Optimization for Accurate Approximations of Operators on a Quantum Computer

by

Hyeonmi Lee

(Under the direction of Andrew Sornborger)

Abstract

As a part of efforts to realize practical uses of quantum computers, the development of the technique of controlling spin systems is an essential element. This study focuses on the issue of optimization for the control of a coupled spin system. Based on the idea of optimal control for the design of pulse sequences in NMR spectroscopy, a mathematical framework is presented for the approximation of the coupled spin operators including the controlled-not(CNOT) operation as an important gate in quantum computing systems. Finally we suggest an algorithm for the periodic control of coupled qubit systems.

Index words: Quantum Computation, optimal control, CNOT, SU(4),
Theses (academic)

Optimization for Accurate Approximations of Operators on a Quantum Computer

by

Hyeonmi Lee

MEd., Hong Ik University, 2000

BS., Hong Ik University, 1991

A Thesis Submitted to the Graduate Faculty

of The University of Georgia in Partial Fulfillment

of the

Requirements for the Degree

Master of Art

Athens, Georgia

2009

Optimization for Accurate Approximations of Operators on a Quantum Computer

by

Hyeonmi Lee

Approved:

Major Professor: Andrew Sornborger

Committee: Malcolm Adams
Robert Varley

Electronic Version Approved:

Maureen Grasso
Dean of the Graduate School
The University of Georgia
May 2009

Acknowledgments

I would like to express my gratitude to all those who gave me the possibility to complete this thesis. Dr. Adams and Dr. Varley in my committee and my friend Sunbok Lee, their thoughtful advice served me a sense of direction in all the time of research for and writing of this thesis. It is difficult to overstate my appreciation to them.

I am deeply indebted to my advisor Dr.Andrew Sornborger. Throughout my thesis-writing period, his enthusiasm and his great efforts to explain things clearly and simply made this whole study fun for me. His unlimited support, encouragement, and stimulating suggestions help me done this work. He is my professor, my mentor, and my friend. I would have been lost without him.

Last, I would like to give my special thanks to my mother and brothers for their loving support. They have been always with me encouraging me to do my best in all matters of life. To them, I dedicate this thesis.

Table of Contents

List of Tables

Chapter 1

Introduction

1.1 Quantum computers as more efficient alternatives to classical computers

The computer is one of the most innovative and useful inventions created by mankind [10]. It is obvious that the computer has accelerated the progress of civilization. According to Robertson[10], the full impact of the computer is beyond our comprehension, just as the full impact of the printing press could not be perceived by people in the 15th century. He contends that computers are causing greater changes in society than the printing press, having brought tremendous advances in mathematics and science, and virtually influencing every field of study. Many new discoveries or advances made today would have been impossible before the computer age. Indeed, computers allow us to see things that could not be seen before. This is also reflected in the fact that the wider society depends on computers and this dependence on computers is greater than ever. Since the world's first programmable, digital, electronic computer, Colossus, appeared during World War II, the computer has been developed at an enormous speed. However, it seems that this speed is never quite fast enough. As one example of the ever-increasing desire for computational power, researchers in Los Alamos National Laboratory (LANL) in New Mexico have been conducting various experiments using the supercomputer nicknamed Blue Mountain, which occupies a 1000 square foot area of the lab building. Allegedly, Blue Mountain takes about 72 hours to finish the same work as a regular computer takes 17.8 years, but the researchers, who worked with this state of the art computer, soon started to complain that it takes too long to see the results. Now LANL is equipped with a new super computer, called Q, which occupies a 4000 square foot area

and is five times faster than Blue Mountain [2]. Such supercomputers, either Blue Mountain or Q, are a collection of classical computers that run in parallel to solve a given problem. These matters of space and cost reveal the limit of conventional approaches to improving computer efficiency and lead us to consider the alternatives. In addition to the development of supercomputers for special uses, computer hardware for general and personal computers has also grown in power at an amazing pace. The growth in the component density on a chip was described by Gordon Moore in 196. His observation, known as Moore's Law which states that the number of components that can be inexpensively placed on an integrated circuit is increasing exponentially and doubling approximately every two years. Surprisingly, Moore's Law has approximately held true since the 1960s. The trend has continued for more than half a century and is unlikely to stop for another decade at least and perhaps much longer [2, 7]. Moore's Law also implies a limit to conventional approaches in developing classical computers. If Moore's Law continues to hold, the theoretical conclusion is that switches in computers will be a single atom in size, and quantum effects will become important at that scale causing difficulties for classical computation methods. Today, performing computations at the quantum scale is an urgent problem for the computer industry. Fortunately, the idea of the quantum computer has appeared as an alternative to the classical computer. It may help solve the technological problem of squeezing more computer components on a chip [2, 4, 7] and, in some cases, provide more efficient algorithms than is possible on a classical computer.

Considering the theoretical background of the quantum computer, Nielsen and Chuang pointed out that the idea of a quantum computer is located at the intersection of several fields such as quantum mechanics, computer science, information theory, and cryptography [9]. However, the idea of a quantum computer was suggested by Richard Feynman in 1982 and Paul Benioff as early as 1980 respectively. As a part of his efforts to enhance the operating speed of computers dramatically, Feynman changed his mindset completely. He considered quantum phenomena as useful tools that can be used in a new computing system. On the other hand, Benioff took advantage of the reversibility of quantum mechanics to resolve the

problem of the heat that classical computers generate due to irreversible operations which increase entropy [7].

Among the range of topics of importance to quantum computation, this study focuses on the issue of optimization for the control of a coupled spin system. Specifically, a mathematical framework is presented for the approximation of the controlled-not (CNOT) operation as an important gate in a quantum computing system as well as other coupled spin operators. It has been shown that any quantum circuit can be simulated to an arbitrary degree of accuracy using a combination of CNOT gates and single qubit rotations. Before presenting the methods and the results of this study, it is natural to discuss the properties of a quantum system and how a quantum state can be represented mathematically and used for computation. The next background section touches all the above-mentioned topics making a comparison between classical computers and quantum computers. In the subsequent section, I will review related literature.

Chapter 2

Background

2.1 The Fundamental Properties of a Quantum State: Superposition and Entanglement

In quantum mechanics, quantum refers to a discrete unit that quantum theory assigns to certain physical quantities such as the energy of an atom. Electrons, protons and neutrons considered as quantum bits of information (qubits) are quantized for the correct description of the relationship between quanta and elementary particles. These qubits are realized as actual physical systems as well as considered as mathematical objects with certain specific properties. Newtonian laws are not applied to this quantum world. In quantum computing systems, a qubit is a unit of information that is described by a state vector, and spin half particles such as electrons, photons, and neutrons may play the role of a qubit. The intrinsically spinning states of these particles may be spin-up, spin-down, or a superposition of both states. The states are not deterministic, and measurement can change the states of qubits. While a classical bit of information takes the value 0 or 1, a qubit can not only take the values 0 and 1 but also a superposition of states representing 0 and 1. This is the first fundamental property of qubits: qubits are represented by linear combinations of two states, that is, a superposition of up and down. The second property is entanglement. Two objects can form a single quantity at the quantum level even if they are separated a long distance away from each other. As the material of sci-fi movies, the idea of quantum teleportation is closely related to this second quantum property. An attempt to see a single entity as a combination of two independent objects is theoretically possible under the condition that the possibility of signal propagation at superluminal speeds is allowed. These two properties

lie together at the core of quantum parallelism, the possibility of performing a large number of operations in parallel [6].

2.2 The Quantum Computer and Basics for a Mathematical formulation for quantum bits

Quantum mechanical phenomena such as superposition and entanglement are directly used in quantum computing systems. Qubits are quantum objects that correspond to the bits of classical computers. A classical computer has a memory made up of bits. A quantum computer operates on a sequence of qubits. In quantum computers, qubits carry the information whereas information is coded as a string of bits in classical computers. The main advantage of quantum computers is the use of quantum parallelism, which allows quantum computations that are more efficient than classical computations. The following Bloch sphere provides a good picture of a single qubit state.

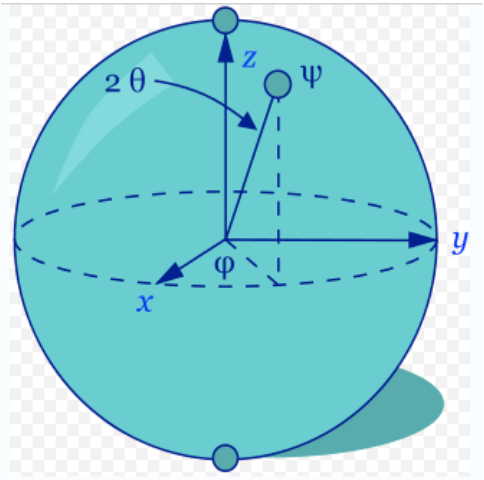

Figure 2.1: Bloch sphere representing a qubit state

retrieved from http://en.wikipedia.org/wiki/Bloch- sphere

As the picture shows, the number of possible states of a qubit is infinitely large, the same as the number of points on the surface of the Bloch sphere. In a mathematical sense, two states of a qubit such as spin up and spin down, which can also be considered as the excited

state and the ground state of an atom, are represented with the notation of $|1\rangle$ and $|0\rangle$ respectively. The notations $\langle \ \ | $ and $| \ \ \rangle$ are called bra and ket respectively, known as Dirac notation. The meanings are,

$$|\psi\rangle = \begin{pmatrix} c_0 \\ c_1 \\ \vdots \end{pmatrix}, \langle\varphi| = \begin{pmatrix} b_0 & b_1 & \cdots \end{pmatrix}$$

and therefore, the bra-ket notation $\langle\varphi \mid \psi\rangle = \sum_i \overline{c}_i b_i$ represents the inner product of two states with complex coefficients. $|1\rangle$ and $|0\rangle$ are considered as computational basis states and each state is a vector in a two-dimensional complex vector space. Thus, $|1\rangle$ and $|0\rangle$, meaning $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, and $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$, form an orthonormal basis for this vector space. An arbitrary state of a qubit is a linear combination of these two states:

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

Quantum mechanics makes predictions about probability distributions depending on the states of qubits at the instant of the measurement, and the values are connected to the coefficients of up and down states. These coefficients are complex numbers representing probability amplitudes, that is, $\|\alpha\|^2 + \|\beta\|^2 = 1$, and $\|\alpha\|^2$ and $\|\beta\|^2$ indicate probabilities of measuring spin up and spin down respectively. That is, the state of a single qubit can be any normalized superposition of the two possible basis states. According to the probabilities $\|\alpha\|^2$ and $\|\beta\|^2$, one of two states, such as spin up or down will be measured. For example, in the Fig 2.1, the state of the given qubit is represented;

$$|\psi\rangle = \cos\theta\,|0\rangle + e^{i\varphi}\sin\theta\,|1\rangle$$

where $\alpha = \cos\theta$, and $\beta = e^{i\varphi}\sin\theta$

$$\|\alpha\|^2 = \|\cos\theta\|^2, \|\beta\|^2 = \left\|e^{i\varphi}\sin\theta\right\|^2 = \left\|e^{i\varphi}\right\|^2 \|\sin\theta\|^2 = \|\sin\theta\|^2$$

$$\|\alpha\|^2 + \|\beta\|^2 = \|\cos\theta\|^2 + \|\sin\theta\|^2 = 1.$$

This quantum state corresponds to the coordinate (x, y, z) in R^3, where $x = \sin 2\theta \times \cos\varphi$, $y = \sin 2\theta \times \sin\varphi$, and $z = \cos 2\theta$. We represent states of a single qubit in the manner shown above, as a wavefunction. Wavefunctions can change as time progresses, and the time dependent *Schrödinger* Equation, $i\frac{\partial}{\partial t}|\psi(t)\rangle = H|\psi(t)\rangle$ describes how wavefunctions change in time, playing the corresponding role of Newton's second law in classical mechanics. The time evolution of wave functions is reflected in changes in measurement probabilities of a quantum state over time. Measurement changes the state of a qubit, when observed, but the final state after measurement is still somewhere on the surface of the Bloch sphere. Thus, qubits are represented by using self-adjoint operators[1] while transformative processes can be represented by unitary linear operators such as rotation or time evolution as in the *Schrödinger* Equation. By multiplying a state vector by a unitary matrix, a new state is given marking a new point on the Bloch sphere. Consider $|\psi\rangle$ as the general state of a two-qubit system. Then the state of the system is any normalized superposition of the four orthogonal states. In the same way, in a quantum computing system that operates on a 3-qubit register, a total of eight states are possible for three qubits, $|000\rangle, |001\rangle, |010\rangle, |011\rangle, |100\rangle, |101\rangle, |110\rangle$, and $|111\rangle$. Three-qubit quantum computers can represent a superposition of all eight possible cases at the same time. Thus, the register is described by the wavefunction:

$$|\psi\rangle = \alpha_1|000\rangle + \alpha_2|001\rangle + \alpha_3|010\rangle + \alpha_4|011\rangle + \alpha_5|100\rangle + \alpha_6|101\rangle + \alpha_7|110\rangle + \alpha_8|111\rangle$$

[1]Self-adjoint operators are usually used in functional analysis and quantum mechanics. In quantum mechanics, they are significant in that physical observables such as position, momentum, and spin, are represented by them on a Hilbert space.

$$= \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \\ \alpha_5 \\ \alpha_6 \\ \alpha_7 \\ \alpha_8 \end{pmatrix} \text{ where } \sum_{i=1}^{8} \|\alpha_i\|^2 = 1 \alpha_i \in C$$

This shows that a three-qubit system can be represented by any normalized superposition of eight orthogonal classical states. In contrast, classical computers that have a three-bit register can represent only one of these eight cases at one time. A quantum computer that has a three-qubit register is tantamount to eight classical computers that have a three-bit register at the same time. 2^n classical computers may be thought of as equivalent to an n-qubit quantum computer. A quantum computer operates by manipulating qubits with a sequence of quantum logic gates, called a quantum algorithm. The matter of manipulating qubits will be discussed in section 2.4.

2.3 Operators to generate the time evolution of quantum states

Consider the *Schrödinger* equation, $i\frac{\partial|\psi(t)\rangle}{\partial t} = H|\psi(t)\rangle$ where H is any constant coefficient Hermitian matrix. $|\psi(t)\rangle = e^{-iHt}|\psi(t_0)\rangle$ is the solution of the *Schrödinger* equation and it represents the state of the system at time t. For the initial state $|\psi_0\rangle$, the states of qubits can be represented by $|\psi(t)\rangle = e^{-iHt}|\psi_0\rangle$. The state of a quantum system is described by a unit vector in a Hilbert space $\boldsymbol{H}$. H is Hermitian guarantees that eigenvalues of any Hermitian matrix are real numbers, which means physical observables have real values. Eigenvectors of a Hermitian matrix form a complete orthogonal basis of the space of state vectors. That allows any point on the Bloch sphere, which means any possible state, can be represented by a linear combination of such a basis with complex coefficients.

Further, The fact that H is Hermitian also supports That e^{-iHt} is a unitary matrix.

$$\| |\psi\rangle \|^2 = |\psi\rangle^\dagger |\psi\rangle = \left\langle e^{-iHt} \mid \psi_0 \right\rangle^\dagger \left\langle e^{-iHt} \mid \psi_0 \right\rangle$$

$$= \left(\langle \psi_0 | \, e^{iHt} \right) \left(e^{-iHt} \, |\psi_0\rangle \right)$$

$$= \langle \psi_0 | \, e^{iHt} e^{-iHt} \, |\psi_0\rangle = \langle \psi_0 \mid \psi_0 \rangle = 1$$

The time-evolution of the state of a closed quantum system is represented by such unitary operators. For the initial state $|\psi_1\rangle$ and the final state $|\psi_2\rangle$, there is a unitary operator U such that $|\psi_2\rangle = U \, |\psi_1\rangle$ [5].

2.4 Manipulating qubits and Pauli matrices

Charged particles have magnetic fields and may be described as quantum spins. Nuclear Magnetic Resonance (NMR), especially MRI (Magnetic Resonance Imaging) is based on the properties of magnetic particles in a magnetic field. Electrons and protons as spin half particles are spinning and charged in nature. For instance, the states of protons are described as superpositions of spin up and down. The following Fig 2.2 explains this.

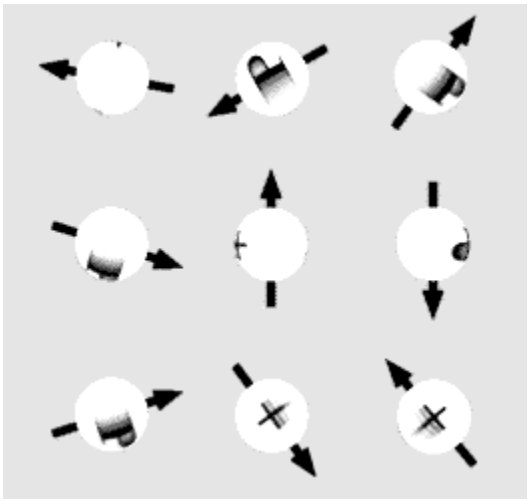

Figure 2.2: Protons that are in different states of superposition
retrieved from http://www.simplyphysics.com/MRI shockwave.html

A magnetic field makes these protons align according to its direction as Fig 2.3 shows.

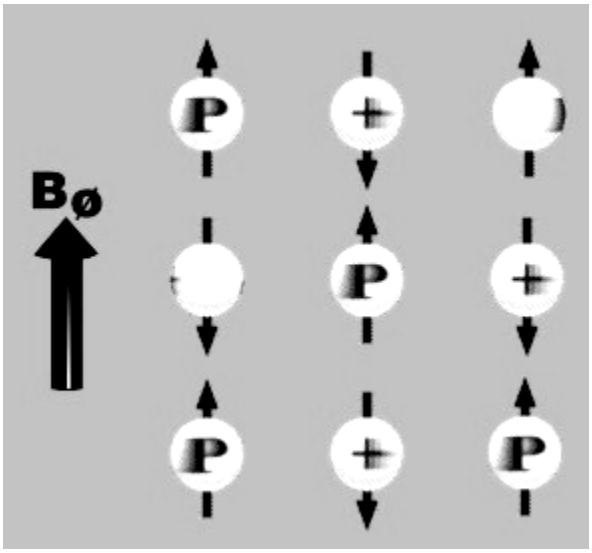

Figure 2.3: Protons' aligning in a magnetic field of the strength B_0
retrieved from http://www.simplyphysics.com/MRI shockwave.html

When given specially designed pulse sequences to the polarized protons in Fig 2.3, the states of protons are changed. This is the one way of manipulating qubits. NMR quantum computing uses the spin states of molecules as qubits. In principle, NMR-like manipulation techniques are applicable to any spinning particles. NMR spectroscopy is technique that uses the magnetic properties of nuclei. Among experiments that show or use quantum properties, the Stern-Gerlach experiment has special significance in quantum mechanics. It can be used to demonstrate quantum properties that electrons and atoms intrinsically have and how measurement affects the system being measured [3]. This is seen in the following picture. Electrons from the furnace are collimated and pass between two poles of the magnet. When electrons are between two poles, they align their directions in parallel with the magnetic field direction as in Figure 2.4 and perpendicular to the ray direction. The ray from the furnace splits into two rays depending on whether the direction of each electron is spin up or spin down when aligned between the N-pole and the S-pole of the magnet. Pauli matrices can be used to formulate such phenomena mathematically. Consider a basis of **H**, a Hilbert space with $|0\rangle$ and $|1\rangle$ as the basis vectors. As already mentioned, the general state

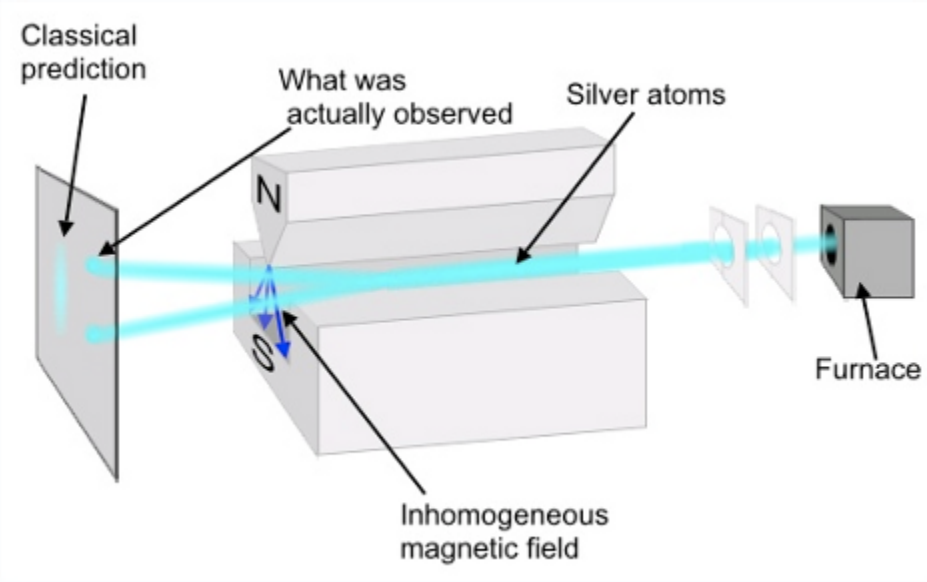

Figure 2.4: Stern-Gerlach experiment
retrieved from http://en.wikipedia.org/wiki/Stern-Gerlach-Experiment

of spin $\frac{1}{2}$ particles, related to the Figure 2.1, is represented by $|\psi\rangle = \cos\theta\,|0\rangle + e^{i\varphi}\sin\theta\,|1\rangle$ corresponding to the vector $(\sin 2\theta \times \cos\varphi, \sin 2\theta \times \sin\varphi, \cos 2\theta)$ in the Bloch sphere. See Section 2.3. The physical properties of qubits can be represented by Hermitian operators in a two-dimensional complex space, and Pauli matrices are the basis for the operators.

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

Pauli matrices are Hermitian and unitary and have important properties:

$$\sigma_i^2 = I, \sigma_x\sigma_y = i\sigma_z, \sigma_z\sigma_x = i\sigma_y, \sigma_y\sigma_z = i\sigma_x$$
$$\det(\sigma_i) = -1$$
$$tr(\sigma_i) = 0$$

The Pauli matrices plus the identity matrix generate the Lie Algebra U(2) under the operation of commutation.

The commutator on a Hilbert space is important since it is related to how well two states can be measured simultaneously. Heisenberg's uncertainty principle, connected to the Baker-Campbell-Hausdorff formula, is a theorem concerning these commutators. The probability distribution of a spin can be acquired by computing the spectral decomposition of the corresponding operator. Consider that the operators $e^{-iH_0t_1}e^{-iH_1t_2}$ evolve the state of a particle over three equal time intervals in a certain period. Then by Baker-Campbell Hausdorff formula;

$$\begin{aligned} & e^{-iH_0t_1}e^{-iH_1t_2} \\ & = e^{-i\{(b_0H_0+b_1H_1)+b_2[H_0,H_1]+b_3[H_0[H_0,H_1]]+\ldots\}} \end{aligned}$$

for real numbers b_i.

But all commutators are closed in the Lie algebra U(2) of Pauli matrices with the identity matrix. Thus, the observed states are finally represented by $e^{-i\left(\sum\limits_{j=x,y,z,I}\alpha_j\sigma_j\right)}$. Operators corresponding to certain spins do not commute, that is, $e^Ae^B \neq e^Be^A$, and $e^Ae^B \neq e^{A+B}$. This is the analog of Heisenberg's uncertainty principle for quantum spins, which states that one cannot measure the position and the momentum of a particle at the same time. Complex Hermitian matrices can be represented as $H = \lambda_0 I + \sum\limits_{i=1}^{3}\lambda_i\sigma_i$ with real coefficients λ_j for $j = 0, 1, 2, 3$. This fact is connected to the Bloch sphere representation of mixed states. The states

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \text{and } |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

are eigenvectors of σ_z. For $\vec{\sigma} = \begin{pmatrix} \sigma_x & \sigma_y & \sigma_z \end{pmatrix}$ and $(\sin 2\theta\cos\varphi, \sin 2\theta\sin\varphi, \cos 2\theta)$,

$$\begin{aligned} & \vec{\sigma}\cdot(\sin 2\theta\cos\varphi, \sin 2\theta\sin\varphi, \cos 2\theta) \\ & = \sigma_x\cdot\sin 2\theta\cos\varphi + \sigma_y\cdot\sin 2\theta\sin\varphi + \sigma_z\cdot\cos 2\theta \\ & = \begin{pmatrix} \cos 2\theta & e^{-i\varphi}\sin 2\theta \\ e^{i\varphi}\sin 2\theta & -\cos 2\theta \end{pmatrix} (*) \end{aligned}$$

The vector $|\psi\rangle$ is an eigenvector of the result matrix $(*)$ with eigenvalue 1.

2.5 Classical computers vs Quantum computers

The following table summarizes the differences and the similarities between classical computers and quantum computers.

Table 2.1: Classical computers vs. Quantum computers

	Classical Computers	Quantum Computers
Information units	Bit	Qubit
Possible states of information unit	Either 0 or 1	Any linear combination of two states represented by $\|0\rangle \, and \, \|1\rangle$
Gate	A set of combined transistors	A mechanism that consists of three elements of input state, output state, and time-evolution
An example of a universal set of gates	NAND gate (or NOR gate) (irreversible)	Single qubit gates and CNOT gate (reversible)
Implementing gates	Semiconductor diodes or transistors	Manipulating qubits

2.6 Algorithm for Optimal Control of Coupled Spin Systems

Navin Khaneja et al. introduced an optimal control algorithm for the design of pulse sequences in NMR spectroscopy [8]. The following subsections from the article provide a methodological idea for this study.

2.6.1 Transfer between Hermitian operators in the absence of relaxation

The state of the spin system can be represented by the density operator $\rho(t)$ derived from the Liouville-von Neumann equation, $\dot{\rho}(t) = -i\left[\left(H_0 + \sum_{k=1}^{m} u_k(t)H_k\right), \rho(t)\right]$, where H_0 is the free evolution Hamiltonian, H_k are the radiofrequency (rf) Hamiltonians corresponding to the available control fields and $u(t) = (u_1(t), u_2(t), \ldots, u_m(t))$ represents the vector of amplitudes that can be changed and also considered as a control vector. To find the optimal amplitudes $u_k(t)$ of the rf fields that steer a given initial density operator $\rho(0) = \rho_0$ in a

certain time T to a density operator $\rho(T)$ which is maximally overlapped to a desired target operator C. This overlap is measured by the standard inner product $\Phi_0 = \langle C \mid \rho(T) \rangle = tr\left\{C^\dagger \rho(T)\right\}$.

For simplicity, consider that the selected time period T is divided into N intervals and for each time interval, the control amplitudes u_k are constant. Then $U_j = \exp\left\{-i\Delta t\left(H_0 + \sum_{k=1}^{m} u_k(j)H_k\right)\right\}$ represents the time-evolution of the spin system for the j^{th} time interval. Thus, the final density operator at time T is $\rho(T) = U_N \cdots U_1 \rho_0 U_1^\dagger \cdots U_N^\dagger$, and the performance function $\Phi_0 = \left\langle C \,\middle|\, U_N \cdots U_1 \rho_0 U_1^\dagger \cdots U_N^\dagger \right\rangle$, is expected to be maximized as desired. This performance function can be rewritten, by the definition of inner product and the fact that the trace of a product is fixed by cyclic permutations of the factors,

$$\begin{aligned}\Phi_0 &= \left\langle U_{j+1}^\dagger \cdots U_N^\dagger C U_N \cdots U_{j+1} \,\middle|\, U_j \cdots U_1 \rho_0 U_1^\dagger \cdots U_j^\dagger \right\rangle \\ &= \langle \lambda_j \mid \rho_j \rangle\end{aligned}$$

where ρ_j is the density operator $\rho(t)$ at time $t = j\Delta t$ and λ_j is the backward propagated target operator C at the same time $t = j\Delta t$. While the control amplitudes $u_k(j)$ are perturbed to $u_k(j) + \delta u_k(j)$ perturbed, the value of the performance Φ_0 changes.

$\delta U_j = -i\Delta t \delta u_k(j) \bar{H}_k U_j$ with $\bar{H}_k \Delta t = \int_0^{\Delta t} U_j(\tau) H_k U_j(\tau) d\tau$ and

$U_j = exp\left\{-i\tau\left(H_0 + \sum_{k=1}^{m} u_k(j)H_k\right)\right\}$. [2]

If Δt is small, that is, $\Delta t \ll \left\| H_0 + \sum_{k=1}^{m} u_k(j)H_k \right\|^{-1}$, then $\bar{H}_k \approx H_k$. From the above Φ_0 and $\delta U_j = -i\Delta t \delta u_k(j) \bar{H}_k U_j$, we have $\frac{\delta \Phi_0}{\delta u_k(j)} = -\langle \lambda_i \mid i\Delta t\,[H_k, \rho_j] \rangle$

for the first order in Δt. As we choose $u_k(j) \to u_k(j) + \varepsilon \frac{\delta \Phi_0}{\delta u_k(j)}$

for the small , we increase the performance function Φ_0.

N. Khaneja et al. suggests that this provides the new way of the gradient ascent pulse engineering (GRAPE) algorithm. The algorithm is given in [8] as follows:

1. Guess initial controls $u_k(j)$

[2] This is derived from the standard formula $\frac{d}{dx} e^{A+Bx} \mid_{x=0} = e^A \int_0^1 e^{A\tau} B e^{-A\tau} d\tau$.

2. Starting from the initial ρ_0, calculate $\rho_j = U_j \cdots U_1 \rho_0 U_1^\dagger \cdots U_j^\dagger$ for all $j \leq N$

3. Starting from $\lambda_N = C$, calculate $\lambda_j = U_{j+1}^\dagger \cdots U_N^\dagger C U_N \cdots U_{j+1}$ for all $j \leq N$

4. Evaluate $\frac{\delta \Phi_0}{\delta u_k(j)}$ and update the $m \times n$ control amplitudes $u_k(j)$

according to $u_k(j) \rightarrow u_k(j) + \varepsilon \frac{\delta \Phi_0}{\delta u_k(j)}$

5. With these as the new controls, go to the second step. The starting value of $u_k(j)$ is random and the algorithm ends up when Φ_0 becomes smaller than a selected initial value.

2.6.2 Synthesis of Unitary Transformations

This problem is for creating a desired unitary operator in a given time T. For a closed quantum system, the motion equation is $\dot{U} = -i \left(H_0 + \sum_{k=1}^{m} u_k(t) H_k \right) U$.

At the initial value for the system is represented as $U(0) = 1$.

For the desired operator U_F, $u_k(t)$ is given to minimize the value of $\|U_F - U(T)\|^2 = \|U_F\|^2 - 2\mathrm{Re}\langle U_F \mid U(T)\rangle + \|U(T)\|^2$.

This is equivalent to the maximization of $\mathrm{Re}\langle U_F \mid U(T)\rangle$.

Thus, the performance index is given as,

$$\begin{aligned}
\Phi_1 &= \mathrm{Re}\langle U_F \mid U(T)\rangle \\
&= \mathrm{Re}\langle U_F \mid U_N \cdots U_1\rangle \\
&= \mathrm{Re}\left\langle U_{j+1}^\dagger \cdots U_N^\dagger U_F \;\middle|\; U_j \cdots U_1 \right\rangle \\
&= \mathrm{Re}\langle P_j \mid X_j\rangle
\end{aligned}$$

The actual application of this idea is to approximate the target operator U_F up to an arbitrary phase factor $e^{i\phi}$ and $\left\|U_F - e^{i\phi} U(T)\right\|^2$ is to be minimized for some ϕ. Since $\left\|U_F - e^{i\phi} U(T)\right\|^2 = \|U_F\|^2 - 2\mathrm{Re}\left\langle U_F \mid e^{i\phi} U(T)\right\rangle + \|U(T)\|^2$, minimizing $\left\|U_F - e^{i\phi} U(T)\right\|^2$ is equivalent to maximizing $\Phi_2 = \|\langle U_F \mid U(T)\rangle\|^2$

Chapter 3

Methods

Attempts to find a better approximation and control the coupled qubit systems are described through five steps in this chapter. The technique we use throughout this chapter is based on the GRAPE algorithm introduced in section 2.4 with the help of the computing tool, MATLAB.

3.1 Pauli two component formalism and Euler rotations from classical mechanics

The ideas for this first step are drawn from the book, Modern Quantum Mechanics by J. J. Sakurai (1994). First , we summarize the fundamental ideas from that book [9].

We discussed in section 2.6 that Pauli matrices can be used as the means to describe how to manipulate the state kets in spin $\frac{1}{2}$ systems. As the matrix representation of the rotation operator $\exp\left(\frac{-i\sigma\cdot\hat{n}\phi}{2}\right)$, acts on a two-component χ. The column matrix $\chi = \begin{pmatrix} c_+ \\ c_- \end{pmatrix}$ can be referred to as a two-component state with the elements of spin up, $\chi_+ = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and spin down, $\chi_- = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ where $c_+^2 + c_-^2 = 1$. That is, $\chi = c_+\chi_+ + c_-\chi_-$ For $\hat{n}$, a unit vector in a certain direction, we can write

$$\exp\left(\tfrac{-i\sigma\cdot\hat{n}\phi}{2}\right) = \left[1 - \tfrac{(\sigma\cdot\hat{n})^2}{2!}\left(\tfrac{\phi}{2}\right)^2 + \tfrac{(\sigma\cdot\hat{n})^4}{4!}\left(\tfrac{\phi}{2}\right)^4 - \cdots\right] - i\left[(\sigma\cdot\hat{n})\tfrac{\phi}{2} - \tfrac{(\sigma\cdot\hat{n})^3}{3!}\left(\tfrac{\phi}{2}\right)^3 + \cdots\right]$$
$$= I\cos\left(\tfrac{\phi}{2}\right) - i\sigma\cdot\hat{n}\sin\left(\tfrac{\phi}{2}\right).$$

Since $\sigma \cdot \hat{n} = \sum_{k=1,2,3} \sigma_k n_k$ where $\sigma_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ $\sigma_2 = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$, and $\sigma_3 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$, we have $\exp\left(\frac{-i\sigma\cdot\hat{n}\phi}{2}\right) = \begin{pmatrix} \cos\left(\frac{\phi}{2}\right) - in_3 \sin\left(\frac{\phi}{2}\right) & (-in_1 - n_2)\sin\left(\frac{\phi}{2}\right) \\ (-in_1 + n_2)\sin\left(\frac{\phi}{2}\right) & \cos\left(\frac{\phi}{2}\right) + in_3 \sin\left(\frac{\phi}{2}\right) \end{pmatrix}$.

That is, χ changes as $\exp\left(\frac{-i\sigma\cdot\hat{n}\phi}{2}\right)\chi$ by the rotation specified with $\hat{n}$ and ϕ. As we discussed in the section 2.3, the rotation operator $\exp\left(\frac{-i\sigma\cdot\hat{n}\phi}{2}\right)$ is unitary. $U(a,b) = \begin{pmatrix} a & b \\ -b^* & a^* \end{pmatrix}$ is unitary where $|a|^2 + |b|^2 = 1$ for complex numbers a and b since $U(a,b)^\dagger U(a,b) = \begin{pmatrix} a^* & -b \\ b^* & a \end{pmatrix}\begin{pmatrix} a & b \\ -b^* & a^* \end{pmatrix} = 1$.

By comparing the corresponding entities of $\exp\left(\frac{-i\sigma\cdot\hat{n}\phi}{2}\right)$ and $U(a,b) = \begin{pmatrix} a & b \\ -b^* & a^* \end{pmatrix}$

$$\mathrm{Re}(a) = \cos\left(\frac{\phi}{2}\right),\ \mathrm{Im}(a) = -n_3 \sin\left(\frac{\phi}{2}\right)$$

$$\mathrm{Re}(b) = -n_2 \sin\left(\frac{\phi}{2}\right),\ \mathrm{Im}(b) = -n_1 \sin\left(\frac{\phi}{2}\right)$$

it is immediate that $\exp\left(\frac{-i\sigma\cdot\hat{n}\phi}{2}\right)$ is unitary. Conversely, the general unitary unimodular matrix, $U(a,b)$ can be considered as a rotation and forms a subgroup of $U(2)$. That is, in two dimensional space, $U = e^{i\gamma}\begin{pmatrix} a & b \\ -b^* & a^* \end{pmatrix}$ forms a unitary group where $|a|^2 + |b|^2 = 1$ $\gamma = \gamma^*$. It is well known from classical mechanics that Euler rotations accomplish an arbitrary rotation of a rigid body in three steps. Figure 3.1 describes this. For the notation, $R_z(\alpha)$ means the z-axis rotation by angle α, $\mathrm{R}(\alpha,\beta,\gamma) \equiv R_Z(\gamma)R_Y(\beta)R_z(\alpha)$.

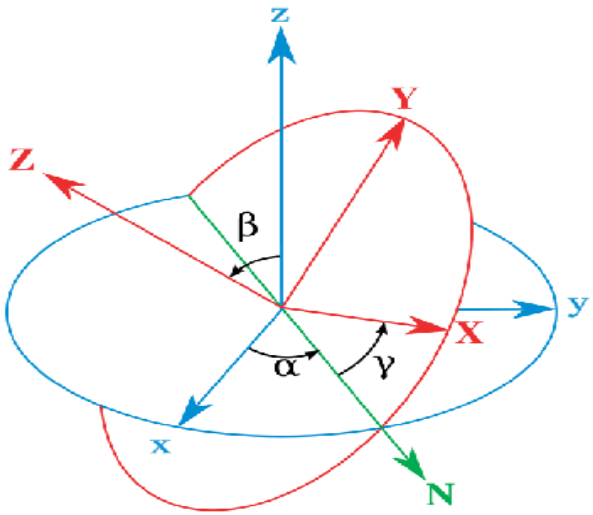

Figure 3.1: Euler rotation

retrieved from http://en.wikipedia.org/wiki/ Euler-rotations

We can rewrite $R(\alpha, \beta, \gamma) \equiv R_Z(\gamma)R_Y(\beta)R_z(\alpha)$ by substituting the following

$$R_Y(\beta) = R_z(\alpha)R_y(\beta)R_z^{-1}(\alpha)$$
$$R_Z(\gamma) = R_Y(\beta)R_z(\gamma)R_Y^{-1}(\beta)$$

That is,

$$R_Z(\gamma)R_Y(\beta)R_z(\alpha) = R_Y(\beta)R_z(\gamma)R_Y^{-1}(\beta)R_Y(\beta)R_z(\alpha)$$
$$= R_z(\alpha)R_y(\beta)R_z^{-1}(\alpha)R_z(\gamma)R_z(\alpha)$$

Since the rotation for the same axis is commutative,

$$R_z(\alpha)R_y(\beta)R_z^{-1}(\alpha)R_z(\gamma)R_z(\alpha)$$
$$= R_z(\alpha)R_y(\beta)R_z(\gamma)$$

We switch this representation with the matrix representation. That is,

$$
\begin{aligned}
&R_z(\alpha)R_y(\beta)R_z(\gamma) = \exp\left(\tfrac{-i\sigma_3\alpha}{2}\right)\exp\left(\tfrac{-i\sigma_2\beta}{2}\right)\exp\left(\tfrac{-i\sigma_3\gamma}{2}\right) \\
&= \begin{pmatrix} e^{-i\frac{\alpha}{2}} & 0 \\ 0 & e^{i\frac{\alpha}{2}} \end{pmatrix} \begin{pmatrix} \cos\left(\frac{\beta}{2}\right) & -\sin\left(\frac{\beta}{2}\right) \\ \sin\left(\frac{\beta}{2}\right) & \cos\left(\frac{\beta}{2}\right) \end{pmatrix} \begin{pmatrix} e^{-i\frac{\gamma}{2}} & 0 \\ 0 & e^{i\frac{\gamma}{2}} \end{pmatrix} \\
&= \begin{pmatrix} e^{\frac{-i(\alpha+\gamma)}{2}}\cos\left(\frac{\beta}{2}\right) & -e^{\frac{-i(\alpha-\gamma)}{2}}\sin\left(\frac{\beta}{2}\right) \\ e^{\frac{i(\alpha-\gamma)}{2}}\sin\left(\frac{\beta}{2}\right) & e^{\frac{i(\alpha+\gamma)}{2}}\cos\left(\frac{\beta}{2}\right) \end{pmatrix}
\end{aligned}
$$

Based on this idea, we try to find an approximation to the target operator with rotation in three steps by varying the rotation angles. As we discussed earlier in Section 2.3 and 2.4, a target operator D in a single qubit system is represented as a unitary matrix. Thus, $D = e^{iC}$ where $C = \begin{pmatrix} a & b \\ b^* & c \end{pmatrix}$ is Hermitian matrix, that is, a and c are real numbers, and b and b^* are complex conjugates. We want to minimize the difference between D and $U(T) = e^{-iAt_1}e^{-iBt_2}e^{-iAt_3}$ for a good approximation, $U(T)$ for each time interval t_i in the time evolution operator $|\psi(t)\rangle = e^{-iHt}|\psi_0\rangle$. Thus, for $C(T) = \|D - U(t)\|^2 = tr[(D-U(T))^\dagger(D-U(T))]$, this amounts to an optimization problem for finding the minimum of $C(T)$. The criterion function $C(T)$ informs us how close $U(T)$ is to the target, and the function $MP(t) = e^{-iAt_1}e^{-iBt_2}e^{-iAt_3}$ as the product of unitary matrices provides $U(T)$ as its output for a time variable t.

3.1.1 The result for the test of the idea of Euler rotation

$C(T) = \mathrm{Re}\{tr(D-MP(t))^\dagger \cdot (D-MP(t))\}$ and $MP(t) = e^{-iAt_1}e^{-iBt_2}e^{-iAt_3}$ were computed for random initial values of t_i and random entries of Hermitian matrices of A, B, and C. For the random values of entries in matrices and fixed value $t_i = 1$, the following table shows the smallest errors between D and $U(T)$ and the number of computing times of $C(T)$ and $MP(T)$ for each smallest error. For instance, in the first row, 10 means $C(T)$ and $MP(T)$ were generated and computed 10 times with random entities of Hermitian matrices, and 0.0065 is the smallest error among them.

Table 3.1: Errors based on the number of optimizing searches (SU(2))

The number of optimizing searches	The smallest error between D and U(T)
10	0.0065
50	8.1223e-004
200	1.0679e-005
2000	7.4931e-010

For instance, the following figure shows all errors from the fifty trials. Among them the smallest error was 8.1223e-004.

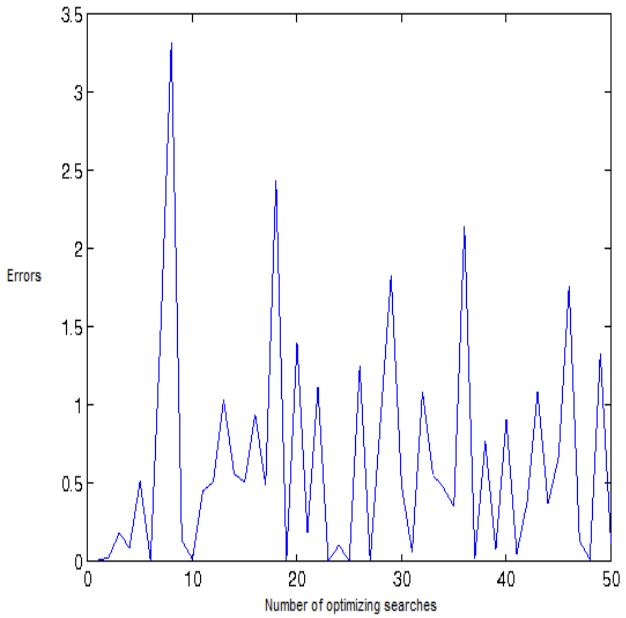

Figure 3.2: The smallest error for each optimizing search (Euler rotation Test)

Obviously the bigger the number of times of computation is the smaller the minimum error is. The approximation, 7.4931e-010 is good enough to check that the ideas work. We move on to the next step in which we attempt to control a more realistic single qubit system where the experimentalist can control a parameter in the Hamiltonian.

3.2 Controlling possible real quantum states

In this second step, we set the experiment to control a single qubit system. In reality, it is unlikely that an experimentalist will be able to control all the parameters in the function of $MP(T)$. So we attempt to control some of parameters in the Hermitian matrices in the function of $MP(T)$. The time variable served as the independent variable corresponding to rotation angle in the previous step, but in this second step, we fix the time interval and vary values of the parameters to find $U(T)$. That is, $MP(T) = \sum_{j=1}^{n} e^{-i\sigma_j \Delta t}$ where $\sigma_j = \begin{pmatrix} 1 & s(j)+2i \\ s(j)-2i & 2 \end{pmatrix}$ and $s(j)'$ s are randomly given initially but $\Delta t = 0.01$. We start with the same number of rotations as in the previous section and increase it to ten times in this experiment. That is, $U(T)$ is determined by n-time rotation from the function $MP(T)$ where n=3 or n=10.

3.2.1 Results from the second step

The same $C(T) = \mathrm{Re}\{tr[D - MP(T)]^{\dagger} \cdot [D - MP(T)]\}$ is also used to evaluate how good approximations are. However, the matrix production function in this step $MP(T)$ is $MP(T) = \sum_{j=1}^{n} e^{-i\sigma_j \Delta t}$ where $\sigma_j = \begin{pmatrix} 1 & s(j)+2i \\ s(j)-2i & 2 \end{pmatrix}$ and $s(j)'$ s are randomly given but $\Delta t = 0.01$ as fixed. We expect to see randomly generated $s(j)'$ s would make the difference, yet good results. For the random values of entities in matrices and fixed value $\Delta t = 0.01$, the following table shows the smallest errors between D and $MP(T)$ and the number of optimizing searches of $C(T)$ and $MP(T)$ for each smallest error.

The above results also show that the bigger the number of optimizing searches is the smaller the minimum error is. However, the approximations are not good compared to the previous step. The following two figures show the difference. They also show all errors from the fifty trials for each of three rotations and 10 rotations. Among them the smallest errors

Table 3.2: Errors based on the number of optimizing searches (SU(2))

The number of optimizing searches	The smallest error between D and U(T) for three times rotation	The smallest error between D and U(T) for ten times rotation
10	0.3745	0.3689
50	0.0192	0.1165
200	0.0490	0.0018
2000	0.0016	1.4590e-004

were 0.0016 at the 42^{nd} and 0.1165 at the 37^{nd} optimizing search for each of three time and ten time rotations.

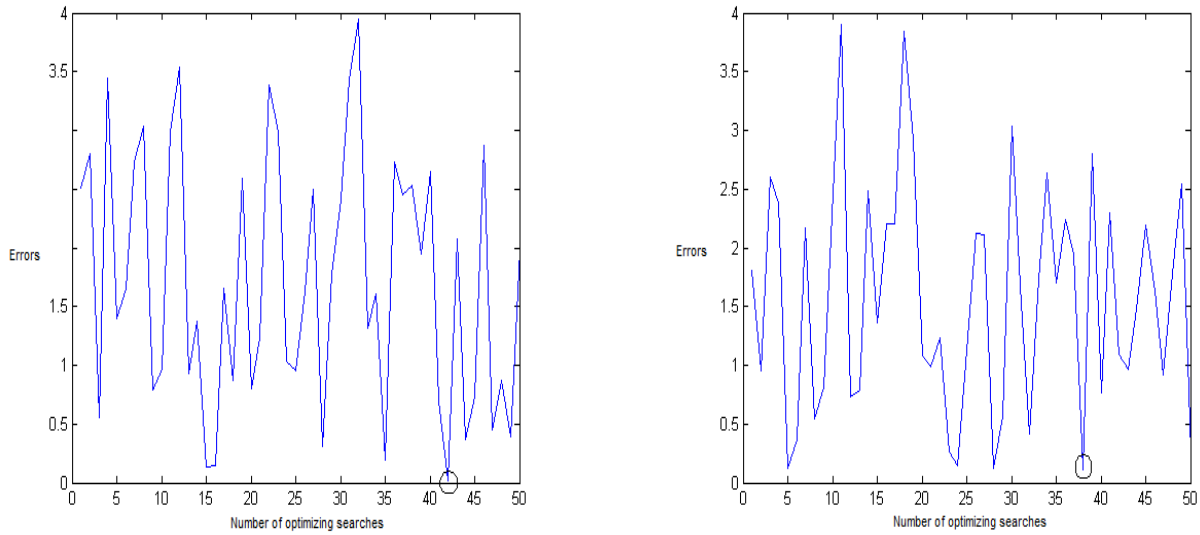

Figure 3.3: The smallest error for each optimizing search (Test for realistic control system)

To this end, we set the experiment with the different number of operators that are multiplied in the matrix product function, $MP(T)$ which means varying the number of rotation operators to approach to the target. In the next step, we also consider a coupled spin system instead of a single qubit system.

3.3 Varying the number of rotation operators in a coupled spin system: the case of $SU(4)$

The third step considers controlling a coupled spin system. Thus, we need a different dimension of rotation operator. We keep the notation σ to represent Hermitian matrices, which substitute the matrices A, B and C in step 1. In step 2, $\sigma_j = \begin{pmatrix} 1 & s(j)+2i \\ s(j)-2i & 2 \end{pmatrix}$ and $s(j)'s$ that is, $MP(t) = e^{-i\sigma_1\Delta t}e^{-i\sigma_2\Delta t}\dots e^{-i\sigma_n\Delta t}$ where random initial $s(j)'s$ and $\Delta t = 0.01$. Now we use 4×4 unitary matrices as operators for a coupled spin system. That is, $\sigma_j = \begin{pmatrix} 1 & s(j)+i & 1+is(j) & i \\ s(j)-i & -1 & 2+is(j) & s(j)+si \\ 1-is(j) & 2-is(j) & 3 & -s(j)+i \\ -i & s(j)-2i & -s(j)-i & 2 \end{pmatrix}\Gamma$

$MP(T) = \sum_{j=1}^{n} e^{-i\sigma_j\Delta t}$ where $\Delta t = 0.01$ as fixed. The locations for the parameter were arbitrary. We also vary the number of allowed rotation operators that are multiplied in $MP(T)$. This is another difference from the previous steps. This approach turned out to require more time to compute providing bigger errors compared to previous steps. The results are as follows.

3.3.1 Result from the third step

Table 3.3: Errors based on different number of rotations (SU(4))

The number of optimizing searches	The number of rotations	The smallest error between D and U(T)
10	20	0.2496
10	40	0.0158
10	50	6.5235e-004
10	100	3.0993e-04
50	10	2.0727
200	10	0.6509

From the table 3.3, at least 50 rotation operators were necessary for a relatively small value of error. Figures below support the fact.

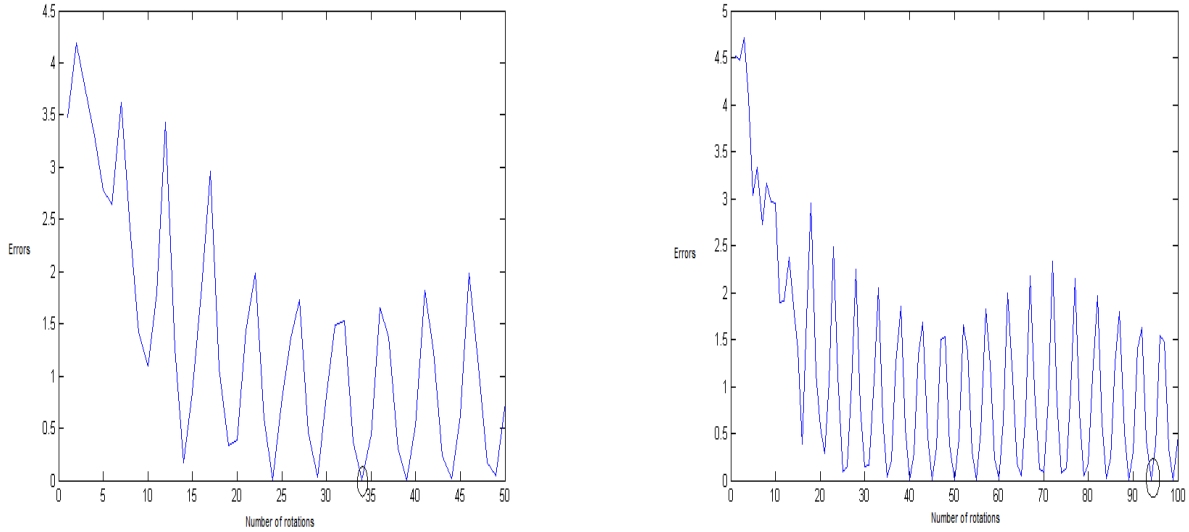

Figure 3.4:

The smallest errors: 6.5235e-004 (left) and 3.0993e-04 (right)

Figure 3.5: Errors and number of rotations for a fixed time interval

The conditions given in this section required much time to find a good approximation. Thus, in the following section, we vary the time variable as well as the number of rotation operators for both a single qubit system and a coupled qubit system.

3.4 Varied time parameters for both a single qubit and a coupled qubit system

As an attempt to find a better approximation at an efficient pace, in this section, we also vary the time values as inputs. First of all, we started with varying the range of the sizes of time intervals in order to see if the size of time intervals to manipulate coupled qubits influences the size of errors between target operators and their approximations. The following is the result for the errors between target operators and their approximations and for corresponding

time values to manipulate. In the table below, T_i represents the time parameter for the i^{th} interval and errors are found by manipulating coupled qubits 15 times.

Table 3.4: Errors based on different ranges of time interval (SU(4))

Time	interval range 0.2	interval range 4	interval range 68
T_1	0.0689	0.7291	43.5302
T_2	0.1809	0.1554	59.6608
T_3	0.0404	2.3222	16.4271
T_4	0.0817	3.8056	61.7564
T_5	0.0432	0.5854	42.9882
T_6	0.1752	0.5456	8.4126
T_7	0.0075	0.3523	0.5619
T_8	0.0381	0.2921	0.0953
T_9	0.0015	1.7881	39.6906
T_{10}	0.0775	1.5494	5.6572
T_{11}	0.1977	1.5706	25.2371
T_{12}	0.0707	0.0261	26.8472
T_{13}	0.0051	1.2984	67.9465
T_{14}	0.0832	2.6000	11.7821
T_{15}	0.1884	0.2122	34.0199
Error	**9.5252**	**9.9878e-010**	**9.9255**

The result show us time parameters in the second column gave the smallest error, and this informs that what initial input time value for optimizing searches produces the best result. We used this finding to optimize approximation and could find good approximations for a single qubit system by rotating four or five times. However, coupled qubit systems required

about 15 rotations to guarantee small errors. The following table and figures illustrate the results.

Table 3.5: Errors between target operators and their approximations in both single qubit and coupled quit systems varying number of rotations without fixing the value of the time variable

The number of rotations	single qubit system, SU(2)	coupled qubit system, SU(4)
5	1.1319e-009	0.1215
10	2.2770e-010	$1.1868e-013$
15	3.7920e-010	1.3635e-012
25	3.2129e-010	2.7259e-014

Figures and tables below show the results for cases of both a single qubit system and a coupled qubit system.

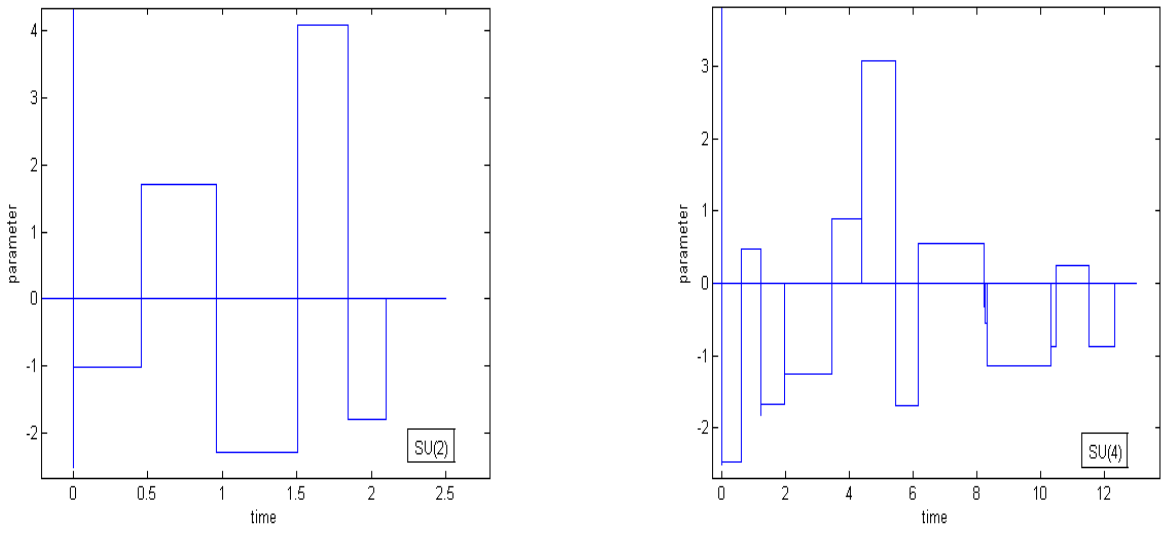

Figure 3.6: Parameters as a function of time

Table 3.6: Time evolution of a qubit

A single qubit system		
The number of rotation times	Time	Parameter
1	0.4575	-1.0120
2	0.9625	1.7143
3	1.5031	-2.2834
4	1.8399	4.0722
5	2.0976	-1.8015

Table 3.7: Time evolution of two qubits

A coupled qubit system		
The number of rotation times	Time	Parameter
1	0.6379	-2.4716
2	1.2462	0.4776
3	1.2529	-1.8176
4	1.9914	-1.6674
5	3.4533	-1.2557
6	4.3917	0.9005
7	5.4425	3.0838
8	6.1682	-1.6987
9	8.2198	0.5540
10	8.2676	-0.3165
11	8.3329	-0.5514
12	10.3142	-1.1481
13	10.4648	-0.8697
14	11.5158	0.2435
15	12.3178	-0.8810

We move on to the last section in which we test the idea of periodic control of coupled qubits system by fixing the number of rotation operators as 15 times.

3.5 Periodic control for coupled Qubit systems

In this section, we seek periodic control of coupled systems. We found 15 rotations produced a good approximation in the previous section, so we fixed the number of rotation operators. Instead of finding the errors between target operators and their approximations, first, we find the approximations for the 50^{th} root both of a target operator D. By taking 50^{th} power to both $D^{\frac{1}{50}}$ and $A^{\frac{1}{50}}$ which means the approximation for $D^{\frac{1}{50}}$, we find the approximation of the actual target operator and the error between them. The following table shows the results acquired by periodically controlling a coupled qubit system.

Table 3.8: Results from periodic controlling a coupled qubit system by rotating 15 times

The number of optimizing searches	Error between $D^{\frac{1}{50}}$ and $A^{\frac{1}{50}}$	Error between D and A
10	6.2329e-013	9.9878e-010
10	7.1377e-013	1.2176e-009

We used optimizing searches to find the approximation $A^{\frac{1}{50}}$ of $D^{\frac{1}{50}}$ where and 15 rotation operators were used per period. The results were from a few those trials.

3.5.1 A test result from periodic controlling coupled qubit systems

From the first case in the above table, the following graph and table specifically show parameters for time evolution of a coupled qubit system by periodically control.

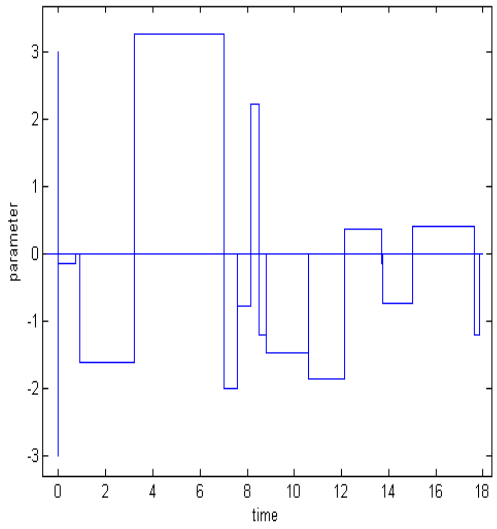

Figure 3.7: Parameters according to time

Table 3.9: Time evolution of periodic control for a coupled qubit system

The number of rotation times	Time	Parameter
1	0.7291	-0.1397
2	0.8845	-0.0056
3	3.2067	-1.6144
4	7.0123	3.2662
5	7.5977	-2.0154
6	8.1433	-0.7795
7	8.4956	2.2324
8	8.7877	-1.2022
9	10.5758	-1.4782
10	12.1252	-1.8717
11	13.6958	0.3673
12	13.7219	-0.1433
13	15.0203	-0.738
14	17.6203	0.4026
15	17.8325	-1.2164

3.5.2 Finding an approximation of the C-not operator

Finally, we periodically controlled a coupled qubit system to find an approximation of the CNOT operation, that is, the CNOT gate.

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

From a total of three attempts, the errors between the CNOT operator and its approximations were 1.3398e-009, 1.8433e-009, and 9.6140e-010. The following graph and table show the process of time evolution of a coupled qubit system to approximate the CNOT operation.

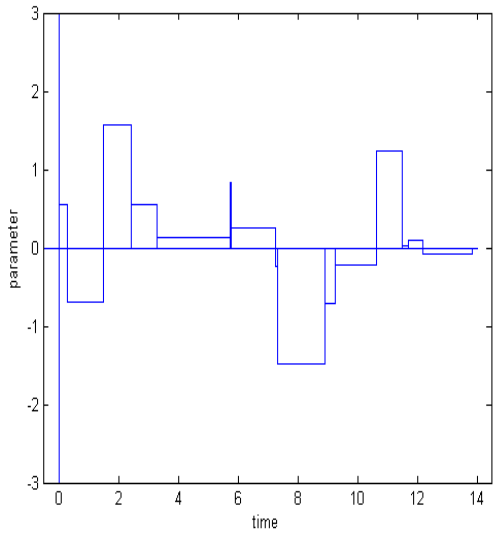

Figure 3.8: The time evolution parameters for the CNOT gate

Table 3.10: Control parameters for a coupled qubit system approximating the CNOT operation

The number of rotation times	Time	Parameter
1	0.2703	0.5462
2	1.4711	-0.6974
3	2.3962	1.5683
4	3.2765	0.5502
5	5.7227	0.1340
6	5.7396	0.8408
7	7.2085	0.2574
8	7.2989	-0.2366
9	8.8857	-1.4776
10	9.2085	-0.7189
11	10.6000	-0.2225
12	11.4783	1.2303
13	11.6661	0.0201
14	12.1661	0.1010
15	13.8166	-0.0840

Chapter 4

Conclusion

In this study, we sought ways to control a coupled qubit system for the approximation of arbitrary SU(2) and SU(4) operations. We increased the complexity of the operators in five steps. Findings from each step informed us how to move forward to the next step. In the last step, we found good approximations of any target operator including the CNOT operator by periodically controlling the qubit system.

Specifically we tested the idea of Euler rotation in the first step. We found the smallest error between a target operator and its approximations was 8.1223e-010 from 2000 optimizing searches.

In the second step we tested realistic qubit control systems, and the smallest errors between targets and their approximations were 0.0016 and 1.4590e-004 from three time rotations and from ten time rotations respectively. The result showed that the number of rotations matters.

Hence, we tested the role of the number of rotations fixing the time variable in the third step. What we found is more than 50 rotation operators are needed under the condition of the fixed time parameters to find a good approximation. We also needed to vary the time parameter to find better approximations.

In the fourth step, we varied time parameters and the number of rotations at the same time. This idea worked very well producing good approximations. However, we found the range of time intervals is the critical element to guarantee a good result. Either too small or too big time intervals couldn't provide good results. Within the range 4 of time, about 1.2 for the average time values guaranteed good approximations. As to the number of rotations,

at least five time rotations for a single qubit system and fifteen time rotations for a coupled qubit system guaranteed good approximations respectively.

In the last step, good approximations were acquired by applying the findings from the fourth step, which are the 15 time rotation and varying time variables, and by using the idea of periodically controlling qubit systems for both general qubit operators and the CNOT operator.

In conclusion, we needed to limit values of time variable within a certain range and use at least fifteen rotation operators with time and experimental control parameters, which means manipulating qubits at least fifteen times, in order to efficiently control a coupled qubit system.

Appendix A

Matlab codes

A.1 Matlab codes in section 3.1

```
function [y,z]=loop2opt(x)
global A; global B; global D;
k=zeros([x,1]);
l=zeros([x,3]);
for j=1:x
a = randn(1);
c = randn(1);
b = randn(1) + i * randn(1);
A = [a b; b' c];
a = randn(1);
c = randn(1);
b = randn(1) + i * randn(1);
B = [a b; b' c];
a = randn(1);
c = randn(1);
b = randn(1) + i * randn(1);
C = [a b; b' c];
D = expm(-i * C);
t0 = [1 2 3];
```

```
20 options = optimset('maxIter', 1e7, 'TolFun',1e-12);
21 l(j,:) = fminunc(@critsu2, t0);
22 k(j)=critsu2(l(j,:));
23 matprodsu2(l(j,:));
24 end
25 [m,n]=min(k);
26 y = k;
27 z=l(n,:);
```

```
1 function out = critsu2(t)
2 global D;
3 out = real(trace((D - matprodsu2(t))' * (D - matprodsu2(t))));
```

```
1 function out = matprodsu2(t)
2 global A; global B;
3 out = expm(-i * A * t(1)) * expm(-i * B * t(2)) * expm(-i * A * t(3));
```

A.2 Matlab codes in section 3.2

```
1 [y,z]=loopstep2opt(x)
2 D;
3 k=zeros([x,1]);
4 l=zeros([x,10]);
5 for j=1:x
6 a = randn(1);
```

```
c = randn(1);
b = randn(1) + i * randn(1);
C = [a b; b' c];
D = expm(-i * C);
e = randn(1);
f = randn(1);
g = randn(1);
s0 = [e f g 1 2 3 1 2 3 1];
options = optimset('maxIter', 1e7, 'TolFun',1e-12);
l(j,:) = fminunc(@critstep2, s0);
k(j)=critstep2(l(j,:));
matprostep2(l(j,:));
end
[m,n]=min(k);
y=k;
z=l(n,:);
```

```
function out = critstep2(s)
global D;
out = real(trace((D - matprostep2(s))' * (D - matprostep2(s))));
```

```
function out = matprostep2(s)
Δt = 0.01;
out = eye(2);
for i = 1:size(s, 2)
sig = [[1 complex(s(i), 2)];[complex(s(i), 2)' 2]];
out = out*expm(-complex(0,1)*sig*Δt);
```

```
7 end;
```

A.3 Matlab codes in section 3.3

```
1  function [y, z] = loop4opt(x, dim)
2  global D;
3  k = zeros([x, 1]);
4  l = zeros([x, dim]);
5  for j = 1:x
6  s0 = 50 * randn([1, dim]);
7  options = optimset('maxIter', 1e7, 'TolFun', 1e-12);
8  l(j,:) = fminunc(@critsu3, s0);
9  k(j) = critsu3(l(j,:));
10 end
11 [m, n] = min(k);
12 y = m;
13 z = l(n,:);
```

```
1  function out = critsu3(t)
2  global D;
3  out = real(trace((D - matprod(t))' * (D - matprod(t))));
```

```
1  function out = matprod(s)
2  ∆t = 1;
3  out = eye([4, 4]);
```

```
4  for i = 1:size(s, 2)
5  sig = [[1 complex(s(i),1) complex(1,s(i)) complex(0,1)];
6  [complex(s(i),1)' −1 complex(2,s(i)) complex(s(i), 2)];
7  [complex(1, s(i))' complex(2, s(i))' 3 complex(−s(i), 1)];
8  [complex(0,1)' complex(s(i), 2)' complex(−s(i), 1)' 2]];
9  out = out * expm(−complex(0,1) * sig * Δt);
10 end;
```

A.4 Matlab codes in section 3.4

analysis.m

```
1  global D;
2  a = randn(1);
3  c = randn(1) + complex(0,1) * randn(1);
4  b = randn(1) + complex(0,1) * randn(1);
5  d = randn(1) + complex(0,1) * randn(1);
6  e = randn(1);
7  f = randn(1) + complex(0,1) * randn(1);
8  g = randn(1) + complex(0,1) * randn(1);
9  h = randn(1);
10 i = randn(1) + complex(0,1) * randn(1);
11 j = randn(1);
12 C = [a b c d; b' e f g; c' f' h i; d' g' i' j];
13 D = expm(−complex(0,1) * C);
14 z = zeros([10 10]);
15 count = 1;
16 for i = 1:10
```

```
warning off all;
[y(count); z(1 : i; count)] = loop4opt(10, i);
warning on all;
count = count + 1;
end;
```

```
function [y, z] = loop4opt(x, dim)
global D;
k = zeros([x, 1]);
l = zeros([x, dim]);
for j = 1:x
s0 = randn([1, dim]);
options = optimset('maxIter', 1e7, 'TolFun', 1e-12);
l(j,:) = fminunc(@critsu3, s0);
k(j) = critsu3(l(j,:));
end
[m, n] = min(k);
y = m;
z = l(n,:);
```

```
function out = critsu3(t)
global D;
out = real(trace((D - matprod(t))' * (D - matprod(t))));
```

```
function out = matprod(s)
Δt = 1;
```

```
out = eye([4, 4]);
for i = 1:2: size(s, 2)-1
sig = [[1 complex(s(i),1) complex(1,s(i)) complex(0,1)];
[complex(s(i),1)' -1 complex(2,s(i)) complex(s(i), 2)];
[complex(1, s(i))' complex(2, s(i))' 3 complex(-s(i), 1)];
[complex(0,1)' complex(s(i), 2)' complex(-s(i), 1)' 2]];
out = out * expm(-complex(0,1) * sig * abs(s(i+1)));
end;
```

A.5 Matlab codes in section 3.5

analysis.m

```
global D; global C; global E;
a = randn(1);
c = randn(1) + complex(0,1) * randn(1);
b = randn(1) + complex(0,1) * randn(1);
d = randn(1) + complex(0,1) * randn(1);
e = randn(1);
f = randn(1) + complex(0,1) * randn(1);
g = randn(1) + complex(0,1) * randn(1);
h = randn(1);
i = randn(1) + complex(0,1) * randn(1);
j = randn(1);
C = [a b c d; b' e f g; c' f' h i; d' g' i' j];
D = expm(-complex(0,1) * C);
[v; l] = eig(D);
E = v*l^{1/50}*v^{-1};
```

```
z = zeros([1 30]);
[y; z] = loop4opt(10; 30) ;
w = matprod(z);
p = w50;
q = E50;
z
real(trace((p-q)'*(p-q)))
```

```
function [y, z] = loop4opt(x, dim)
k = zeros([x 1]);
l = zeros([x dim]);
for j = 1:10
s0 = randn([1, dim]);
options = optimset('maxIter', 1e7, 'TolFun', 1e-12);
l(j,:) = fminunc(@critsu3, s0);
k(j) = critsu3(l(j,:));
end
[m, n] = min(k);
y = m;
z = l(n,:);
```

```
function out = critsu3(t)
global D, global E;
out = real(trace((E - matprod(t))' * (E - matprod(t))));
```

```
function out = matprod(s)
```

```
2 out = eye([4, 4]);
3 for i = 1:2:size(s,2)-1
4  sig = [[1 complex(s(i),1) complex(1,s(i)) complex(0,1)];
5  [complex(s(i),1)' -1 complex(2,s(i)) complex(s(i), 2)];
6  [complex(1, s(i))' complex(2, s(i))' 3 complex(-s(i), 1)];
7  [complex(0,1)' complex(s(i), 2)' complex(-s(i), 1)' 2]];
8 out = out * expm(-complex(0,1) * sig * abs(s(i+1)));
9 end;
```

Bibliography

[1] J. J. Sakurai (1997). *Modern Quantum Mechanics*. Addison-Wesley Longman. New York.

[2] George Johnson (2003) *A Shorcut Through Time*. Alfred A. Knopf. New York.

[3] Hassell, N. David Mermin (2007). *Quantum Computer Science*. Cambridge University Press. New York.

[4] Gennady P. Berman, Gary D. Doolen, Ronnie Mainieri, and Vladimir I. Tsifrinovich. (1998). *Introduction to Quantum Computers*. World Scientific. New Jersey.

[5] Phillip Kaye, Raymond Laflamme, and Michele Mosca. (2007). *An Introduction to Quantum Computing*. Oxford University Press Inc. New York.

[6] Michel Le Bellac. (2006). *A Short Introduction to Quantum Information and Quantum Computation*. Cambridge University Press. New York.

[7] Soonchil Lee. (2003). *Quantum Computer: The 21st Century Science Revolution*. Sallim. Seoul.

[8] Navin Khaneja, Timo Reiss, Cindie Kehlet, Thomas Schulte- Herbrüggen, and Steffen J. Glaser. (2005). Optimal control of coupled spin dynamics: design of NMR pulse sequences by gradient ascent algorithms. *Journal of Magnetic Resonance* Vol 172, pp. 296-305.

[9] Michael A. Nielsen and Isaac L. Chuang. (2000). Quantum computation and quantum information. Cambridge University Press. New York.

[10] Douglas Robertson (2003). Impact of Computers Is Just Beginning, CU-Boulder Futurist Says. *News Events* May 20, 2003. Web: http://cires.colorado.edu/news/press/2003/03-05-20.html.