary evolution of the technology to be simulated. From a practical point of view, the native two-qubit gate of this generic technology is that of NMR quantum computers and, assuming an interaction strength $J_{k,l}$ between qubits k and l, the angle ϕ is mapped onto the free evolution time such that

$$\tau = \begin{cases} \left| \frac{\phi}{2\pi J_{k,l}} \right| & \text{if } \text{sign}(J_{k,l}) = \text{sign}(\phi) \\ 2 \left| \frac{\text{sign}(J_{k,l}) \cdot 2\pi - \text{sign}(J_{k,l}) \cdot \left| \frac{\phi}{2} \right|}{2\pi J_{k,l}} \right| & \text{otherwise} \end{cases}$$

When $\phi = \pi$, the duration $\frac{1}{2|J_{k,l}|}$ required for the implementation of the CZ gate (Figure 5.10) is obtained. In other words, ϕ is mapped onto a time duration $\tau > 0$ such that $U_J(\tau) = R_{zz}\left(\frac{\phi}{2}\right)$[1].

The quantum gates methods of `QuantumComputingSystem` are public and they are responsible to call the methods available in each technology compact model to manipulate the state vector and the density matrix with U_{real}, R, D and U_{ideal}.
In addition to `QuantumComputingTechnology`, the directory `devices` also contains a sub-directories tree associated with molecular spin qubits. As observable in Figure B.5, compact models for trapped ions, superconducting qubits and solid-state spin qubits (such as silicon quantum dots and NV-centers in diamond) are expected to be integrated in future. Starting from `devices/spin/molecular`, files can be organized in three paths, each one with a characteristic color:

- `Molecule.m`, the mother class of molecular spin qubits. It is substantially equivalent to `QuantumComputingTechnology.m`, with the only difference that it contains an additional property related to the spin quantum number of each qubit, since it is theoretically possible to take into account systems with spin greater than $\frac{1}{2}$. However, in all the examined cases it is equal to $\frac{1}{2}$.

[1] Properly, the evolution corresponds to $e^{i(\text{sign}(J_{k,l}) - \text{sign}(\phi))\frac{\pi}{2}} R_{zz}\left(\frac{\phi}{2}\right)$, but the global phase is irrelevant.

- `switch_technology`, containing the file of the class `SwitchBasedMolecule`, associated with the compact model of switch-based devices and two directories `cr7ni` and `vanadyl`, with the parameters files for the automatic instatiation of an object associated with Cr_7Ni or a vanadyl devices. In the Cr_7Ni case, the directory also contains `GenerateCr7NiResidualSwitchHamiltonian.m`, with a function which permits to construct the Hamiltonian of unwanted interaction of qubits in open switch configuration $\mathcal{H}_{\text{eff}}$, to be employed for computing the unwanted interaction unitary evolution according to Equation 5.16.

- `nmr`, which contains the file of the class `NMR`, associated with the compact model of NMR devices and the parameters configuration files of heteronuclear and homonuclear devices, divided in two directories according to their nuclear type.

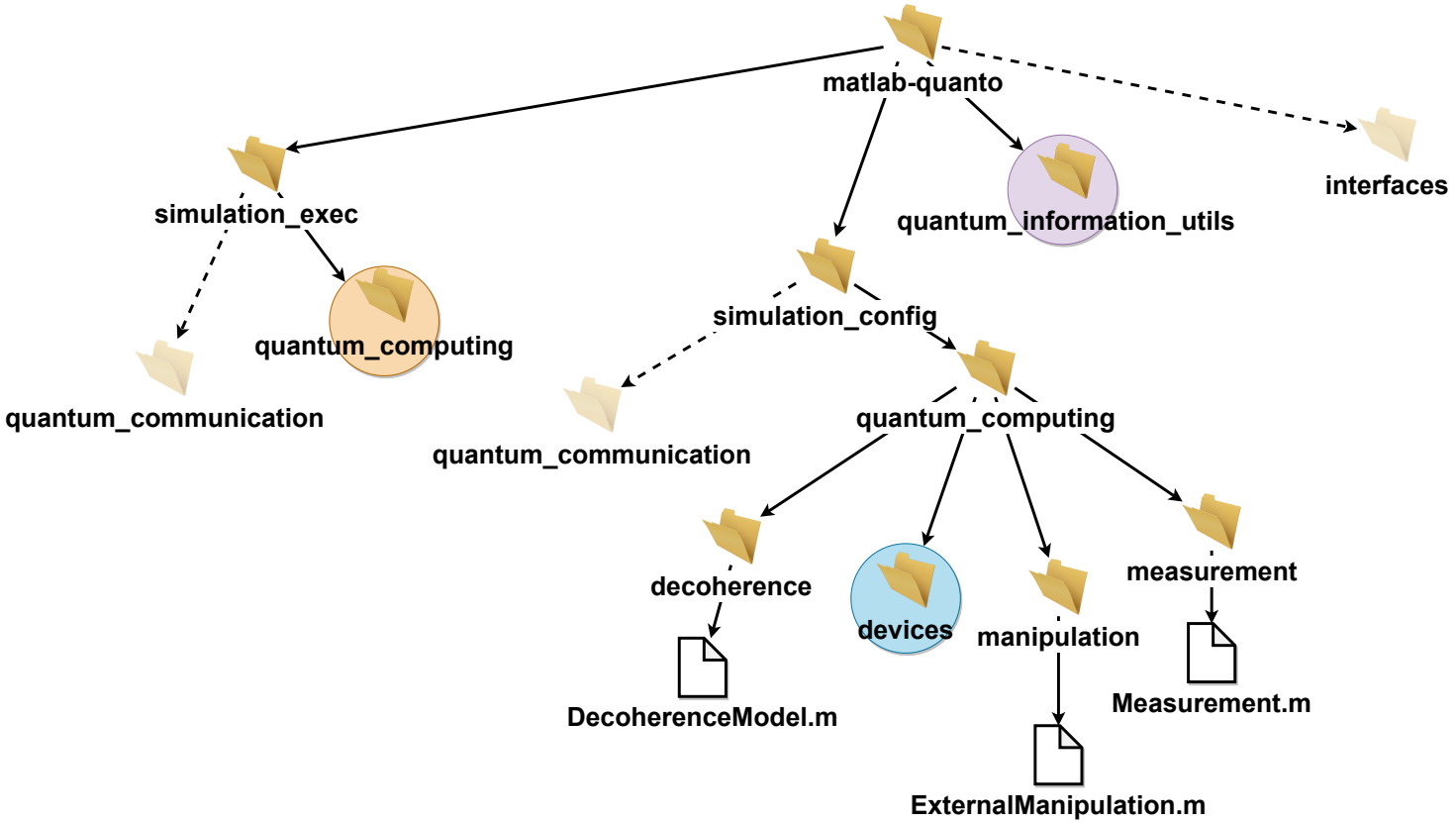


Figure B.4: Overview of the main directories of QuanTO simulator.

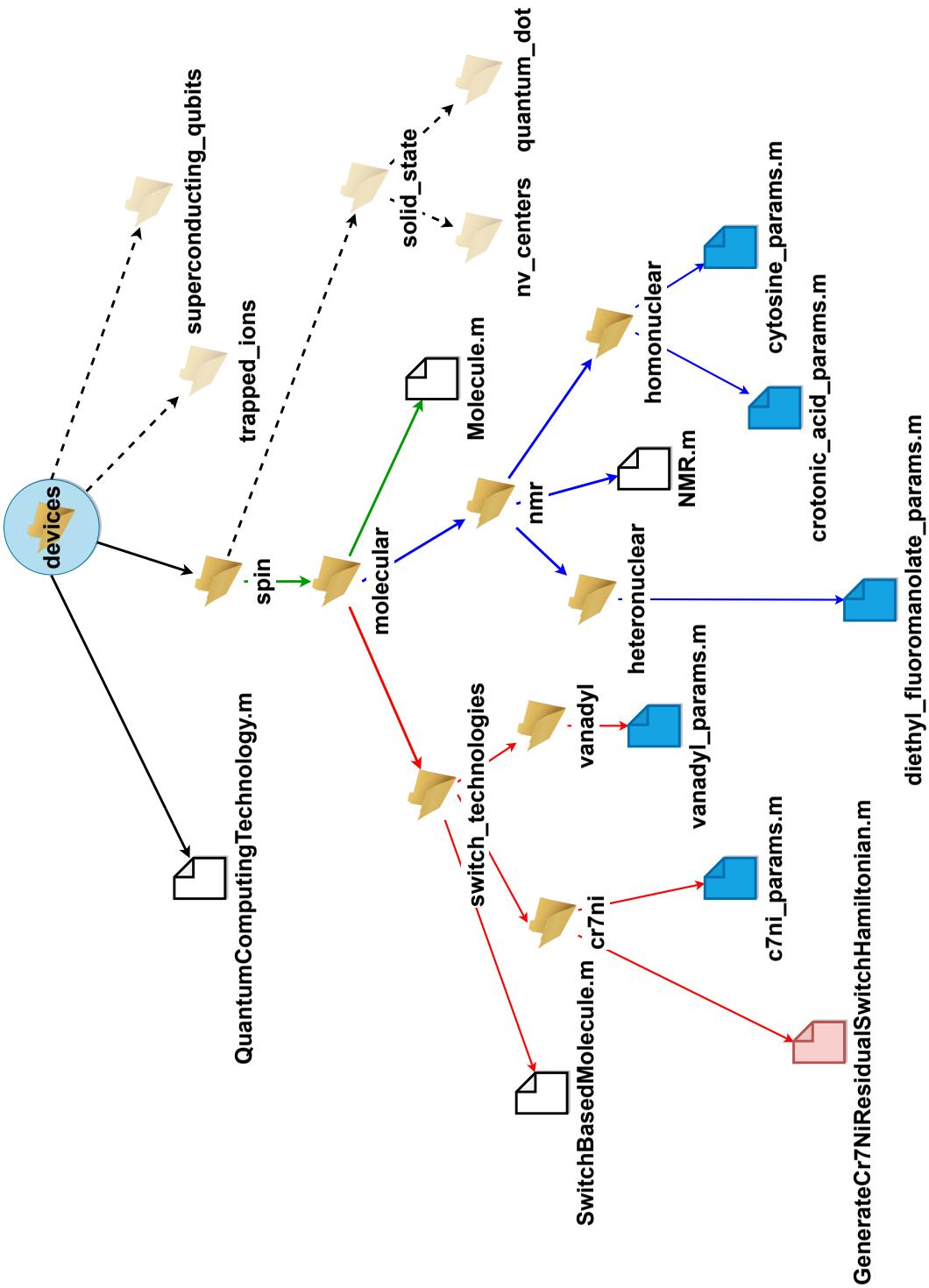


Figure B.5: Contents of `devices` sub-directory.

Bibliography

[1] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. *A Quantum Approximate Optimization Algorithm*. 2014. DOI: 10.48550/ARXIV.1411.4028. URL: https://arxiv.org/abs/1411.4028.

[2] Amira Abbas et al. *Learn Quantum Computation Using Qiskit*. 2020. URL: http://community.qiskit.org/textbook.

[3] Nicolas Gisin et al. "Quantum cryptography". In: *Rev. Mod. Phys.* 74 (1 Mar. 2002), pp. 145–195. DOI: 10.1103/RevModPhys.74.145. URL: https://link.aps.org/doi/10.1103/RevModPhys.74.145.

[4] David Peral García, Juan Cruz-Benito, and Francisco José García-Peñalvo. *Systematic Literature Review: Quantum Machine Learning and its applications*. 2022. DOI: 10.48550/ARXIV.2201.04093. URL: https://arxiv.org/abs/2201.04093.

[5] Giovanni Amedeo Cirillo et al. "Quantum Computing tutorial". In: *Notiziario Tecnico TIM* (2020). https://www.gruppotim.it/content/dam/telecomitalia/it/archivio/documenti/Innovazione/MnisitoNotiziario/2020/2020-2/cap03_tutorial-quantum-computing.pdf, pp. 58–77.

[6] Lov K. Grover. "A Fast Quantum Mechanical Algorithm for Database Search". In: *Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing*. STOC '96. Philadelphia, Pennsylvania, USA: ACM, 1996, pp. 212–219. ISBN: 0-89791-785-5. DOI: 10.1145/237814.237866.

[7] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010. DOI: 10.1017/CBO9780511976667.

[8] Jules Tilly et al. *The Variational Quantum Eigensolver: a review of methods and best practices*. 2021. DOI: 10.48550/ARXIV.2111.05176. URL: https://arxiv.org/abs/2111.05176.

[9] Leo Zhou et al. "Quantum Approximate Optimization Algorithm: Performance, Mechanism, and Implementation on Near-Term Devices". In: *Phys. Rev. X* 10 (2 June 2020), p. 021067. DOI: 10.1103/PhysRevX.10.021067. URL: https://link.aps.org/doi/10.1103/PhysRevX.10.021067.

[10] Tadashi Kadowaki and Hidetoshi Nishimori. "Quantum annealing in the transverse Ising model". In: *Phys. Rev. E* 58 (5 Nov. 1998), pp. 5355–5363. DOI: 10.1103/PhysRevE.58.5355. URL: https://link.aps.org/doi/10.1103/PhysRevE.58.5355.

[11] Fred Glover, Gary Kochenberger, and Yu Du. "Quantum Bridge Analytics I: a tutorial on formulating and using QUBO models". In: *4OR* 17.4 (2019), pp. 335–371. DOI: 10.1007/s10288-019-00424-y.

[12] Frank Arute et al. *Quantum supremacy using a programmable superconducting processor*. Oct. 2019. DOI: 10.1038/s41586-019-1666-5. URL: http://dx.doi.org/10.1038/s41586-019-1666-5.

[13] Edwin Pednault et al. *Leveraging Secondary Storage to Simulate Deep 54-qubit Sycamore Circuits*. 2019. DOI: 10.48550/ARXIV.1910.09534. URL: https://arxiv.org/abs/1910.09534.

[14] John Preskill. "Quantum Computing in the NISQ era and beyond". In: *Quantum* 2 (Aug. 2018), p. 79. ISSN: 2521-327X. DOI: 10.22331/q-2018-08-06-79. URL: https://doi.org/10.22331/q-2018-08-06-79.

[15] Omar Costa Hamido, Giovanni Amedeo Cirillo, and Edoardo Giusto. "Quantum Synth: A Quantum-Computing-Based Synthesizer". In: *Proceedings of the 15th International Conference on Audio Mostly*. AM '20. Graz, Austria: Association for Computing Machinery, 2020, pp. 265–268. ISBN: 9781450375634. DOI: 10.1145/3411109.3411135. URL: https://doi.org/10.1145/3411109.3411135.

[16] Omar Costa Hamido. *Adventures in Quantumland*. https://escholarship.org/uc/item/93c1t8vx. University of California, Irvine, 2021.

[17] Janusz Kusyk, Samah M. Saeed, and Muharrem Umit Uyar. "Survey on Quantum Circuit Compilation for Noisy Intermediate-Scale Quantum Computers: Artificial Intelligence to Heuristics". In: *IEEE Transactions on Quantum Engineering* 2 (2021), pp. 1–16. DOI: 10.1109/TQE.2021.3068355.

[18] Stefanie Zbinden et al. "Embedding algorithms for quantum annealers with chimera and pegasus connection topologies". In: *International Conference on High Performance Computing*. Springer. 2020, pp. 187–206. DOI: 10.1007/978-3-030-50743-5_10.

[19] Alwin Zulehner and Robert Wille. *Introducing Design Automation for Quantum Computing*. Springer Nature, 2020.

[20] *Qiskit: An Open-source Framework for Quantum Computing*. 2019. DOI: 10.5281/zenodo.2562110.

[21] Guido Burkard et al. *Semiconductor Spin Qubits*. 2021. DOI: 10.48550/ARXIV.2112.08863. URL: https://arxiv.org/abs/2112.08863.

[22] M. H. Abobeih et al. *Fault-tolerant operation of a logical qubit in a diamond quantum processor*. en. May 2022. DOI: 10.1038/s41586-022-04819-6. URL: http://dx.doi.org/10.1038/s41586-022-04819-6.

[23] Shi-Yao Hou et al. *SpinQ Gemini: a desktop quantum computing platform for education and research*. en. July 2021. DOI: 10.1140/epjqt/s40507-021-00109-8. URL: http://dx.doi.org/10.1140/epjqt/s40507-021-00109-8.

[24] Elizabeth Crosson and Aram W. Harrow. "Simulated Quantum Annealing Can Be Exponentially Faster Than Classical Simulated Annealing". In: *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*. 2016, pp. 714–723. DOI: 10.1109/FOCS.2016.81.

[25] Mikio Nakahara and Tetsuo Ohmi. *Quantum Computing From Linear Algebra to Physical Realizations*. 1st ed. Boca Raton, Florida: CRC Press, 2008.

[26] Silvio Greco and Paolo Valabrega. *Lezioni di algebra lineare e geometria*. Libreria editrice universitaria Levrotto & Bella, 1992.

[27] National Institute of Standards and Technology. *The NIST Reference on Constants, Units and Uncertainty*. [Online at https://physics.nist.gov/cgi-bin/cuu/Value?h,; accessed 17-May-2022]. 2019.

[28] National Institute of Standards and Technology. *The NIST Reference on Constants, Units and Uncertainty*. [Online at https://physics.nist.gov/cgi-bin/cuu/Value?hbar,; accessed 17-May-2022]. 2019.

[29] S. Kullback and R. A. Leibler. "On Information and Sufficiency". In: *Ann. Math. Statist.* 22.1 (Mar. 1951), pp. 79–86. DOI: 10.1214/aoms/1177729694. URL: https://doi.org/10.1214/aoms/1177729694.

[30] Daniel D Stancil and Gregory T Byrd. *Principles of Superconducting Quantum Computers*. John Wiley & Sons, 2022.

[31] David C. McKay et al. "Efficient Z gates for quantum computing". In: *Phys. Rev. A* 96 (2 Aug. 2017), p. 022330. DOI: 10.1103/PhysRevA.96.022330.

[32] Andrew W. Cross et al. *Open Quantum Assembly Language*. 2017. DOI: 10.48550/ARXIV.1707.03429. URL: https://arxiv.org/abs/1707.03429.

[33] Ang Li et al. *QASMBench: A Low-level QASM Benchmark Suite for NISQ Evaluation and Simulation*. https://arxiv.org/abs/2005.13018. 2021.

[34] *JKU IIC circuits repository*. [Online] https://github.com/iic-jku/ibm_qx_mapping/tree/master/examples, accessed 23-May-2022.

[35] David P. DiVincenzo. "The Physical Implementation of Quantum Computation". In: *Fortschritte der Physik* 48.9-11 (2000), pp. 771–783. DOI: 10.1002/1521-3978(200009)48:9/11<771::AID-PROP771>3.0.CO;2-E.

[36] Mark Fox. *Quantum optics: an introduction*. Vol. 15. Oxford university press, 2006.

[37] Siegmund Brandt and Hans Dieter Dahmen. *The Picture Book of Quantum Mechanics*. 4th. Springer Publishing Company, Incorporated, 2012. ISBN: 1461439507.

[38] Philip Krantz et al. "A quantum engineer's guide to superconducting qubits". In: *Applied Physics Reviews* 6.2 (2019), p. 021318. DOI: 10.1063/1.5089550.

[39] Uri Vool and Michel Devoret. "Introduction to quantum electromagnetic circuits". en. In: *International Journal of Circuit Theory and Applications* 45.7 (June 2017), pp. 897–934. DOI: 10.1002/cta.2359. URL: http://dx.doi.org/10.1002/cta.2359.

[40] Thomas E. Roth, Ruichao Ma, and Weng C. Chew. "An Introduction to the Transmon Qubit for Electromagnetic Engineers". In: (2021). DOI: 10.48550/ARXIV.2106.11352. URL: https://arxiv.org/abs/2106.11352.

[41] Colin D Bruzewicz et al. "Trapped-ion quantum computing: Progress and challenges". In: *Applied Physics Reviews* 6.2 (2019), p. 021314. DOI: 10.1063/1.5088164.

[42] T. P. Harty et al. "High-Fidelity Preparation, Gates, Memory, and Readout of a Trapped-Ion Quantum Bit". In: *Phys. Rev. Lett.* 113 (22 Nov. 2014), p. 220501. DOI: 10.1103/PhysRevLett.113.220501. URL: https://link.aps.org/doi/10.1103/PhysRevLett.113.220501.

[43] Stewart Allen et al. "Reconfigurable and Programmable Ion Trap Quantum Computer". In: *2017 IEEE International Conference on Rebooting Computing (ICRC)*. 2017, pp. 1–3. DOI: 10.1109/ICRC.2017.8123665.

[44] Justin E. Christensen et al. *High-fidelity manipulation of a qubit enabled by a manufactured nucleus*. en. Apr. 2020. DOI: 10.1038/s41534-020-0265-5. URL: http://dx.doi.org/10.1038/s41534-020-0265-5.

[45] Thomas Monz et al. "Realization of a scalable Shor algorithm". In: *Science* 351.6277 (2016), pp. 1068–1070. DOI: 10.1126/science.aad9480.

[46] L Petit et al. "Universal quantum logic in hot silicon qubits". In: *Nature* 580.7803 (2020), pp. 355–359. DOI: 10.4121/uuid:22653416-85b0-4d7d-ad48-65967f9ea7ad.

[47] Alejandro Gaita-Ariño et al. "Molecular spins for quantum computation". In: *Nature chemistry* 11.4 (2019), pp. 301–309. DOI: 10.1038/s41557-019-0232-y.

[48] National Institute of Standards and Technology. *The NIST Reference on Constants, Units and Uncertainty*. [Online at https://physics.nist.gov/cgi-bin/cuu/Value?gammaebar,; accessed 5-May-2022]. 2019.

[49] National Institute of Standards and Technology. *The NIST Reference on Constants, Units and Uncertainty*. [Online at https://physics.nist.gov/cgi-bin/cuu/Value?gammapbar,; accessed 5-May-2022]. 2019.

[50] National Institute of Standards and Technology. *The NIST Reference on Constants, Units and Uncertainty*. [Online at https://physics.nist.gov/cgi-bin/cuu/Value?gammanbar,; accessed 5-May-2022]. 2019.

[51] National Institute of Standards and Technology. *The NIST Reference on Constants, Units and Uncertainty*. [Online at https://physics.nist.gov/cgi-bin/cuu/Value?gem,; accessed 20-May-2022]. 2019.

[52] National Institute of Standards and Technology. *The NIST Reference on Constants, Units and Uncertainty*. [Online at https://physics.nist.gov/cgi-bin/cuu/CCValue?mubshhz,; accessed 20-May-2022]. 2019.

[53] Thomas Vosegaard. *NMR Periodic Table*. [Online at https://periodic.pastis.dk/,; accessed 5-May-2022]. 2022.

[54] J Robert Johansson, Paul D Nation, and Franco Nori. "QuTiP: An open-source Python framework for the dynamics of open quantum systems". In: *Computer Physics Communications* 183.8 (2012), pp. 1760–1772. DOI: 10.1016/j.cpc.2012.02.021.

[55] Alwin Zulehner and Robert Wille. "Advanced Simulation of Quantum Computations". In: *Trans. on CAD of Integrated Circuits and Systems* 38.5 (2019), pp. 848–859. DOI: 10.1109/TCAD.2018.2834427.

[56] Gian Giacomo Guerreschi et al. "Intel Quantum Simulator: a cloud-ready high-performance simulator of quantum circuits". In: *Quantum Science and Technology* 5.3 (May 2020), p. 034007. DOI: 10.1088/2058-9565/ab8505. URL: https://doi.org/10.1088%2F2058-9565%2Fab8505.

[57] Nader Khammassi et al. "QX: A high-performance quantum computer simulation platform". In: *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017*. IEEE, 2017, pp. 464–469. DOI: 10.23919/DATE.2017.7927034.

[58] G. A. Cirillo, G. Turvani, and M. Graziano. "A Quantum Computation Model for Molecular Nanomagnets". In: *IEEE Transactions on Nanotechnology* 18 (2019), pp. 1027–1039. ISSN: 1941-0085. DOI: 10.1109/TNANO.2019.2939910.

[59] G. A. Cirillo et al. "Advances in Molecular Quantum Computing: from Technological Modeling to Circuit Design". In: *2020 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*. IEEE, 2020, pp. 132–137. DOI: 10.1109/ISVLSI49217.2020.00033.

[60] Mario Simoni et al. "Towards Compact Modeling of Noisy Quantum Computers: A Molecular-Spin-Qubit Case of Study". In: 18.1 (Sept. 2021). https://doi.org/10.1145/3474223. ISSN: 1550-4832. DOI: 10.1145/3474223.

[61] Samar K Saha. *Compact models for integrated circuit design: conventional transistors and beyond*. Taylor & Francis, 2015. DOI: 10.1201/b19117.

[62] Abdullah Ash- Saki, Mahabubul Alam, and Swaroop Ghosh. "Experimental Characterization, Modeling, and Analysis of Crosstalk in a Quantum Computer". In: *IEEE Transactions on Quantum Engineering* 1 (2020), pp. 1–6. DOI: 10.1109/TQE.2020.3023338.

[63] G Massimo Palma, Kalle-Antti Suominen, and Artur Ekert. "Quantum computers and dissipation". In: *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences* 452.1946 (1996), pp. 567–584. DOI: 10.1098/rspa.1996.0029.

[64] Lieven MK Vandersypen et al. "Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance". In: *Nature* 414.6866 (2001), pp. 883–887.

[65] Jesús Ferrando-Soria et al. "A modular design of molecular qubits to implement universal quantum gates". In: *Nature Communications* 7.1 (2016), p. 11377. ISSN: 2041-1723. DOI: 10.1038/ncomms11377.

[66] Alessandro Chiesa, Paolo Santini, and Stefano Carretta. "Supramolecular complexes for quantum simulation". In: *Magnetochemistry* 2.4 (2016), p. 37.

[67] Matteo Atzori et al. "A two-qubit molecular architecture for electron-mediated nuclear quantum simulation". In: *Chem. Sci.* 9 (29 2018), pp. 6183–6192. DOI: 10.1039/C8SC01695J.

[68] Peter Atkins and Julio de Paula. *Physical Chemistry*. 9th ed. New York City, New York: W H Freeman and Co, 2010.

[69] Ivan S. Oliveira et al. *NMR Quantum Information Processing*. 1st ed. Oxford, UK: Elsevier, 2007.

[70] National Institute of Advanced Industrial Science and Technology. *Spectral Database for Organic Compounds*. [Online at https://sdbs.db.aist.go.jp,; accessed 5-May-2022]. 2022.

[71] Jingwei Wen et al. "Experimental realization of quantum algorithms for a linear system inspired by adiabatic quantum computing". In: *Phys. Rev. A* 99.1 (Jan. 2019), p. 012320. DOI: 10.1103/PhysRevA.99.012320. URL: https://link.aps.org/doi/10.1103/PhysRevA.99.012320.

[72] Keren Li et al. "Quantum spacetime on a quantum simulator". In: *Communications Physics* 2.1 (Dec. 2019), pp. 122–128. DOI: 10.1038/s42005-019-0218-5.

[73] Yasushi Kondo et al. *Hamiltonian of Homonucleus Molecules for NMR Quantum Computing*. 2005. eprint: arXiv:quant-ph/0503067.

[74] Amandeep Singh et al. "Experimental classification of entanglement in arbitrary three-qubit pure states on an NMR quantum information processor". In: *Phys. Rev. A* 98 (3 Sept. 2018), p. 032301. DOI: 10.1103/PhysRevA.98.032301. URL: https://link.aps.org/doi/10.1103/PhysRevA.98.032301.

[75] Mikio Nakahara et al. *Physical realizations of quantum computing : are the DiVincenzo criteria fulfilled in 2004?* 1st ed. Singapore Hackensack, N.J: World Scientific, 2006.

[76] Lieven MK Vandersypen and Isaac L Chuang. "NMR techniques for quantum control and computation". In: *Reviews of modern physics* 76.4 (2005), p. 1037. DOI: 10.1103/RevModPhys.76.1037.

[77] S Siddhaye et al. "Pharmaceutical product design using combinatorial optimization". In: *Computers & Chemical Engineering* 28.3 (2004), pp. 425–434. DOI: 10.1016/j.compchemeng.2003.08.011.

[78] Frauke Liers, Tim Nieberg, and Gregor Pardella. *Via Minimization in VLSI Chip Design Application of a Planar Max-Cut Algorithm*. 2011. URL: http://e-archive.informatik.uni-koeln.de/630/.

[79] Carlos D Gonzalez Calaza, Dennis Willsch, and Kristel Michielsen. "Garden optimization problems for benchmarking quantum annealers". In: *Quantum Information Processing* 20.9 (2021). https://doi.org/10.1007/s11128-021-03226-6, pp. 1–22.

[80] Kotaro Tanahashi et al. "Application of Ising machines and a software development for Ising machines". In: *Journal of the Physical Society of Japan* 88.6 (2019), p. 061010. DOI: 10.7566/JPSJ.88.061010.

[81] Joseph T. Iosue. *qubovert Documentation*. [Online at https://qubovert.readthedocs.io/en/stable/,; accessed 13-September-2021]. 2019.

[82] Andrey Popov. "Genetic algorithms for optimization". In: *User Manual, Hamburg* 2013 (2005). URL: https://www.p0p0v.com/science/downloads/Popov05a.pdf.

[83] Dongshu Wang, Dapei Tan, and Lei Liu. "Particle swarm optimization algorithm: an overview". In: *Soft Computing* 22.2 (2018), pp. 387–408. DOI: 10.1007/s00500-016-2474-6.

[84] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. "Optimization by Simulated Annealing". In: *Science* 220.4598 (1983), pp. 671–680. ISSN: 0036-8075. DOI: 10.1126/science.220.4598.671.

[85] David S Johnson et al. "Optimization by simulated annealing: An experimental evaluation; part I, graph partitioning". In: *Operations research* 37.6 (1989), pp. 865–892. DOI: 10.1287/opre.37.6.865.

[86] W. K. Hastings. "Monte Carlo sampling methods using Markov chains and their applications". In: *Biometrika* 57.1 (Apr. 1970), pp. 97–109. ISSN: 0006-3444. DOI: 10.1093/biomet/57.1.97.

[87] Stuart Geman and Donald Geman. "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-6.6 (1984), pp. 721–741. DOI: 10.1109/TPAMI.1984.4767596.

[88] Hasitha Muthumala Waidyasooriya and Masanori Hariyama. "A GPU-Based Quantum Annealing Simulator for Fully-Connected Ising Models Utilizing Spatial and Temporal Parallelism". In: *IEEE Access* 8 (2020), pp. 67929–67939. DOI: 10.1109/ACCESS.2020.2985699.

[89] Hasitha Waidyasooriya and Masanori Hariyama. "Highly-Parallel FPGA Accelerator for Simulated Quantum Annealing". In: *IEEE Transactions on Emerging Topics in Computing* (2019), pp. 1–1. DOI: 10.1109/TETC.2019.2957177.

[90] Hayato Goto, Kosuke Tatsumura, and Alexander R Dixon. "Combinatorial optimization by simulating adiabatic bifurcations in nonlinear Hamiltonian systems". In: *Science advances* 5.4 (2019), eaav2372. DOI: 10.1126/sciadv.aav2372.

[91] Hayato Goto et al. "High-performance combinatorial optimization based on classical mechanics". In: *Science Advances* 7.6 (2021), eabe7953. DOI: 10.1126/sciadv.abe7953.

[92] Sanroku Tsukamoto et al. "An accelerator architecture for combinatorial optimization problems". In: *Fujitsu Sci. Tech. J* 53.5 (2017), pp. 8–13. URL: https://www.fujitsu.com/global/documents/about/resources/publications/fstj/archives/vol53-5/paper02.pdf.

[93] Maliheh Aramon et al. "Physics-Inspired Optimization for Quadratic Unconstrained Problems Using a Digital Annealer". In: *Frontiers in Physics* 7 (2019), p. 48. DOI: 10.3389/fphy.2019.00048.

[94] Jingya Li. *Downlink resource allocation in cooperative wireless networks*. Doctoral Thesis, https://www.dissertations.se/dissertation/4d087c8401/. Chalmers Tekniska Hogskola (Sweden), 2015.

[95] Michael Grant and Stephen Boyd. *CVX: Matlab Software for Disciplined Convex Programming, version 2.1*. http://cvxr.com/cvx. Mar. 2014.

[96] Steven Diamond and Stephen Boyd. "CVXPY: A Python-embedded modeling language for convex optimization". In: *Journal of Machine Learning Research* 17.83 (2016), pp. 1–5.

[97] B. O'Donoghue et al. "Conic Optimization via Operator Splitting and Homogeneous Self-Dual Embedding". In: *Journal of Optimization Theory and Applications* 169.3 (June 2016), pp. 1042–1068. URL: http://stanford.edu/~boyd/papers/scs.html.

[98] Lilian Childress and Ronald Hanson. "Diamond NV centers for quantum computing and quantum networks". In: *MRS Bulletin* 38.2 (2013), pp. 134–138. DOI: 10.1557/mrs.2013.20.

[99] Thomas Grurl, Jürgen Fuß, and Robert Wille. "Considering decoherence errors in the simulation of quantum circuits using decision diagrams". In: *Proceedings of the 39th International Conference on Computer-Aided Design*. New York, NY, USA: Association for Computing Machinery, 2020, pp. 1–7. DOI: 10.1145/3400302.3415622.

[100] Rui Pereira et al. "Energy efficiency across programming languages: how do energy, time, and memory relate?" In: *Proceedings of the 10th ACM SIGPLAN International Conference on Software Language Engineering*. https://doi.org/10.1145/3136014.3136031. 2017, pp. 256–267.

[101] Adriano Barenco et al. "Elementary gates for quantum computation". In: *Phys. Rev. A* 52 (5 Nov. 1995), pp. 3457–3467. DOI: 10.1103/PhysRevA.52.3457. URL: https://link.aps.org/doi/10.1103/PhysRevA.52.3457.

[102] Peter W Shor. "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer". In: *SIAM review* 41.2 (1999), pp. 303–332. DOI: 10.1137/S0036144598347011.

[103] Malcolm H Levitt. *Spin dynamics: basics of nuclear magnetic resonance*. John Wiley & Sons, 2013.

[104] Andrew W. Cross et al. *Open Quantum Assembly Language*. Arxiv repository: arXiv:1707.03429. 2017. eprint: arXiv:1707.03429v2. URL: https://arxiv.org/abs/1707.03429v2.

[105] Tyler Andrews. "Computation time comparison between Matlab and C++ using launch windows". In: (2012).

[106] Jacob Biamonte et al. "Quantum machine learning". In: *Nature* 549.7671 (2017), pp. 195–202. DOI: 10.1038/nature23474.

This Ph.D. thesis has been typeset by means of the TEX-system facilities. The typesetting engine was pdfLATEX. The document class was `toptesi`, by Claudio Beccari, with option `tipotesi=scudo`. This class is available in every up-to-date and complete TEX-system installation.