View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Aaltodoc Publication Archive

TKK Dissertations 3
Espoo 2005

UNITARY TRANSFORMATIONS FOR QUANTUM COMPUTING

Doctoral Dissertation

Juha J. Vartiainen

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

TKK Dissertations 3
Espoo 2005

UNITARY TRANSFORMATIONS FOR QUANTUM COMPUTING

Doctoral Dissertation

Juha J. Vartiainen

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 F1 at Helsinki University of Technology (Espoo, Finland) on the 8th of April, 2005, at 12 noon.

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

Teknillinen korkeakoulu
Teknillisen fysiikan ja matematiikan osasto
Materiaalifysiikan laboratorio

Distribution:
Helsinki University of Technology
Department of Engineering Physics and Mathematics
Materials Physics Laboratory
P.O.Box 2200
FI - 02015 TKK
URL: http://www.hut.fi/Yksikot/TeknillinenFysiikka/
Tel. +358 9 451 3153
Fax. +358 9 451 3164
E-mail: juhav@iki.fi

© 2005 Juha J. Vartiainen

ISBN 951-22-7611-9
ISBN 951-22-7612-7 (PDF)
ISSN 1795-2239
ISSN 1795-4584 (PDF)
URL: http://lib.hut.fi/Diss/2005/isbn9512276127/

TKK-DISS-1982

Otamedia Oy
Espoo 2005

HELSINKI UNIVERSITY OF TECHNOLOGY P.O. BOX 1000, FIN-02015 HUT http://www.hut.fi	ABSTRACT OF DOCTORAL DISSERTATION

Author Juha Johannes Vartiainen

Name of the dissertation

Unitary Transformations for Quantum Computing

Date of manuscript 24.1.2005	Date of the dissertation 8.4.2005
□ Monograph	☑ Article dissertation (summary + original articles)

Department	Engineering Physics and Mathematics
Laboratory	Materials Physics Laboratory
Field of research	Quantum Computing
Opponent(s)	Dr. Yuriy Makhlin
Supervisor	Prof. Jukka Pekola
(Instructor)	Prof. Mikio Nakahara

Abstract

The last two decades have seen an enormous increase in the computational power of digital computers. This was due to the rapid technical development in manufacturing processes and controlling semiconducting structures on submicron scale. Concurrently, the electric circuits have encountered the first signs of the realm of quantum mechanics. Those effects may induce noise and thus they are typically considered harmful. However, the manipulation of the coherent quantum states might turn out be the basis of powerful computers - quantum computers. There, the computation is encoded into the unitary time evolution of a quantum mechanical state vector. Eventually, quantum mechanics could enable one, for example, to read secret electronic messages which are encrypted by the widely employed RSA cryptosystem - a task which is extremely laborious for the current digital computers.

This thesis presents a theoretical study of the coherent manipulations of pure quantum states in a quantum register, that is, quantum algorithms. An implementation of a quantum algorithm involves the initialization of the input state and its manipulation with quantum gates followed by the measurements. The physical implementation of each gate requires that it is decomposed into low-level gates whose physical realizations are explicitly known. Here, the problem is examined from two directions. Firstly, the numerical optimization scheme for controlling time-evolution of a closed quantum system is discussed. This yields a method for implementing quantum gates acting on up to three quantum bits, qubits. The approach is independent of the physical realization of the quantum computer, but it is considered explicitly for a proposed inductively coupled Josephson charge qubit register. Secondly, the techniques of numerical matrix computation are utilized to find a general method for decomposing an arbitrary n-qubit gate into a sequence of elementary gates,which act on one or two qubits.

The results of this thesis help to improve the implementation of quantum algorithms. The quantum circuit construction developed in the thesis is the first one to achieve the asymptotically minimal complexity in the number of elementary gates. In context of acceleration of quantum algorithms we present a gate-level study of Shor's algorithm and show how to accelerate the algorithm by merging several elementary gates into multiqubit gates. Finally,the requirements set by the resulting gate array are compared to the properties of superconducting qubits. This allows us to discuss the feasibility of the Josephson charge qubit register,for instance, as hardware for breaking the RSA cryptosystem.

Keywords quantum computing, unitary transformations, gate decompositions

UDC	530.145:519.7:681.3	Number of pages	64
ISBN (printed)	951-22-7611-9	ISBN (pdf)	951-22-7612-7
ISBN (others)		ISSN	1795-2239 (print); 1795-4584 (pdf)

Publisher Helsinki University of Technology, Materials Physics Laboratory

Print distribution Helsinki University of Technology, Materials Physics Laboratory, P.O.Box 2200, FI - 02015 TKK

☑ The dissertation can be read at http://lib.hut.fi/Diss/

Preface

The work described in this thesis has been carried out in the Materials Physics Laboratory at the Helsinki University of Technology during the years 2002–2004. Academy of Finland has funded this work first through Graduate school in Technical Physics and later by the research project in quantum computation for some nine moths in total. The funding for the main part of the work is by two one-year scholarships from the Foundation of Technology and Research Foundation of Helsinki University of Technology. This work was completed by funding from Jenny and Antti Wihuri Foundation. Emil Aaltonen Foundation and Magnus Ehrnrooth Foundation are acknowledged for travel grants to Japan. I am also grateful to the Heikki ja Hilma Honkanen Foundation and Nokia foundation for financial support. I thank CSC Finland for providing the parallel computing resources which are extensively utilized in this work.

My interest to quantum computing was launched in 2001 on a course given by Prof. Mikio Nakahara (Kinki university, Osaka) at Helsinki University of Technology. After the course I have been privileged to have Prof. Nakahara as my advisor. I am grateful to him for teaching me mathematics, physics, and Japanese culture. I wish I could express gratitude for Prof. Martti Salomaa, supervisor of this thesis, who provided excellent working conditions in Materials Physics Laboratory - facilities and great atmosphere to carry out research. After his departure, Acad. Prof. Jukka Pekola has been acting as supervisor of this thesis. I thank him for proof reading this manuscript and help in arranging numerous practical things. Furthermore, I thank Dr.Tech. Antti Niskanen, M.Sc. Mikko Möttönen, M.Sc. Ville Bergholm and M.Sc. Teemu Ojanen for fruitful collaboration, discussions on scientific issues, and all the joyful moments during these years.

I want also to thank you guys: Anssi Mennala, Juha Karvanen, Jukka Reitmaa, Jani Lindqvist, Teijo Salonen, Matti Öhman, and Hannu Pulakka for keeping me active with your interesting off-duty projects. Especially, I thank Teemu Mäki-Patola for sharing the residence and all the fun in the study years. Those years were great. Finally I thank my parents, Ritva and Erkki, and Johanna for support and love.

Espoo, December 2004

Juha Vartiainen

List of Publications

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

I A. O. Niskanen, J. J. Vartiainen, and M. M. Salomaa, *Optimal multiqubit operations for Josephson charge qubits*, Phys. Rev. Lett. **90**(19), 197901 (2003). © 2003 American Physical Society.

II J. J. Vartiainen, A. O. Niskanen, M. Nakahara and M. M. Salomaa, *Acceleration of quantum algorithms using three-qubit gates*, The International Journal of Quantum Information Science **2**(1), 1-10 (2004). © 2004 World Scientific Publishing Company.

III J. J. Vartiainen, A. O Niskanen, M. Nakahara, and M. M. Salomaa, *Implementing Shor's algorithm of Josephson Charge Qubits*, Phys. Rev. A **70**(1), 012319 (2004). © 2004 American Physical Society.

IV J. J. Vartiainen, M. Möttönen, and M. M. Salomaa, *Efficient decomposition of quantum gates*, Phys. Rev. Lett. **92**(17), 177902 (2004). © 2004 American Physical Society.

V M. Möttönen, J. Vartiainen, V. Bergholm, and M. M. Salomaa, *Quantum Circuits for General Multiqubit Gates,* Phys. Rev. Lett. **93**(13), 130502 (2004). © 2004 American Physical Society.

VI M. Möttönen, J. J. Vartiainen, V. Bergholm, and M. M. Salomaa, *Transformation of quantum states using uniformly controlled rotations*, Submitted to Quantum Information and Computation (2004). Preprint: quant-ph/0407010.

VII V. Bergholm, J. J. Vartiainen, M. Möttönen, M. M. Salomaa, *Quantum circuits with uniformly controlled one-qubit gates*, Accepted for publication in Physical Review A (2005). Preprint: quant-ph/0410066.

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

Author's Contribution

I have had a central role in all aspects of the work reported in this thesis and strongly influenced on initiating the research reported in Publications [**II-V**] and [**VII**]. I have mainly written the manuscripts for Publications [**II–IV**] and also actively participated in writing publications [**I**],[**V–VII**].

Publications [**I-III**] describe a method for finding control sequences for a hypothetical superconducting Josephson charge qubit register. I designed and partially programmed an optimization software, which was employed for finding optimal control parameter sequences numerically on a parallel computer. Publications [**II**] and [**III**] discuss the possibility to accelerate a quantum algorithm using three-qubit gates. The idea of acceleration emerged after a wide literature study that I conducted. I also produced all of the numerical results for Publications [**II**] and [**III**].

In the publications [**IV-VII**] new quantum circuit topologies for implementing quantum gates are presented. For these publications I found the perfect match of the uniformly controlled rotation and Cosine-Sine decomposition [**V**] and strongly contributed to discovering the idea to utilize the modified order of basis vectors [**IV**] and modification of the quantum multiplexor construction to implement block diagonal matrices [**VII**]. Publication [**VI**] extends the ideas of [**V**] for preparing a quantum state. There my role was mainly to improve and clarify the manuscript.

I have produced all the figures in the publications, except Figs. 3-5 in [**I**]. I have also presented the results of publication [**II**] in ERATO Conference on Quantum Information Science 2003 (EQIS'03) in Kyoto, Japan and results of [**IV**,**V**] in EQIS'04 in Tokyo, Japan.

Contents

1 Introduction

Quantum mechanics, predominately established in 1927 [1], strongly contributes to our modern philosophy of life. It gives a peculiar but precise explanation for the physical phenomena on the atomic scale. In addition, it helps us to understand the properties of certain macroscopic systems such as superconductors and lasers [2]. The development in techniques of microscopy, nano-fabrication, and accurate control of high frequencies and low temperatures continuously build up technological potential. Eventually, it leads to a possibility of accurate engineering of macroscopic quantum mechanical systems. This raises strong prospects towards devices whose functionality essentially relies on the coherent properties of the quantum states, such as a quantum mechanical computer [3].

Several different physical systems allow for controlled manipulation of their quantum state [4, 5]. Thus far the most extensive controlled discrete state-space comprising of collection of seven two-state systems has been demonstrated using nuclear magnetic resonance (NMR) in a liquid solution [6]. The other techniques studied for this purpose involve nuclear spins [6, 7], trapped ions [8], cavity quantum electrodynamics [9], and electrons in quantum dots [10, 11]. Especially, the recently established possibility to manipulate nanolectronic superconducting circuits [12–25] has considerably broadened the quantum realm.

One of the promising applications for the manipulation of quantum states is quantum computation [3, 4, 26–28]. Quantum computers are supposed to be useful for solving certain mathematical problems efficiently. Especially, Shor's integer factorization [29] and Grover's database search [30] show considerable speed-up compared to the algorithms on classical digital computers. In particular, Shor's algorithm could be utilized to break the widely employed RSA cryptosystem [31] in polynomial time. This would strongly influence our society that relies on the safety of information encryption protocols.

In addition to applications in computation, the framework of coherent manipulation of quantum states can be used to describe other entanglement-related phenomena of quantum mechanics, such as quantum teleportation [32] and quantum cryptography [33–35]. The quantum teleportation transports the initial state of a source system into separate destination system while the quantum cryptography provides revolutionary secure communication protocol. In the communication the transported information is protected at fundamental level by the laws of physics instead of relying on the limitations of any mathematical techniques or computing technology. Both quantum teleportation and cryptography are successfully demonstrated experimentally, and devices for quantum cryptography are even commercially available [36].

In the classical sense the quantum computers are universal: a quantum computer allows for emulating a classical computer but the quantum computer cannot be efficiently emulated by a classical computer [37]. This is due to two distinctive features of the quantum computer: quantum mechanical superposition and entanglement. According to the superposition principle a quantum system can virtually lie in two or more distinct states simultaneously. The entanglement is also a purely quantum phenomenon that allows the states of two or more subsystems to be described in comparison to each other. This leads to non-classical correlations between observable physical properties of the systems.

Physically, any computation must be encoded into a temporal evolution of a physical system. In the quantum circuit model [38], which is the current paradigm in quantum computation, the amenable system for computation is a collection of two-level systems called a quantum register. In this context, the two-level systems are called quantum bits or qubits. Due to the laws of quantum mechanics their properties are richer than the ones of classical bits. When isolated from the dissipative environment, the physical state of the quantum register corresponds to a vector in a very high-dimensional Hilbert space. The temporal evolution of the state vector is described by the Schrödinger equation [39] whose formal solution is called a propagator. The propagator is a unitary operator, which combines quantum mechanics to the computation. In addition to the quantum circuit model which we use here, quantum computation can be formulated using various other approaches, for example, quantum random walk [26], quantum Turing machine [40, 41], quantum computing by measurements [42–44], and quantum cellular automata [45].

In the context of quantum computing the unitary transformations are called gates. The role of bits and gates feature a remarkable difference between the most proposals of a quantum computer and a classical computer based on semiconducting transistors. The classical logic gates are implemented by static semiconductor structures and the bits are flying objects between them, whereas the quantum bits are often static entities and the gates are operations which are actively applied on them. A quantum gate may involve in the simplest form only adjustment of the occupation probabilities of given quantum states. In general, the full unitary transformation acting on all the possible states of a qubit register can be implemented.

The form of the propagator depends on the interactions appearing between the physical qubits and the transition probabilities within the states of a single qubit. In typical realizations of quantum register, the form of the inter-qubit interactions is fixed and the Hamiltonian allows only limited tuning of the interaction parameters. This naturally leads to restrictions for the realizable gate arrays. Let us consider two examples. An array of two-electron quantum dots [11] may be considered as a quantum register. In this register two different spatial charge distributions within a quantum dot will act as qubit states. The Coulomb force between the electrons in the quantum dots results in inter-qubit coupling. The range of the Coulomb force is, in principle, infinite. However, in practice the interactions couple only spatially neighboring qubits. In an NMR quantum computer [7] the qubits are encoded into nuclear spins of a molecule, which acts as a quantum register. The coupling between the qubits is due to the magnetic dipole-dipole forces. Since the dipole-dipole coupling strongly reduces as the separation of the atoms grows, the interaction part of the Hamiltonian consists of, to a good approximation, only the nearest neighbor couplings. Therefore the native gate library of a quantum computer based on a quantum dot or NMR register involves only two-qubit gates acting on a pair of nearest-neighbor qubits. However, any gate can be build of them as shown below. Although several realizations of quantum computer provide only limited library of gates, there are also proposals, such as Josephson charge qubit register [24], where the coupling between arbitrary two-qubits is controllable.

The current level of technology provides only limited possibilities to make use of coherent quantum states in computing. The quantum states featuring coherent properties can be achieved in the experiments but they are fragile and easily destroyed by undesired interactions with the environment. This is called decoherence [46]. Due to the omnipresent decoherence, efficiency in using the computational resources is of prime importance in the implementation of quantum computations. DiVincenzo [47] gives a concrete list of criteria which a quantum system has to fulfill in order to be amenable for execution of reasonably extensive algorithms.

In this thesis I study the efficient implementation of quantum computation. The mathematical structure of the gate operations and the form of the interactions of the physical system set the lower bound for the execution time needed for implementing each of the gates. I show how to find for quantum gates implementations which almost achieve the theoretically estimated lower bounds for the need of the computational resources. Those implementations take advantage of techniques developed for numerical optimization and matrix computation. Publications [**I-III**] study the physical implementation of quantum gates, while the publications [**IV-VII**] discuss the mathematical structure of the gates and how it helps to find an efficient implementation for them.

From the physical point of view, one needs to be able to control the system strongly enough in order to obtain the desired propagators. Typically this is accomplished using external fields, such as laser pulses or electric and magnetic fields. Subsequently, the fundamental problem in manipulating the quantum state is to find the proper setting for the external fields as a function of time. This can be considered as an optimization problem. Properly conducted optimization provides, in general, the most effective implementation for any unitary transformation. Publications [**I-III**] show a straightforward optimization technique for finding the control parameters of a Josephson charge qubit register. In addition to quantum computation the similar strategy can be applied to more practical applications, such as creating an accurate standard for electric current [48].

The dimension of the state space and thus the optimization problem grow exponentially with the number of qubits in the register. Therefore, the numerical optimization becomes computationally unrewarding for transformations acting on many qubits. To efficiently implement a unitary transformation for arbitrarily many qubits, their internal structure must be studied and utilized in the implementation.

Intuitively, unitary transformations can be understood as straightforward extensions of rotations in a multidimensional complex vector space. Their mathematical properties are studied extensively in the theory of the Lie groups [49]. They also play a significant role in the field of matrix computation [50] which is the foundation of numerical computation in engineering. One of the characteristic properties of unitary transformations is that each of them can be expressed as a product of consecutive unitary transformations which are simpler in form than the original one was. In the context of quantum computation this is called a quantum gate decomposition where the expression gate refers to transformations restricted into a subspace of a few qubits. The unitary transformation corresponding to the simplest quantum gates acting on one or two qubits are of tensor product form. Surprisingly, decompositions of this kind have not been studied in mathematics in detail. However, a series of physics papers [51–53] show that any transformation can be exactly achieved using a rather limited set of gates. For example, the set of one-qubit gates with almost any fixed two-qubit gate form a universal set of elementary gates [52]. Publications [**IV**],[**V**], and [**VII**] present efficient methods for finding implementation of a general n-qubit gate using an array of elementary gates.

Apparently, the theoretical research in quantum computing has proceeded much further than the experimentally realizable devices. Undoubtedly, the strong progress in both fields continues. However, even if a large-scale quantum computer would never be built, the research of quantum computing helps us to better understand the universe we live in. In a broad sense, the whole universe can be considered to be gigantic quantum computer [54]. The question is: what is the problem it is solving.

This thesis is organized as follows. Section 2 introduces the basic concepts of quantum computation and studies briefly its physical grounds. Starting from the Schrödinger equation of the isolated quantum system a method for finding the propagator corresponding to a desired quantum gate by numerical optimization is introduced. However, due to the practical limitations of computational resources the optimization approach is applicable only for gates acting on less than four qubits. In Sec. 3, methods for decomposing a general quantum gate acting on arbitrarily many qubits are presented. This section also presents implementations for various generic quantum gates which can be used as a basic building blocks of a quantum compiler. Section 4 applies the presented methods to Shor's integer factorization algorithm on a Josephson charge qubit register. Finally, Sec. 5 summarizes the most important results of this thesis.

2 Quantum state engineering

This section briefly reviews the essentials of the quantum mechanics needed for understanding quantum computation. A more comprehensive discussion of the fundamentals of quantum mechanics is given, for example, by Ballentine [2], while Nielsen and Chuang [3] present a thorough introduction to quantum computing.

Publications [**I-III**] describe a method for finding the control parameter sequences for a quantum mechanical system which realize a desired unitary transformation on it. The method is briefly reviewed here. In addition to this method, several authors have presented other approaches to the problem in literature. In particular, the time-optimal implementation of two-qubit gates using Cartan decomposition has been solved [55] and a method based on optimal control theory has been introduced, for instance, in Ref. [56]. However, the most fundamental problem of the field — finding the time-optimal implementation of an arbitrary n-qubit gate with a given Hamiltonian — remains unsolved.

2.1 State space of a quantum register

Nine philosophically different but physically equivalent formulations have been found for quantum mechanics [1]. We follow the one formulated by Schrödinger [39] which states that the physical state of a quantum system can be fully described by a temporally evolving vector $|\psi(t)\rangle$ in a complete complex inner product space $\mathbb{H}$ called a Hilbert space. This requires that the system is isolated from the environment. In this formalism, each of the measurable physical quantities corresponds to a certain Hermitian operator $\mathcal{A} = \mathcal{A}^\dagger$. The expectation value of the measurable quantity associated with $\mathcal{A}$ is obtained as the inner product

$$\langle \mathcal{A} \rangle_t = \langle \psi(t)| \mathcal{A} |\psi(t)\rangle , \tag{2.1}$$

where we have assumed that the system is in the state $|\psi(t)\rangle$ at time t. Equation (2.1) immediately implies that the global phase of the state vector does not carry any physical information; states $|\psi(t)\rangle$ and $e^{i\theta} |\psi(t)\rangle$, where $\theta \in \mathbb{R}$, are equivalent with respect to all physical observables.

Any bounded Hermitian operator $\mathcal{A}$ has a complete set of orthonormal eigenvectors $\{|\lambda_i\rangle\}$ which span the Hilbert space $\mathbb{H}$. Here the curly braces denote the set whose elements are characterized by index i. Especially, any state $|\psi(t)\rangle$ can be presented as

$$|\psi(t)\rangle = \sum_i a_i(t) |\lambda_i\rangle , \tag{2.2}$$

where $\{a_i(t)\}$ are complex numbers. The formalism describes a projective measurement $\mathcal{M}$ of the state with respect to basis $\{|\lambda_i\rangle\}$; the probability of finding state λ_i occupied at time t is $|a_i(t)|^2$. To be consistent with the probabilistic interpretation, the state $|\psi(t)\rangle$ is normalized such that $\sum_i |a_i(t)|^2 = 1$.

An operator of special interest is the Hamiltonian $\mathcal{H}$, whose expectation value corresponds to the total energy of the system. The eigenvectors of the Hamiltonian, $\{|\phi_i\rangle\}$, are the energy eigenstates, while the corresponding eigenvalues are called the energy levels of the system. The eigenstates of $\mathcal{H}$, or any other Hermitian operator, yield a natural basis for the state space of the system under consideration. This basis can be utilized as a computational basis in quantum computation.

Let us restrict ourselves to the system consisting of d distinct quantum states. A complex vector of unit length in a d-dimensional complex space fully describes the state of this system: $|\psi(t)\rangle = \sum_{i=1}^{d} a_i(t)\,|\lambda_i\rangle$. Similarly the physical state of a register consisting of n of those systems can be represented with a vector having d^n components

$$|\phi(t)\rangle_n = \sum_{i=1}^{d^n} a_i(t)\,|e_i\rangle\,, \tag{2.3}$$

where $\{|e_i\rangle\}$ is a set of basis vectors which spans $\mathbb{H}$.

In this overview I study only the properties of a register of n qubits ,*i.e.*, a system where $d = 2$, and denote the qubit states as

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \text{and} \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \tag{2.4}$$

The state space of an n-qubit register is $N = 2^n$-dimensional. For purposes of quantum computation, it is convenient to choose the basis vectors for this space $\{|e_k\rangle\}$, where $k = 1, ..., N$, according to $|e_k\rangle = \bigotimes_i |x_i^k\rangle$, where $x_i^k \in \{0, 1\}$ and the index $i = 1, ..., n$ refers to the physical qubits. In this basis the state vector of the system is of the form

$$|\Psi\rangle = \sum_{i=1}^{N} a_i\,|e_i\rangle \quad \text{and} \quad \sum_{i=1}^{N} |a_i|^2 = 1. \tag{2.5}$$

Conventionally in quantum computing, the order of the basis vectors has been chosen such that the values x_i^k essentially form the binary representation of the number $k-1$, *i.e.*, $k = 1 + \sum_{i=0}^{n} 2^i x_i^k$. We note that the order of the basis vectors in the computational basis is not fixed. Publication [**IV**] takes advantage of this freedom to reorder the basis vectors, which simplifies the gate arrays, as explained in Sec. 3.3.5.

The fundamental difference of the quantum computer compared to the classical one arises from the utilization of the properties of a high-dimensional Hilbert space $\mathbb{H}$. For comparison, the states accessible for a classical computer are limited to states which can be presented by vectors $|\Psi\rangle = |e_i\rangle$, *i.e.*, by the vectors in which all of the weight factors except the one for state i vanish. The quantum mechanical superposition principle allows several weight factors to be simultaneously non-zero. This makes the quantum mechanical state space considerably larger than the classical one. Especially, a set of non-classical states called the entangled states have interesting properties since they yield non-classical correlations between the qubits. For example, the EPR-state (after Einstein, Podolsky and Rosen [57]) of two qubits

$$|\Psi\rangle_{\mathrm{EPR}} = \frac{1}{\sqrt{2}} \left(|01\rangle - |10\rangle \right), \tag{2.6}$$

has extraordinary properties. When the qubits are measured independently, each of them has a 50% probability of being in state zero (or one). When the coordinate system or the local parameters of the measurement systems are changed, the measurements still remain correlated. Especially, when the qubits are moved far from each other after initializing the EPR-state, the changes in the measurement setup of the first qubit can be easily understood to involve non-local interactions which change the state of the second qubit instantaneously. This easily leads to paradoxes, since there is no classical counterpart for this correlation. Theoretically, the discrepancy of classical and quantum worlds can described by famous Bell's inequality [58,59]. However, the measurement of the quantum correlations still require state of art experiments [60].

2.2 Unitary time-evolution

The temporal evolution of a quantum register is described by the Schrödinger equation

$$i\hbar \frac{\partial}{\partial t} |\Phi(t)\rangle = \mathcal{H}(\gamma) |\Phi(t)\rangle, \tag{2.7}$$

where $\mathcal{H}(\gamma)$ is the Hamiltonian of the system. Here, the Hamiltonian is taken to depend on a set of tunable parameters, formally denoted as γ. In general, the parameters are time-dependent, $\gamma = \gamma(t)$. The formal solution of Eq. (2.7) at time t for an initial state $|\Phi(0)\rangle$ is

$$|\Phi(t)\rangle = \mathcal{U}(t,0) |\Phi(0)\rangle, \tag{2.8}$$

where the propagator $\mathcal{U}(t,0)$ is

$$\mathcal{U}(t,0) = \mathcal{T} \exp\left(-\frac{i}{\hbar} \int_{\gamma(t)} \mathcal{H}(\gamma(t)) dt \right). \tag{2.9}$$

Above $\mathcal{T}$ stands for the time-ordering operator. The time ordering is needed, since the Hamiltonian operators associated with different moments of time do not typically commute. From the form of Eq. (2.8) and the properties of the unitary transformation one notes that the norm of the state vector is constant, unity, in time. The second important note is that the temporal evolution of the quantum system is reversible, *i.e.*, $|\Phi(0)\rangle = \mathcal{U}^\dagger(t,0) |\Phi(t)\rangle$.

Since the state space $\mathbb{H}$ of the register is 2^n-dimensional and all the operators in the formalism are linear, the operators can be represented as $2^n \times 2^n$ matrices once the basis is fixed. The Hamiltonian $\mathcal{H}$ takes the form of a $2^n \times 2^n$ Hermitian matrix, H. Similarly the propagator $\mathcal{U}$ takes the form of a $2^n \times 2^n$ unitary matrix, U. This unitary matrix is the connection between quantum mechanics and quantum computing. Namely, for properly chosen control sequences $\gamma(t)$, the matrix U corresponds to a certain quantum gate which may be used as a building block of quantum algorithms.

The analysis of Eq. (2.9) shows that the global phase of the state vector $|\Phi(t)\rangle$, the determinant of U, and the zero-level of energy represent the same phenomenon called gauge freedom which does not have any measurable consequences. Without loss of generality, the Hamiltonian can be regarded traceless. The traceless Hamiltonian generates propagators having unit determinant. Consequently, instead of considering the full unitary group $U(2^n)$, we restrict ourselves to study the propagators that belong to the group $SU(2^n)$.

The model of a quantum register presented above is an idealization which assumes that the quantum system lies in a pure state, *i.e.*, it lacks the interactions with the environment. Thus the model totally omits the numerous degrees of freedom of the environment which tend to become entangled with those of the register. This results in inevitable decoherence [46,61] in any experimental setup. Moreover, one certainly applies external forces to the system when tuning the parameters of the Hamiltonian. Strictly speaking this is not consistent with the isolation requirement. Neglecting the environment we thus ignore the fact that the coherent lifetime of the quantum state is limited. The shortness of the coherent lifetime may present fundamental difficulties in scaling the quantum register up to large sizes; this scalability is the basic requirement for the execution of meaningful quantum algorithms [47]. Nevertheless, ingenious qubit architectures combined with powerful theoretical tools, such as decoherence free subspaces and quantum error correction [62], may be used in order to reduce the detrimental influence of the environment.

2.3 Finding the control parameters by numerical optimization

The temporal evolution of a quantum register resulting from the applied Hamiltonian $\mathcal{H}$ with control parameter sequences $\gamma(t)$ can be straightforwardly evaluated using Eq. (2.9). In contrast, to determine the physical realization of a quantum gate, the parameter sequences $\gamma(t)$, the inverse problem must be solved. This section presents numerical optimization procedure which gives a solution for the inverse problem.

Let us start by discretizing the path $\gamma(t)$ of control parameters which, in general, involves infinitely many degrees of freedom. Denote the finite set of real numbers which describe the path $\gamma(t)$ by X_γ. Consequently, for a given arbitrary unitary matrix U, the solution of the inverse problem is to find proper values of the parameters X_γ. When the parameter path associated with X_γ is applied to a quantum register, the time-evolution of the register results in a quantum gate U_{X_γ}.

To be more concrete, the numerical optimization problem is to find the zeroes of the error function

$$p(X_\gamma) = \|U - U_{X_\gamma}\|. \tag{2.10}$$

Minimization of $p(X_\gamma)$ over parameters X_γ produces an approximation U_{X_γ} for the desired matrix U. Above $\|\cdot\|$ denotes a matrix norm [50], which can be taken to be the Frobenius trace norm, which is numerically efficient to compute. Since all the matrix norms are mathematically equivalent, a small value of a Frobenius trace norm implies a small value in all other norms as well.

In practice, polygons in the parameter space provide a suitable discretization for path $\gamma(t)$. Let us consider a polygonal path having $\nu + 1$ edges, which starts and ends at the origin, *i.e.*, the same fixed point for all gates. This makes it possible to execute a sequence of gates one after the other, without need of abrupt changes in the control parameter values. The resulting control-parameter path $\gamma(t)$ for an n-qubit register is of the vector form

$$\gamma(t) = \left[\gamma^1(t), \quad \ldots \quad, \gamma^k(t)\right]^T, \tag{2.11}$$

where $\gamma^i(t)$ $(i = 1, ..., k)$ are continuous piecewise linear functions of time for the chosen parametrization. In order to evaluate the propagator in Eq. (2.9) with path $\gamma(t)$ one only needs to specify the k coordinates for the ν vertices of the polygon. On the other hand, a $2^n \times 2^n$ unitary matrix belongs to the group $SU(2^n)$ and the number of the real degrees of freedom of group $SU(2^n)$ is $4^n - 1$. This sets the lower bound for the number of discretization parameters needed for realizing the n-qubit gate:

$$k\nu \geq 4^n - 1, \tag{2.12}$$

which is the necessary requirement for being able to parameterize all the degrees of freedom in the matrix U.

For the calculations we choose the time spent in traversing each edge of the polygon be constant, say, unity. Thus the execution time of the gate depends linearly on the number ν of the vertices in the parameter path. This allows us to compare the execution times of different realizations of a gate. In a particular physical implementation. The actual time that corresponds to the traversing of one edge of the polygon depends on the available energy scales of the Hamiltonian.

To evaluate the unitary operator U_{X_γ} for a parameter set X_γ we need a numerical method which is efficient, yet numerically stable. We divide the path $\gamma(t)$ into short intervals that take time Δt to traverse. If γ_i collectively denotes the values of all the parameters in the midpoint of the i^{th} interval, and m is the number of such intervals, we then find to a good approximation

$$U_{X_\gamma} \approx \exp(-i\mathcal{H}(\gamma_m)\Delta t) \ldots \exp(-i\mathcal{H}(\gamma_1)\Delta t)\,. \tag{2.13}$$

Especially, the form of the Eq. (2.13) takes into account the time ordering of the non-commuting propagators in the different moments of time. The evaluation of the U_{X_γ} using the above approximation consists of independent matrix multiplications which can be evaluated simultaneously. This allows straightforward parallelization of the computation.

To evaluate each of the exponential factors in Eq. (2.13) we may employ the truncated Taylor series expansion

$$e^{-i\mathcal{H}\Delta t} \approx \sum_{k=0}^{l} \frac{(-i\mathcal{H}\Delta t)^k}{k!}. \tag{2.14}$$

Since Δt is small the eigenvalues of the anti-Hermitian matrix $A = -iH\Delta t$ are significantly less than unity and the expansion converges rapidly. The applicability of the approximation is confirmed for the optimized path using finer discretization.

The Taylor series provides a fast and accurate but by not unique method to evaluate the propagator. For example we could use the Cayley form

$$e^{-i\mathcal{H}\Delta t} \approx (1 - i\mathcal{H}\Delta t/2)(1 + i\mathcal{H}\Delta t/2)^{-1} \tag{2.15}$$

to evaluate the matrix exponential as well. The special property of the Cayley form is that the approximation it yields is unitary by nature. Another applicable approach to evaluating the propagator would be to directly integrate the Schrödinger equation using some of the adaptive Runge-Kutta methods [63].

Using the above methods the minimization of the error function in Eq. (2.10) is possible, provided the Hamiltonian under consideration allows for implementation of the desired unitary transformations. Still, finding of the realization for any quantum gate may be computationally hard; the size of the unitary matrices and thus the number of dimensions of the optimization problem grows exponentially with the number of qubits n on which the gate acts. Due to the restricted availability of computational capacity the implementation of gates acting on a large number of qubits is limited. In practice gates for more than three or four qubits have turned out to be computationally too demanding using current supercomputers.

3 Gate-efficient decompositions

One of the main results in quantum computation is that a properly constructed array of elementary gates implements any unitary transformation [52]. In particular, this provides us with a method to implement a quantum gate even though the numerical optimization method described in the previous section would fail. The generation of the elementary gate sequence for a given computational task is called quantum compiling. A quantum compiler is a computer program which converts a given quantum algorithm into an array of elementary quantum gates analogously to classical compilers which are used to synthesize logic gate circuits for microprocessors. Although constructing an elementary gate array for any unitary transformation is, in principle, straightforward procedure using the results by Barenco *et al.* [52], finding of the most favorable gate array is still a highly non-trivial task. The minimization of the number of elementary gates as well as the number of extra qubits in the compilation helps one to better implement algorithms on any physical realization of a quantum computer. Achieving gate arrays of lower gate count is interesting not only because it results in shorter execution time in general but also because it may introduce less errors.

Publications [**IV-VII**] introduce several efficient techniques which can be utilized in quantum compiling. In all these techniques, the strategy is to first find an elementary gate construction for a certain generic intermediate level quantum gate. In the second step the intermediate level gates serve as building blocks for unitary matrix decompositions [50]. In addition to matrix decompositions the quantum compiler may employ the algorithmic definition of the implemented computation, when it is available, see *e.g.*, Ref. [64]. Moreover, the compiler may manipulate and simplify the obtained gate circuits by peephole optimization. For this purpose, Ref. [65] gives a comprehensive list of simplification rules for the elementary gates.

In this section three categories of quantum gates are considered: gates permuting the basis vectors, Sec. 3.2; gates corresponding to sparse matrices or involving high internal symmetries, Sec. 3.3; and unstructured n-qubit gates, Sec. 3.3.5.

3.1 Elementary gates and notation

The set of unitary transformations which is available on a quantum computer is called a quantum gate library. The gate library is universal when it is sufficient for exactly implementing or arbitrarily closely approximating any n-qubit unitary transformation [38,41]. The result of a series of papers [51–53] is that a library involving certain one-qubit gates and almost any fixed two-qubit gate is universal. Below I consider a gate library which consists of the controlled-`NOT` gate (CNOT) and all the one-qubit gates, $U \in SU(2)$. The simple form of CNOT allows it to be used as a building block of logical functions but otherwise it does not have any special status among the two-qubit gates.

The form of the interaction Hamiltonian of the quantum register dictates the time required for the physical realization of each of the gates. Typically, the two-qubit interactions are much weaker compared to those needed in manipulations of a single qubit. Thus CNOT or any other two-qubit gate takes considerably more time to realize than any of the one-qubit gates. Accordingly, minimization of the number of two-qubit gates, instead of the total gate count, is of prime importance in quantum compiling. Furthermore, the form of interactions may restrict the two-qubit gates to act only on spatially neighboring qubits. This calls for specialized gate decompositions. In addition to the minimal set of elementary gates, certain complicated multiqubit gates may also be included in the library. This may help the compiler to accelerate the implementation of the quantum algorithm.

A one-qubit gate $U \in SU(2)$ acting on the k^{th} qubit in an n-qubit register is represented by a unitary matrix

$$\tilde{U} = \underbrace{I \otimes \ldots \otimes I}_{k-1 \text{ times}} \otimes U \underbrace{\otimes I \ldots \otimes I}_{n-k \text{ times}}, \tag{3.1}$$

where $\otimes$ stand for the Kronecker product of matrices. For simplicity we omit below the qubits that are unaffected in the transformation. Accordingly, the matrix representation for gate U is

$$U = \begin{pmatrix} a & b \\ -\bar{b} & \bar{a} \end{pmatrix}, \tag{3.2}$$

where a and b are two complex numbers satisfying $|a|^2 + |b|^2 = 1$. Another parametrization for the matrices in $SU(2)$ is obtained through its group theoretical structure

$$U = \exp\{i(a_1\sigma_x + a_2\sigma_y + a_3\sigma_z)\}, \tag{3.3}$$

where $\{a_i\}$ are real numbers and $\{\sigma_i\}$ are the generators of group $SU(2)$. We fix the basis for the two-state system such that the operator σ_z is diagonal. Furthermore we call the vectors corresponding to the eigenvalues 1 and -1 by $|0\rangle$ and $|1\rangle$, respectively. In this basis the matrix representations of the operators $\{\sigma_i\}$ are called the Pauli spin matrices:

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma_z = \begin{pmatrix} +1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{3.4}$$

In particular we define a one-parameter rotation:

$$R_{\mathbf{a}}(\theta) = e^{i(a_x\sigma_x + a_y\sigma_y + a_z\sigma_z)\theta/2}, \tag{3.5}$$

where θ stands for the rotation angle around the unit vector $\mathbf{a}$. Here we note that $R_{\mathbf{a}}(\theta)$ can be made diagonal by a similarity transformation

$$R_{\mathbf{a}}(\theta) = V_{\mathbf{a}} R_z(\theta) V_{\mathbf{a}}^{\dagger}, \tag{3.6}$$

where $V_{\mathbf{a}}$ stands for a unitary matrix depending only on the direction of the vector $\mathbf{a}$. Here, we note that the rotations about any single axis are additive

$$R_{\mathbf{a}}(\theta_1) R_{\mathbf{a}}(\theta_2) = R_{\mathbf{a}}(\theta_1 + \theta_2) \tag{3.7}$$

and for $\mathbf{a}\perp\mathbf{e}_x$ the rotation angle is reversed by conjugation with σ_x:

$$a_x = 0 \implies \sigma_x R_{\mathbf{a}}(\theta)\sigma_x = R_{\mathbf{a}}(-\theta). \tag{3.8}$$

The above identities play an important role in the elementary gate decompositions below.

When the rotation axis points towards any of the coordinate axes, the corresponding one-parameter rotation is called an elementary rotation. The matrix representations for the elementary rotations are

$$R_x(\theta) = e^{i\sigma_x\theta/2} = \begin{pmatrix} \cos\theta/2 & i\sin\theta/2 \\ i\sin\theta/2 & \cos\theta/2 \end{pmatrix}, \tag{3.9}$$

$$R_y(\theta) = e^{i\sigma_y\theta/2} = \begin{pmatrix} \cos\theta/2 & \sin\theta/2 \\ -\sin\theta/2 & \cos\theta/2 \end{pmatrix}, \tag{3.10}$$

$$R_z(\theta) = e^{i\sigma_z\theta/2} = \begin{pmatrix} e^{i\theta/2} & 0 \\ 0 & e^{-i\theta/2} \end{pmatrix}. \tag{3.11}$$

Any element in $U \in SU(2)$ can be accessed using at most three consequent rotations:

$$U = R_z(\alpha)R_y(\beta)R_z(\gamma), \tag{3.12}$$

where angles α, β, γ are called the Euler angles. Instead of z and y axes any other two rotation axes which are mutually orthogonal will qualify the above equation. This sets one requirement for the controllability of the Hamiltonian of the quantum register: it has to provide control over two independent generators of $SU(2)$.

In addition to the one-qubit gates, the universal gate library must include at least one two-qubit gate; almost any gate will qualify [53]. To keep the analysis of the circuits simple, CNOT is typically chosen to be the only two-qubit gate in the gate library. The matrix presentation for CNOT in basis $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$ is

$$U_{\text{CNOT}} = I \oplus \sigma_x = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \tag{3.13}$$

The CNOT is our first example of the family of controlled gates, see Fig 3.1. In the quantum circuit diagrams the qubits are denoted by horizontal lines and the gates as rectangles on them. The control nodes are marked by circles which are connected to the associated gate by a vertical line. The action of the CNOT is σ_x, *i.e.*, the logical `NOT` in the subspace spanned by $\{|10\rangle, |11\rangle\}$. In contrast, the subspace spanned by $\{|00\rangle, |01\rangle\}$ remains untouched. In general, the effect of the control nodes is to limit the gates to act only on a certain subspace. The nodes in the quantum circuit diagram can be white or black corresponding to the control qubit states $|0\rangle$ or $|1\rangle$, respectively. Thereafter we refer by C^kV to k-fold controlled one-qubit gate V. When applied to an n-qubit register,

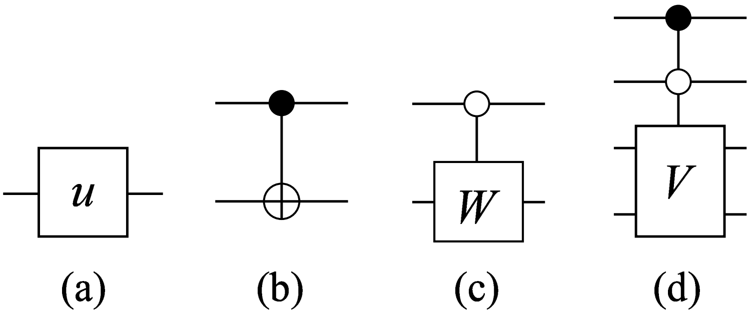

Figure 3.1: Quantum circuit symbols for (a) one-qubit gate , (b) CNOT, (c) controlled one-qubit gate, (d) two-fold controlled two-qubit gate. The white and black control nodes indicate that the gate acts non-trivially only in the subspace of $\mathbb{H}$ in which the corresponding qubit lies in the state $|0\rangle$ or $|1\rangle$, respectively.

this gate operates non-trivially in 2^{n-k}-dimensional target subspace consisting of those basis vectors for which the values of the controlled qubits match with those of the control nodes.

The universal gate library of CNOT and one-qubit gates efficiently implements any C^kV gate, see Ref. [52]. As an illustration of this, Fig. 3.2 shows how an array of elementary gates may emulate a controlled rotation. Basically, the structure of the gate array takes advantage of Eqs. (3.7) and (3.8). In general, for the implementation of a C^kV gate, where $V \in U(2)$, a quantum circuit of $O(n^2)$ elementary gates is required. However, the gate C^kW requires only $O(n)$ gates provided that $W \in SU(2)$. One possibility to implement a gate C^kU with more than one target qubit in U is to first decompose gate U. Decomposition reduces it into array of elementary gates each having k control nodes. In the second step they can be implemented as described above.

We say that an elementary gate array which implements a certain unitary transformation U is efficient if the number of degrees of freedom in U and the number of gates in the gate array are on the same order. Furthermore we use the product symbol for matrices with indices to denote certain gate decompositions. In these products the order of the matrices is always taken to be from left to right.

3.2 Reversible implementation of arithmetic functions

The permutation matrices have a special role among the unitary matrices since they represent reversible digital computation. Particularly, they provide an implementation for the reversible arithmetic functions, such as adding and multiplying [64, 66, 67]. This is important since quantum computation is always reversible; by definition the inverse $U^{-1} = U^\dagger$ exists for any unitary operator U. In other words, the output for each input is uniquely determined, and vice versa.

Let us consider an example. A two-qubit register can contain a quantum state, which

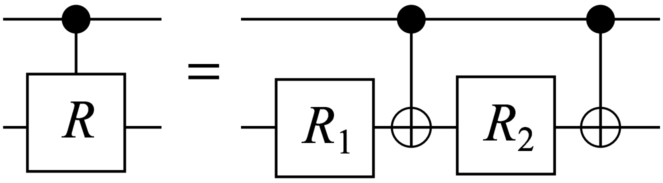

Figure 3.2: Implementation of a controlled one-parameter rotation R using the elementary gates. For example, choose $R_1 = R_z(\alpha/2)$ and $R_2 = R_z(-\alpha/2)$to obtain $R = R_z(\alpha)$. Here we have assumed the rotation axis to be perpendicular to x axis.

Table 1: Truth table for ADD 1 (MOD 4)

output \ input	0	1	2	3
0	0	0	0	1
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0

is a superposition of four basis vectors $|00\rangle$,$|01\rangle$,$|10\rangle$, and $|11\rangle$. Let us relabel them as $|0\rangle$,...,$|3\rangle$. In this basis, an adder that increases the value stored in the register by one corresponds to a matrix

$$U_{\mathrm{ADD}\,1} = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \tag{3.14}$$

To make the adder reversible we have taken the modulo four of the result. Say that the register is initially in state $|i\rangle$. The matrix $U_{\mathrm{ADD}\,1}$ transforms it into state $|i+1 \pmod 4\rangle$, *i.e.*, the unitary transformation works like the classical logic circuit whose truth table is represented in Table 1. The gate array implementing the algorithm U_{ADD1} is simple — it involves only one CNOT.

The arithmetic functions [68] are the basic ingredients of quantum algorithms almost similarly as they are for classical computations; only the superposition principle makes a difference. It allows the quantum computers to simultaneously evaluate the same function for several inputs using a single processor, which is classically impossible. The classical digital computers rely on the fast and reliable evaluation of long sequences of elementary operations one by one; parallelism can be obtained only by employing several processors.

Shor's factorization algorithm [29] is based on the quantum Fourier transformation (QFT) and the evaluation of the modular exponential function $f(x) = a^x \pmod N$, where a, x, and N are integers. The quantum computer evaluates $f(x)$ for all the integer values of x from 0 to $2^n - 1$, where the value of n is on the order of several hundreds. This is massive parallelism which is not achievable without a quantum computer. The QFT helps one to analyze the resulting output state and deduce the period of the function $f(x)$.

The complexity of the gate array needed for modular exponential function sets the requirements on computational resources of Shor's algorithm. Publication [**II**] discusses the structure of Shor's algorithm and presents a detailed method to construct the elementary gate array for it. In addition to the algorithm discussed there, several other techniques to implement a modular exponential function are represented in literature. Reference [69] gives a recent review of them. Besides, Draper *et al.* [70–72] discuss effective implementations of an adder, which is an important ingredient of the exponential function.

3.3 Building blocks of a quantum computer

The set of operations available on a quantum computer is larger than that available on a classical one. The following sections are devoted to discussion about gates which appear only in quantum computers, that is, gates that generate entangled or superposed states. Let us approach this enormous subject through a few important examples especially considering uniformly controlled gates. For them we know several applications and, most importantly, an elementary gate array which efficiently implements them. Due to the efficient implementation these particular gates should be considered as a part of a quantum compiler — tools for implementing quantum algorithms [73].

3.3.1 Quantum Fourier transformation

Quantum Fourier transformation is the best known quantum gate, or low-level algorithm, which is known to involve only a polynomial number of elementary gates in the number of qubits n. The elements of the matrix representing an n-qubit QFT are given by

$$U_{l,m} = \frac{1}{\sqrt{2^n}} e^{2\pi i l m/2^n}. \tag{3.15}$$

References [43, 74] present a decomposition of QFT into one- and two-qubit gates following the construction of the classical Fast Fourier Transformation (FFT) algorithm [68]. Very recently Tucci [75] found that this construction can also be produced by using a more general method, namely, Cosine-Sine matrix decomposition which can also be utilized to find implementations of unstructured n-qubit gates.

3.3.2 Uniformly controlled gates

A sequence of consequent controlled gates with slightly different control node configurations appear frequently in the circuit diagrams of quantum algorithms. Let us call a sequence of 2^k such gates, each having k control nodes, a uniformly controlled U gate,

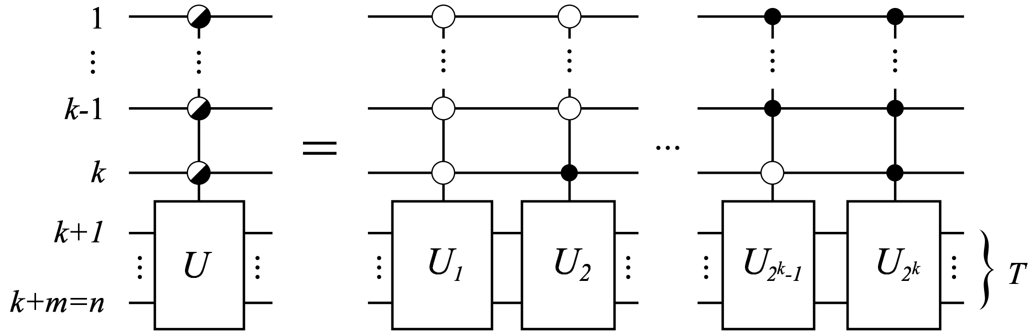

Figure 3.3: k-fold uniformly controlled m-qubit gate, $F_T^k\,(U(2^m))$, stands for a sequence of k-fold controlled gates U_i. Each of the gates acts on the set of target qubits T. Here $U_i \in U(2^m)$, where $i = 1, \ldots, 2^k$.

see Fig. 3.3. The gate shown acts on an n qubit register. Thus it has $m = n - k$ target qubits which we denote collectively by T. Let us denote a gate of this kind by a symbol $F_T^k\,(U(2^m))$.

The concept, quantum circuit symbol, and an efficient implementation of a uniformly controlled gate was for the first time introduced in Publication [**V**] in the context of uniformly controlled rotations. After that it has been utilized in decomposing general n-qubit gates in [**VII**], [76], and [77], and in preparation of quantum states in [**VI**], and [76],[**VII**]. Bullock *et al.* have generalized uniformly controlled gates for a quantum register which is built of qudits, d-level ($d > 2$) quantum systems [78]. The methods to implement uniformly controlled z rotations are also closely related to the earlier work by Bullock and Markov [79], and Schuch and Siewert [80].

Reducing the number of control and target qubits

Above $F_T^k\,(U(2^m))$ is defined straightforwardly as a sequence of the gates $\mathrm{C}^k U_i$. The sequence consists of all possible control node combinations, thus involving 2^k $\mathrm{C}^k U_i$ gates in total. Clearly this definition of $F_T^k\,(U(2^m))$ does not provide an economical implementation for it. To efficiently implement $F_T^k\,(U(2^m))$ in terms of CNOTs and one-qubit gates we need methods to manipulate the uniformly controlled gates.

In literature two different techniques are presented to manipulate the uniformly controlled gates: Cosine-Sine Decomposition [**V**] (CSD) and the quantum multiplexor [76] (QM). The effect of CSD is to produce a decomposition which involves gates with a reduced number of target bits. Similarly, the effect of QM is to produce a decomposition which involves gates with a reduced number of control nodes, see Fig. 3.4.

Tucci [81] was the first to consider CSD [82] in the context of quantum computation. Bullock discussed its relation to the Kheneja-Glaser decomposition in Ref. [83]. CSD of

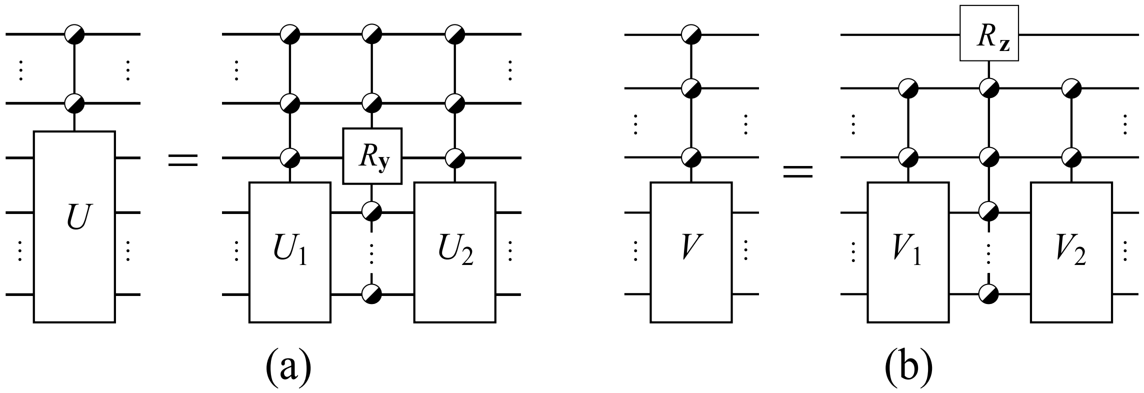

Figure 3.4: Methods to manipulate a uniformly controlled gate: (a) Cosine-Sine decomposition, (b) Quantum multiplexor.

a unitary $2^n \times 2^n$ matrix U is

$$U = \begin{pmatrix} u_{11} & 0 \\ 0 & u_{12} \end{pmatrix} \begin{pmatrix} c & s \\ -s & c \end{pmatrix} \begin{pmatrix} u_{21} & 0 \\ 0 & u_{22} \end{pmatrix}, \tag{3.16}$$

where u_{jk}, $j, k \in \{1, 2\}$ are $2^{n-1} \times 2^{n-1}$ unitary matrices. The real diagonal matrices c and s satisfy $s^2 + c^2 = I$ that justifies the name Cosine-Sine decomposition. The matrices $u_{11} \oplus u_{12}$ and $u_{21} \oplus u_{22}$ correspond to the rightmost and leftmost uniformly controlled gates in rhs of Fig. 3.4(a), respectively. The central matrix involving s and c parts corresponds to the uniformly controlled y-rotation. Reference [82], for example, explains details of evaluating the numerical values of the matrices in the decomposition. One should note that the matrices in CSD are not unique. This leaves room for optimization since for properly chosen matrices the resulting elementary gate sequences may simplify, see Ref. [75].

Quantum multiplexor is a clever trick to eliminate the control nodes from a uniformly controlled gate. The QM circuit then takes the form

$$\begin{pmatrix} v_1 & \\ & v_2 \end{pmatrix} = \begin{pmatrix} b & \\ & b \end{pmatrix} \begin{pmatrix} d & \\ & d^\dagger \end{pmatrix} \begin{pmatrix} a & \\ & a \end{pmatrix}, \tag{3.17}$$

where $v_1 \oplus v_2$ is the matrix presentation of the decomposed $F_T^1(V)$ gate and a and b are the matrix representations for the gates V_1 and V_2 in Fig. 3.4(b). The central matrix $d \oplus d^\dagger$, where d is a unitary diagonal matrix of size $2^{m-1} \times 2^{m-1}$, corresponds to the uniformly controlled z-rotation. All the matrices in the decomposition are unitary. Shende *et al.* [76] have introduced a method for determining the matrices of the right side in Eq. (3.17) for a given $F_T^k(V)$ gate.

When employed together, the CSD and QM techniques reduce any $F_T^k(U(2^m))$ gate into several uniformly controlled one-qubit gates. Below we consider how to implement those gates using a library of elementary gates.

Uniformly controlled one-qubit gates

Let us construct an elementary gate circuit for a uniformly controlled one-qubit gate. Consider the controlled one-parameter rotations, $F_t^k(R)$, separately since they need less gates to implement compared to general $F_t^k(U(2))$ gates. In $F_t^k(R)$ the rotation angle may vary, but the rotation axis is the same for each of the subrotations. The decompositions are designed assuming that the rotation axis is perpendicular to the x axis. This is due to the structure of CNOT: $I \oplus \sigma_x$. However, the choice of the axis of rotation is virtual — the rotation axis can be changed straightforwardly using the result of Eq. (3.6).

The elementary gate decompositions for the uniformly controlled gates are found through recursive technique involving reflections of the gate sequences. This leads to gate arrays whose properties can be described by the binary reflected Gray codes [63, 84, 85], or by the ruler function given by Sloane's sequence A001511 [86]. The ruler function gives the changing bit in the binary reflected Gray code, where the bit string i_g is obtained from the binary representation $i_{\rm b}$ of the number i as $i_g = i_{\rm b}\, \mathrm{XOR}\, (i_{\rm b}/2)$. This particular coding scheme was patented by Frank Gray [85] in 1953 for a communication system for railways. The useful property of the Gray codes is that the adjacent bit strings differ only in single bit, by definition.

A gate $F_t^k(U(2))$ decomposes into gates involving less control nodes efficiently, see Fig. 3.5. The decompositions shown in Figs. 3.5(a) and (b) strongly resemble each other although they are based on different mathematical principles. Publication [**V**] discusses the derivation of an elementary gate decomposition of $F_t^k(R)$ gate. There the main point is that by sandwiching a gate $F_t^k(R)$ with two CNOTs selectively reverses half of the rotation angles. Consequently the decomposition shown in Fig. 3.5(a) leads to a linear system of equations for the rotation angles, which always has a solution. Publication [**VII**] discusses the implementation of the uniformly controlled one-qubit gate. The implementation takes advantage of tuning of the eigenvalues of 2×2 matrix U. The tuning is performed by multiplying U by a diagonal matrix Γ. The eigenvalues of the 2×2 matrix $A = \Gamma U$ can be solved analytically, which fixes the matrix Γ. The analytic solution should be possible, even if tedious, also for 4×4 matrices, *i.e.*, for uniformly controlled two-qubit gates. Arising out of this, the interesting question for the future work is if this technique can be generalized for uniformly controlled m-qubit gates, where $m \geq 2$.

The recursive application of decompositions represented in Fig. 3.5 yield an elementary gate decomposition of $F_t^k(R)$ and $F_t^k(U(2))$. Figure 3.6 presents quantum circuits obtained for gates $F_4^3(R)$ and $F_4^3(U(2))$ using three-level recursion. In general, the decomposition of a gate $F_t^k(U(2))$ includes an alternating sequence of 2^k one-qubit gates and $2^k - 1$ CNOTs which we denote by $\tilde{F}_t^k(U(2))$. Moreover, the implementation involves a cascade of k distinct uniformly controlled z rotations which corresponds to a single diagonal $(k+1)$-qubit gate Δ_{k+1}, see Sec. 3.3.3. The implementation of the diagonal

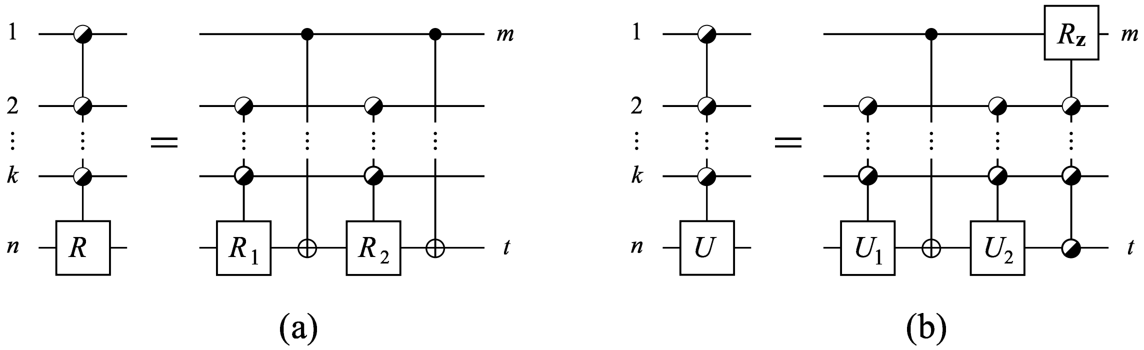

Figure 3.5: Decomposition of a uniformly controlled one-qubit gate. (a) One parameter rotation, (b) general one-qubit gate $U \in SU(2)$.

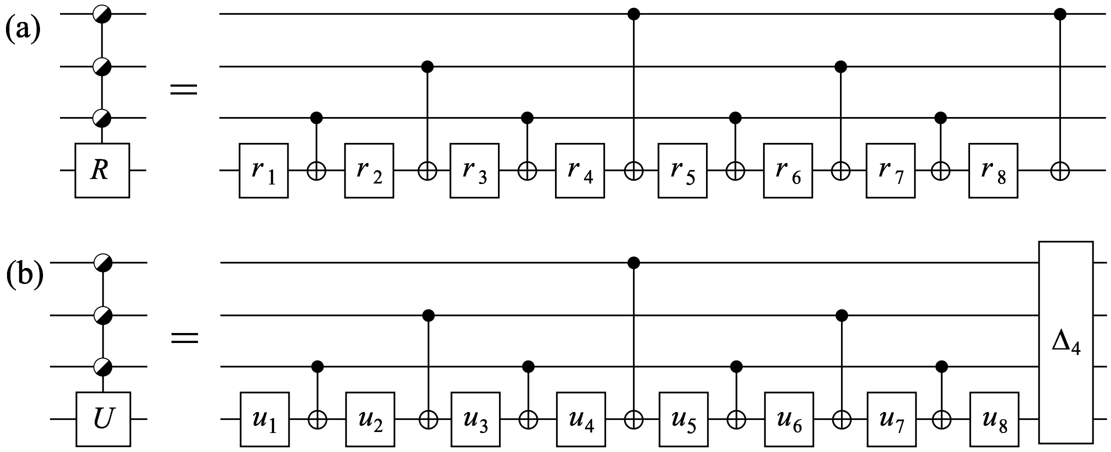

Figure 3.6: Quantum circuit realizing a three-fold uniformly controlled one-qubit gate (a) one-parameter rotation, (b) general one-qubit gate. Here r_i stands for a one-parameter rotation and in u_i is a general one-qubit gate. The alternating sequence of CNOTs and u_i gates is denoted by $\tilde{F}_4^3\,(U(2))$ while the gate Δ_4 corresponds to a diagonal 16×16 unitary matrix. Its implementation is discussed in the text.

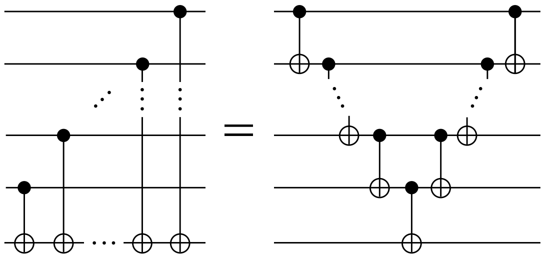

Figure 3.7: CNOT cascade which can be efficiently implemented using nearest-neighbor CNOTs.

part of the gate sequence can often be circumvented by merging it to the adjacent gate. Compared to the gate $F_t^k(U)$ the implementation of the gate $F_t^k(R)$ requires fewer elementary gates. Namely, an alternating sequence of 2^k CNOTs and 2^k one-qubit rotations implements the gate.

Linear chain of qubits with nearest-neighbor couplings

In the practical realization of a quantum computer, the spatial arrangement of the quantum register or other reasons may limit the interactions between the qubits. Let us consider a quantum register whose topology corresponds to that of a linear chain and which allows the gates to act only on nearest-neighboring qubits. This topology turns out to be amenable for implementing a uniformly controlled gate. This may have important consequences for experimentally realizing quantum computing.

The quantum circuit presented for a uniformly controlled gate can be translated efficiently into an array of nearest-neighbor gates. The technique is based on the circuit identity shown in Fig. 3.7. The strategy is to modify the decomposition shown in Fig. 3.5 by inserting an identity in the form of a CNOT cascade and its inverse, a similar cascade, into the circuit next to the central CNOTs. The other of the cascades is absorbed into the following uniformly controlled gate. The remaining cascade, together with the central CNOT, can be efficiently implemented using nearest-neighbor CNOTs as illustrated in Fig. 3.8.

The complexity of the nearest-neighbor implementation depends on the relative order of the target and control qubits, and the order in which the control qubits are eliminated. An efficient strategy is to first eliminate the control nodes that are furthest apart from the target. Furthermore, for the gates with numerous control nodes, it is advantageous to use a sequence of swap gates to move the target qubit next to the center of the chain before the operation and back after it. A swap gate can be realized using three consecutive CNOTs [3].

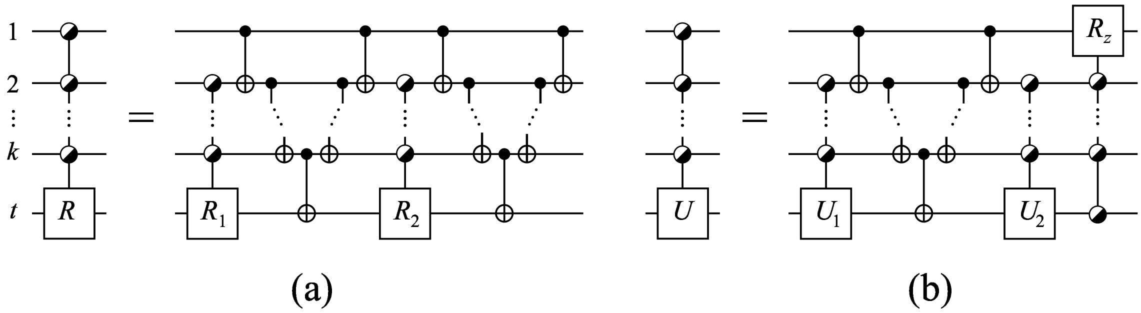

Figure 3.8: Reducing a uniformly controlled gate into nearest-neighbor gates: (a) uniformly controlled rotation and (b) general one-qubit gate. Here the circuit diagrams may also be mirrored horizontally.

Using this strategy a gate $\tilde{F}_t^{n-1}(U(2))$ can be implemented with at most

$$C_{U(2)}(n,s) = \frac{5}{6}2^n + 2n - 6s - \begin{cases} \frac{1}{3}, & n \text{ even} \\ \frac{5}{3}, & n \text{ odd} \end{cases} \tag{3.18}$$

nearest-neighbor CNOTs. Here $s = 1, \ldots, \lceil \frac{n}{2} \rceil$ is the distance of the target qubit t from the end of the chain. Figure 3.9(a) depicts the resulting circuit for the case $k = 4$ and $s = 1$.

Similar treatment for gate $F_t^{n-1}(R_{\mathbf{a}})$ yields a quantum gate array with

$$C_R(n,s) = \frac{5}{6}2^n + 3n - 6s - \begin{cases} \frac{4}{3}, & n \text{ even} \\ \frac{5}{3}, & n \text{ odd} \end{cases} \tag{3.19}$$

nearest-neighbor CNOTs. Figure 3.9(b) displays an example circuit for the case $k = 4$ and $s = 1$.

A Uniformly controlled one-qubit gate carries $3 \cdot 2^k$ degrees of freedom, and requires roughly the same number of elementary gates for its implementation. We conclude that arrays of nearest-neighbor CNOTs provide efficient implementations for $F_t^k(R)$ and $F_t^k(U(2))$ gates, and therefore for any uniformly controlled gate. In particular this can be utilized to efficiently implement unstructured unitary transformations as discussed below. Furthermore, the structure of the nearest-neighbor circuit allows several gate operations to be executed in parallel which may further reduce the execution time of the algorithm.

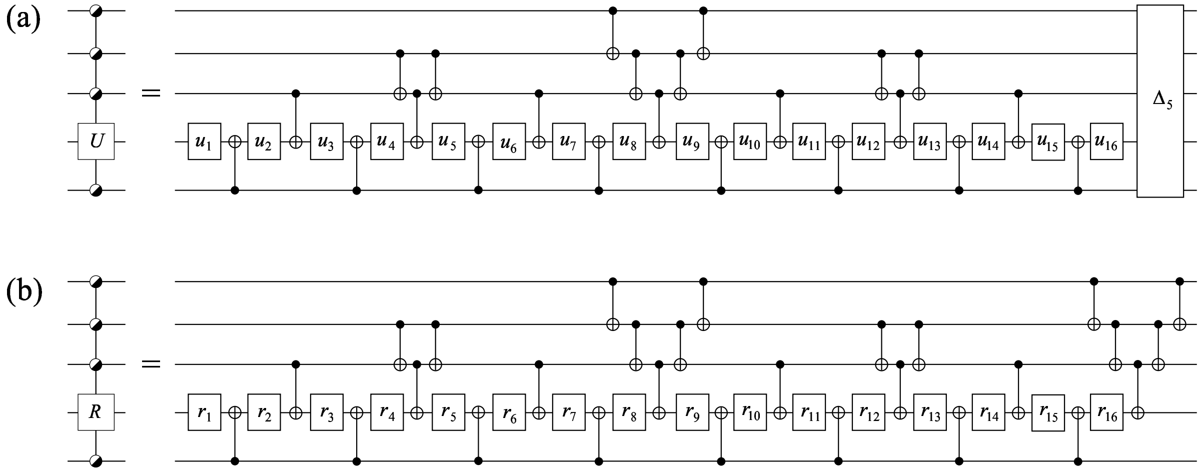

Figure 3.9: Implementation of three-fold uniformly controlled one-qubit gate: (a) general one-qubit gate and (b) one-parameter rotation. Here r_i gates are generic rotations in plane perpendicular to x-axis and u_i gates belong to $SU(2)$. A five-qubit diagonal gate is denoted by Δ_5, see Sec.3.3.3.

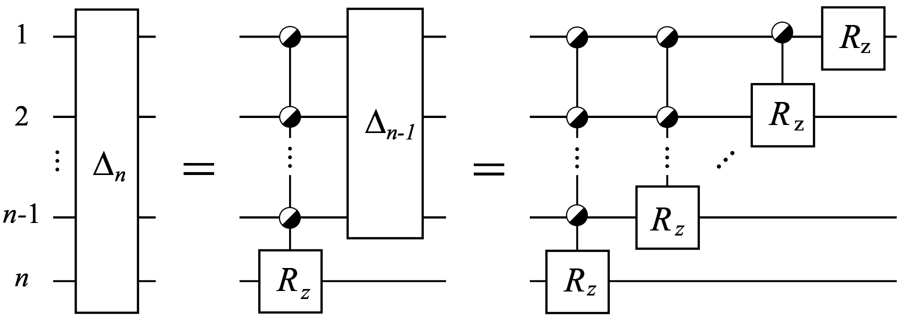

Figure 3.10: Quantum circuit for a diagonal quantum gate.

3.3.3 Diagonal quantum gate

Several gate decompositions involve parts which correspond to a diagonal unitary matrix, as explained above. A diagonal gate adjusts the phases of the state vector, and hence does not feature any practical properties alone. In contrast, when diagonal gate is connected to other gates, it starts to play a significant role as seen below.

Bullock and Markov [79] were the first to find the efficient implementation for an arbitrary diagonal gate using an array of CNOTs and elementary rotations about z axis. The array they propose for an n-qubit diagonal gate Δ_n consists of $2^n - 2$ CNOTs and $2^n - 1$ one-parameter rotations. Comparing this to $2^n - 1$, that is the number of degrees of freedom in the diagonal unitary matrix having determinant 1, we find that the gate array efficiently implements it. Schuch and Siewert posted their paper [80] on the preprint server four days after Bullock and Markov. Their solution to the problem resembles that given in Ref. [79] providing, however, a considerably different gate sequence, which is amenable for the parallel execution of the elementary gates.

The uniformly controlled rotations provide a framework which straightforwardly explains the operational principle of the optimal circuit for diagonal gates obtained by Bullock and Markov using other methods. Figure 3.10 illustrates the decomposition of a diagonal gate Δ_n into $F^i_{i+1}(R_z)$ gates. The first equality in Fig. 3.10 shows how $F^{n-1}_n(R_z)$ and Δ_{n-1} gates implement Δ_n gate. The recursive application of this decomposition yields a cascade of n uniformly controlled z rotations which exactly corresponds to a diagonal gate acting on n qubits.

3.3.4 State preparation

Here, the state preparation means a unitary transformation which converts a single known n-qubit quantum state $|a\rangle_n$ into another well specified state $|b\rangle_n$. Knill [87] has discussed a quantum gate array needed for this transformation and shown that it is of $O(n2^n)$ complexity. Recently, Shende *et al.* [88] found a minimal circuit which implements the state preparation for a two-qubit register. Publication [**VI**] describes an explicit method for constructing an elementary gate array of complexity $O(2^n)$ for the state preparation of an n-qubit register. Again, the implementation takes advantage of the uniformly controlled rotations and their gate-efficient implementation. Almost simultaneously with [**VI**] Shende *et al.* [76] come up with the same construction independently. Furthermore, Publication [**VII**] describes improvements to the construction and achieves a gate count which is only a factor of two away from the theoretical minimum.

In all the reported schemes the state preparation follows the same strategy; the state $|a\rangle_n$ is first transformed into one of the basis vectors $|e_1\rangle_n$ and then using the same strategy backwards from $|e_1\rangle_n$ to $|b\rangle_n$. The quantum circuit to transform a state $|a\rangle_n$ into $|e_1\rangle_n$ consists of a sequence of gate pairs [**VI**]

$$S_a = \prod_{i=1}^{n} \left[\left(F_i^{i-1}(R_y)\, F_i^{i-1}(R_z) \right) \otimes I_{2^{n-i}} \right]. \tag{3.20}$$

The effect of a gate pair $F_i^{i-1}(R_y)\, F_i^{i-1}(R_z)$ on the state $|a\rangle_i$ is to nullify half of the elements:

$$F_i^{i-1}(R_y)\, F_i^{i-1}(R_z)\, |a\rangle_i = |a'\rangle_{i-1} \otimes |0\rangle_1 . \tag{3.21}$$

Hence, each successive gate pair nullifies half of the elements of the state vector that have not yet been zeroed. Eventually, all the elements except one have been zeroed, and we have the desired transformation

$$S_a\, |a\rangle_n = |e_1\rangle_n \tag{3.22}$$

up to a global phase. Similarly we obtain $S_b\, |b\rangle_n = |e_1\rangle_n$. Combining these result we get

$$S_b^\dagger S_a\, |a\rangle_n = |b\rangle_n , \tag{3.23}$$

where $S_b^\dagger S_a$ is the desired transformation which maps $|a\rangle_n$ to $|b\rangle_n$.

The gate array producing the transformation $S_b^\dagger S_a$ can be improved [**VII**] by making on each of the gate pairs in Eq. (3.20) a replacement

$$F_n^{i-1}(R_y)\, F_n^{i-1}(R_z) = \Delta_i \tilde{F}_i^{i-1}(U(2)). \tag{3.24}$$

The reduction of the gate array comes from the fact that the gate Δ_i can be compensated by the neighboring gates, and thus requires no gates to implement. Figure 3.11 shows the circuit for this transformation. The entire circuit for transforming $|a\rangle_n$ to $|b\rangle_n$ requires $2 \cdot 2^n - 2n - 2$ CNOTs and $2 \cdot 2^n - n - 2$ one-qubit gates. The transformation is very efficient since a vector of unit length in 2^n-dimensional complex space with arbitrary global phase involves describes $2^{n+1} - 2$ real degrees of freedom.

3.3.5 Unstructured unitary transformations

An interesting result in quantum computation is that a relatively simple universal gate library exists. Universal gate library includes those elementary gates that are needed

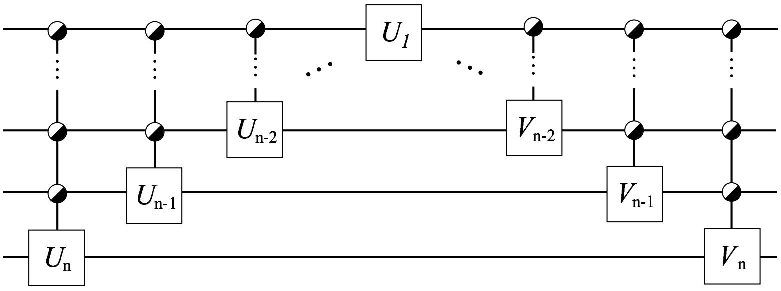

Figure 3.11: Quantum circuit for transforming an arbitrary n-qubit state vector $|a\rangle_n$ into a desired state vector $|b\rangle_n$. The resulting gates are of the form $\tilde{F}_i^{i-1}(U(2))$ which is efficient to implement, see Fig. 3.6.

for implementing an arbitrary transformation in the state space of the qubit register. Barenco *et al.* [52] presented the first detailed decomposition of an unstructured n-qubit gate U into elementary gates in 1995. The construction they presented requires $O(n^3 4^n)$ elementary gates. Afterwards, Knill [87] in 1995 and Cybenko [89] in 2001 presented decompositions which involve $O(n4^n)$ gates. However, we can easily argue that an array of $O(4^n)$ gates should implement U, since any unitary transformation of n qubits is represented by a unitary matrix U of size $2^n \times 2^n$, which has 4^n real degrees of freedom.

Before 2004, there was an annoying gap between the known decompositions and the known highest lower bounds for the gate counts. Although the techniques which eventually lead to minimal gate arrays were considered already in 1999 [81], the explicit gate array yielding even the $O(4^n)$ complexity was first presented in [**IV**]. Publication [**IV**] presents two improvements to Barenco's construction, which eventually results in $O(4^n)$ complexity. Although the presented implementation together with previous results confirms that asymptotic scaling for the construction is $\Theta(4^n)$ the actual gate count for the circuit is considerably high even for a small number of qubits.

A partial problem, the implementation of a two-qubit gate using minimal number of elementary gates was fully solved in Refs. [88, 90–92]. The main problem, the implementation of an unstructured n-qubit unitary transformation using minimal number of elementary gates was solved in [**V**], [76] ,[**VII**], [77]. The gate arrays that these papers suggest do not only achieve the asymptotically minimal complexity $O(4^n)$, but they also almost reach the theoretical minimum in the gate counts: 4^n in number of one-parameter rotations and $\lceil \frac{1}{4}(4^n - 3n - 1) \rceil$ [88] in number of CNOTs. Recently, Bullock *et al.* [78] generalized the result obtained for a register involving d-level systems, qudits, instead of qubits.

Some recently discovered quantum algorithms [93–95] embody unstructured unitary transformations as a part of their structure and hence call for techniques for their efficient implementation. However, from algorithmic viewpoint it is important to note that an unstructured n-qubit gate indeed requires a non-polynomial, $O(4^n)$, amount of gates to realize and thus cannot be considered solely an efficient implementation of any quantum algorithm.

QR Decompositions

Numerical matrix computation [50] is a field of mathematics that provides excellent tools to construct and manipulate quantum gate arrays. For example, theorem of QR decomposition proves that for each matrix A there exists unitary matrix Q and upper diagonal matrix R such that $A = QR$. Here Q may be be realized as a product of the Givens rotations [96]. Especially, if the matrix A is unitary, the matrix R is essentially an identity. Consequently, the sequence of Givens rotations yields a useful decomposition of a unitary matrix. Traditionally [52,89,97] a technique based on this principle was employed in quantum computation to find the elementary decomposition of an unstructured unitary matrix. Publication [**IV**] presents improvements to the traditional construction that eventually lead to the quantum gate decomposition of minimal complexity $O(4^n)$.

Let us outline how to find the sequence of Givens rotation matrices whose product implements any (special) unitary matrix $U \in SU(2^n)$. A Givens rotation ${}^iG_{j,k}$ is a two-level complex matrix which operates non-trivially only on two basis vectors $|e_j\rangle$ and $|e_k\rangle$. A numeric values of the elements of a matrix ${}^iG_{j,k}$ can be chosen such that it selectively nullifies the element on the i^{th} column and the j^{th} row of a matrix U when multiplied from left: $\tilde{U} =^i G_{j,k}U$. For example,

$${}^1G_{N,N-1}U = \begin{pmatrix} u_{1,1} & u_{1,2} & \cdots & u_{1,N} \\ \vdots & \vdots & \ddots & \vdots \\ u_{N-2,1} & u_{N-2,2} & \cdots & u_{N-2,N} \\ \tilde{u}_{N-1,1} & \tilde{u}_{N-1,2} & \cdots & \tilde{u}_{N-1,N} \\ 0 & \tilde{u}_{N,2} & \cdots & \tilde{u}_{N,N} \end{pmatrix},$$

where the elements of $\tilde{U}$ that differ from those of U are indicated with the tilde.

Applying ${}^1G_{N-1,N-2}$ to the modified matrix $\tilde{U}$ we can nullify the element $\tilde{u}_{N-1,1}$ and similarly the whole first column, except the diagonal element. The unitarity of the matrix U fixes its absolute value to unity. Furthermore, the Givens rotation can be defined such that the argument of the resulting diagonal element vanishes, *i.e.*, it obtains value 1. When further applied to the columns 2 to $N-1$ the process results in an identity matrix. Thus we obtain the factorization

$$\left(\prod_{i=1}^{2^n-1}\prod_{j=i+1}^{2^n} {}^{2^n-i}G_{j,j-1}\right)U = I \quad\Longleftrightarrow\quad U = \left(\prod_{i=1}^{2^n-1}\prod_{j=i+1}^{2^n} {}^{i+1}G^\dagger_{2^n-j+2,2^n-j+1}\right), \tag{3.25}$$

which yields the implementation of an arbitrary quantum gate provided that an elementary gate implementation of each of the Givens rotations is known.

Let us consider the most convenient basis which would allow us to implement the QR decomposition with the simplest gate array. In Publication [**IV**] we choose the order of

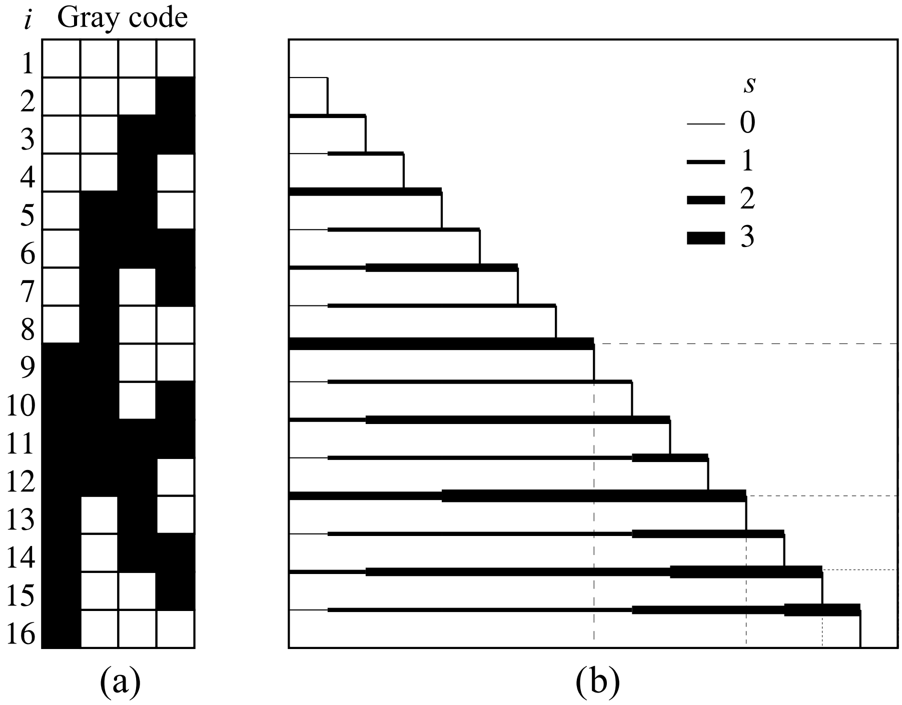

Figure 3.12: (a) Four bit Gray code. White squares stand for bit value 0 and black squares denote 1. (b) The number of control nodes needed for the Givens rotation nullifying the elements of the matrix U. The width of the line s between the matrix elements represents the number of control nodes required to zero the element below the line.

the basis vectors $\{|e_i\rangle\}$ of the quantum register according to the binary reflected Gray code. The special property of any Gray code ordered basis is that only one bit changes between the adjacent basis vectors $|e_i\rangle$ and $|e_{i+1}\rangle$. The important consequence of this is that the operations limited to the subspace spanned by $|e_i\rangle$ and $|e_i + 1\rangle$ take the form of $\mathrm{C}^{n-1}G$ gates, where $G \in SU(2)$. Consequently, each of the Givens rotations can be implemented using only one $\mathrm{C}^{n-1}G$ gate which saves gates compared to the earlier schemes [52,89,97].

Furthermore, we find that only a small fraction of the control nodes in the $\mathrm{C}^{n-1}G$ gates appears to be essential for the final result of the decomposition. If s control nodes are removed from a $\mathrm{C}^{n-1}G$ gate, the matrix representation of such an operation is no more two-level, but rather 2^{s+1}-level, *i.e.*, the matrix operates non-trivially to all pairs of basis vectors which satisfy the remaining control conditions. The strategy of eliminating the control nodes is following: once some element of U becomes zero in the diagonalization process, use control nodes in such a way that it does not mix with the non-zero elements.

The number of control nodes necessary for each of the gate depends on the position of

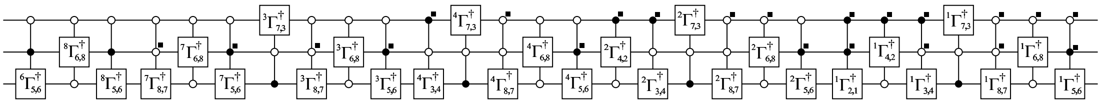
$^{6}\Gamma^{\dagger}_{5,6}$
$^{8}\Gamma^{\dagger}_{6,8}$
$^{8}\Gamma^{\dagger}_{5,6}$
$^{7}\Gamma^{\dagger}_{8,7}$
$^{7}\Gamma^{\dagger}_{6,8}$
$^{7}\Gamma^{\dagger}_{5,6}$
$^{3}\Gamma^{\dagger}_{7,3}$
$^{3}\Gamma^{\dagger}_{8,7}$
$^{3}\Gamma^{\dagger}_{6,8}$
$^{3}\Gamma^{\dagger}_{5,6}$
$^{4}\Gamma^{\dagger}_{3,4}$
$^{4}\Gamma^{\dagger}_{7,3}$
$^{4}\Gamma^{\dagger}_{8,7}$
$^{4}\Gamma^{\dagger}_{6,8}$
$^{4}\Gamma^{\dagger}_{5,6}$
$^{2}\Gamma^{\dagger}_{4,2}$
$^{2}\Gamma^{\dagger}_{3,4}$
$^{2}\Gamma^{\dagger}_{7,3}$
$^{2}\Gamma^{\dagger}_{8,7}$
$^{2}\Gamma^{\dagger}_{6,8}$
$^{2}\Gamma^{\dagger}_{5,6}$
$^{1}\Gamma^{\dagger}_{2,1}$
$^{1}\Gamma^{\dagger}_{4,2}$
$^{1}\Gamma^{\dagger}_{3,4}$
$^{1}\Gamma^{\dagger}_{7,3}$
$^{1}\Gamma^{\dagger}_{8,7}$
$^{1}\Gamma^{\dagger}_{6,8}$
$^{1}\Gamma^{\dagger}_{5,6}$

Figure 3.13: Quantum circuit equivalent to an arbitrary three-qubit quantum gate U up to a global phase. Here ${}^{i}\Gamma_{j,k}$ stands for Given's rotation matrix which nullifies element (i,j) of U with element (i,k) of U. The control nodes indicated with a black square on the upper right hand side corner are superfluous and may be omitted to decrease the complexity of the decomposition.

the nullified element in the matrix. Figure 3.12(a) illustrates the Gray code involved in a diagonalization process of a four-qubit gate $U \in SU(2^4)$. Figure 3.12(b) illustrates the number of control nodes necessary in the diagonalization. The total number of gates in the implementation depends on the number of the control nodes in each of the involved gates. In Publication [IV] we found that the number of k-fold controlled gates decreases exponentially with the number of control nodes. On the other hand, gate $\mathrm{C}^k V$ takes $O(n)$ gates to implement [52]. These results together imply that the gate array for an n-qubit unitary gate involves $O(4^n)$ elementary gates.

To calculate the number of elementary gates, we use the decompositions described in Ref. [52]. For large n, the leading contribution to the number of CNOTs is approximately 8.7×4^n. We note that neither one of the two techniques alone, the Gray code ordered basis vectors nor the elimination of the control nodes suffices to decrease the circuit complexity to $O(4^n)$. As a curiosity, the technique presented here has recently been generalized and adopted again to numerical matrix computation [98].

Cosine-Sine Decomposition

We are proceeding toward the elementary gate array with minimal complexity. The approach taken here is presented in detail in [**V**] and [**VII**]. In this approach, the Cosine-Sine decomposition is applied recursively to an unstructured n-qubit gate. This method yields a gate array

$$U(2^n) = F_n^{n-1}\left(U(2)\right) \prod_{i=1}^{2^{n-1}-1} F_{n-\gamma(i)}^{n-1}\left(R_y\right) F_n^{n-1}\left(U(2)\right), \tag{3.26}$$

where γ is the ruler function [86].

In publication [**V**] we decompose each of the $F_n^{n-1}\left(U(2)\right)$ gates starting from the left as

$$F_n^{n-1}\left(U(2)\right) = F_n^{n-1}\left(R_z\right) F_n^{n-1}\left(R_y\right) F_n^{n-1}\left(D\right), \tag{3.27}$$

where D is a 2×2 diagonal unitary matrix. For each $F_n^{n-1}\left(U(2)\right)$ gate the rightmost $F_n^{n-1}\left(D\right)$ that emerges from the decomposition can be merged into the neighboring

$F^{n-1}_{n-\gamma(i)}\left(R_y\right)$ gate

$$F^{n-1}_{n-\gamma(i)}\left(R_y\right)F^{n-1}_{n}\left(D\right) \quad\longrightarrow\quad F^{n-1}_{n-\gamma(i)}\left(R_y\right)F^{n-1}_{n-\gamma(i)}\left(R_z\right). \tag{3.28}$$

Furthermore, this merging requires that the next $F^{n-1}_{n}\left(U(2)\right)$ gate at right is multiplied by a diagonal gate. When applied to the entire sequence of gates, it takes the form

$$U(2^n) = F^{n-1}_{n}\left(R_y\right)F^{n-1}_{n}\left(R_z\right)\prod_{i=1}^{2^{n-1}-1}F^{n-1}_{n-\gamma(i)}\left(R_y\right)F^{n-1}_{n-\gamma(i)}\left(R_z\right)F^{n-1}_{n}\left(R_y\right)F^{n-1}_{n}\left(R_z\right)\Delta_n, \tag{3.29}$$

where Δ_n corresponds to an n-qubit diagonal matrix. The total complexity of the decomposition is $4^n - 2^{n+1}$ CNOTs and 4^n one-qubit rotations.

Publication [**VII**] presents a second variant of the decomposition. Starting from the last gate in Eq. (3.26), we write the diagonal part Δ_n separately:

$$F^{n-1}_{n}\left(U(2)\right) = \Delta_n\tilde{F}^{n-1}_{n}\left(U(2)\right). \tag{3.30}$$

The diagonal part Δ_n can then be merged with the neighboring $F^{n-1}_{n-\gamma(i)}\left(R_y\right)$ gate, which is transformed into a general gate of type $F^{n-1}_{n}\left(U(2)\right)$. Again, the diagonal part can be separated and merged into the next gate $F^{n-1}_{n}\left(U(2)\right)$. Continuing this process sequentially, we finally obtain

$$U(2^n) = \Delta_n\tilde{F}^{n-1}_{n}\left(U(2)\right)\prod_{i=1}^{2^{n-1}-1}\tilde{F}^{n-1}_{n-\gamma(i)}\left(U(2)\right)\tilde{F}^{n-1}_{n}\left(U(2)\right), \tag{3.31}$$

which is an efficient implementation for any unstructured unitary transformation. This decomposition involves in total $\frac{1}{2}4^n - \frac{1}{2}2^n - 2$ CNOTs and $\frac{1}{2}4^n + \frac{1}{2}2^n - n - 1$ one-qubit gates.

The $F^k_t\left(R\right)$ and $F^k_t\left(U(2)\right)$ gates can be implemented efficiently using gates nearest-neighbor gate arrays. This is the most interesting feature in the quantum circuits presented above, since due to that they are amenable to experimental realizations allowing only the nearest neighbor gates. To obtain the minimal nearest neighbor gate arrays it is favorable to have the target qubit of a uniformly controlled one-qubit gate as close to the center of the chain as possible. Consequently, we start the CS decomposition from the ends of the qubit chain, moving alternatingly towards the center. In this fashion, a general n-qubit gate can be implemented using at most

$$C_{U(n)} = \frac{5}{6}4^n - n2^n - 2n + \begin{cases} \frac{5}{6}2^n - \frac{5}{3}, & n \text{ even} \\ \frac{1}{2}2^n - \frac{1}{3}, & n \text{ odd} \end{cases} \tag{3.32}$$

nearest-neighbor CNOTs.

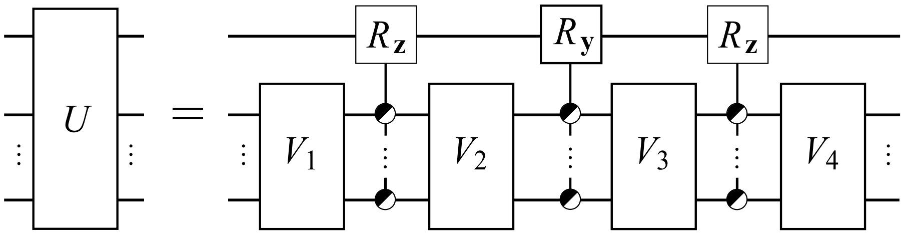

Figure 3.14: Combination of Cosine-Sine decomposition and quantum multiplexor yields a method to reduce the number of qubits the gates operates on.

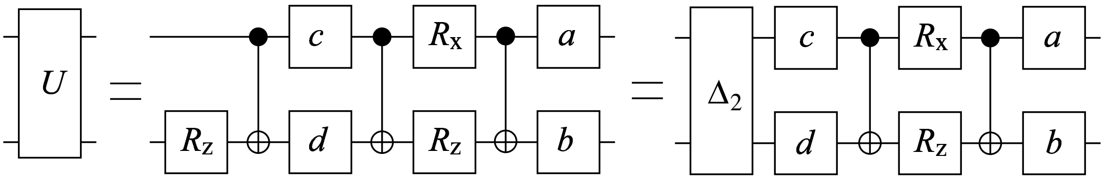

Figure 3.15: The minimal elementary gate implementation of a two-qubit gate [88].

NQ recursion

A combination of CSD and QM techniques provides the NQ recursion [76]. Each step of that recursion reduces a n-qubit gate into four $(n-1)$-qubit gates and three uniformly controlled rotations, see Fig. 3.14. To decompose an n-qubit gate the recursion is continued until the level of two-qubit gates is encountered. Here we note that each of the uniformly controlled rotations commutes with a diagonal gate. Thus all the two-qubit gates in the decomposition — except the rightmost one — can be implemented using two CNOTs, see Fig. 3.15. This leads to the complexity of $\frac{1}{2}4^n - \frac{3}{2}2^n + 1$ CNOTs and $\frac{9}{8}4^n - \frac{3}{2}2^n + 3$ elementary rotations.

Example: three-qubit gate

Let us briefly summarize how to find elementary gate arrays for unstructured n-qubit gates. Above we have discussed four different methods: improved QR decomposition, CS decomposition (CSD1), improved CS decomposition (CSD2), and NQ method. Figure 3.16 illustrates the quantum circuit diagrams obtained using the different methods for an arbitrary three-qubit gate. Tables 2 and 3 compare in the decompositions the number of gates needed for an n-qubit gate, where n=1,...,9.

Certain quantum gates that are likely to be useful in quantum computation comprise internal symmetries and can thus be implemented using only a polynomial number of elementary gates. For example, $O(n^2)$ gates are needed to implement a quantum Fourier transformation of n qubits [3]. Although the method presented apparently requires $O(4^n)$ elementary gates, it is still possible that using proper optimizations the gate array will

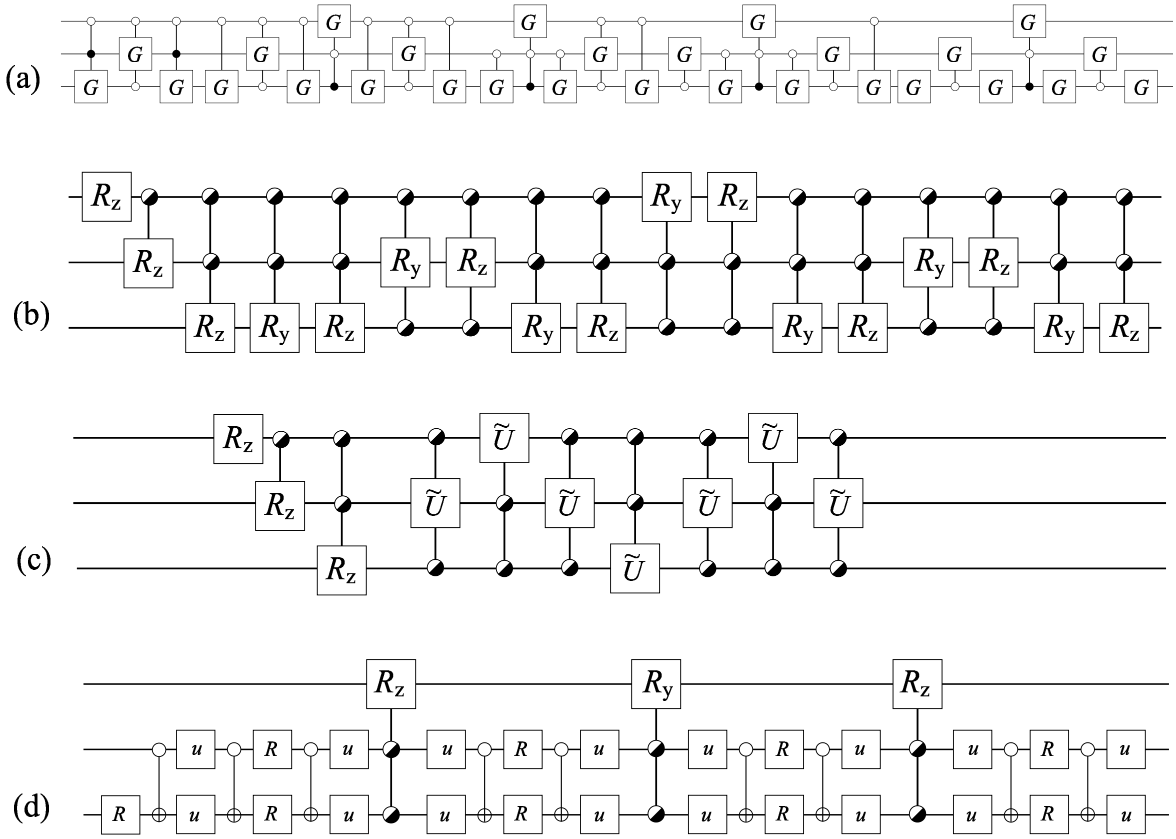

Figure 3.16: Quantum circuit for a three-qubit gate obtained using (a) QR decomposition, (b) CS decomposition, (c) improved CS decomposition, and (d) NQ method. Gates marked by G denote the Givens rotations, uniformly controlled R_y and R_z have their standard meaning, one-qubit R and u gates denote one-qubit rotations and general one-qubit gates, and the gates with symbol $\widetilde{U}$ stand for the uniformly controlled one-qubit gates without diagonal part.

appreciably simplify, and the result will resemble that of polynomial decompositions, as was found in Ref. [75].

Table 2: Comparison of the decompositions with respect to the number of CNOTs needed in different decompositions.

n	1	2	3	4	5	6	7	8	9
QR	0	4	64	536	4156	22618	108760	486052	2078668
CSD1	0	8	48	224	960	3968	16128	65024	261120
CSD2	0	4	26	118	494	2014	8126	32638	130814
NQ	0	3	21	105	465	1953	8001	32385	130305

Table 3: Comparison of the decompositions with respect to the total number of gates needed in different decompositions.

n	1	2	3	4	5	6	7	8	9
QR	1	14	136	980	7384	42390	208820	944280	4062520
CSD1	1	23	111	479	1983	8063	32511	130559	523263
CSD2	1	11	58	249	1016	4087	16374	65525	262132
NQ	1	10	54	262	1142	4758	19414	78422	315222

4 Shor's algorithm on a Josephson qubit register

Superconductivity [99] belongs to the most extraordinary phenomena in nature. It occurs in metals at very low temperatures, and also in certain ceramic materials at temperatures exceeding the boiling point of liquid nitrogen 77 K. Superconducting structures feature macroscopic quantum effects such as the quantization of the magnetic flux through a superconducting loop. Using nanolithographic techniques one may fabricate submicron scale superconducting electric circuits which involve weak links, *e.g.*, thin insulator gaps called Josephson junctions. They provide a potentially excellent housing for a quantum register [25].

A qubit implemented in a Josephson junction circuit is a two-state system whose basis states may correspond to distinct macroscopic variables: either the charge on superconducting islands or the flux through superconducting loops. The recent development on qubits operating on the charge regime is reported in Refs. [12, 16–20, 22] while the qubits mainly taking advantage of the flux degree of freedom are considered in [14, 100–102]. In addition, other promising scenarios such as phase qubits [15, 21] and tetrahedral qubits [103] have been suggested. We consider here a so-called inductively coupled Josephson charge qubit register. This model as well as others related realizations of quantum computing are carefully analyzed in Ref. [24].

Publication [**III**] studies the implementation of Shor's factorization algorithm using a Josephson charge qubit register. The presented study combines the algorithmic issues of Shor's algorithm to those of physical realization, bridging a gap between these parallel but rarely meeting research fields. The main problem is that the computational resources available for Shor's algorithm are strictly limited due to the strong decoherence that plagues Josephson junction circuits [24, 104]. To meet these requirements we employ a specialized implementation of the quantum algorithm which minimizes the number of qubits and the execution time needed. In particular, the arithmetic functions are implemented by methods that involve minimal number of extra qubits. To fight the decoherence we accelerate the execution of the quantum algorithms. This is performed by manipulating the algorithm on quantum gate level by replacing complicated elementary gate sequences by tailored two- and three-qubit gates.

4.1 Inductively coupled Josephson charge qubits

An ideal Josephson charge qubit register is a homogeneous array of mesoscopic superconducting islands, see Fig. 4.1. Each of the islands is coupled to a gate voltage V_g^i through capacitance C_g. In addition, each island is coupled to a magnetic field with flux Φ_i through a mesoscopic SQUID (Superconducting QUantum Interference Device) with identical junctions, each having the same Josephson energy $E_\mathrm{J}/2$, and capacitance $C_\mathrm{J}/2$. All the islands are coupled in parallel with an inductor L. In this setup, the Cooper pairs can tunnel coherently between an island and the superconducting electrode. The qubit

i is coded into the charge state of the island i. The basis states of a qubit correspond to either zero or one extra Cooper pairs residing on the island, denoted by $|0\rangle$ and $|1\rangle$, respectively.

The externally controllable variables of the model are the gate voltages $\{V_{\mathrm{g}}^i\}$ and the time-dependent fluxes $\{\Phi_i\}$ through each of the SQUID loops. The gate voltage V_g^i tunes the effective gate charge n_{g}^i of the island whereas the external magnetic flux controls the effective Josephson energy $E_{\mathrm{J}}(\Phi_i)$. The dynamical variables of the model are the flux φ through the inductor L and the time-integral of voltage $\dot{\phi}_i$ over the left junction of the i^{th} SQUID. The elementary circuit analysis [105] yields the Lagrangian of this qubit register

$$\begin{aligned}\mathcal{L} = \frac{1}{2}\sum_{i=1}^{n}&\left[\frac{C_{\mathrm{J}}}{2}\dot{\phi}_i^2 + \frac{C_{\mathrm{J}}}{2}(\dot{\phi}_i - \dot{\Phi}_i)^2 + C_{\mathrm{g}}(\dot{\phi}_i + \dot{\varphi} - V_{\mathrm{g}}^i)^2\right] \\ &- \frac{\varphi^2}{2L} + \frac{1}{2}\sum_{i=1}^{n}\left[E_{\mathrm{J}}\cos\left(\frac{2e}{\hbar}\phi_i\right) + E_{\mathrm{J}}\cos\left(\frac{2e}{\hbar}(\phi_i - \Phi_i)\right)\right]. \qquad (4.1)\end{aligned}$$

The quantization of the system yields the Hamiltonian for the low-energy spectrum of the n qubit-register [**III**]:

$$\mathcal{H} = -C\sum_{i<j} B_x^i B_x^j \sigma_y^i \otimes \sigma_y^j - \sum_i \left\{\frac{1}{2}B_z^i\sigma_z^i + \frac{1}{2}B_x^i\sigma_x^i\right\} \qquad (4.2)$$

where $\sigma_k^i = \overbrace{I \otimes \ldots \otimes I}^{i-1 \text{ times}} \otimes \sigma_k \otimes \overbrace{I \otimes I \ldots \otimes I}^{n-i \text{ times}}$, k=$\{x, z\}$. Above $B_x^i = E_{\mathrm{J}}\cos\left(\pi\frac{\Phi_i}{\Phi_0}\right)$ and $B_z^i = E_{\mathrm{C}}(1 - 2n_{\mathrm{g}}^i)$ describe the control parameters of the qubit register and the constant $C = \pi^2 L/\Phi_0^2\,(C_{\mathrm{qb}}/C_{\mathrm{J}})^2$ denotes the strength of the coupling between the qubits. Here $\Phi_0 = h/2e$ is the flux quantum and $C_{\mathrm{qb}} = C_{\mathrm{J}} + C_{\mathrm{g}}$ is the total capacitance of a qubit in the circuit. We have denoted charging energy of the island by $E_{\mathrm{C}} = \frac{(2e)^2}{C_J}$ and the effective gate charge by $n_{\mathrm{g}}^i = \frac{C_{\mathrm{g}}}{2e}\left(V_{\mathrm{g}}^i - \frac{\dot{\Phi}_i}{2}\right)$. The approach taken is to deal with the parameters $B_z^i(t)$ and $B_x^i(t)$ as dimensionless control parameters. Above, we have used the natural units, $\hbar = 1$. Furthermore we rescale the time such that $C = 1$.

The BCS gap Δ for the typical fabrication material thin-film aluminium is $\sim$ 200 μeV $\sim$ 2.5 K. However, niobium is known to provide a gap as high as 1.5 meV. The typical junction resistance is on range $1 - 100$ kΩ while capacitance is on the order of 1 fF. This gives the Coulomb charging energy $E_C/k_B \sim 4K$. The lowest relevant energy scale of the system is set by the thermal energy $k_{\mathrm{B}}T$ and the highest scale by the BCS gap Δ_{BCS}. A crucial assumption is that $k_{\mathrm{B}}T\ln N_{\mathrm{qp}} \ll E_{\mathrm{J}} \ll E_{\mathrm{C}} \ll \Delta_{\mathrm{BCS}}$, where N_{qp} is the number of quasiparticle modes. The lowest achievable operation temperature is tens of mK. Since N_{qp} is a small integer $\sim$ 5, the above chain of inequalities for E_J limits the critical current to tens of nA. The operation frequencies of this register would be in the range of several GHz, which is within the operation range of the available signal generators.

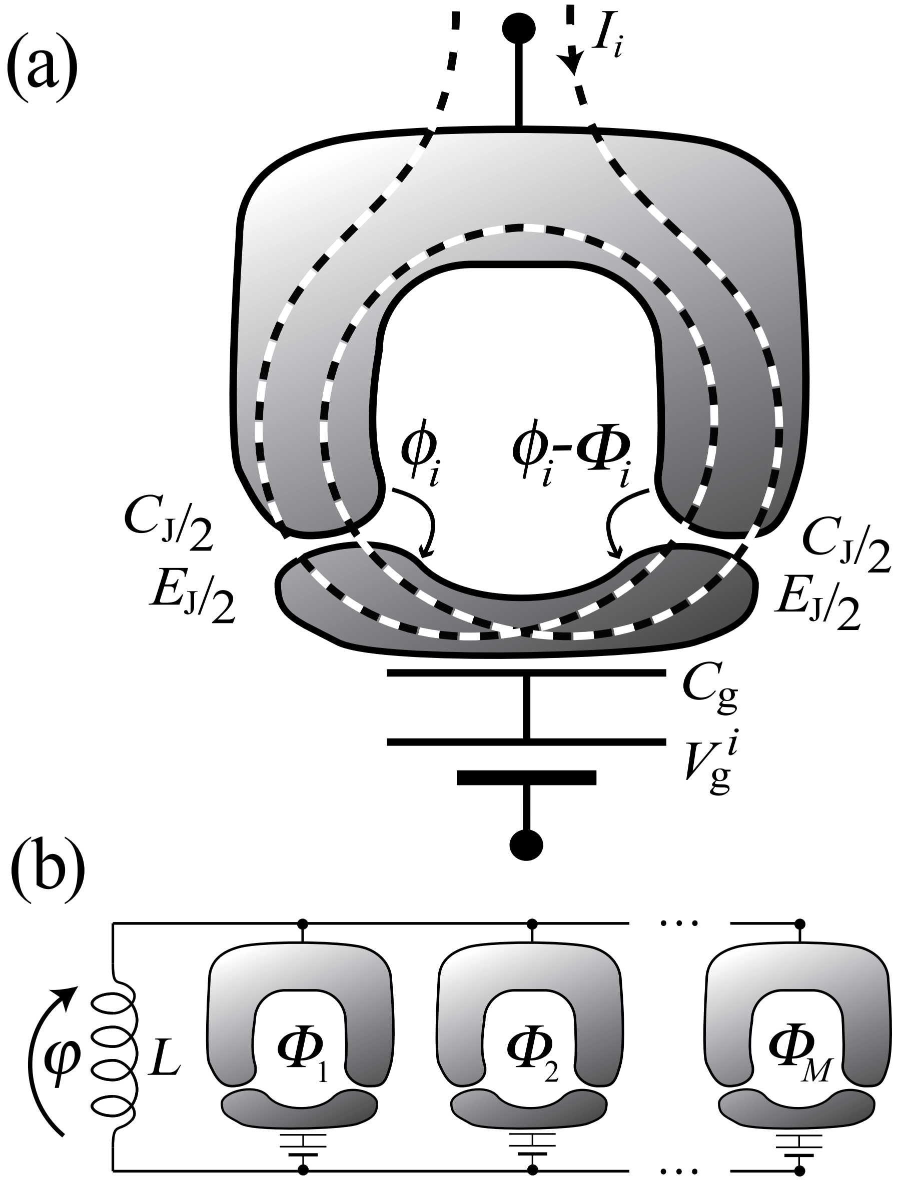

Figure 4.1: (a) Schematic of a Josephson charge qubit and its relevant parameters. The dashed line stands for an external coil which produces flux Φ_i through the SQUID loop, when current I_i is applied. (b) An array of Josephson charge qubits coupled in parallel with an inductor L.

The Hamiltonian of Eq. (4.2) is a convenient model for studying the construction of quantum algorithms since it allows full control over the one-qubit part and the coupling between the qubits. In particular, the coupling between any pair of qubits is available; there are no restrictions of nearest-neighbor gates. Furthermore, the total Hamiltonian can be set to zero by setting $B_z^i = B_x^i = 0$ for all i, thereby eliminating all temporal evolution. Equally, the control parameter values $B_x^i = 0$ for all $i \neq j$ eliminates all the interqubit couplings and allows a straightforward implementation of one-qubit operations on qubit j. However, if the parameter B_x is simultaneously nonzero for any two qubits, there will automatically emerge a coupling between them. The interaction Hamiltonian involves second order terms in control parameters, which complicates the finding of the control sequences analytically. Hence numerical methods are convenient and may even be necessary for finding the control-parameter sequences for multiqubit gates.

4.2 Control parameters of quantum gates

Numerical optimization provides us with a method to find control parameter sequences which produce approximate quantum gates. Publications [**I-III**] show how to solve the optimization task specified in Sec. 2.3 for the Josephson charge qubit register. In this scheme the parameter path $\gamma(t)$ is of the vector form

$$\gamma(t) = \left[B_z^1(t), \quad \ldots \quad , B_z^n(t); \;\; B_x^1(t), \quad \ldots \quad , B_x^n(t)\right]^T . \tag{4.3}$$

We let the control parameters $\{B_x^j(t)\}$ and $\{B_z^j(t)\}$ be piecewise linear functions of time, and set the parameter loop to start at the origin, *i.e.*, at the degeneracy point where no time development takes place. Consequently the parameter paths become polygonal. To fully specify the path only the $2n$ coordinates for the ν vertices of the polygon are needed. Let us denote these parameters collectively as X_γ. We have used $\nu = 4$ for the two-, and $\nu = 11$ for the three-qubit gates.

The piecewise linear parameter paths make this scheme experimentally viable since the parameters are adjusted in such a way that no fields are switched instantaneously. For practical applications it may turn out to be useful to try and describe the parameter paths using a collection of smooth functions instead of a piecewise linear parametrization.

The realization of a gate U is now reduced to finding the set of numerical parameters X_γ. First we note that the Hamiltonian of Eq. (4.2) allows us to tune the one-qubit generators σ_x and σ_z independently, and thus to realize any R_x or R_z rotation. Consequently, the realization of any one-qubit gate is straightforward using Eq. (3.12). The remaining optimization task is to find the parameters X_γ for multiqubit gates which minimize the error function

$$p(X_\gamma) = \|U - U_{X_\gamma}\|_F, \tag{4.4}$$

where U is the desired gate and U_{X_γ} is the gate obtained for control parameters X_γ. To evaluate the propagators in Eq. (2.9) we utilize the Taylor expansion. We find that the three-term expansion is fast to evaluate and yields enough precision for our purposes.

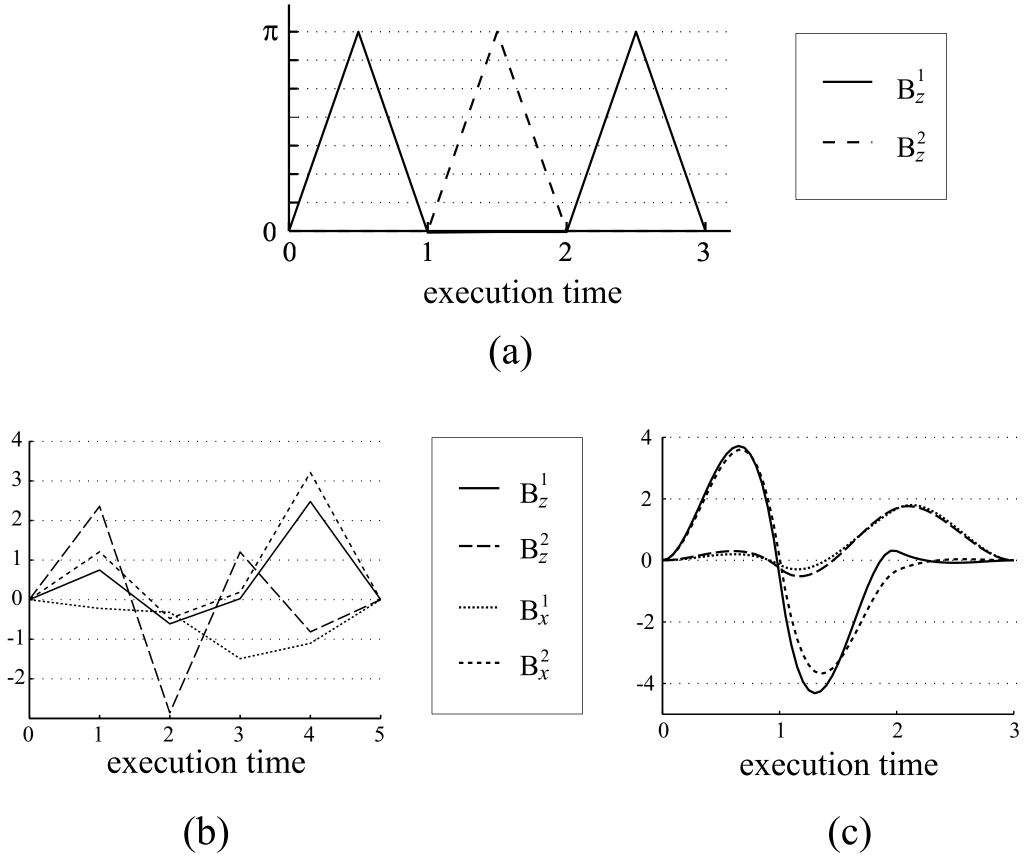

Figure 4.2: Parameter sequences implementing (a) Hadamard gate (b) CNOT up to a global phase. (c) Solution of the optimization problem using the path with qubic splines parametrization.

Figure 4.2 illustrates the control sequences for the Hadamard and CNOT gates. Analytic calculation provides the sequence for the Hadamard gate, while numerical optimization must be used for the CNOT. In this thesis we have considered the numerical optimization procedure through piecewise linear sequences. However, the parametrization is possible also using, for example, smooth spline sequences, see Fig. 4.2(c). The smooth parameter sequences may be experimentally more easily reachable than piecewise linear ones. The qubic spline path with three control points contains enough free parameters to describe any two-qubit gate. However, the resulting control parameter fields obtain higher values, compare Figs. 4.2(a) and (c). On the other hand, the maximum field values are limited by the physical parameters of the quantum register. Thus the execution time of the spline sequence is necessary not shorter than that of piecewise linear sequence.

For this minimization problem, the error-function landscape is rough consisting of many local minima, see Fig. 4.3. This is why any gradient-based minimization algorithm will encounter serious problems. Thus a robust polytope search algorithm [106] is employed for the minimization. Typical convergence of the search algorithm versus the

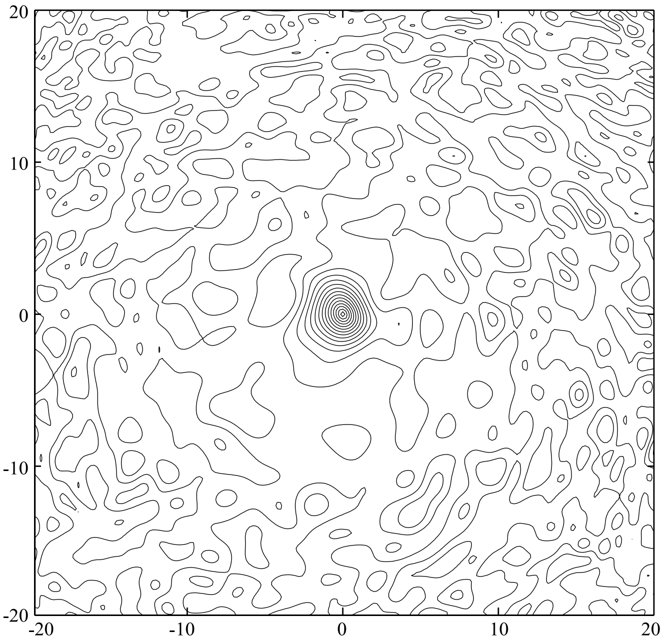

Figure 4.3: Planar cut of the error function space illustrating its roughness. The plane through the minimum point X_{min} has been chosen arbitrarily in the parameter space. The irregular shape of the landscape easily reveals the complexity of finding the global minimum and the reason why the gradient-based methods fail.

number of function evaluations is illustrated in Fig. 4.4. The natural question is the goal for the error function minimization. We have assumed that a sufficient accuracy for the gate operations is

$$\|U_{X_\gamma} - U\|_{\mathrm{F}} < 10^{-4}. \tag{4.5}$$

The presented minimization routine takes on the order of 10^6 function evaluations to reach this accuracy. Below a certain threshold level quantum error correction can, in principle, be utilized to reduce the accumulated errors. However, it is still an open question what the highest threshold level is or if there is any [107].

The above numerical optimization technique provides us with the realizations, not only for any two-qubit, but also for any three-qubit gates. Hence, the gates acting on many qubits need not necessarily be decomposed down to the level of CNOTs and one-qubit gates. This yields a possibility of compressing the required quantum gate array and thus accelerating the quantum algorithm, see Fig. 4.5. The utilization of three-qubit gates in implementation of a quantum algorithm results in a shorter execution time and smaller errors.

Let us illustrate the acceleration obtained using the three-qubit gates. We have chosen the time spent in traversing each edge of the polygonal path $\gamma(t)$ to be unity to make the different gate realizations comparable. In this scheme, any three-qubit gate requires

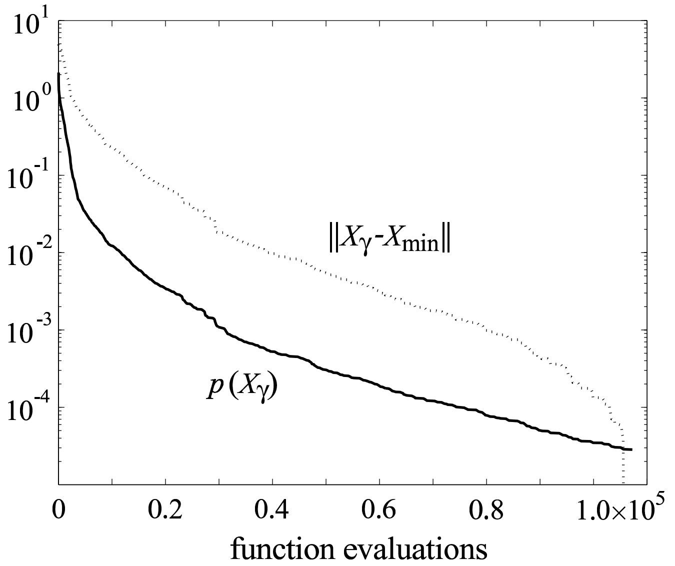

Figure 4.4: Convergence of the algorithm for the Fredkin gate. The error function values are indicated by the solid line and the distance of the parameter sequence from the numerical optimum X_{min} by the dashed line.

an integration path $\gamma(t)$ with 11 control points so that $6 \times 11 > 63$. This path takes 12 units of time to execute. Similarly, a two-qubit gate takes 5 units of time to execute. Table 4 summarizes our results by comparing the number of steps that are required to carry out a single three-qubit gate to the number of steps required for a sequence of two-qubit gates which implements the three-qubit gate. In all studied examples the optimized three-qubit gates provides a shorter execution time. The results in Table 4 are calculated assuming that the physical realization for any two qubit gates is available through some scheme similar to the one which is employed above and one-qubit gates are merged into two-qubit modules.

The possibility to implement nontrivial multiqubit gates in an efficient way may well turn out to be a crucial improvement in making quantum computing experimentally realizable. The acceleration of algorithm using multiqubit gates is discussed in Publications [**II**] and [**III**]. For further discussion on the implementation of non-standard gates as the building blocks for quantum circuits, see Refs. [80, 108, 109].

4.3 Shor's algorithm

Shor's algorithm is an important example of a quantum algorithm owing to its potential applications in breaking the otherwise secure RSA cryptosystem. Many widely applied methods of public-key cryptography are currently based on the RSA algorithm [110] which relies on the computational difficulty of factorizing large integers.

The strategy in factoring a number $N = pq$, both p and q being primes, using a

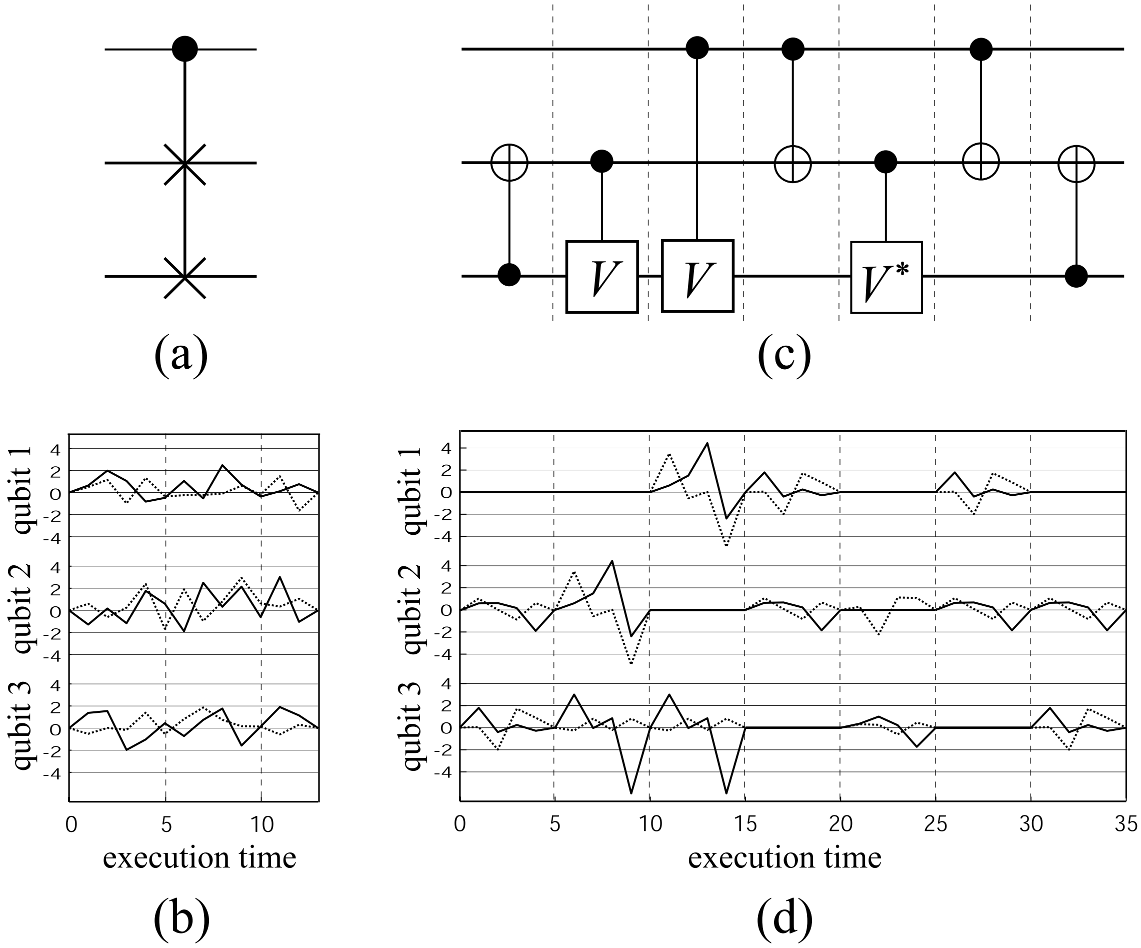

Figure 4.5: Acceleration of quantum computing via numerically optimized three-qubit gates on the Josephson charge qubit model. (a) The quantum circuit symbol of the Fredkin gate, and (b) its physical implementation by controlling all three qubits simultaneously. (c) The two-qubit gate decomposition of the Fredkin gate. Here $V = \sqrt{\sigma_x}$ and V^* stands for its Hermitian conjugate. (d) The physical implementation of the gate sequence; note that during each gate operation, one of the qubits is in the idle state. The vertical axis in figures (b) and (d) stands for the field amplitudes of the control parameters; the solid line describes the parameter B_z^i and the dashed line the parameter B_x^i, see text.

quantum computer relies on finding the period r of the modular exponential function $f(x) = a^x \pmod N$, where $0 < a < N$ is a random number coprime to N. With a high probability [29] one finds at least one prime factor of N is given by the greatest common divisor of the numbers $a^{r/2} \pm 1$ and N.

Shor's algorithm has the following stages: (1) Initializing the quantum registers. (2)

Table 4: Comparison of the execution times needed for various quantum gates. The execution time of three qubit gates, the second row, can be compared to the 12 edges, which is the time needed for the optimized implementation of a three-qubit gate.

gate	Toffoli	QFT	Fredkin	$U \in SU(2^3)$ [76]
number of two-qubit gates	3	3	5	14
execution time in edges	15	15	25	70

Generating the enormous superposition state. The number N takes $n = \lceil \log_2(N+1) \rceil$ bits to store into memory. Thus we need in a quantum register $|x\rangle_{2n}$ to store the integer range $0 \leq x \leq 2^{2n} - 1$. (3) Executing the algorithm U_f. This entangles each input value x with the corresponding value of $f(x)$:

$$U_f \sum_{x=0}^{2^n-1} |x\rangle|1\rangle = \sum_{x=0}^{2^n-1} |x\rangle|a^x \pmod{N}\rangle. \tag{4.6}$$

(4) The quantum Fourier transformation (QFT) is applied to the register $|x\rangle_{2n}$, which squeezes the probability amplitudes into peaks due to the periodicity $f(x) = f(x+r)$. (5) A measurement of the register $|x\rangle_{2n}$ finally yields an indication of the period r. A repetitive execution of the algorithm reveals the probability distribution which is peaked at the value $2^{2n}/r$ and its integer multiples of output values in the register $|x\rangle_{2n}$.

The evaluation of $f(x)$ on a quantum computer can be implemented using several different techniques [69], also efficiently in a linear nearest-neighbor qubit array [111]. To obtain the implementation which involves the minimal number of qubits, one assumes that the numbers a and N are hardwired in the quantum circuit. This means that the quantum circuit must be redesigned for each N uniquely.

The approach taken in [**II**] follows a technique called the longhand multiplication algorithm. It takes advantage of the fast powers trick, as well as the construction of a multiplier suggested by Beauregard [112], which in part employs the adder by Draper [70]. In this scheme, to extract the period of $f(x)$, we need at least two registers: $2n$ qubits for the register $|x\rangle_{2n}$ to store numbers x and n qubits for the register $|y\rangle_n$ to store the values of $f(x)$. In addition for the scratch space we need an $n+1$-qubit register $|z\rangle_{n+1}$ and one ancilla qubit $|a\rangle$. Thus $4n+2$ qubits are required in total. The register $|x\rangle_{2n}$ is initialized as $|0\rangle_{2n}$, whereas $|y\rangle_n = |1\rangle_n$.

Besides the quantum algorithm which is used to find r, a considerable amount of classical precomputing and postprocessing is required as well. However, all this computing can be performed in a polynomial time.

4.3.1 Factorizing number 21

To demonstrate the level of the complexity of the quantum circuit and the demands on the execution time, we explicitly present the quantum circuit needed for Shor's algorithm to factor the number $N = 21$.

Figure 4.6 illustrates the structure of the quantum part of the factorization algorithm for $N = 21$. We choose $a = 11$ and hardwire this into the quantum circuit. Since it takes $n = 5$ bits to store the number 21, we need $4n + 2 = 22$ qubits to implement the circuit.

The modular exponential function is decomposed into controlled modular multipliers acting on thirteen qubits. Each of them can be further decomposed into controlled modular adders as indicated in Fig. 4.6. They are implemented by controlled phase-shifts and QFT gates. A ten-qubit QFT breaks down to 42 two-qubit gates and one three-qubit QFT. Similarly, the six-qubit QFT can be equivalently implemented as a sequence of 18 two-qubit gates and one three-qubit QFT. In this manner we can implement the entire algorithm using only one-, two- and three-qubit gates. The control parameter sequence realizing each of them can then be found using the scheme outlined in Sec. 2.3. Two examples of the pulse sequences are also shown in the bottom insets of Fig. 4.6.

Following the above construction of the quantum circuit, the full Shor algorithm to factor 21 requires about 2300 three-qubit gates and some 5900 two-qubit gates, in total. Also a few one-qubit gates are needed but alternatively they can all be merged into the multi-qubit gates. If only two-qubit gates are available, about 16400 of them are required. If only a minimal set of elementary gates, say the CNOT and one-qubit rotations are available, the total number of gates is remarkably higher. In our scheme the execution time of the algorithm is proportional to the total length of the piecewise linear parameter path which governs the physical implementation of the gate operations. Each of the three-qubit gates requires at least a 12-edged polygonal path $\gamma(t)$ whereas two-qubit gates can be implemented with 5 edges. Consequently, on the order of 57100 edges are required for the whole algorithm if arbitrary three-qubit gates are available, whereas $\sim$ 82000 edges would be required for an implementation with only two-qubit gates.

Let us consider the experimental feasibility of our scheme. To factor the number 21, we need on the order of 10^4 edges along the control-parameter path. Currently, typical coherence time of single Josephson qubit is on the order of of 10^{-6} s. Let us optimistically assume that the n-qubit register could be fabricated without introducing any extra sources of decoherence. This sets the upper limit for the duration of each edge to be 10^{-10} s. Since our dimensionless control parameters in the examples are on the order of unity, the energy scale in angular frequencies must be at least on the order of 10^{10} s^{-1}. Typical charging energies for, say, thin-film aluminium structures may be on the order of 10^{-23} J which corresponds to 10^{11} s^{-1}. The ultimate limiting energy scale is the BCS gap, which for thin-film aluminium corresponds to an angular frequency of about 3×10^{11} s^{-1}. Based on these rough estimates, we argue that factoring the number 21 on Josephson charge qubits might be, in principle, experimentally accessible - although extremely demanding. The utilization of quantum error correction or some other of the related methods may significantly affect the presented analysis when applied

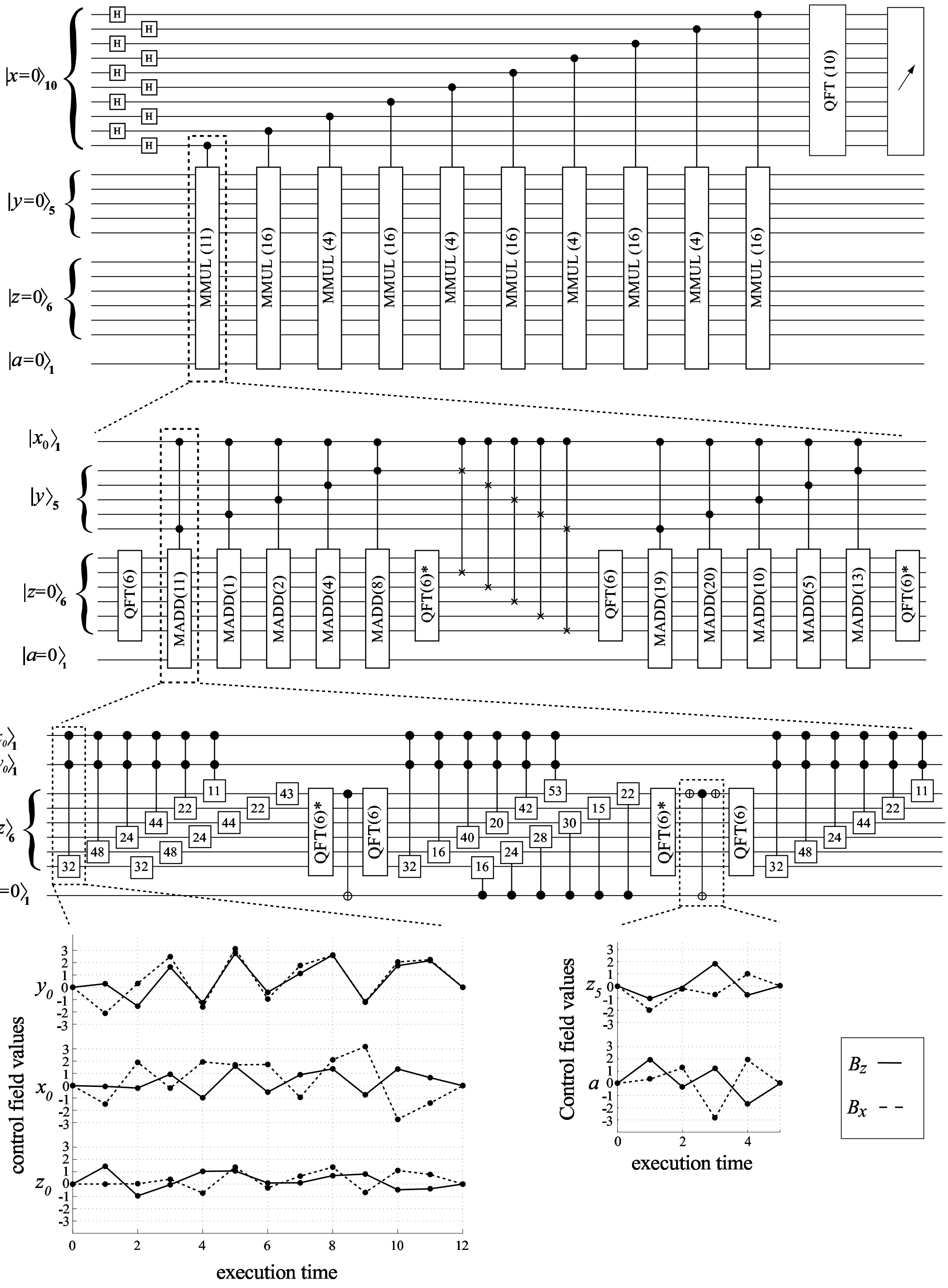

Figure 4.6: Quantum circuit for Shor's algorithm factoring the number 21 with the parameter value $a = 11$. The full circuit is shown topmost and the decompositions of the modular multiplier and adder blocks are indicated with dashed lines. We denote a phase-shift gate by a box with a single number ϕ in it meaning that the phase of the state $|1\rangle$ is shifted by $e^{2\pi i\phi/2^n}$ with respect to the state $|0\rangle$. Two examples of numerically optimized parameter sequences are also shown.

for hundreds of qubits.

Finally, let us consider a hacker with a quantum computer trying to break a message encrypted with RSA-155 coding which is widely employed in electronic communication. Let us note as background for this discussion that it has been demonstrated that 8000 MIPS (Million Instructions Per Second) years of classical computing power is needed to decrypt the code using the general number field sieve technique [4]. Since RSA-155 involves a 512-bit integer N, we would require on the order of 2000 qubits for our quantum computer. For the execution of the algorithm, decoherence time of tens of seconds is needed. This estimation agrees with Ref. [113] and apparently poses a huge experimental challenge. Thus Shor's algorithm does appear impractical for decrypting RSA-155 since it is considerably easier to build a huge classical computing system which yields the required computing power. In contrast, the quantum algorithm, owing to its polynomial scalability, provides the only known potentially feasible method to break RSA encryption involving 1024 or more bits.

5 Summary

This thesis proposes theoretical methods to efficiently implement unitary transformations on a quantum computer. As an application, Shor's algorithm on an inductively coupled Josephson charge qubit register is studied.

The complexity of elementary gate arrays of various generic n-qubit quantum gates is discussed. Introduced improvements [**IV**] to the former gate decompositions show that unstructured n-qubit gates belong to the complexity class $\Theta(4^n)$. In addition, a new versatile concept of uniformly controlled gates is introduced [**V**,**VII**]. For unstructured unitary transformations, a combination of the uniformly controlled gates and the Cosine-Sine matrix decomposition provides elementary gate sequences with almost minimal gate count. Furthermore, the uniformly controlled rotation can be used in the efficient preparation of a quantum state [**VI**],[**VII**]. The uniformly controlled gates are efficiently implemented in a qubit register similar to a one-dimensional chain of qubits with nearest-neighbor interactions [**VII**]. This makes the developed techniques suitable for experimental realizations.

The numerical optimization algorithm based on the polytope search has been shown to be useful for finding the control parameters for the Josephson charge qubit register [**I** – **III**]. The proposed scheme allows implementation of desired gates acting on up to three qubits. The three-qubit gates allow the merging of several one- and two-qubit gates together. This shortens the required execution time and thus accelerates the quantum algorithms.

The potential killer application for quantum computers is Shor's algorithm due to the possibility to break RSA encrypted messages. An explicit quantum gate construction implementing Shor's algorithm is considered in [**II**]. When applied to Shor's algorithm, the proposed acceleration scheme reduces the execution time roughly by a factor of two. This reduction of execution time may turn out to be crucial since the decoherence limits the time available for the execution of the algorithm. Still, the realization of a general factorization algorithm for a large integer N will be extremely challenging, at least in the near future.

In conclusion, this thesis presents a useful optimization scheme to find realization of quantum gates, suggests a method of optimization of quantum algorithms using multiqubit gates, puts forward a new family of gates which may be utilized as building blocks of a quantum compiler, and introduces decomposition techniques for general n-qubit gates which involve appreciably smaller elementary gate counts than previously reported.The results may be used in the quantum compilers to considerably shorten the gate sequences of quantum algorithms or, in general, to optimize the operation of any device that deals with the manipulation of coherent quantum states.

References

[1] D. F. Styer *et al.*, *Nine formulations of quantum mechanics*, American Journal of Physics **70**, 288 (2002).

[2] L. E. Ballentine, *Quantum Mechanics: a Modern Development* (World Scientific, Singapore, 1998).

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

[4] A. Galindo and M. A. Martin-Delgado, *Information and computation: Classical and quantum aspects*, Rev. Mod. Phys. **74**, 347 (2002).

[5] J. I. Cirac, L. M. Duan, and P. Zoller, in *"Experimental Quantum Computation and Information" Proceedings of the International School of Physics "Enrico Fermi", Course CXLVIII*, edited by F. D. Martini and C. Monroe (IOS Press, Amsterdam, 2002), p. 263.

[6] L. Vandersypen *et al.*, *Experimental realization of Shor's quantum factoring algorithm using magnetic resonance*, Nature **414**, 883 (2001).

[7] L. Vandersypen and I. L. Chuang, *NMR techniques for quantum control and computation*, quant-ph/0404064, (2004).

[8] D. Wineland *et al.*, *Experimental issues in coherent quantum-state manipulation of trapped atomic ions*, Journal of Research of the National Institute of Standards and Technology **103**, 259 (1998).

[9] C.-P. Yang, S.-I. Chu, and S. Han, *Possible realization of entanglement, logical gates, and quantum-information transfer with superconducting-quantum-interference-device qubits in cavity QED*, Phys. Rev. A **67**, 042311 (2003).

[10] W. G. van der Wiel *et al.*, *Electron transport through double quantum dots*, Rev. Mod. Phys. **75**, 1 (2003).

[11] J. H. Jefferson, M. Fearn, D. L. J. Tipton, and T. P. Spiller, *Two-electron quantum dots as scalable qubits*, Phys. Rev. A **66**, 042328 (2002).

[12] D. Vion *et al.*, *Manipulating the Quantum State of an Electrical Circuit*, Science **296**, 886 (2002).

[13] A. Shnirman, G. Schön, and Z. Hermon, *Quantum Manipulations of Small Josephson Junctions*, Phys. Rev. Lett. **79**, 2371 (1997).

[14] T. P. Orlando *et al.*, *Superconducting persistent-current qubit*, Phys. Rev. B **60**, 15398 (1999).

[15] Y. Yu *et al.*, *Coherent temporal oscillations of macroscopic quantum states in a Josephson junction*, Science **296**, 889 (2002).

[16] Y. Pashkin *et al.*, *Quantum oscillations in two coupled charge qubits*, Nature **421**, 823 (2003).

[17] T. Yamamoto *et al.*, *Demonstration of conditional gate operation using superconducting charge qubits*, Nature **425**, 941 (2003).

[18] Y. Nakamura, Y. A. Pashkin, and J. S. Tsai, *Coherent Control of Macroscopic Quantum State in a Single-Cooper-pair Box*, Nature **398**, 786 (1999).

[19] Y. Nakamura, Y. A. Pashkin, and J. S. Tsai, *Rabi Oscillations in a Josephson-Junction Charge Two-Level System*, Phys. Rev. Lett. **87**, 246601 (2002).

[20] Y. Nakamura, Y. A. Pashkin, T. Yamamoto, and J. S. Tsai, *Charge echo in a Cooper-pair box*, Phys. Rev. Lett. **88**, 047901 (2002).

[21] J. M. Martinis, S. Nam, J. Aumentado, and C. Urbina, *Rabi Oscillations in a Large Josephson-Junction Qubit*, Phys. Rev. Lett. **89**, 117901 (2002).

[22] D. V. Averin and C. Bruder, *Variable Electrostatic Transformer: Controllable Coupling of Two Charge Qubits*, Phys. Rev. Lett. **91**, 057003 (2003).

[23] J. Q. You, J. S. Tsai, and F. Nori, *Scalable Quantum Computing with Josephson Charge Qubits*, Phys. Rev. Lett. **89**, 197902 (2002).

[24] Y. Makhlin, G. Schön, and A. Shnirman, *Quantum-state engineering with Josephson-junction devices*, Rev. Mod. Phys. **73**, 357 (2001).

[25] M. H. Devoret, A. Wallraff, and J. M. Martinis, *Superconducting Qubits: A Short Review*, cond-mat/0411174, (2004).

[26] R. P. Feynman, *Quantum Mechanical Computers*, Foundations of Physics **16**, 507 (1986).

[27] J. Gruska, *Quantum computing* (McGraw-Hill, New York, 1999).

[28] M. Hirvensalo, *Quantum computing* (Springer-Verlag, Berlin, 2001).

[29] P. W. Shor, *Algorithms for Quantum Computation: Discrete Logarithms and Factoring*, IEEE Proc. 35nd Annual Symposium on Foundations of Computer Science 124 (1994).

[30] L. K. Grover, in *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing* (ACM Press, Philadelphia, 1996), pp. 212–219.

[31] R. Rivest, A. Shamir, and L. Adleman, *A method for obtaining digital signatures and public-key cryptosystems*, Comm. ACM **21**, 120 (1978).

[32] C. Bennett *et al.*, *Teleporting an Unknown Quantum State via Dual Classical and EPR Channels*, Phys. Rev. Lett. **70**, 1895 (1993).

[33] S. Wiesner, *Conjugate coding*, Sigact News **15**, 78 (1983).

[34] C. H. Bennett *et al.*, *Experimental quantum cryptography*, Journal of Cryptology **5**, 2 (1992).

[35] N. Gisin, G. Ribordy, W. Tittel, and H. Zbinden, *Quantum cryptography*, Rev. Mod. Phys. **74**, 145 (2002).

[36] *http://www.magiqtech.com/press/Magiq_Navajo_Launch.pdf,*
http://www.idquantique.com/crypto-press-engl.html.

[37] R. P. Feynman, *Simulating physics with computers*, International Journal of Theoretical Physics **21**, 467 (1982).

[38] D. Deutsch, *Quantum computational networks*, Proc. R. Soc. of Lond. A **425**, 73 (1989).

[39] E. Schrödinger, *Quantisierung als Eigenwertproblem*, Annalen der Physik **79**, 361 (1926).

[40] C. H. Bennett, *Logical Reversibility of Computation*, IBM Journal of Research and Development **17**, 525 (1973).

[41] D. Deutsch, *Quantum Theory, the Church-Turing Principle, and the Universal Quantum Computer*, Proc. R. Soc. of Lond. A **400**, 97 (1985).

[42] D. Gottesman and I. L. Chuang, *Quantum Teleportation is a Universal Computational Primitive*, Nature **402**, 390 (1999).

[43] M. A. Nielsen, *Universal quantum computation using only projective measurement, quantum memory, and preparation of the 0 state*, Phys. Lett. A. **308**, 96 (2003).

[44] D. Leung, *Quantum computation by measurements*, Int. J. Quant. Inf. **2**, 33 (2004).

[45] S. Lloyd, *A potentially realizable quantum computer*, Science **261**, 1569 (1993).

[46] W. H. Zurek, *Decoherence, einselection, and the quantum origins of the classical*, Rev. Mod. Phys. **75**, 715 (2003).

[47] D. P. DiVincenzo, *The Physical Implementation of Quantum Computation*, Fortschr. Phys. **48**, 771 (2000).

[48] A. O. Niskanen, J. P. Pekola, and H. Seppä, *Fast and Accurate Single-Island Charge Pump: Implementation of a Cooper Pair Pump*, Phys. Rev. Lett. **91**, 177003 (2003).

[49] A. W. Knapp, *Lie Groups Beyond an Introduction*, Vol. 140 of *Lecture Notes in Physics*, 2nd ed. (Birkhauser, Berlin, 2002).

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

[51] D. P. DiVincenzo, *Two-bit gates are universal for quantum computation*, Phys. Rev. A **51**, 1015 (1995).

[52] A. Barenco *et al.*, *Elementary gates for quantum computation*, Phys. Rev. A **52**, 3457 (1995).

[53] S. Lloyd, *Almost Any Quantum Logic Gate is Universal*, Phys. Rev. Lett. **75**, 346 (1995).

[54] S. Lloyd, *Black Hole Computers*, Scientific American **292**, 52 (2004).

[55] N. Khaneja, R. Brockett, and S. J. Glaser, *Time optimal control in spin systems*, Phys. Rev. A **63**, 032308 (2001).

[56] J. P. Palao and R. Kosloff, *Optimal control theory for unitary transformations*, Phys. Rev. A **68**, 062308 (2003).

[57] A. Einstein, B. Podolsky, and N. Rosen, *Can quantum-mechanical description of physical reality be considered complete?*, Phys. Rev. **41**, 777 (1935).

[58] J. Bell, *On the Einstein-Poldolsky-Rosen paradox*, Physics **1**, 195 (1964).

[59] J. Bell, *A General Method of Empirical State Determination in Quantum Physics*, Found. Phys. **1**, 339 (1971).

[60] A. Aspect, *Bell's inequality tests : more ideal than ever*, **398**, 189 (1999).

[61] A. O. Caldeira and A. J. Leggett, *Path integral approach to quantum Brownian motion*, Physica A **121**, 587 (1983).

[62] K. David, R. Laflamme, and D. Poulin, *A Unified and Generalized Approach to Quantum Error Correction*, quant-ph/0412076, (2004).

[63] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes in FORTRAN: The Art of Scientific Computing*, 2nd ed. (Cambridge University Press, Cambridge, 1992).

[64] V. Vedral, A. Barenco, and A. Ekert, *Quantum networks for elementary arithmetic operations*, Phys. Rev. A **54**, 147 (1996).

[65] R. R. Tucci, *QC Paulinesia*, quant-ph/0407215, (2004).

[66] D. Beckman, A. N. Chari, S. Devabhaktuni, and J. Preskill, *Efficient networks for quantum factoring*, Phys. Rev. A **54**, 1034 (1996).

[67] G. Florio and D. Picca, *Implementation of analytic functions with quantum gates*, quant-ph/0407079, (2004).

[68] D. E. Knuth, *The Art of Computer Programming, Vol. 2: Seminumerical Algorithms*, 3rd edition ed. (Addison-Wesley, Reading, 1998).

[69] R. Van Meter and K. M. Itoh, *Fast Quantum Modular Exponentiation*, quant-ph/0408006, (2004).

[70] T. G. Draper, *Addition on a Quantum Computer*, quant-ph/0008033, (2000).

[71] T. G. Draper, S. A. Kutin, E. M. Rains, and K. M. Svore, *A logarithmic-depth quantum carry-lookahead adder*, quant-ph/0406142, (2004).

[72] S. A. Cuccaro, S. A. Draper, Thomas G. amd Kutin, and D. P. Moulton, *A new quantum ripple-carry addition circuit*, quant-ph/0410184, (2004).

[73] T. Hogg, C. Mochon, W. Polak, and E. Rieffel, *Tools for quantum algorithms*, quant-ph/9811073, Int. J. Mod. Phys. **C10**, 1347 (1999).

[74] D. Coppersmith, *An approximate Fourier transform useful in quantum factoring*, quant-ph/0201067, (2002), (1994 IBM Internal Report).

[75] R. R. Tucci, *Quantum Fast Fourier Transform Viewed as a Special Case of Recursive Application of Cosine-Sine Decomposition*, quant-ph/0411097, (2004).

[76] V. V. Shende, S. S. Bullock, and I. L. Markov, *A Practical Top-down Approach to Quantum Circuit Synthesis*, quant-ph/0406176, (2004).

[77] R. R. Tucci, *Qubiter Algorithm Modification, Expressing Unstructured Unitary Matrices*, quant-ph/0411027, (2004).

[78] S. S. Bullock, G. K. Brennen, and D. P. O'Leary, *Asymptotically Optimal Quantum Circuits for d-level Systems*, quant-ph/0410116, (2004).

[79] S. S. Bullock and I. L. Markov, *Asymptotically optimal circuits for arbitrary n-qubit diagonal computations*, Quant. Inf. and Comp. **4**, 27 (2004).

[80] N. Schuch and J. Siewert, *Programmable Networks for Quantum Algorithms*, Phys. Rev. Lett. **91**, 027902 (2003).

[81] R. R. Tucci, *A Rudimentary Quantum Compiler*, quant-ph/9902062, (1999), 2nd Edition.

[82] C. C. Paige and M. Wei, *History and generality of the CS decomposition*, Linear Algebra and Appl. **208**, 303 (1994).

[83] S. S. Bullock, *Note on the Khaneja Glaser Decomposition*, quant-ph/0403141, (2004).

[84] C. Savage, *A survey of combinatorial Gray codes*, SIAM Rev. **39**, 605 (1997).

[85] F. Gray, *Pulse code communication*, U.S. patent no. 2,632,058 (1953).

[86] R. K. Guy, *Unsolved Problems in Number Theory, 2nd ed.* (Springer-Verlag, New York, 1994), p. 224.

[87] E. Knill, *Approximation by Quantum Circuits*, quant-ph/9508006, (1995).

[88] V. V. Shende, I. L. Markov, and S. S. Bullock, *Minimal Universal Two-qubit Quantum Circuits*, Phys. Rev. A **69**, 062321 (2004).

[89] G. Cybenko, *Reducing Quantum Computations to Elementary Unitary Operations*, Computing in Science and Engineering **3**, 27 (2001).

[90] J. Zhang, J. Vala, S. Sastry, and K. B. Whaley, *Exact Two-Qubit Universal Quantum Circuit*, Phys. Rev. Lett. **91**, 027903 (2003).

[91] F. Vatan and C. P. Williams, *Optimal quantum circuits for general two-qubit gates*, Phys. Rev. A **69**, 032315 (2004).

[92] G. Vidal and C. M. Dawson, *Universal quantum circuit for two-qubit transformations with three controlled-NOT gates*, Phys. Rev. A **69**, 010301 (2004).

[93] D. S. Abrams and S. Lloyd, *Quantum Algorithm Providing an Exponential Speed Increase for Finding Eigenvalues and Eigenvectors*, Phys. Rev. Lett. **83**, 5162 (1999).

[94] P. Jaksch and A. Papageorgiou, *Approximation Leading to Exponential Speedup of Quantum Eigenvalue Calculation*, Phys. Rev. Lett. **91**, 257902 (2003).

[95] J. P. Paz and A. Roncaglia, *Quantum gate arrays can be programmed to evaluate the expectation value of any operator*, Phys. Rev. A **68**, 052316 (2003).

[96] W. Givens, *Computation of Plane Unitary Rotations Transforming a General Matrix to Triangular Form*, J. Soc. Ind. Appl. Math **6**, 26 (1958).

[97] A. V. Aho and K. M. Svore, *Compiling Quantum Circuits using the Palindrome Transform*, quant-ph/0322008, (2003).

[98] D. P. O'Leary and S. S. Bullock, *QR Factorizations Using a Restricted Set of Rotations*, unpublished, (2004).

[99] M. Tinkham, *Introduction to Superconductivity*, 2nd ed. (McGraw Hill, New York, USA, 1996).

[100] J. E. Mooij *et al.*, *Josephson persistent-current qubit*, Science **285**, 1036 (1999).

[101] C. H. van der Wal *et al.*, *Quantum superposition of macroscopic persistent current states*, Science **290**, 773 (2000).

[102] I. Chiorescu, Y. Nakamura, M. Harmans, and J. E. Mooij, *Coherent quantum dynamics of a superconducting flux qubit*, Science **296**, 889 (2003).

[103] M. V. Feigel'man *et al.*, *Superconducting Tetrahedral Quantum Bits*, Phys. Rev. Lett. **92**, 098301 (2004).

[104] J. M. Martinis *et al.*, *Decoherence of a superconducting qubit due to bias noise*, Phys. Rev. B **67**, 094510 (2003).

[105] M. H. Devoret, in *Quantum Fluctuations in Electrical Circuits*, *Quantum Fluctuations*, edited by S. Reynaud, E. Giacobino, and J. Zinn-Justin (Elsevier Science, Amsterdam, 1997), pp. 351–382.

[106] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright, *Convergence properties of the Nelder-Mead simplex algorithm in low dimensions*, SIAM Journal on Optimization **9**, 112 (1998).

[107] I. Chuang, in *From Qubit Physics to Quantum Architectures*, *ERATO conference on Quantum Information Science 2004* (ERATO, JST, Tokyo, 2004), p. 4.

[108] G. Burkard, D. Loss, D. P. DiVincenzo, and J. A. Smolin, *Physical optimization of quantum error correction circuit*, Phys. Rev. B **60**, 11404 (1999).

[109] M. D. Price, T. F. Havel, and D. G. Cory, *Multiqubit logic gates in NMR quantum computing*, New Journal of Physics **2**, 10 (2000).

[110] D. R. Stinson, *Cryptography: Theory and Practice, Second Edition* (C&H / CRC Press, London, 2002).

[111] A. G. Fowler, S. J. Devitt, and L. C. L. Hollenberg, *Implementation of Shor's algorithm on a Linear Nearest Neighbor Qubit Array*, Quant. Inf. and Comp. **4**, 237 (2004).

[112] S. Beauregard, *Circuit for Shor's algorithm using 2n+3 qubits*, Quantum Inf. Comput. **3**, 175 (2003).

[113] R. J. Hughes, *Cryptography, quantum computation and trapped ions*, Phil. Trans. Roy. Soc. A **356**, 1853 (1998).

Abstracts of Publications I–VI

I We introduce a method for finding the required control parameters for a quantum computer that yields the desired quantum algorithm without invoking elementary gates. We concentrate on the Josephson charge qubit model, but the scenario is readily extended to other physical realizations. Our strategy is to numerically find any desired double- or triple-qubit gate. The motivation is the need to significantly accelerate quantum algorithms in order to fight decoherence.

II Quantum-circuit optimization is essential for any practical realization of quantum computation, in order to beat decoherence. We present a scheme for implementing the final stage in the compilation of quantum circuits, i.e. for finding the actual physical realizations of the individual modules in the quantum-gate library. We find that numerical optimization can be efficiently utilized in order to generate the appropriate control-parameter sequences which produce the desired three-qubit modules within the Josephson charge qubit model. Our work suggests ways in which one can in fact considerably reduce the number of gates required to implement a given quantum circuit, hence diminishing idle time and significantly accelerating the execution of quantum algorithms.

III We investigate the physical implementation of Shor's factorization algorithm on a Josephson charge qubit register. While we pursue a universal method to factor a composite integer of any size, the scheme is demonstrated for the number 21. We consider both the physical and algorithmic requirements for an optimal implementation when only a small number of qubits are available. These aspects of quantum computation are usually the topics of separate research communities; we present a unifying discussion of both of these fundamental features bridging Shor's algorithm to its physical realization using Josephson junction qubits. In order to meet the stringent requirements set by a short decoherence time, we accelerate the algorithm by decomposing the quantum circuit into tailored two- and three-qubit gates and we find their physical realizations through numerical optimization.

IV Optimal implementation of quantum gates is crucial for designing a quantum computer.We consider the matrix representation of an arbitrary multiqubit gate. By ordering the basis vectors using the Gray code, we construct the quantum circuit which is optimal in the sense of fully controlled single-qubit gates and yet is equivalent with the multiqubit gate. In the second step of the optimization, superfluous control nodes are eliminated, which eventually results in a smaller total number of the elementary gates. In our scheme the number of controlled-NOT gates is $\mathrm{O}(4^n)$ which coincides with the theoretical lower bound.

V We consider a generic elementary gate sequence which is needed to implement a general quantum gate acting on n qubits - a unitary transformation with $4n$

degrees of freedom. For synthesizing the gate sequence, a method based on the so-called Cosine-Sine matrix decomposition is presented. The result is optimal in the number of elementary one-qubit gates, 4^n, and scales more favorably than the previously reported decompositions requiring $4^n - 2^{n+1}$ controlled NOT gates.

VI We consider a unitary transformation which maps any given state of an n-qubit quantum register into another one. This transformation has applications in the initialization of a quantum computer, and also in some quantum algorithms. Employing uniformly controlled rotations, we present a quantum circuit of $2^n + 2 - 4n - 4$ CNOTs and $2^n + 2 - 5$ one-qubit elementary rotations that effects the state transformation. The complexity of the circuit is noticeably lower than the previously published results. Moreover, we present an analytic expression for the rotation angles needed for the transformation.

VII Uniformly controlled one-qubit gates are quantum gates which can be represented as direct sums of two-dimensional unitary operators acting on a single qubit. We present a quantum gate array which implements any n-qubit gate of this type using at most $2^{n-1} - 1$ controlled-NOT gates, 2^{n-1} one-qubit gates and a single diagonal n-qubit gate. The circuit is based on the so-called quantum multiplexor, for which we provide a modified construction. We illustrate the versatility of these gates by applying them to the decomposition of a general n-qubit gate and a local state preparation procedure. Moreover, we study their implementation using only nearest-neighbor gates. We give upper bounds for the one-qubit and controlled-NOT gate counts for all the aforementioned applications. In all four cases, the proposed circuit topologies either improve on or achieve the previously reported upper bounds for the gate counts. Thus, they provide the most efficient method for general gate decompositions currently known.

ISBN 951-22-7611-9
ISBN 951-22-7612-7 (PDF)
ISSN 1795-2239
ISSN 1795-4584 (PDF)