Helsinki University of Technology Laboratory of Physics
Teknillisen korkeakoulun fysiikan laboratorio
Espoo 2007 Dissertation 151

FROM THE CONTROL OF QUANTUM SYSTEMS TO MULTIQUBIT LOGIC

Ville Bergholm

Dissertation for the degree of Doctor of Science in Technology to be presented with due permission of the Department of Engineering Physics and Mathematics for public examination and debate in Auditorium TU2 at Helsinki University of Technology (Espoo, Finland) on the 4th of December, 2007, at 12 o´clock noon.

Helsinki University of Technology
Department of Engineering Physics and Mathematics
Laboratory of Physics

Teknillinen korkeakoulu
Teknillisen fysiikan ja matematiikan osasto
Fysiikan laboratorio

Dissertations of Laboratory of Physics, Helsinki University of Technology
ISSN 1455-1802

Dissertation 151 (2007):
Ville Bergholm: From the control of quantum systems to multiqubit logic

Opponent:
Dr. Jeremy O'Brien, University of Bristol, UK

Pre-examiners:
Prof. Alán Aspuru-Guzik, Harvard University, USA
Dr. Jyrki Piilo, University of Turku, Finland

ISBN 978-951-22-9090-1 (print)
ISBN 978-951-22-9091-8 (electronic)

Multiprint Oy
Espoo 2007

ABSTRACT OF DOCTORAL DISSERTATION	HELSINKI UNIVERSITY OF TECHNOLOGY P. O. BOX 1000, FI-02015 TKK http://www.tkk.fi

Author Ville Axel Bergholm

Name of the dissertation

From the control of quantum systems to multiqubit logic

Manuscript submitted 20.08.2007	Manuscript revised
Date of the defence 04.12.2007	
□ Monograph	☒ Article dissertation (summary + original articles)

Department	Engineering Physics and Mathematics
Laboratory	Laboratory of Physics
Field of research	quantum computing
Opponent(s)	Dr. Jeremy O'Brien, University of Bristol, UK
Supervisor	Prof. Risto Nieminen
Instructor	Doc. Mikko Möttönen

Abstract

Quantum computing and quantum information science are two recently discovered and rapidly growing fields of physics that show substantial promise in providing new and valuable technologies in the foreseeable future. Large-scale quantum computers, if ever realized experimentally, are likely to outperform their classical counterparts in a number of important computational tasks, the most important of which may be the accurate simulation of many-body quantum systems such as the ones encountered in physics, chemistry and life sciences.

This thesis investigates the problem of controlling quantum systems for the purpose of performing quantum information processing tasks. The problem is approached from a theoretical and simulational viewpoint. The work contained here encompasses a range of levels of abstraction. Firstly, we discuss the decomposition of abstract multiqubit logic gates into sequences of simple elementary gates. Secondly, we study the local commutational properties of two-qubit gates using local gate invariants. Thirdly, we develop methods for the physical implementation of the elementary gates through the control of specific quantum systems, possibly in the presence of noise and decoherence.

We present a new, almost optimal n-qubit gate decomposition based on the cosine-sine decomposition, which utilizes a likewise new intermediate quantum circuit structure we call a uniformly controlled gate. We then show how they can be used in constructing a general state transformation circuit. Both of the resulting circuits can be efficiently implemented using nearest-neighbor gates which makes their physical realization simpler. A local gate invariant is introduced which can be used to assess the suitability of two-qubit gates for serving as the entangling gate in elementary gate libraries. Finally, we develop numerical optimization methods for finding near-optimal control sequences for generating one- and two-qubit gates, both in closed quantum systems and in the presence of Markovian noise.

Keywords quantum computing, quantum gate, entanglement, decoherence, quantum control

ISBN (printed) 978-951-22-9090-1	ISSN (printed) 1455-1802
ISBN (pdf) 978-951-22-9091-8	ISSN (pdf)
Language English	Number of pages 60 p. + app. 58 p.

Publisher Laboratory of Physics, Helsinki University of Technology

Print distribution Laboratory of Physics, Helsinki University of Technology

☒ The dissertation can be read at http://lib.tkk.fi/Diss/2007/isbn9789512290918/

VÄITÖSKIRJAN TIIVISTELMÄ	TEKNILLINEN KORKEAKOULU PL 1000, 02015 TKK http://www.tkk.fi

Tekijä Ville Axel Bergholm

Väitöskirjan nimi

Kvanttimekaanisten systeemien kontrollista monen qubitin logiikkaan

Käsikirjoituksen päivämäärä 20.08.2007	Korjatun käsikirjoituksen päivämäärä

Väitöstilaisuuden ajankohta 04.12.2007

□ Monografia	☒ Yhdistelmäväitöskirja (yhteenveto + erillisartikkelit)

Osasto	Teknillisen fysiikan ja matematiikan osasto
Laboratorio	Fysiikan laboratorio
Tutkimusala	kvanttilaskenta
Vastaväittäjä(t)	Dr. Jeremy O’Brien, University of Bristol, UK
Työn valvoja	Prof. Risto Nieminen
Työn ohjaaja	Dos. Mikko Möttönen

Tiivistelmä

Kvanttilaskenta ja kvantti-informaatiotiede ovat hiljattain kehitettyjä ja nopeasti kasvavia fysiikan aloja, jotka saattavat tuottaa uusia hyödyllisiä teknologioita jo lähitulevaisuudessa. Jos suuren mittakaavan kvanttitietokoneita kyetään joskus valmistamaan, ne todennäköisesti lyövät klassiset vastineensa useissa tärkeissä laskentatehtävissä, joista tärkein saattaa olla fysiikassa, kemiassa ja biotieteissä esiintyvien monihiukkaskvanttisysteemien tarkka simulaatio.

Tämä väitöskirja käsittelee kvanttimekaanisten systeemien kontrollointia informaation käsittelemiseksi. Ongelmaa lähestytään teoreettisesta ja simulationaalisesta näkökulmasta tavoilla, jotka kattavat useita eri abstraktiotasoja. Aluksi käsittelemme abstraktien moniqubittiporttien hajottamista jonoksi elementaariportteja. Seuraavaksi tutkimme kaksiqubittiporttien lokaaleja kommutatiivisia ominaisuuksia käyttäen apuna lokaaleja portti-invariantteja. Lopulta kehitämme menetelmiä elementaariporttien toteuttamiseksi kontrolloimalla kvanttisysteemejä, joissa saattaa myös esiintyä kohinaa tai dekoherenssia.

Esittelemme uuden, lähes optimaalisen n:n qubitin porttihajotelman, joka perustuu kosini-sini -hajotelmaan ja hyödyntää niinikään uutta tasaisesti kontrolloiduksi portiksi kutsumaamme keskitason kvanttipiirirakennetta. Näytämme myös kuinka näiden porttien avulla voidaan muodostaa yleinen tilamuunnospiiri. Kumpikin edellämainituista piireistä voidaan implementoida tehokkaasti käyttäen ainoastaan vierekkäisiin qubitteihin operoivia elementaariportteja, mikä tekee niiden fysikaalisesta realisaatiosta yksinkertaisempaa. Johdamme uuden lokaalin portti-invariantin, jonka avulla voi arvioida kaksiqubittiporttien soveltuvuutta elementaariporttikirjastojen lomittavaksi portiksi. Lisäksi kehitämme numeerisia optimointimenetelmiä miltei optimaalisten yksi- ja kaksiqubittikontrollisekvenssien muodostamiseksi sekä suljetuissa kvanttisysteemeissä että markovisen kohinan vaikutuksen alla.

Asiasanat kvanttilaskenta, kvanttiportti, lomittuminen, dekoherenssi, kvanttisäätö

ISBN (painettu) 978-951-22-9090-1	ISSN (painettu) 1455-1802
ISBN (pdf) 978-951-22-9091-8	ISSN (pdf)
Kieli englanti	Sivumäärä 60 s. + liit. 58 s.

Julkaisija Fysiikan laboratorio, Teknillinen korkeakoulu

Painetun väitöskirjan jakelu Fysiikan laboratorio, Teknillinen korkeakoulu

☒ Luettavissa verkossa osoitteessa http://lib.tkk.fi/Diss/2007/isbn9789512290918/

Preface

The work described within this dissertation has been carried out in the Materials Physics Laboratory and later in the quantum many-body physics group of the Laboratory of Physics at the Helsinki University of Technology during the years 2004–2007. It was largely funded by a three-year grant from the Finnish Cultural Foundation and the Quantum Computing project of the Academy of Finland, which I gratefully acknowledge. The travel grant I received from the Research Foundation of the Helsinki University of Technology which enabled me to present our work in the EQIS'2005 conference in Tokyo is also greatly appreciated.

I would like to give thanks my original supervisor and instructor, the late Prof. Martti Salomaa, for his enthusiastic and encouraging support for my research, as well as my present supervisor, Prof. Risto Nieminen for picking things up after Martti passed away. I'd also like to thank my co-authors Mikko Möttönen (who also served as my instructor after Martti's departure), Juha Vartiainen, Martti Salomaa, Laura Koponen, Andreas Spörl, Thomas Schulte-Herbrüggen, Steffen Glaser, Markus Storcz, Johannes Ferber, Frank Wilhelm, Olli-Pentti Saira, and Teemu Ojanen for their important contributions to our research. I'm indebted to Prof. Mikio Nakahara for introducing me to quantum computing. Colleagues in the Materials Physics Laboratory and later in the QMP group of the Laboratory of Physics, as well as the people with whom I enjoyed pizza, laughs and interesting conversations on numerous Fridays, have my gratitude for providing a pleasant and inspiring working environment.

Last but not least, I would like to thank my family and friends for their inexhaustible encouragement and love.

Otaniemi, November 2007

Ville Bergholm

List of Publications

This thesis is a review of the author's work in the field of quantum information and computation. It consists of an overview and the following publications in this field:

I M. Möttönen, J. J. Vartiainen, V. Bergholm, M. M. Salomaa, *Quantum circuits for general multiqubit gates*, Phys. Rev. Lett. **93**, 130502 (2004).

II M. Möttönen, J. J. Vartiainen, V. Bergholm, M. M. Salomaa, *Transformation of quantum states using uniformly controlled rotations*, Quantum Information and Computation **5**, 467 (2005).

III V. Bergholm, J. J. Vartiainen, M. Möttönen, M. M. Salomaa, *Quantum circuits with uniformly controlled one-qubit gates*, Phys. Rev. A **71**, 052330 (2005).

IV L. Koponen, V. Bergholm, M. M. Salomaa, *A discrete local invariant for quantum gates*, Quantum Information and Computation **6**, 58 (2006).

V A. K. Spörl, T. Schulte-Herbrüggen, S. J. Glaser, V. Bergholm, M. J. Storcz, J. Ferber, F. K. Wilhelm, *Optimal control of coupled Josephson qubits*, Phys. Rev. A **75**, 012302 (2007).

VI O.-P. Saira, V. Bergholm, T. Ojanen, M. Möttönen, *Equivalent qubit dynamics under classical and quantum noise*, Phys. Rev. A **75**, 012308 (2007).

Throughout the overview, these papers are referred to by their Roman numerals.

Author's contribution

The research presented in this dissertation has been carried out in the Materials Physics Laboratory and the Laboratory of Physics at the Helsinki University of Technology during the years 2004–2007. The author has had a central role in all the work included in this thesis, having mainly written the manuscripts for publications **III** and **IV**, and actively participated in writing the manuscripts for publications **I**, **II** and **VI**.

Publications **I** and **III** deal with decomposing arbitrary quantum gates into a circuit of one- and two-qubit gates using a recursive cosine-sine decomposition, implemented using a likewise recursive construction for an intermediate gate we call a uniformly controlled rotation, whereas publication **II** utilizes uniformly controlled rotations to implement a general state preparation procedure. The author strongly participated in the development of these ideas and performed much of the mathematical analysis and the numerical verification of the results in these publications. Publication **IV** introduces a local gate invariant which describes the local commutational properties of multiqubit gates, based on the author's idea. Publication **V** describes an optimization method for generating control sequences for noiseless quantum systems and, using an existing device as an example, demonstrates how these sequences could be realistically implemented. Here the author's contribution was in the original optimization work and analysis which resulted in the discovery of the particularly simple control sequences, and in the finishing of the manuscript. Publication **VI** presents a method for simulating quantum systems subject to Markovian classical noise, and shows how this type of noise can arise from a genuine quantum mechanical environment under the Born-Markov approximation. As an example, the method is used to derive noise-resistant one-qubit control sequences. The author strongly participated in the development of the mathematical formalism for the simulation method, as well as performed the numerical calculations behind the plots.

List of Abbreviations

CNOT	controlled NOT
CSD	cosine-sine decomposition
DFS	decoherence-free subspace/subsystem
EM	electromagnetic
EPR	Einstein-Podolsky-Rosen
GRAPE	gradient ascent pulse engineering
HSP	hidden subgroup problem
LOCC	local operations and classical communication
NMR	nuclear magnetic resonance
RTN	random telegraph noise
QC	quantum computing
QCM	quantum circuit model
QFT	quantum Fourier transform
QIP	quantum information processing
QTM	quantum Turing machine
TM	Turing machine

Contents

Preface **vii**

List of Publications **viii**

Author's contribution **ix**

List of Abbreviations **x**

Contents **xi**

1 Introduction **1**

2 Quantum mechanics **7**

2.1 Fundamentals 7

2.2 Entanglement and decoherence 10

3 Quantum computing **13**

3.1 Basics 13

3.2 Quantum algorithms 16

3.3 Models of quantum computing 17

3.4 Proposed architectures 19

4 Quantum circuits **25**

4.1 Gates and circuit diagrams 25

4.2 Gate decompositions 29

4.3 Local gate invariants 30

5 Control sequences **33**

5.1 Optimization 33

5.2 Simulation of noise 36

6 Summary **39**

References **41**

Appendix A Notation

A.1 Pauli matrices

1 Introduction

An algorithm is a concept that is easy to understand intuitively but difficult to rigorously define. Loosely speaking, an algorithm is a finite list of simple instructions that enables one to solve a specific type of a problem. The process of applying an algorithm is called a *computation.* Algorithms can vary greatly in complexity and generality; for example, the algorithm to divide an angle into two equal parts using a compass and a straightedge can be written down in a few lines of English, whereas the algorithm for winning a game of chess (or forcing it into a stalemate), even though it theoretically must exist, would be extremely complicated and totally infeasible to derive using present-day means.

Due to its intuitive nature, algorithm is by no means a new concept. Detailed descriptions of algorithms for solving various nontrivial mathematical and geometrical problems can be found in preserved works from antiquity. A well known example is the Euclidean algorithm, a method for finding the greatest common divisor of two integers. It appears in Euclid's Elements [1], written around 300 BCE, but most likely predates it by a hundred years or more.

The rapid development of computability theory in the 1930s, through the work of Alonzo Church [2, 3, 4], Stephen C. Kleene [5, 6], Alan Turing [7] and others, brought about the first rigorous mathematical definitions of the concept of algorithm. It was understood that algorithms need to be defined in the context of a specific model of computation. During this era, many superficially different models were presented, but none of them turned out to be fundamentally more powerful than the others. Instead, it was found out that they all could simulate each other perfectly. This observation was condensed into the Church-Turing thesis, which gives the current definition of algorithmic computability using the Turing machine (TM) model of computation as a basis.

Church-Turing thesis

Any problem that is computable can be solved using a Turing machine.

A TM is a hypothetical device designed to run a single algorithm, or solve a given class of problems. It consists of an infinite one-dimensional memory tape and a moving read/write head that is a finite state machine controlled by an action table. The tape consists of cells, each containing a symbol from a finite alphabet. The choice of the tape alphabet, the set of states, and the contents of the action table define a TM. At the beginning of the computation, the memory is initialized to the finite-sized input for the algorithm, and the read/write head sits at memory cell zero in its startup state. The computation consists of a series of steps. During each step, the head finds an entry in the action table corresponding to its current state and the symbol read from the currently addressed memory cell and then, following the directions in the entry, writes a symbol in the current cell, moves one step right or left, and changes its state. When the head reaches a special halting state, the computation is finished and the result can be read from the memory.

This model of computation seems a bit limited in the sense that each TM can only run a single algorithm. However, it has been proven that there exists a universal Turing machine (UTM), which is capable of simulating any other Turing machine. For a UTM, the input consists of a series of symbols defining the TM to be simulated, followed by the actual input data for the algorithm. The concept is very similar to modern programmable computers. Given sufficient memory and time, in the sense of the Church-Turing thesis, such a device is capable of solving any computational problem that *can* be solved. There is a catch, however; the Church-Turing thesis does not say anything about the efficiency of the computation, i.e. the resources of memory space and computation time it requires.

In computational complexity theory, a problem is said to be *tractable* or *efficiently solvable* if it has an algorithm with a running time that is at most polynomial in the size of the problem. Problems that are not efficiently solvable in this sense are called *hard.* From this convention stems the notion of polynomial equivalence: two models of computation are considered equally efficient if they can simulate each other with an overhead that is at most polynomial. Several new models of computation have been developed since the 1930s, but none of them have been found fundamentally more efficient than the probabilistic Turing machine model, which is simply a TM with a random number generator. This has led to the formulation of the *strong* Church-Turing thesis, which is actually an unverifiable hypothesis.

> **Strong Church-Turing thesis**
> Any algorithmic process can be simulated using a probabilistic Turing machine at a polynomial overhead.

The group or *complexity class* of algorithms that are considered efficient within this hypothesis is known as BPP (Bounded error, Probabilistic resources, Polynomial time).

During the 1980s, it was realized that all the computational models proposed thus far either intentionally or accidentally made the implicit assumption that the computation had to follow the rules of classical physics. This had made sense, since one of the goals of computation theory is to design machines for performing computation for us using technology mostly based on classical physics. However, since the beginning of the 20^{th} century it had been known that classical physics could not accurately describe the small-scale workings of our Universe. Beyond a certain limit, quantum mechanics had to be used instead. Moore's Law [8], the well-known observation that the number of transistors on an integrated circuit of a given price doubles roughly every two years, has held remarkably true since its inception in 1965. If this process continues unabated, in the near future the size of a single transistor will reach the scale where quantum mechanical effects dominate its behaviour. Since quantum mechanics will eventually intrude into the field of classical computing anyway, why not try to take advantage of it?

In 1980 Paul Benioff introduced the concept of a quantum Turing machine (QTM) [9]. It is basically a Turing machine where the position of the read/write head, its internal state and the states of the memory cells are all quantum mechanical observables, and the action table is replaced by a fixed local unitary propagator. This idea was further developed by David Deutsch [10] into a new model of computation. The

QTM is by no means the only possible computational model utilizing quantum mechanics. The quantum circuit model (QCM) [11], for example, is much more accessible and useful, and has been proven equivalent to the QTM [12]. All the computational models equivalent to the QTM are collectively known as models of *quantum computing.*

The new model of computation necessitated the introduction of a host of new complexity classes, most importantly BQP (Bounded error, Quantum resources, Polynomial time) [13]. Colloquially speaking it is the class of efficient quantum algorithms. Considering that classical physics can be obtained as a limiting case of quantum mechanics, it is not surprising that quantum computers were determined to be able to simulate all classical probabilistic algorithms with a polynomial overhead. In the language of complexity classes, this fact is stated as BPP $\subset$ BQP. Also, classical computers can simulate quantum algoritms with an exponential overhead, so from a computability theory perspective quantum computing is fundamentally no more powerful than classical computing. However, quantum computing may yet turn out to be a far more *efficient* model, capable of solving certain types of problems much *faster* than its classical counterpart.

The first example that quantum computing is capable of doing something that classical computing cannot was the discovery of the Deutsch-Jozsa algorithm in 1992 [14]. It can determine with full certainty whether a given binary function with n-bit input is constant or balanced by evaluating it only once. It has no practical importance as there is a classical probabilistic algorithm with comparable performance, but it served as an inspiration for things to come. In 1994, Peter Shor published efficient quantum algorithms for factoring integers and computing discrete logarithms [15]. What makes these algorithms important is the fact that many common public key cryptographic algorithms such as the Diffie-Hellman [16] and Rivest-Shamir-Adleman [17] key exchange protocols, as well as their elliptic curve variants, rely on the assumed hardness of one of these problems. As these protocols underlie most popular cryptosystems such as RSA, DSA, and ElGamal [18], the Shor algorithms could conceivably be used to break them all if the proper hardware was available.

This success was continued in 1996 when Lov Grover published a quantum algorithm for unsorted database search [19]. The best possible classical algorithm for this task requires $O(n)$ oracle queries, but the Grover algorithm manages to complete the task in $O(\sqrt{n})$ queries. Even though this is only a polynomial improvement, the database search is such an ubiquitous problem that the Grover algorithm may prove to be extremely valuable.

Despite these attractive applications, the practical viability of quantum computing was in doubt, since there was no known way to perform error correction in a quantum computer. Simple redundancy could not be used because quantum information cannot be cloned [20, 21]. The situation was remedied in 1995 when Peter Shor [22] and Andrew Steane [23] presented the first quantum error correcting codes. In 1996 it was shown that the entire process of quantum computing can be made fault-tolerant [24] if the error probability of a single operation can be made low enough.

So, is quantum computing fundamentally more efficient than classical computing or, in terms of complexity classes, is BPP $\neq$ BQP? Presently, we do not know. However, considering how important a tool automated computing has become to our civilization, this question certainly merits an answer.

Be this as it may, the use of the rules of quantum mechanics to process information has also other important uses besides computing. In a communication context quantum information processing (QIP) presents many unintuitive and even startling results, many of which have something to do with a phenomenon known as quantum entanglement. For example, two parties sharing an entangled quantum state may use a protocol known as superdense coding [25] to communicate two bits of information by transmitting only a single bit on a classical channel, the entanglement taking care of the rest. Even more importantly, a shared entangled state and a classical communication channel can be used to faithfully transfer an arbitrary quantum state from one party to the other, a process known as quantum teleportation [26]. Quantum communication protocols can even offset the damage done to the world of applied cryptography by the Shor algorithms. If a quantum channel exists between two parties, there are key distribution protocols such as the BB84 [27] which ensure full security against eavesdropping and thus make public key cryptography unnecessary.

Yet another application of QIP is the simulation of quantum systems, as suggested by Richard Feynman in 1981 [28]. Running a quantum simulation in a classical computer is inherently ineffective due to the exponential scaling of the memory and time requirements with respect to the size of the problem. With a quantum computer, the computational power scales in principle just as quickly. Feynman's conjecture was proven by Seth Lloyd in 1996: A quantum computer can efficiently simulate all local quantum systems of corresponding size [29]. As the availability of efficient large-scale quantum simulations would be of immense value to many fields of science and technology such as physics, chemistry, and life sciences, this may yet be the most important contribution to humanity QIP has to offer.

This thesis investigates the problem of controlling quantum systems in order to perform quantum information processing tasks. The overview is organized as follows. Chapter 2 presents the fundamentals of quantum mechanics from the viewpoint of quantum information science. Chapter 3 is a brief introduction to the field of quantum computing. Chapter 4 is devoted to a discussion of quantum circuits. The concepts of quantum gates and gate decompositions, which are the main topics of publications **I**–**III**, are introduced. The subject of local gate invariants is also approached as we review the contents of publication **IV**. Chapter 5 examines the subject matter of publications **V** and **VI**, namely the problem of controlling quantum systems both in the presence and in the absence of noise. Finally, Chapter 6 contains a summary of the main results of this thesis.

2 Quantum mechanics

> Those who are not shocked when they first come across quantum mechanics cannot possibly have understood it.
>
> Niels Bohr

Quantum mechanics [30, 31] is in its most basic form a mathematical framework for constructing physical theories. Unlike most previous fundamental theories of physics, theories based on quantum mechanics are inherently non-deterministic in nature, and many find them quite counterintuitive. Despite this, they have been profoundly successful in explaining the small-scale structure of the universe. This chapter consist of a brief presentation of quantum mechanics, and discussion of some of its more curious and important features from the viewpoint of quantum computing.

2.1 Fundamentals

> I do not like it, and I am sorry I ever had anything to do with it.
>
> Erwin Schrödinger

With every physical system, one can associate a complex Hilbert space $\mathcal{H}$, called the state space. All the possible pure states of the system correspond to normalized vectors $|\psi\rangle$ or *kets* in this space (disregarding the unimportant global phase), and vice versa. The time evolution of the state is given by the Schrödinger equation:

$$i\hbar\frac{\partial}{\partial t}|\psi\rangle = H|\psi\rangle\,. \tag{2.1}$$

Here H is a Hermitian operator called the Hamiltonian which, in conjunction with the space $\mathcal{H}$, contains the entire physics of the system. One may also place additional demands on the form of the theory, such as causality or symmetries.

The solution of the Schrödinger equation may be written as

$$|\psi(t_1)\rangle = U(t_1, t_0)\,|\psi(t_0)\rangle\,, \tag{2.2}$$

where $U(t_1, t_0)$ is the unitary *propagator* of the system from t_0 to t_1, obtained as the time-ordered integral of $H(t)$:

$$U(t_1, t_0) = \mathcal{T} \exp\left(\frac{1}{i\hbar}\int_{t_0}^{t_1} H(t)\,\mathrm{d}t\right). \tag{2.3}$$

A linear combination of two state vectors, properly normalized, is again a valid state of the system. This is known as the superposition principle. In addition to pure states, there are mixed states which represent classical ensembles of pure states with known probabilities. A mixed state is described by a state operator $\rho \in \mathrm{End}(\mathcal{H})$ [1]:

$$\rho = \sum_i p_i\,|\psi_i\rangle\,\langle\psi_i|\,, \quad \text{with} \quad \sum_i p_i = 1, \tag{2.4}$$

where p_i is the classical probability of the state $|\psi_i\rangle$. By construction, a state operator is always Hermitian, semipositive and has trace one. A pure state $|\psi\rangle$ can thus also be represented by the state operator $\rho = |\psi\rangle\,\langle\psi|$. However, the decomposition of the state operator into a convex combination of pure states is not unique, and thus one can have different classical interpretations for a single mixed quantum state. The time evolution of the state operator is described by the quantum Liouville equation,

$$i\hbar\frac{\partial\rho}{\partial t} = [H, \rho]\,, \tag{2.5}$$

which can be immediately obtained from the Schrödinger equation.

Additionally, the rules of quantum mechanics postulate the existence of *measurements.* A measurement M is defined by a set of operators $\{M_i\}$, where i indexes all possible results of the measurement. Assuming the system is in the state $|\psi\rangle$, the probability of obtaining the result i is

$$p(i) = \langle\psi|\,M_i^\dagger M_i\,|\psi\rangle = \mathrm{Tr}\left(M_i\rho M_i^\dagger\right) \tag{2.6}$$

[1] $\mathrm{End}(V)$ is the set of endomorphisms of the set V. If V is a vector space, this corresponds to the set of all linear operators $A : V \to V$.

and the state of the system, after obtaining the result i, is $|\psi'\rangle \propto M_i |\psi\rangle$ or equivalently $\rho' \propto M_i \rho M_i^\dagger$. In order for the probabilities to sum up to unity, the the set of the measurement operators must be complete: $\sum_i M_i^\dagger M_i = I$.

There are specific types of measurements that hold particular interest. A *projective measurement* of a real scalar quantity is defined by a Hermitian operator A on $\mathcal{H}$ having the spectral decomposition $A = \sum_i a_i P_i$. The measurement operators are set to be equal to the orthogonal projection operators of the decomposition, $M_i = P_i$, whereas the corresponding spectral values a_i define the measurement results. Calculating the expectation value of the projective measurement, we obtain

$$\langle A \rangle = \sum_i a_i p(i) = \sum_i a_i \operatorname{Tr}\left(P_i \rho P_i^\dagger\right) = \sum_i a_i \operatorname{Tr}\left(P_i \rho\right) = \operatorname{Tr}\left(A\rho\right). \tag{2.7}$$

All physical *observables* of the system are represented by Hermitian operators such as A.

Individual quantum systems can be combined to form a larger one by taking a tensor product of their respective Hilbert spaces: $\mathcal{H} = \mathcal{H}_1 \otimes \ldots \otimes \mathcal{H}_N$. The resulting state vectors describe the combined state of all the constituent degrees of freedom. If we are interested in only a part of the full system, we may take a partial trace over the uninteresting parts of $\mathcal{H}$ to obtain a *reduced state operator*,

$$\rho_A = \operatorname{Tr}_B(\rho), \quad \text{i.e.} \quad (\rho_A)_{ij} = \sum_k \left(\left(\langle i| \otimes \langle k|\right) \rho \left(|j\rangle \otimes |k\rangle\right)\right), \tag{2.8}$$

which functions otherwise exactly like a full state operator, except that its time evolution is not necessarily unitary.

There is a convenient mathematical formalism called the *operator sum representation* for describing any possible evolution or *quantum operation* $\mathcal{E}$ in a quantum system or part thereof, including both discrete and infinitesimal time evolutions as well as measurements. In this formalism, a quantum operation is defined by a set of operators $\{E_i\}$ [2] such that $\sum_i E_i^\dagger E_i \leq I$. The result of the quantum operation is given by the mapping

$$\mathcal{E}(\rho) = \sum_i E_i \rho E_i^\dagger. \tag{2.9}$$

[2] The E_i are often called Kraus operators.

The probability of the operation taking place is given by $\mathrm{Tr}\left(\mathcal{E}\left(\rho\right)\right)$ and may be less than unity.

2.2 Entanglement and decoherence

Two photons, close-coupled at start,
Flew several parsecs apart.
Said one, in distress,
"What you're forced to express
Removes any choice on my part."

"Einstein, Podolsky and Rosen" by David Halliday

When I hear of Schrödinger's cat, I reach for my gun.

Stephen Hawking

Entanglement is the property of quantum systems which enables them to have non-local correlations that cannot be explained classically. It was originally used as an argument against the completeness of quantum mechanics as a physical theory in the famous Einstein-Podolsky-Rosen "paradox" [32]. Since then it has been shown that not only is quantum mechanics incompatible with local realism [33, 34, 35], but so is Nature itself [36, 37] [3].

The pure state of a multipart quantum system is *entangled* if it cannot be expressed as a tensor product of the states of the parts. A state that is not entangled is called *separable*. It is not simple to turn this qualitative idea into a rigorous quantitative definition. To make this task easier, the concept of Local Operations and Classical Communication (LOCC) is usually employed. Since entanglement is a nonlocal

[3] It should be noted that experiments on local realism are notoriously hard to make entirely foolproof, and hence the matter is not entirely settled yet.

phenomenon, one should not be able to generate it through local quantum operations, i.e. by operating only on the individual parts of the multipart system, even if the measured result of one operation is allowed to affect another. With these requirements in mind we may now define what we mean by entanglement.

A function $E : \mathrm{End}(\mathcal{H}) \to \mathbb{R}$ is a scalar measure of entanglement or an *entanglement monotone* if and only if it fulfills the following properties: Given a division of $\mathcal{H}$ to two or more "local" parts,

P1 $E(\rho) \geq 0$, and $E(\rho) = 0$ if and only if ρ is separable. Furthermore, the entanglement monotone is usually normalized by demanding that if ρ is a Bell state, $E(\rho) = 1$. (For a definition of the Bell states, see Eq. (3.4).)

P2 E is on the average not increased by LOCC. As a corollary, invertible local operations cannot decrease it either.

P3 E is convex under the loss of information about the state:
$\sum_i p_i E(\rho_i) \geq E\left(\sum_i p_i \rho_i\right)$.

For bipartite states, the simplest measure of entanglement is the von Neumann entropy of either of the parts:

$$E(\rho) = -\operatorname{Tr}(\rho_A \log_2 \rho_A) = -\operatorname{Tr}(\rho_B \log_2 \rho_B). \tag{2.10}$$

Entanglement is an essential ingredient for many quantum information processing tasks such as superdense coding [25], quantum teleportation [26], and most quantum algorithms. However, it is also responsible for one of the greatest challenges for quantum computing. A quantum system which is isolated from its environment is said to be *closed* and exhibits strictly unitary behavior. This is, of course, an idealization. In reality, all quantum systems interact with their environment and hence are *open* to a greater or lesser degree. The nonunitary behavior of an open system, caused by the entanglement of the system with the environment, is called *decoherence*. If the environment is much larger than the system, as is usually the case, decoherence becomes an irreversible process that destroys the quantum information contained in the system through the decay of the offdiagonal elements of the state operator.

3 Quantum computing

> Nothing shocks me. I'm a scientist.
>
> Henry Jones Jr.

In this chapter the theoretical basis of quantum computing is presented, followed by a short discussion of some of the suggested physical implementations.

3.1 Basics

> Young man, in mathematics you don't understand things. You just get used to them.
>
> John von Neumann

Quantum computing is an umbrella term for a number of computational models utilizing the properties of quantum mechanics [38]. Even though there are many different models and implementations of quantum computing, there are certain concepts which appear frequently throughout the field.

A *qubit* [39] is a quantum two-state system. It is the simplest nontrivial quantum system and, much like the classical bit, turns out to be sufficient for any kind of quantum information processing task. Spin-$\frac{1}{2}$ particles can be regarded as "natural" qubits, but a qubit need not be ideal. If, for example, the two lowest energy states of a discrete energy spectrum are sufficiently well separated from the rest, the system can be used to emulate a qubit. Qubits can be either *stationary* or *flying*. A stationary qubit must stay in place, whereas a flying qubit may (or, with some implementations such as photons, indeed must) move relative to the laboratory in some controlled fashion and can thus be used to transmit quantum information between spatially separated locations.

A *logical qubit* is a two-dimensional subspace within the larger Hilbert space of a physical system. The choice of the subspace that stores the quantum information is called an *encoding* or a quantum code. It is possible to use redundancy to enhance the resistance of the quantum information against decoherence and noise, essentially by storing a single logical qubit in multiple physical qubits using an encoding which enables one to actively detect and correct noise-induced errors through the periodic measurement of ancilla qubits followed by corrective operations. These encodings are collectively called *quantum error correcting codes* [40, 41, 42, 43]. Alternatively one can choose a passive error avoidance strategy by encoding the logical qubits into a *decoherence-free subspace* (DFS) [44, 45, 46], which is protected against decoherence for symmetry reasons.

Within the state space of the logical qubit, we define a *computational basis*, labeling the orthogonal, normalized basis vectors as $|0\rangle$ and $|1\rangle$. Disregarding the nonphysical global phase and normalization, all pure states of a qubit can be represented in the form

$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\phi}\sin\left(\frac{\theta}{2}\right)|1\rangle\,, \tag{3.1}$$

where θ and ϕ are real parameters. When they are interpreted as polar coordinates, we obtain the *Bloch sphere* representation of the qubit where the pure states of the qubit are mapped on the surface of the unit sphere in three dimensions, as shown in Fig. 3.1. In the state operator representation, this is equivalent to

$$\rho = |\psi\rangle\langle\psi| = \frac{1}{2}\left(I + \sin\theta\cos\phi\,\sigma_x + \sin\theta\sin\phi\,\sigma_y + \cos\theta\,\sigma_z\right) = \frac{1}{2}\left(I + \vec{a}\cdot\vec{\sigma}\right), \tag{3.2}$$

where $|\vec{a}| = 1$ and $\{\sigma_i\}$ are the Pauli matrices (see Eq. (A.1)). Since all the states of the system are obtained as the convex hull of the set of pure states in the state operator representation, we find that the nonpure one-qubit states must lie inside the Bloch sphere.

In a system consisting of n qubits, commonly called an n-qubit *quantum register*, the computational basis of the register is obtained as a tensor product of the single-qubit basis vectors, i.e.

$$|x_{n-1}x_{n-2}\dots x_0\rangle := |x_{n-1}\rangle \otimes |x_{n-2}\rangle \otimes \dots \otimes |x_0\rangle\,. \tag{3.3}$$

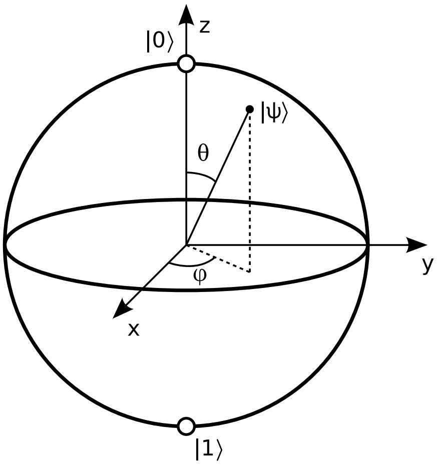

Figure 3.1: Bloch sphere representation of a qubit.

For the purposes of the computation, the labels are usually interpreted as binary numbers.

An *EPR pair* is essentially any fully entangled bipartite quantum state. However, in the context of quantum computing it usually refers to one of the four two-qubit *Bell states*:

$$\begin{aligned} \left|\Phi^+\right\rangle &= \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) & \left|\Phi^-\right\rangle &= \frac{i}{\sqrt{2}}(|00\rangle - |11\rangle) \\ \left|\Psi^+\right\rangle &= \frac{i}{\sqrt{2}}(|01\rangle + |10\rangle) & \left|\Psi^-\right\rangle &= \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle), \end{aligned} \tag{3.4}$$

most typically the singlet state $|\Psi^-\rangle$. The Bell states form a basis for two-qubit states, not surprisingly called the *Bell basis*. They are also used in *Bell measurements*, which are simply projective measurements of two qubits in this basis. Since the Bell states are entangled, a Bell measurement is necessarily a nonlocal entangling operation.

3.2 Quantum algorithms

> Deutsch's Law: Every problem that is interesting is also soluble.
>
> David Deutsch

For quantum computing to be interesting, there must be something that it can do better than classical computing. The currently known quantum algorithms can be divided into three families: hidden subgroup problem, amplitude amplification and quantum simulation.

The hidden subgroup problem (HSP) family contains the most successful quantum algorithms known to date, including the Deutsch-Jozsa [14], Simon [47], and the Shor algorithms for solving the discrete logarithm and integer factorization problems [15]. The HSP can be stated as follows. Given a group G, a finite set X and a function $f : G \to X$ that separates the cosets of an unknown subgroup $H < G$, find the generating set of H. Many important instances of the HSP have no known efficient classical solution, whereas an efficient quantum solution has been found. More generally, a quantum algorithm for solving the HSP has been derived for all finitely generated Abelian groups [48]. The "holy grail" of quantum algorithm design, a general algorithm for the non-Abelian HSP, still evades us and indeed may not exist at all.

The amplitude amplification [49] family contains quantum algorithms which do not provide a superpolynomial speedup, but still outperform their classical counterparts. This family includes i.a. the Grover search algorithm [19] and the quantum counting algorithm [50]. The primary application area of these algorithms seems to be in solving hard problems from the complexity class NP (Non-deterministic, Polynomial time) through an exhaustive search of all the possible solutions.

Finally, a quantum computer can efficiently simulate all local quantum systems of corresponding size, a task which is impossible for a classical computer. This was first suggested by Feynman [28] and later shown to be possible by Lloyd [29]. If feasible, this may yet prove to be the most important application of quantum computing, as nuclear and materials physics, nanotechnology, chemistry, and molecular biology alike would benefit enormously from efficient and accurate molecular and solid state simulations.

3.3 Models of quantum computing

> The first principle is that you must not fool yourself—and you are the easiest person to fool.
>
> Richard Feynman

There are several different, polynomially equivalent models for quantum computing. Some of them are merely abstract mathematical devices. For example the quantum Turing machine [9, 10], much like its classical counterpart, can be useful in constructing proofs, but nobody expects to actually build one. Other models, however, may yet avail themselves to physical implementation. There is an important property that all useful models of quantum computing must share. Namely, one must be able to present quantum algorithms within the model in a way that is polynomial both in time and space with respect to the size of the problem. This is what separates an actual quantum computer from e.g. a classical computer simulating a quantum computer.

The first and thus far predominant model with a possible physical implementation is the *quantum circuit* model [11]. Here, the algorithm is encoded into a circuit of quantum gates acting on the register, interspersed with measurements of the resulting state. The gates correspond to arbitrary unitary operations in the computational basis. Consequently this model automatically incorporates classical reversible logic (which consists of permutations within the computational basis). It is straightforward to implement, but requires rather precise control of the system Hamiltonian. The quantum circuit model will be described in detail in the next chapter. The publications **I**–**IV** deal almost exclusively with this model.

If the Hamiltonian of a quantum system is changed sufficiently slowly, the different energy eigenstates retain their respective populations, assuming that there are no level crossings. This result is called the adiabatic theorem [51], and there are models of quantum computing that exploit it. *Holonomic quantum computing* [52] encodes the algorithm into a loop in the parameter space of the system Hamiltonian, along which the system is adiabatically moved. The system is assumed to have a degenerate ground state which, during the loop, experiences an unitary quantum holonomy which corresponds to a quantum gate. The resulting holonomy, being a geometrical effect, does not depend on the speed with which the parameter loop is traversed, which makes this scheme somewhat easier to control than quantum circuits. Another model utilizing adiabatic evolution is the succinctly named *adiabatic quantum computing* [53]. In this model the system is initialized to the ground state of its initial Hamiltonian. Then the Hamiltonian is adiabatically changed to the final Hamiltonian, whose ground state represents the solution to the problem at hand. If this is possible without violating the conditions of the adiabatic theorem, we may obtain our result by measuring the final state.

An alternative to the abovementioned models, where the computation requires precise control of the system Hamiltonian, is *measurement-based quantum computing.* Here the idea is to have a source of entangled quantum states with known properties, and the algorithm is encoded into measurements performed on these states. There are two main variants of this model. The first one, *teleportation quantum computing* (TQC) [54], replaces quantum gates with pregenerated EPR pairs and Greenberger-Horne-Zeilinger states [35] subjected to the required one-qubit operations. The register qubits are then teleported “through” these gates using only Bell measurements and Pauli gates. The other variant, *one-way quantum computing* (1WQC) [55, 56], requires a massive fully entangled state of qubits arranged in a two-dimensional grid, called a cluster state. This state can be produced using fixed nearest-neighbor interactions within the grid. After the preparation of the cluster state, the interactions are turned off. The computation only requires single-qubit measurements which disentangle the state one qubit at a time, while the entanglement propagates the result of the computation forward in the grid.

The preceding models, while involved, are rather concrete. More abstract proposals exist as well, such as *topological quantum computing* [57, 58]. Here, the computation happens by braiding the (2+1)-dimensional spacetime trajectories of anyon quasiparticles that have a non-Abelian braid group. This approach has the advantage that since the quantum information is encoded into nonlocalized topological degrees of freedom, it is quite resistant against noise. However, so far there are no known experimental realizations of non-Abelian anyons.

3.4 Proposed architectures

> Quantum phenomena do not occur in a Hilbert space, they occur in a laboratory.
>
> Asher Peres

When designing a physical implementation for a quantum computer, many things need to be taken into account. The DiVincenzo criteria [59] are necessary conditions which any candidate technology should fulfill. Namely, any viable implementation must provide [60]

1. a scalable system of well-defined qubits (usually called the register)
2. a way to initialize the register to a simple, useful quantum state
3. decoherence times for the register that are much longer than the required operation time
4. a controllable universal set of quantum operations
5. a way to measure the state of the register in some useful basis.

Of course, there are also further factors to be considered when designing an implementation. Instead of a single register, one may use a huge ensemble of identical registers, which tends to make the measurement easier but introduces complications to the initialization procedure, forcing us to use a *pseudopure state* [61] instead of a pure one.

In many implementations, single qubits can at least in principle be fully controlled. To obtain an universal set of multiqubit operations, one also needs a way to entangle qubits with one another. Usually this is accomplished using interqubit interactions, even though they tend to be much harder to control than single qubits. An implementation may have fully tunable nearest-neighbor interactions, fixed nearest-neighbor interactions that can be dynamically suppressed using single-qubit controls, or one or more special qubits which mediate the interaction between the actual data qubits, a design known as the quantum bus. Some designs avoid the interactions altogether and generate the required entanglement using ancilla states and measurements.

If we want our setup to allow for quantum networkability, i.e. the input/output and transmission of quantum information, we must augment the DiVincenzo criteria by two additional requirements:

6. a way to interconvert stationary and flying qubits
7. a way to faithfully transmit flying qubits between specified locations.

The flying qubits are usually photons, but some designs also use atoms or ions manipulated using electromagnetic (EM) fields. A quantum state can also be transferred over a distance using teleportation, but this requires the sender and receiver to share entanglement, for example in the form of previously transmitted members of EPR pairs. These members can be considered flying qubits in their own right.

Different designs may have vastly different sources of error. They may stem from unwanted interactions between the environment and the register, noisy controls or the measurement procedure. An important practical point to keep in mind are the fabrication issues, such as how accurately we can specify the physical parameters of our qubits. Especially in solid state systems it is not given that our qubits will have identical characteristics, even if we design them that way. Hence their properties need to be measured before accurate control sequences can be derived.

The rest of this chapter briefly presents some of the architectures proposed thus far. This is by no means meant to be a complete, rigorous literature study but rather a simple listing of the most popular and promising implementations. It should also be noted that the architectures are not isolated. There are significant overlaps in the ideas and techniques involved, and a future “winning” implementation may well be a hybrid model incorporating several of them.

Josephson circuits

These solid-state superconducting circuits use Josephson junctions (JJs) and quantum interference loops to create macroscopic quantum states that can be manipulated using external currents, voltages, and magnetic fields. They are readily manufactured and controlled, but suffer from high decoherence rates. Depending on whether the information is stored in charge [62, 63] or phase eigenstates [64] or some superposition of them [65], the design is called a Josephson charge, flux or hybrid qubit, respectively. More involved designs employ directional superconductors [66] or complex circuit topologies and geometries [67] to make the control easier and to enhance resistance against noise and imperfections in the fabrication.

Trapped ions

In this design, often called the Cirac-Zoller model [68], ionized atoms are placed in a linear Paul trap [69] and cooled to their motional ground state. The qubits are formed either by the hyperfine levels in the ground state (hyperfine qubit) or the ground and excited states with a weak transition (optical qubit) within the electronic structure of the ions. Individual ions can be manipulated by lasers, and the qubits can be coupled through their collective motional mode which serves as a bus qubit. A more recent variant of this design proposes to trap the ions in a semiconductor chip microtrap [70] which gives much better scaling properties.

Quantum dots

Quantum dots are semiconductor nanostructures which generate a spatially localized potential well capable of trapping and confining individual electrons. In the Loss-DiVincenzo -model [71], the spins of the trapped electrons serve as the qubits. The exchange interaction between the spins in neighboring dots can be controlled by varying the tunneling barrier between them using surface electrodes, whereas the single-qubit operations are performed using local magnetic fields. A more recent proposal due to Levy [72] gains full control of the register using the exchange interaction alone by combining two electron spins in neighboring dots into a single logical qubit. This scheme also admits a feasible initialization and measurement procedure [73].

Optics

Using photons as vessels for the quantum information seems attractive as they are easily controlled and measured, and interact weakly with most matter. This is offset by the fact that they cannot remain stationary with respect to the laboratory. Single photons can be obtained from attenuated lasers or parametric down-conversion, manipulated using mirrors, phase shifters, and beamsplitters, and measured with photodetectors. Interactions between photons are produced using a nonlinear optical element such as a Kerr medium [74] or a high-quality optical cavity containing one or more atoms mediating the interaction [75]. Another scheme, called the linear optics quantum computing (LOQC) [76], uses ancilla states and measurements instead of photon-photon interactions.

NMR

Nuclear magnetic resonance (NMR) [77] is a well-known and widely utilized phenomenon, in which the nuclear spins of a material sample are aligned using a strong external magnetic field and then coherently manipulated using radiofrequency EM fields. In 1997 it was found that room-temperature liquid-state NMR could also be used for quantum computing [61, 78] through the use of pseudopure states. Due to the relative sophistication and availability of NMR technology, this approach has provided the most powerful quantum computers to date [79].

The liquid-state NMR setup consists of a liquid sample of molecules placed in a strong homogeneous magnetic field. Individual nuclear spins within the molecule function as qubits. Chemical shift is used to make two otherwise identical spins with different surroundings individually addressable. Dipole-dipole interactions between the spins provide the entangling interaction. However, due to the pseudopure state used in the computation, this scheme is inherently nonscalable as the output signal amplitude is halved with each qubit added to the system.

There are also proposals for solid-state NMR quantum computing. The Kane quantum computer [80] consists of individual ^{31}P donor atoms with nuclear spin embedded in a ^{28}Si substrate. The qubits, formed by the phosphorus nuclear spins, are well isolated and can be addressed using time-dependent radiofrequency EM fields. For two-qubit interactions, the states of two neighboring nuclear spins are transferred to the spins of the respective donor electrons, which are then brought close to each other using lithographic surface electrodes so that a dipolar spin-spin interaction can take place. Afterwards, the states are again transferred to the nuclear spins. The donor electron spins are also used for readout.

Neutral atoms in optical lattices

An optical lattice is a periodic n-dimensional potential lattice created by $n+1$ (or more) lasers. It can be used to trap cold neutral atoms, whose electronic states can be used as qubits. The atoms can be manipulated using lasers, and the qubit-qubit interactions are generated using either electric dipole-dipole interactions [81] or cold atomic collisions [82]. This scheme is particularly well-suited for measurement-based quantum computing.

4 Quantum circuits

The quantum circuit model [11] is the currently predominant theoretical approach to quantum computing. In this model, the computation takes place in a *quantum register* which consists of n local units. Typically these local units are qubits, but in principle they could be any kind of discrete quantum systems. They only serve as a definition for locality. The model imposes no particular physical realization on the register.

In the beginning of a computation, the register is initialized into a known pure state, most often $|00\ldots0\rangle$. The qubits are operated upon by unitary operations which, in this context, are called quantum gates. It is understood that the gates are to be implemented as propagators of the register. Finally, the state of the register is measured, which concludes the quantum computation. Since the state space of an n-qubit register is 2^n-dimensional, the gates can be represented as $2^n \times 2^n$ unitary matrices. Disregarding the unphysical global phase, together they constitute $SU(2^n)$, the special unitary group in 2^n dimensions.

4.1 Gates and circuit diagrams

The quantum circuit model allows for a rather compact and illustrative way of describing quantum algorithms and their parts, namely quantum circuit diagrams such as the one shown in Fig. 4.1. In the diagrams, time advances from left to right and the individual qubits are represented by horizontal lines. When interpreted as a binary number, the topmost qubit is usually the most significant one. In other aspects their relative ordering is unimportant, but sometimes may be related to the topology of the register. Rectangular boxes (perhaps connected by vertical lines) touching one or more horizontal lines represent quantum gates acting on these qubits. For certain common gate types specific symbols are used.

Any n-qubit quantum algorithm can be represented as one or more n-qubit gates

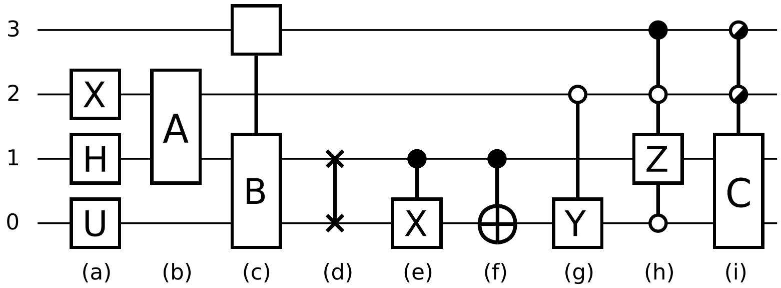

Figure 4.1: Example of a quantum circuit diagram describing a four-qubit system. The qubits are numbered 0–3. Explanation of the gate symbols: (a) one-qubit gates: identity, σ_x (NOT), Hadamard, unspecified one-qubit gate U, (b) two-qubit gate acting on qubits 1 and 2, (c) three-qubit gate acting on qubits 0, 1 and 3, (d) SWAP gate, (e) controlled NOT (CNOT), (f) another symbol for CNOT, (g) controlled σ_y with a reversed control node, (h) multiply controlled σ_z, (i) uniformly controlled two-qubit gate.

interspersed with measurements. This is not an useful representation in itself, since computing the corresponding matrices is equivalent to running the actual algorithm. Instead, the algorithm is decomposed into a series (or circuit) of gates acting only on a small number of qubits at a time. There is an important result which states that two-qubit gates are universal [83]; any n-qubit gate can be decomposed into a sequence of two-qubit gates. In fact we do not even need all two-qubit gates. A properly chosen entangling two-qubit gate together with all one-qubit gates forms an universal gate library of *elementary gates* capable of exactly synthesizing any n-qubit gate. Usually the two-qubit gate in the library is chosen to be the controlled-NOT (CNOT) due to its straightforward logical interpretation [84]. All efficient quantum algorithms have a decomposition consisting of a number of gates from this library that is polynomial in n.

Elementary one-qubit rotations about the x, y, and z axes are defined as the one-parameter subgroups of $SU(2)$ generated by the Pauli matrices σ_x, σ_y, and σ_z:

$$R_x(\theta) := e^{i\sigma_x\theta/2} = \begin{pmatrix} \cos\frac{\theta}{2} & i\sin\frac{\theta}{2} \\ i\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix} \tag{4.1}$$

$$R_y(\theta) := e^{i\sigma_y\theta/2} = \begin{pmatrix} \cos\frac{\theta}{2} & \sin\frac{\theta}{2} \\ -\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix} \tag{4.2}$$

$$R_z(\theta) := e^{i\sigma_z\theta/2} = \begin{pmatrix} e^{i\theta/2} & 0 \\ 0 & e^{-i\theta/2} \end{pmatrix}. \tag{4.3}$$

Together the Pauli matrices form a basis for the underlying Lie algebra $\mathfrak{su}(2)$. More generally we may define a rotation about the vector $\vec{a}$,

$$R_{\vec{a}}(\theta) := e^{i\,\vec{a}\cdot\vec{\sigma}\,\theta/2} = I\cos(\theta/2) + i\,\vec{a}\cdot\vec{\sigma}\,\sin(\theta/2), \tag{4.4}$$

where $|\vec{a}| = 1$. Other important one-qubit gates include the logical NOT gate σ_x, and the Hadamard gate

$$H = \frac{\sigma_x + \sigma_z}{\sqrt{2}} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{4.5}$$

which is used i.a. to prepare the register to an equal superposition of all the states in the computational basis. Since all one-qubit gates correspond to $SU(2)$ matrices, we may also present any such gate U using three consecutive Euler rotations about any two perpendicular directions, usually the z and y axes:

$$U = R_z(\alpha)R_y(\beta)R_z(\gamma) \quad \text{for some} \quad \alpha, \beta, \gamma \in \mathbb{R}. \tag{4.6}$$

The SWAP gate is the conceptually most straightforward one of the two-qubit gates. It is defined as

$$\text{SWAP} = |00\rangle\langle 00| + |10\rangle\langle 01| + |01\rangle\langle 10| + |11\rangle\langle 11|, \tag{4.7}$$

and simply swaps the states of its target qubits:

$$\text{SWAP}\,|a\rangle \otimes |b\rangle = |b\rangle \otimes |a\rangle. \tag{4.8}$$

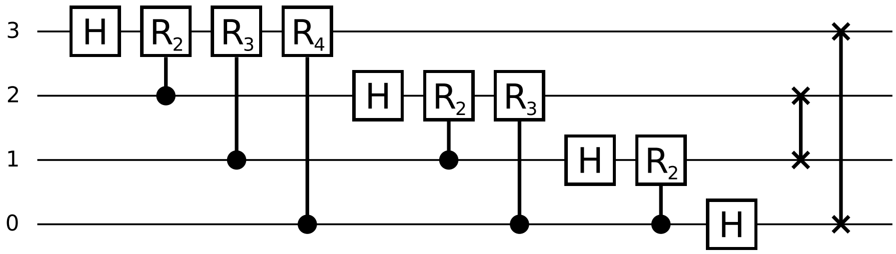

Figure 4.2: Quantum circuit implementing the four-qubit quantum Fourier transform. Here R_k is shorthand for $R_z(-2\pi/2^k)$.

The CNOT gate is an example of a class of gates known as controlled gates. A k-fold controlled gate $C^k(U)$ is defined by the set of k control nodes corresponding to the k-bit binary string c, and a target gate U. Arranging the qubits such that the control qubits precede the target qubits, the gate is given by

$$C^k(U) = \sum_{x \neq c} |x\rangle \langle x| \otimes I + |c\rangle \langle c| \otimes U, \tag{4.9}$$

i.e. the gate U is applied on the target qubits if and only if the control qubits are in the state $|c\rangle$. It is not hard to see that CNOT performs a XOR operation between its control and target qubits, and stores the result in the target:

$$C(\sigma_x) |a\rangle \otimes |b\rangle = |a\rangle \otimes |a \oplus b\rangle . \tag{4.10}$$

The uniformly controlled gates introduced in publications **I** and **III** extend this idea by applying a different gate U_i for each possible combination c_i of the control bits, and thus are equivalent to a sequence of 2^k k-fold controlled gates. However, they require much fewer elementary gates to implement, which makes them a very useful intermediate circuit structure e.g. in the construction of recursive gate decompositions. They also have other applications; publication **II** uses the uniformly controlled gates to construct a circuit which transforms a given pure state into another. In publication **III** we show how they can be implemented efficiently using only nearest-neighbor gates, which makes their physical implementation simpler.

Figure 4.2 is an example of a quantum circuit that does something useful. This

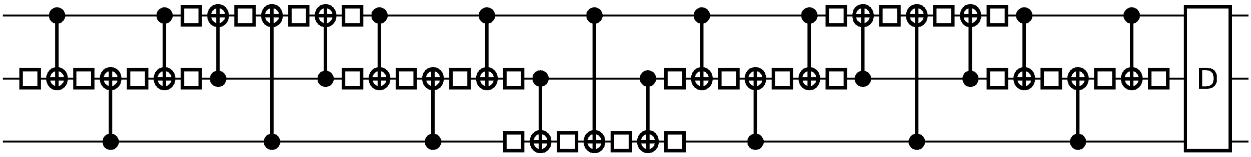

Figure 4.3: Quantum circuit of the three-qubit cosine-sine decomposition from publication **III**. The white boxes denote $SU(2)$ gates, and the gate D is a diagonal residue.

particular circuit design implements the n-qubit quantum Fourier transform (QFT) using $O(n^2)$ elementary gates. The QFT is the quantum computing version of the discrete Fourier transform, and an essential part of the quantum algorithms of the HSP family. The matrix elements of the corresponding gate in the computational basis are given by

$$U_{jk}^{\text{QFT}} = \frac{1}{\sqrt{2^n}} e^{2\pi i jk/2^n}. \tag{4.11}$$

The physical implementation of the elementary gates requires that the Hamiltonian of the register can be controlled well enough, a problem that is addressed in Ch. 5.

4.2 Gate decompositions

The problem of decomposing an arbitrary n-qubit gate into a sequence of elementary gates was first addressed in Ref. [84], where the decomposition was based on the QR matrix decomposition [85] expressed as a sequence of Givens rotations, each implemented using a number of $n-1$-fold controlled one-qubit gates. It results in a circuit requiring $\Theta(n^3 4^n)$ CNOTs. This decomposition was later improved to give an asymptotically optimal circuit requiring just $O(4^n)$ CNOTs [86]. However, the multiplicative constant hidden by the O notation is on the order of 8, which is quite high.

In publication **I** we introduce a new gate decomposition based on a recursive cosine-sine matrix decomposition (CSD) [87]. The CSD of the $SU(2^n)$ matrix U is given by

$$U = \begin{pmatrix} u_{11} & 0 \\ 0 & u_{12} \end{pmatrix} \begin{pmatrix} c & s \\ -s & c \end{pmatrix} \begin{pmatrix} u_{21} & 0 \\ 0 & u_{22} \end{pmatrix}, \tag{4.12}$$

where $c^2 + s^2 = I$ and the u_{ij} are $SU(2^{n-1})$ matrices. The decomposition can be continued recursively until only $n-1$-fold uniformly controlled gates are left. Using this construction we manage to push the CNOT count down to approximately 4^n gates. The new decomposition is improved in publication **III** where we further halve the CNOT count by using a more compact implementation for the uniformly controlled gates. Figure 4.3 presents the final elementary gate structure of the improved CSD for an arbitrary three-qubit gate.

4.3 Local gate invariants

In many of the physical realizations of quantum computing, one-qubit gates are much easier and faster to implement than two-qubit ones, due to the fact that individual qubits are more easily controllable than the interqubit interactions. In some implementations such as the liquid-state NMR, the difference is so great that one-qubit operations can be regarded as essentially free. In these cases it is often useful to consider the local equivalence classes of multiqubit gates instead of the gates themselves. Two gates, U_1 and U_2, are locally equivalent if and only if they can be transformed to each other using only local unitaries:

$$U_1 \sim U_2 \quad \Leftrightarrow \quad U_1 = A\, U_2\, B \quad \text{for some} \quad A, B \in SU(2)^{\otimes n}. \tag{4.13}$$

A local gate invariant is a quantity associated with a quantum gate that is not affected by local unitaries. It can be shown that the local equivalence classes of two-qubit gates can be parametrized with three real parameters. Figure 4.4 shows the set of all equivalence classes of two-qubit gates using the Makhlin parametrization [88].

Publication **IV** introduces a local gate invariant η which describes the local commutational properties of multiqubit gates. Roughly speaking, if a continuous k-parameter family of local gates on one side of a multiqubit gate U can be replaced

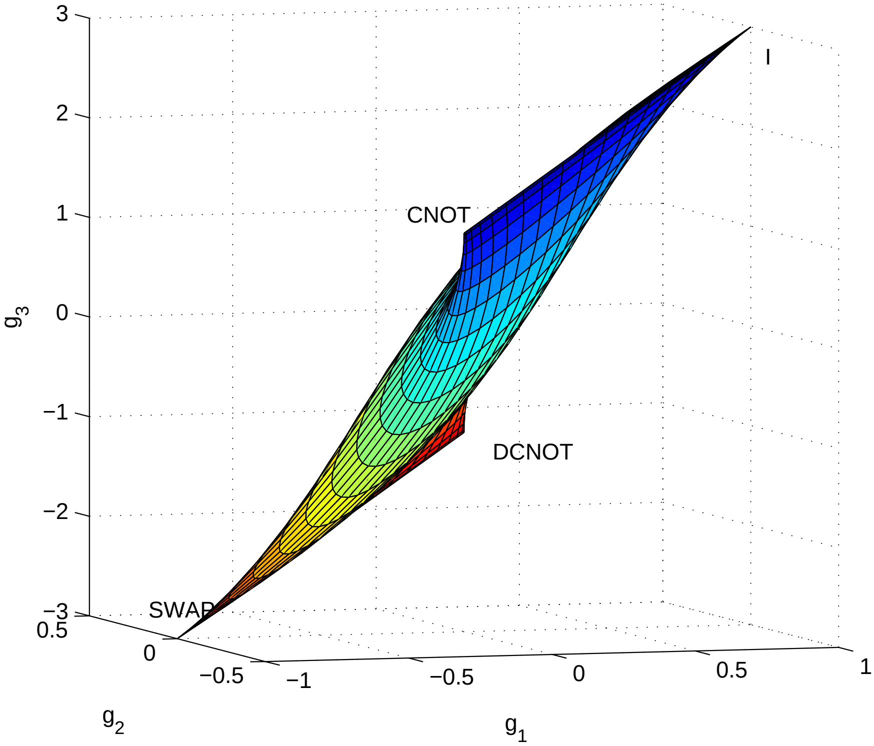

Figure 4.4: Set of all local equivalence classes of two-qubit gates in the Makhlin parametrization. For the definition of the parameters g_1, g_2 and g_3, see publication **IV**.

by another such family on the other side, the gate U is said to leak k local degrees of freedom (LDOFs). The invariant η describes how many LDOFs a gate can bind, i.e. *not* leak. Since one-qubit gates can be parametrized using three reals, an n-qubit gate can bind at most $3n$ LDOFs.

A CNOT, for example, can bind four LDOFs. This is easily seen by placing two one-qubit gates next to a CNOT and decomposing them into Euler rotations about the x and z axes. As the control node of the CNOT commutes with z rotations and the target node with x rotations, two of the six LDOFs involved can be commuted through the CNOT.

In publication **IV** we explicitly calculate the values of the binding invariant for all two-qubit gates, and observe that almost all two-qubit gates can in principle bind the full six LDOFs. If we want to decompose an n-qubit gate into elementary one- and two-qubit gates using a library consisting of all the one-qubit gates and a single two-qubit gate U, an exact general gate decomposition requires at least $\lceil (4^n - 3n - 1)/\eta(U) \rceil$ applications of the gate U.

5 Control sequences

As explained in Sec. 3.4, the controllability of the system serving as our quantum computer is likely to be far from perfect. Typically the system Hamiltonian $H(\vec{c}, t)$ depends on a number of external, bounded control fields $\{c_i\}$, which may correspond to voltages, currents or applied EM field amplitudes and phases in the experimental apparatus. These control fields can be adjusted in real time which gives us a limited ability to steer the time evolution of the system.

Let us denote the number of qubits in our system by n and the dimension of the corresponding Hilbert space by $N = 2^n$. The controllable Hamiltonian is capable of generating a group of propagators G that is a subgroup of $SU(N)$. If $G = SU(N)$, the system is said to be fully controllable. In a typical architecture the individual qubits are fully controllable, but the interqubit interactions are either fixed or can only be turned on and off. This can make the problem of steering the system towards the required gate in an optimal fashion quite hard. If either the system is subject to decoherence or the controls are noisy the situation becomes even more complicated as the evolution may no longer be unitary.

5.1 Optimization

The problem we wish to address here can be stated as follows: Given a Hamiltonian $H(\vec{c}, t)$, we want to derive a control sequence $\vec{c}(t)$ which evolves the system as close to a given target propagator U as possible, as rapidly as possible. In the simplest cases this can be done analytically, but in practice it is not feasible. Instead, we must obtain the control sequence through some other means such as numerical optimization.

For this purpose we need an error measure for our controlled evolution $\mathcal{E}$. There are two relevant measures to consider. The first one, squared distance between two unitary propagators operating in an N-dimensional Hilbert space, can be used when the evolution is unitary, $\mathcal{E}(\rho) = V\rho V^\dagger$ with $V^\dagger = V^{-1}$. It is given by

$$\begin{aligned} E_1(U,V) &:= \inf_{\phi\in\mathbb{R}} \left\| U - e^{i\phi}V \right\|^2 = 2N - 2\sup_{\phi\in\mathbb{R}} \operatorname{Re}\left(e^{i\phi}\operatorname{Tr}\left(U^\dagger V\right)\right) \\ &= 2N - 2\left|\operatorname{Tr}\left(U^\dagger V\right)\right| = 2N - 2f(U,V), \end{aligned} \tag{5.1}$$

since the global phase ϕ has no physical meaning. The expression $f(U,V) := \left|\operatorname{Tr}\left(U^\dagger V\right)\right|$ is called the gate fidelity.

When the evolution $\mathcal{E}$ is not unitary, we replace the measure Eq. (5.1) with the squared distance of an evolved state from the result of the ideal propagation by U, averaged over all pure states:

$$E_2(\mathcal{E},U) := \int_{\rho \text{ pure}} d^2(U\rho U^\dagger, \mathcal{E}(\rho))\,\mathrm{d}\mu(\rho). \tag{5.2}$$

The squared distance between the states χ and ρ is given by

$$d^2(\chi,\rho) := \|\chi-\rho\|^2 = \operatorname{Tr}\left(\chi^2\right) + \operatorname{Tr}\left(\rho^2\right) - 2\operatorname{Tr}\left(\chi\rho\right), \tag{5.3}$$

where the expression $F(\chi,\rho) := \operatorname{Tr}(\chi\rho)$ is called the state fidelity. The integration measure $\mu(\rho)$ is defined below. For pure states, we may replace the squared distance between states by $1 - F(\chi,\rho)$ and thus obtain a third error measure:

$$E_3(\mathcal{E},U) := 1 - \mathcal{F}(\mathcal{E},U) := 1 - \int_{\rho \text{ pure}} F(U\rho U^\dagger, \mathcal{E}(\rho))\,\mathrm{d}\mu(\rho). \tag{5.4}$$

Let us expand N-dimensional state operators using the parametrization

$$\rho = \frac{I_N}{N} + r_k X^k, \tag{5.5}$$

where the operators $\{X^k\}$ along with $\frac{I_N}{\sqrt{N}}$ form an orthonormal Hermitian basis with respect to the Hilbert-Schmidt inner product $\langle A,B\rangle := \operatorname{Tr}\left(A^\dagger B\right)$, r_k are real coefficients, and the Einstein summation convention is used. In an n-qubit system we use the tensor basis, where the basis vectors are tensor products of the single-qubit basis vectors $\frac{1}{\sqrt{2}}\{I, \sigma_x, \sigma_y, \sigma_z\}$.

Using the fact that the time evolution mapping $\mathcal{E}$ is linear and unital we obtain

$$
\begin{aligned}
d^2(U\rho U^\dagger, \mathcal{E}(\rho)) &= |\vec{r}|^2 + r_k r_l \underbrace{\operatorname{Tr}\left(\mathcal{E}\left(X^k\right)\mathcal{E}\left(X^l\right)\right)}_{B^{kl}:=} - 2 r_k r_l \underbrace{\operatorname{Tr}\left(U X^k U^\dagger \mathcal{E}\left(X^l\right)\right)}_{A^{kl}:=} \\
&= |\vec{r}|^2 + r_k (B - 2A)^{kl} r_l, \qquad (5.6)
\end{aligned}
$$

and correspondingly for the fidelity

$$
F(U\rho U^\dagger, \mathcal{E}(\rho)) = \frac{1}{N} + r_k A^{kl} r_l. \qquad (5.7)
$$

The set of all pure states is obtained as the orbit of any single pure state under the action of $SU(N)$. The proper integration measure $\mu(\rho)$ is thus induced by the normalized Haar measure of $SU(N)$ [89]. In a single-qubit system, the set of pure states is simply the surface of the Bloch sphere with the usual Euclidean measure. In this case we obtain the average error

$$
E_2(\mathcal{E}, U) = \frac{1}{2} + \frac{1}{12} \sum_{k=1}^{3} \operatorname{Tr}\left(\left(\mathcal{E}(\sigma_k) - 2U\sigma_k U^\dagger\right)\mathcal{E}(\sigma_k)\right) \qquad (5.8)
$$

and the average fidelity

$$
\mathcal{F}(\mathcal{E}, U) = \frac{1}{2} + \frac{1}{6} \sum_{k=1}^{3} \operatorname{Tr}\left(U\sigma_k U^\dagger \mathcal{E}(\sigma_k)\right). \qquad (5.9)
$$

Another relevant accuracy measure would be the minimum fidelity in the set of pure states. This can be obtained from the smallest eigenvalue of A.

Having obtained an error measure to minimize, one could now proceed to apply an optimization algorithm to the problem. However, given a control sequence $\vec{c}(t)$, evaluating the corresponding $\mathcal{E}(\rho)$ can be very expensive. With K control fields, the trajectory of each parametrized with M optimization parameters, a naïve gradient-based optimization algorithm would require $(KM + 1)$ such evaluations to estimate the value of the gradient.

In publication **V** we employ an optimization algorithm called GRadient Ascent Pulse Engineering (GRAPE) [90]. It requires the control sequences to be piecewise constant with the piece durations fixed and sufficiently short, but can estimate the gradient with only two evaluations of the optimized function.

GRAPE is well suited for the optimization of complex sequences with hundreds of pieces (and curiously enough, more often than not the optimized piecewise constant solution is seen to approximate a smooth continuous sequence) but in some cases, such as with a bad initial guess, it converges quite slowly. Also, like all gradient-based optimization methods, it may get stuck in a local extremum instead of the global one. These drawbacks can be ameliorated by using educated guesses in choosing the initial sequence.

The convergence of the optimization can be significantly sped up if the piece durations are treated as optimization parameters as well. However, if a duration grows too large, the approximation used in deriving GRAPE is no longer valid and the part of the gradient for that particular piece has to be calculated using the usual difference method. This is costly, however, and limits the number of parameters used in the optimization. Hence this second method is better suited for shorter and simpler control sequences, such as the ones in publication **VI**.

5.2 Simulation of noise

An open quantum system can be simulated in a number of different ways. The most accurate and, unfortunately, computationally most intensive method is to explicitly simulate the quantum degrees of freedom constituting the environment. This of course places a limit on the size of the environment that can be simulated because of the exponential scaling. A more frugal but less universal method is to invoke the Born-Markov approximation which results in a Lindblad equation describing the evolution of the system. In some cases this approach is equivalent to a semiclassical model, in which the Hamiltonian of the system contains stochastic Markovian noise terms.

In publication **VI** we present an efficient method for simulating such a system. Assume the Hamiltonian is coupled to a continuous-time discrete Markovian process with N states. At time t, the probability of the noise state k is given by $P_k(t)$. The probabilities evolve according to

$$\vec{P}(t) = e^{\Gamma(t-t_0)}\vec{P}(t_0), \tag{5.10}$$

where Γ is a Markovian transition rate matrix. Corresponding to each noise state we have a system Hamiltonian $H_k(t)$. Also, each noise state is associated with a subnormalized state operator $\rho_k(t)$, with $\mathrm{Tr}(\rho_k(t)) = P_k(t)$, and $\sum_k P_k(t) = 1$. The conditional state operator $\rho_k(t)$ represents the ensemble average over all noise realizations that are in the state k at the time t. The total state operator of the system is obtained as $\rho(t) = \sum_k \rho_k(t)$.

The evolution of $\rho_k(t)$ under a time interval $\mathrm{d}t$ is given by the sum of all possible quantum evolutions weighted by their probabilities:

$$\rho_k(t+\mathrm{d}t) = \sum_j \int_{\eta_{j\to k}} \mathcal{E}_{H_\eta}\left(\rho_j(t)\right)\mathrm{d}P(\eta), \tag{5.11}$$

where the set $\eta_{j\to k}$ contains all noise trajectories that are in state j at t and in state k at $t+\mathrm{d}t$. Under an infinitesimal timestep $\mathrm{d}t$ the evolution effected by a Hamiltonian H is given by

$$\mathcal{E}_H\left(\rho(t)\right) \overset{O(\mathrm{d}t)}{=} \rho(t) + \frac{1}{i\hbar}\mathrm{d}t\left[H(t),\rho(t)\right], \tag{5.12}$$

which gives us

$$\rho_k(t+\mathrm{d}t) \overset{O(\mathrm{d}t)}{=} \sum_j \int_{\eta_{j\to k}} \left(\rho_j(t) + \frac{1}{i\hbar}\mathrm{d}t\left[H_j(t),\rho_j(t)\right]\right)\mathrm{d}P(\eta). \tag{5.13}$$

Under an infinitesimal timestep, we have $P_k(t+\mathrm{d}t) = (\delta_{kj} + \Gamma_{kj}\mathrm{d}t)P_j(t)$. Hence

$$\begin{aligned}\rho_k(t+\mathrm{d}t) &\overset{O(\mathrm{d}t)}{=} \sum_j \left(\rho_j(t) + \frac{1}{i\hbar}\mathrm{d}t\left[H_j(t),\rho_j(t)\right]\right)(\delta_{kj} + \Gamma_{kj}\mathrm{d}t)\\ &\overset{O(\mathrm{d}t)}{=} \rho_k(t) + \frac{1}{i\hbar}\left[H_k,\rho_k(t)\right]\mathrm{d}t + \sum_j \Gamma_{kj}\rho_j(t)\mathrm{d}t. \end{aligned}\tag{5.14}$$

Rearranging terms, dividing by $\mathrm{d}t$ and taking the infinitesimal limit gives us the dynamics of the conditional state operators ρ_k:

$$\partial_t\rho_k(t) = \frac{1}{i\hbar}\left[H_k(t),\rho_k(t)\right] + \sum_j \Gamma_{kj}\rho_j(t). \tag{5.15}$$

To obtain the ensemble average state operator at any time t, we need only sum together all the conditional state operators. We can thus simulate the average evolution generated by the stochastic Hamiltonian using a set of deterministic linear differential equations. The method can be expressed in a superoperator formalism which makes it straightforward to use it with the GRAPE optimization algorithm.

In publication **VI** this method is used to simulate the effect of random telegraph noise (RTN) on a qubit. RTN is the simplest nontrivial discrete noise model, in which the noise flips randomly between two levels and the times between successive flips are exponentially distributed. Since it has only two states, it is very light to simulate, yet it is found to result from a realistic quantum mechanical decoherence model consisting of a single bistable fluctuator coupled to the system under the Born-Markov approximation.

We use this noise simulation together with the modified GRAPE optimization algorithm described in Sec. 5.1 to derive simple one-qubit control sequences which can be used to suppress the effects of the noise when performing gate operations. There is no practical reason why this method could not be used in systems consisting of more than one qubit, or with more complex noise models, other than the limits set by finite computational resources.

6 Summary

> We can only see a short distance ahead, but we can see plenty there that needs to be done.
>
> Alan Turing

The objective of this thesis is to investigate methods for controlling quantum systems for the purpose of performing quantum information processing tasks in them. The problem is approached in the context of the quantum circuit model.

In publications **I** and **III** we study the problem of decomposing an arbitrary n-qubit gate into a series of elementary one-qubit and CNOT gates. In publication **I** we employ a recursive cosine-sine decomposition, along with a new circuit structure we call a uniformly controlled gate, to obtain a gate decomposition that is optimal in the number of elementary one-qubit gates and requires at most four times the optimal number of CNOTs.

Publication **II** is an application of uniformly controlled gates to state transformations, in which we derive an efficient circuit construction for transforming arbitrary pure n-qubit states to each other.

In publication **III** we derive an improved recursive construction for uniformly controlled gates which enables us to further reduce the CNOT counts of both the n-qubit gate decomposition and the state transform circuit by a factor of two. We also show how these circuits can be implemented using only nearest-neighbor gates at a very small overhead, making them easier to realize experimentally.

Publication **IV** introduces an easily computable local gate invariant which describes the local commutational properties of multiqubit gates, and explicitly presents the values of this invariant for all local equivalence classes of two-qubit gates. The results suggest that CNOT may not be the optimal choice for the entangling gate in elementary gate libraries.

Publications **V** and **VI** study the implementation of quantum gates in physical systems using external controls coupled to the Hamiltonian. In publication **V** we show how numerical optimization methods can be utilized to obtain near-optimal control sequences for two-qubit systems in the absence of noise. Using an existing Josephson device as an example, we derive an efficient CNOT sequence, and show how to implement it using almost-contemporary hardware.

Publication **VI** introduces an efficient deterministic method for simulating quantum systems subject to Markovian classical noise, and shows how this type of noise can arise from a genuine quantum mechanical environment under the Born-Markov approximation. As an example application, the method is used to derive RTN-resistant one-qubit control sequences.

In conclusion, the research within this thesis outlines a chain of methods for implementing quantum algorithms and performing other quantum information processing tasks, starting from the level of a controllable noisy Hamiltonian and finishing at abstract n-qubit quantum gates. These methods operate on a rather general level and are not limited to any specific physical realization of a quantum computer. Hence, within the scope this work we have not considered the otherwise important problems of state initialization and measurement. Other possible directions for future research could include specialized polynomial gate decompositions for specific quantum algorithms, more realistic, physically motivated noise models, and incorporating error correcting codes or decoherence-free subspace encodings to the analysis.

References

[1] Euclid of Alexandria, $\Sigma\tau o\iota\chi\epsilon\tilde{\iota}\alpha$ (Alexandria, ca. 300 BCE).

[2] A. Church, The Annals of Mathematics **2:33**, 346 (1932).

[3] A. Church, The Annals of Mathematics **2:34**, 839 (1933).

[4] A. Church, American Journal of Mathematics **58**, 345 (1936).

[5] S. C. Kleene, American Journal of Mathematics **57**, 153 (1935).

[6] S. C. Kleene, Duke Math. J. **2**, 340 (1936).

[7] A. M. Turing, Proceedings of the London Mathematical Society **2:42**, 230 (1936).

[8] G. E. Moore, Electronics **38**, 114 (1965).

[9] P. Benioff, J. Stat. Phys. **22**, 563 (1980).

[10] D. Deutsch, Proc. R. Soc. London A **400**, 97 (1985).

[11] D. Deutsch, Proc. R. Soc. London A **425**, 73 (1989).

[12] A. C.-C. Yao, in *Quantum circuit complexity*, Proc. of the 34th Ann. Symp. on Foundations of Computer Science (IEEE Press, 1993), pp. 352–361.

[13] E. Bernstein and U. Vazirani, SIAM Journal on Computing **26**, 1411 (1997).

[14] D. Deutsch and R. Jozsa, Proc. R. Soc. Lond. A **439**, 553 (1992).

[15] P. W. Shor, in *Proc. of the 35th Ann. Symp. on Foundations of Computer Science* (IEEE Computer Society, Los Alamitos, 1994), pp. 124–139.

[16] W. Diffie and M. E. Hellman, IEEE Transactions on Information Theory **22**, 644 (1976).

[17] R. L. Rivest, A. Shamir, and L. Adleman, Communications of the ACM **21**, 120 (1978).

[18] B. Schneier, *Applied Cryptography*, 2nd ed. (John Wiley & Sons, New York, 1995).

[19] L. K. Grover, in *Proceedings, 28th Annual ACM Symposium on the Theory of Computing (STOC)* (ACM, New York, 1996), p. 212.

[20] D. Dieks, Physics Letters A **92**, 271 (1982).

[21] W. K. Wootters and W. H. Zurek, Nature **299**, 802 (1982).

[22] P. W. Shor, Phys. Rev. A **52**, R2493 (1995).

[23] A. M. Steane, Phys. Rev. Lett. **77**, 793 (1996).

[24] P. W. Shor, in *Proc. of the 37th Ann. Symp. on Foundations of Computer Science* (IEEE Press, Los Alamitos, 1996), pp. 56–65.

[25] C. H. Bennett and S. J. Wiesner, Phys. Rev. Lett. **69**, 2881 (1992).

[26] C. H. Bennett, G. Brassard, C. Crépeau, R. Jozsa, A. Peres, and W. K. Wootters, Phys. Rev. Lett. **70**, 1895 (1993).

[27] C. H. Bennett and G. Brassard, in *Proceedings of IEEE International Conference on Computers, Systems and Signal Processing* (IEEE, New York, 1984), pp. 175–179.

[28] R. P. Feynman, Int. J. Theor. Phys. **21**, 467 (1982).

[29] S. Lloyd, Science **273**, 1073 (1996).

[30] M. Le Bellac, *Quantum Physics* (Cambridge University Press, New York, 2006).

[31] L. E. Ballentine, *Quantum Mechanics: A Modern Development* (World Scientific Publishing Company, Singapore, 1998).

[32] A. Einstein, B. Podolsky, and N. Rosen, Physical Review **41**, 777 (1935).

[33] J. S. Bell, Physics **1**, 195 (1964).

[34] J. F. Clauser, M. A. Horne, A. Shimony, and R. A. Holt, Phys. Rev. Lett. **23**, 880 (1969).

[35] D. M. Greenberger, M. A. Horne, A. Shimony, and A. Zeilinger, American Journal of Physics **58**, 1131 (1990).

[36] A. Aspect, P. Grangier, and G. Roger, Phys. Rev. Lett. **49**, 91 (1982).

[37] S. Gröblacher, T. Paterek, R. Kaltenbaek, Č. Brukner, M. Żukowski, M. Aspelmeyer, and A. Zeilinger, Nature **446**, 871 (2007).

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

[39] B. Schumacher, Phys. Rev. A **51**, 2738 (1995).

[40] A. Steane, Proc. R. Soc. Lond. A **452**, 2551 (1996).

[41] A. R. Calderbank and P. W. Shor, Phys. Rev. A **54**, 1098 (1996).

[42] D. Gottesman, Phys. Rev. A **54**, 1862 (1996).

[43] A. R. Calderbank, E. M. Rains, P. W. Shor, and N. J. A. Sloane, Phys. Rev. Lett. **78**, 405 (1997).

[44] P. Zanardi and M. Rasetti, Phys. Rev. Lett. **79**, 3306 (1997).

[45] P. Zanardi and M. Rasetti, Modern Physics Letters B **11**, 1085 (1997).

[46] E. Knill, R. Laflamme, and L. Viola, Phys. Rev. Lett. **84**, 2525 (2000).

[47] D. R. Simon, in *Proc. of the 35th Ann. Symp. on Foundations of Computer Science* (IEEE Computer Society, Los Alamitos, 1994), pp. 116–123.

[48] A. Y. Kitaev, Report TR96-003, Electronic Colloquium on Computational Complexity (ECCC) (1996).

[49] G. Brassard, P. Hoyer, M. Mosca, and A. Tapp, Quantum Amplitude Amplification and Estimation, arXiv:quant-ph/0005055 (2000).

[50] G. Brassard, P. Hoyer, and A. Tapp, Quantum Counting, arXiv:quant-ph/9805082 (1998).

[51] M. Born and V. Fock, Zeitschrift für Physik A **51**, 165 (1928).

[52] P. Zanardi and M. Rasetti, Physics Letters A **264**, 94 (1999).

[53] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser, Report MIT-CTP-2936, Massachusetts Institute of Technology (2000).

[54] D. Gottesman and I. L. Chuang, Nature **402**, 390 (1999).

[55] R. Raussendorf and H. J. Briegel, Phys. Rev. Lett. **86**, 5188 (2001).

[56] R. Raussendorf, D. E. Browne, and H. J. Briegel, Phys. Rev. A **68**, 022312 (2003).

[57] A. Y. Kitaev, Annals of Physics **303**, 2 (2003).

[58] M. H. Freedman, A. Kitaev, M. J. Larsen, and Z. Wang, Bull. Amer. Math. Soc. **40**, 31 (2003).

[59] D. P. DiVincenzo, Fortschritte der Physik **48**, 771 (2000).

[60] R. Hughes et al., *A Quantum Information Science and Technology Roadmap, version 2.0*, Report LA-UR-04-1778, Advanced Research and Development Activity (ARDA) (2004), `http://qist.lanl.gov/`.

[61] N. A. Gershenfeld and I. L. Chuang, Science **275**, 350 (1997).

[62] A. Shnirman, G. Schön, and Z. Hermon, Phys. Rev. Lett. **79**, 2371 (1997).

[63] D. V. Averin, Solid State Commun. **105**, 659 (1998).

[64] J. E. Mooij, T. P. Orlando, L. Levitov, L. Tian, C. H. van der Wal, and S. Lloyd, Science **285**, 1036 (1999).

[65] D. Vion, A. Aassime, A. Cottet, P. Joyez, H. Pothier, C. Urbina, D. Esteve, and M. H. Devoret, Science **296**, 886 (2002).

[66] L. B. Ioffe, V. B. Geshkenbein, M. V. Feigel'man, A. L. Fauchere, and G. Blatter, Nature **398**, 679 (1999).

[67] M. V. Feigel'man, L. B. Ioffe, V. B. Geshkenbein, P. Dayal, and G. Blatter, Phys. Rev. Lett. **92**, 098301 (2004).

[68] J. I. Cirac and P. Zoller, Phys. Rev. Lett. **74**, 4091 (1995).

[69] W. Paul, Rev. Mod. Phys. **62**, 531 (1990).

[70] D. Stick, W. K. Hensinger, S. Olmschenk, M. J. Madsen, K. Schwab, and C. Monroe, Nature Physics **2**, 36 (2006).

[71] D. Loss and D. P. DiVincenzo, Phys. Rev. A **57**, 120 (1998).

[72] J. Levy, Phys. Rev. Lett. **89**, 147902 (2002).

[73] J. R. Petta, A. C. Johnson, J. M. Taylor, E. A. Laird, A. Yacoby, M. D. Lukin, C. M. Marcus, M. P. Hanson, and A. C. Gossard, Science **309**, 2180 (2005).

[74] G. J. Milburn, Phys. Rev. Lett. **62**, 2124 (1989).

[75] Q. A. Turchette, C. J. Hood, W. Lange, H. Mabuchi, and H. J. Kimble, Phys. Rev. Lett. **75**, 4710 (1995).

[76] E. Knill, R. Laflamme, and G. J. Milburn, Nature **409**, 46 (2001).

[77] I. I. Rabi, J. R. Zacharias, S. Millman, and P. Kusch, Phys. Rev. **53**, 318 (1938).

[78] D. G. Cory, A. F. Fahmy, and T. F. Havel, Proc. Natl. Acad. Sci. USA **94**, 1634 (1997).

[79] L. M. Vandersypen, M. Steffen, G. Breyta, C. S. Yannoni, M. H. Sherwood, and I. L. Chuang, Nature **414**, 883 (2001).

[80] B. E. Kane, Nature **393**, 133 (1998).

[81] G. K. Brennen, C. M. Caves, P. S. Jessen, and I. H. Deutsch, Phys. Rev. Lett. **82**, 1060 (1999).

[82] D. Jaksch, H.-J. Briegel, J. I. Cirac, C. W. Gardiner, and P. Zoller, Phys. Rev. Lett. **82**, 1975 (1999).

[83] D. P. DiVincenzo, Phys. Rev. A **51**, 1015 (1995).

[84] A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. A. Smolin, and H. Weinfurter, Phys. Rev. A **52**, 3457 (1995).

[85] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed. (Johns Hopkins Press, Baltimore, 1996).

[86] J. J. Vartiainen, M. Möttönen, and M. M. Salomaa, Phys. Rev. Lett. **92**, 177902 (2004).

[87] C. C. Paige and M. Wei, Linear Algebra and Appl. **208**, 303 (1994).

[88] Y. Makhlin, Quantum Inf. Process. **1**, 243 (2002).

[89] S. Sternberg, *Group theory and physics* (Cambridge University Press, Cambridge, 1994).

[90] N. Khaneja, T. Reiss, C. Kehlet, T. Schulte-Herbrüggen, and S. J. Glaser, J. Magn. Reson. **172**, 296 (2005).

Appendix A Notation

A.1 Pauli matrices

Pauli matrices are the set $\{\sigma_x, \sigma_y, \sigma_z\}$ of three 2×2 traceless Hermitian matrices,

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \tag{A.1}$$

which obey the following algebra:

$$\sigma_i \sigma_j = I\delta_{ij} + i\epsilon_{ijk}\sigma_k. \tag{A.2}$$

Multiplied by the imaginary unit i they constitute a basis for the Lie algebra $\mathfrak{su}(2)$.